A NOVEL EMBEDDED FEATURE SELECTION FRAMEWORK FOR PROBABILISTIC LOAD FORECASTING WITH SPARSE DATA VIA BAYESIAN INFERENCE

A Thesis Submitted to the
College of Graduate and Postdoctoral Studies
In Partial Fulfillment of the Requirements
For the Degree of Doctor of Philosophy
In the Department of Electrical and Computer Engineering
University of Saskatchewan
Saskatoon

By

Bingzhi Wang

PERMISSION TO USE

In presenting this thesis in partial fulfillment of the requirements for a Postgraduate degree from the University of Saskatchewan, I agree that the Libraries of this University may make it freely available for inspection. I further agree that permission for copying of this thesis in any manner, in whole or in part, for scholarly purposes may be granted by the professor or professors who supervised my thesis work or, in their absence, by the Head of the Department or the Dean of the College in which my thesis work was done. It is understood that any copying or publication or use of this thesis/dissertation or parts thereof for financial gain shall not be allowed without my written permission. It is also understood that due recognition shall be given to me and to the University of Saskatchewan in any scholarly use which may be made of any material in my thesis.

Requests for permission to copy or to make other uses of materials in this thesis in whole or part should be addressed to:

Head of the Department of Electrical and Computer Engineering

57 Campus Drive

University of Saskatchewan

Saskatoon, Saskatchewan S7N 5A9 Canada

OR

Dean

College of Graduate and Postdoctoral Studies

University of Saskatchewan

116 Thorvaldson Building, 110 Science Place

Saskatoon, Saskatchewan S7N 5C9 Canada

ABSTRACT

With the modernization of power industry over recent decades, diverse smart technologies have been introduced to the power systems. Such transition has brought in a significant level of variability and uncertainty to the networks, resulting in less predictable electricity demand. In this regard, load forecasting stands in the breach and is even more challenging. Urgent needs have been raised from different sections, especially for probabilistic analysis for industrial applications. Hence, attentions have been shifted from point load forecasting to probabilistic load forecasting (PLF) in recent years.

This research proposes a novel embedded feature selection method for PLF to deal with sparse features and thus to improve PLF performance. Firstly, the proposed method employs quantile regression to connect the predictor variables and each quantile of the distribution of the load. Thereafter, an embedded feature selection structure is incorporated to identify and select subsets of input features by introducing an inclusion indicator variable for each feature. Then, Bayesian inference is applied to the model with a sparseness favoring prior endowed over the inclusion indicator variables. A Markov Chain Monte Carlo (MCMC) approach is adopted to sample the parameters from the posterior. Finally, the samples are used to approximate the posterior distribution, which is achieved by using discrete formulas applied to these samples to approximate the integrals of interest. The proposed approach allows each quantile of the distribution of the dependent load to be affected by different sets of features, and also allows all features to take a chance to show their impact on the load. Consequently, this methodology leads to the improved estimation of more complex predictive densities. The proposed framework has been successfully applied to a linear model, the quantile linear regression, and been extended to improve the performance of a nonlinear model.

Three case studies have been designed to validate the effectiveness of the proposed method. The first case study performed on an open dataset validates that the proposed feature selection technique can improve the performance of PLF based on quantile linear regression and outperforms the selected comparable benchmarks. This case study does not consider any recency effect. The second case study further examines the impact of recency effect using another open dataset which contains historical load and weather records of 10 different regions. The third case

study explores the potential of extending the application of the proposed framework for nonlinear models. In this case study, the proposed method is used as a wrapper approach and applied to a nonlinear model. The simulation results show that the proposed method has the best overall performance among all the tested methods with and without considering recency effect, and it could slightly improve the performance of other models when applied as a wrapper approach.

ACKNOWLEDGMENTS

I would first like to express my sincere gratitude to my supervisor Prof. C.Y. Chung, for his patience and continuous support during my Ph.D. study. His immense knowledge and guidance greatly facilitate my research. Besides academic study, he also encourages me to actively join industry activities, for instance, taking responsibility for an engage grant project with Saskatoon Light & Power, and joining the Engineer in Training program in SaskPower. These valuable experiences motivate me to have a deeper insight into my research and make my work more practical and beneficial to the power industry.

I would also like to thank my co-supervisor, Prof. Xiaodong Liang, for her kind help during the last year of my Ph.D. study. She gives me incredible support for the administrative affairs of my last year and provides valuable suggestions for my academic study and career life.

Besides my supervisors, I would like to thank the rest of my graduation committee, Prof. Xiaozhe Wang, Prof. Chris Zhang, Prof. Chen Li, and Prof. Sherif Faried for their insightful review comments and suggestions, and also their questions which provide great help in improving my thesis.

I also would like to thank my lab mates, for accompanying me through those very long very long days. We share happiness together, take care of each other and help each other. Without their support I cannot get through all the difficulties from both life and research.

Specifically, I would like to express my true love to my fluffy friends, Erpang Niu and MuscleMumao, for their companionship day and night throughout those beautiful summers and chilling winters in Saskatoon.

Last but not the least, I would like to express my sincere thanks to my family members: my mother, my older brother and aunt parents, for their spiritual and financial support throughout my life overbroad. They are the headspring of my self-belief and sense of being loved.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

BQLRFS	Bayesian quantile linear regression with feature selection
CNN	convolutional neural network
CRPS	continuous ranked probability score
EMS	energy management system
FTEST	***F***-test
GEFCom	Global Energy Forecasting Competition
HPC	high-performance computing
KDE	kernel density estimation
LASSO	least absolute shrinkage and selection operator
LSTM	long short-term memory
MCMC	Markov Chain Monte Carlo
NCA	neighborhood component analysis
VVO	Volt-Var optimization
PDF	probability density function
PF	probabilistic forecast
PI	predictive interval
PLF	probabilistic load forecasting
QLR	quantile linear regression
QRLASSO	quantile regression least absolute shrinkage and selection operator
QRNN	quantile regression neural network

RRF	RReliefF
SFS	sequential forward search

1. Introduction

1.1. Electric Load Forecasting

To guarantee a reliable and secure power supply, the produced electricity must constantly fulfill the load and system loss requirements within acceptable limits, given the inefficiency of large-scale electricity storage. Over decades electric load forecasting has been a critical component of power system operation and planning, delivering considerable benefits to both power utilities and their consumers. The definition of electric load forecasting is straightforward. Basically, it can be defined as the prediction of anticipated load for a predetermined period, ranging from a few hours to several years into the future. Load forecasting provides utilities with rich information on a wide range of decision-making processes. It can tell when, where and how electricity is demanded, and thus assist system operators to make decisions on different operation and planning actions such as adjusting output of generators, interchanging power with neighboring systems, and installing extra capacity to meet the increasing demand, etc. Hence, on the one hand, utilities can maximize their revenues under the promise of a secure system. One the other hand, the customers can benefit from a secure and reliable power supply.

With the modernization of power systems over recent decades, load forecasting has grabbed increasing attention. New requirements arise in different sections including transmission and distribution planning, secure and optimal operation, and system investments [1]. Due to the stochastic nature of load and the presence of diverse exogenous factors like weather conditions, calendar effects, and others, achieving complete accuracy in load forecasting is not feasible. Inaccurate load forecasts can lead to an increase in cost. For example, overestimating the load will require extra generation, which requires increasing output or committing more units, resulting in augmented operational costs. Underestimating the load can cause even more severe problems.

Inadequate generation can lead to the failure of supplying the required reserve and stability to the system, which may ultimately result in a system breakdown [2]. Besides, failing to meet the demand will cause even more complicated impact on end users. For example, a lack of generation will force the utilities to buy power from the market, where the price is super expensive if it is close to the last minute. As per the estimates provided in [3], 1% increase in the national load forecasting error can cost around £10 million a year at 1984 in U.K. because of inefficient plant scheduling. As indicated in [4], a reduction of 1% in the load forecasting error for a 10,000 MW capacity system can potentially result in savings of approximately $1.6 million per year. Therefore, it cannot be overemphasized that accurate load forecasting is at the core of operating and planning a reliance, secure and economical power system.

1.2. From Point Load Forecasting to Probabilistic Load Forecasting

Over the past few decades, electric load forecasting has facilitated a wide range of planning and operation tasks for power utilities. Conventionally, most of the applications in the past relied on point load forecasting, such as unit commitment, load switching, economic dispatch, etc. However, new challenge arises from the power industry modernization. In recent decades, the widespread diffusion of new facilities, such as the introduction of advanced smart grid technologies and intermittent renewable energy resources, have brought in a significant level of variability and uncertainty to the networks, resulting in less predictable electricity demand and the urgent needs for probabilistic analysis for the industrial applications.

A wide range of applications need probabilistic load forecasting (PLF) for the derivations of their probabilistic analysis. A typical one is probabilistic load flow, the well-known methodology used to evaluate the impact of the uncertainties on a set of electrical indices and the operational risks of the system. The load uncertainty is commonly modeled by a stationary statistical distribution based on historical records, and thus failed to construct future scenarios. To consider the sequence order of the system events, the calculation of time-varying load flow has seen increasing interests very recently [5], which takes into account the time series models of demand and generation. Hence, PLF can be adopted to generate a time-varying injection to benefit the calculation of time-varying flows and voltages for future scenarios. An extension of the power

flow-based application is the Volt-Var optimization (VVO), which is a control function used in distribution systems to keep the load voltages within the standards. From the operational planning perspective of distribution networks, VVO can be employed to help determine the best operation condition for the control equipment for a period of time in the future (e.g., 1 day ahead). To guard against uncertainty in the load, PLF is used to support the computation of the voltage variation and consequently help choose the optimal settings that are robust against load fluctuations [6]. Another important applicable scenario of PLF is for microgrid, where a high penetration of intermittent distributed resources is integrated. PLF is frequently used to quantify the upcoming uncertainties in the demand in the microgrid to mitigate several operation issues and support the energy management system. One case is to use it for the assessment of operational reliability. Variation in the demand is one of the main uncertain sources that impacts the reliability metrics. Conventionally, reliability evaluation assumes a constant distribution based on statistical data for the load uncertainty, which, however, could vary in an operational time frame. Hence, the operational reliability evaluation for the microgrid has drawn increasing attention [7], [8], which requires PLF to capture the time-varying probabilities of the load in the short term. Another important application in the microgrid is the optimal operation of the energy management system (EMS) under variable generation and demand [9]. A well-designed EMS can lessen the impact of uncertainty and facilitate the integration of distributed resources with the support of an accurate forecasting system. For the purpose of optimal dispatch operation, the EMS is fed with relevant information of the generation and demand, as well as the corresponding forecasts. To ensure a specific level of reliability, the forecasting system should not only give the expected value but also accounts for the associated uncertainty. In this case, PLF plays a critical role in the estimation of the future load, and thus benefiting the probabilistic analysis.

Beyond the above-mentioned scenarios, other applications include but are not limited to stochastic unit commitment, probabilistic load flow, reliability planning, etc. [10], which rely on accurate PLF.

Among the limited literature on PLF in the scope of technical and methodological development, quantile regression is widely used to directly generate probabilistic forecasts (PFs). It is usually combined with machine learning techniques including neural network [11], [12], random forests [13], etc. Another approach that can be used to directly generate PFs is kernel density

estimation [14]. PFs can also be indirectly generated from point forecasts, for instance, by modeling and simulating the residuals of the underlying point forecast [15], [16], or by feeding temperature scenarios to point forecasting models [17]. To manifest uncertainty, these methods provide PFs in the form of confidential intervals, quantiles or the whole probability density function (PDF), which provides more information of the predicted load than point forecasts thus enhancing the decision-making process in operation and planning of the system. A systematic tutorial review on PLF can be found in [10]. Most of the relevant works focus on establishing and optimizing the forecasting model, while very few attentions have been paid to the feature selection phase, particularly in the area of PLF.

1.3. Feature Selection for Load Forecasting

Feature selection is the process of selecting a subset of relevant input features when constructing a predictive model. It is aimed to avoid the curse of dimensionality, reduce modeling complexity, reduce the risk of over-fitting and improve the forecasting performance. A majority of the feature selection algorithms come with an evaluation metric which scores the selected features, thus offering better interpretability.

From a taxonomic standpoint, feature selection techniques are typically classified into three groups: supervised, unsupervised, and semi-supervised feature selection. Supervised feature selection algorithms can further be categorized into filter methods, wrapper methods, embedded methods and hybrid methods [18]. A comprehensive review is given in Chapter 2. It has not been a long time since researchers began using various feature selection techniques for their predictive model construction, however, with most of the efforts made on deterministic load forecasting. To list a few examples, [19] proposes a hybrid filter-wrapper approach considering relevancy, redundancy and interaction of the candidate features for short-term load forecasting. A conditional mutual information-based feature selection method is developed in [20], which can carry out relevance and redundancy analysis. To extract the deep features from multivariate data, [21] incorporates an embedded feature selection process into a Long Short-Term Memory based network model through a hybrid ensemble approach for ultra-short-term industrial load forecasting. However, very few papers introduce feature selection to PLF. It is a prevalent

approach in the literature to employ heuristic methods such as filter and wrapper methods, which use a point error measure for feature selection in PLF. Specifically, features are initially selected based on a point error measure and subsequently employed for PLF. Nonetheless, these methods neglect the inherent mechanism of PLF and are not appropriate for the task. After conducting a thorough search, I was able to locate only two publications that have investigated feature selection for PLF using holistic methods. [22] proposes a wrapper method by using a probabilistic error measure for feature selection and compare it with method using a point error measure in the context of PLF. However, this method performs feature selection only for part of all available features, utilizing exhaustive search to explore all possible combinations of this subset. As a result, it fails to conduct a comprehensive evaluation of all features and does not demonstrate robust generalization. [23] introduces L_1-norm sparse penalty to quantile regression model based on least absolute shrinkage and selection operator (LASSO) to select features and to the best of the authors' knowledge, is the only recent paper that falls in the scope of embedding feature selection into a PLF model. This method uses a probabilistic error measure for feature selection and the error measure is consistent with the probabilistic error measure used for the final PLF evaluation. This approach is holistic and considers the forecasting process as a whole, instead of individual parts, to address the problem of feature selection for PLF. This method allows the selected features to vary among different quantiles, thus showing its potential to capture complex densities more accurately. However, this method has to go through a model selection process for every quantile to search for the optimal adjustment parameter and it is not equipped with the ability of handling sparse input feature space. In this regard, our proposed method is expected to be a holistic embedded method that surpasses the limitations of the most recent state-of-the-art, by inherently better modeling the uncertainty while selecting features. The capability of the proposed method is described in detail in the following subsection.

1.4. Research Objectives and Contributions

The basic objective of this research work is to propose a new embedded feature selection framework for PLF via Bayesian inference. The main contributions of the research are four-fold:

1) **An embedded feature selection framework via Bayesian quantile regression is added to the limited literature for PLF.**

   The predominant research in the field of load forecasting has concentrated primarily on feature selection for deterministic load forecasting models. However, the body of literature on feature selection for probabilistic load forecasting is quite sparse. In practical applications, researchers often resort to heuristic approaches such as filter or wrapper methods. Nevertheless, these methods rely on a point estimation error metric for feature selection, rendering them unsuitable for probabilistic models. To address this issue, I have proposed a novel embedded feature selection method for probabilistic load forecasting, which offers a more comprehensive approach. The framework can significantly improve the forecasting performance of a certain predictive model and outperforms comparable benchmarks.

2) **The proposed framework is capable of handling high-dimensional datasets and sparse feature space.**

   The dummy coding scheme used for categorical variables in load forecasting models make the problem a high-dimensional one and introduces great sparsity to the models. Hence, we introduce a sparseness-favoring prior for the prior probability of the inclusion indicator variable. This sparseness-favoring prior is used to encourage the model to favor solutions that are sparser. In other words, it encourages sparsity in the model. This prior is added to our Bayesian model as a penalty term in the posterior distribution. By incorporating this penalty term, the model is less likely to overfit the data and is more likely to select the most relevant features, leading to a simpler and more interpretable model.

   The idea behind this prior is that in many real-world problems, the underlying true model is often sparse, meaning that only a small number of features or variables are relevant for predicting the outcome. By using a sparseness-favoring prior, the model can be induced to automatically identify and select the most relevant features, while suppressing the effects of irrelevant or noisy features.

3) **The model can estimate complex distributions more accurately.**

   Through the use of Bayesian inference, the proposed approach enables selected features to vary across different quantiles and allows all features to take a chance to show its influence.

In contrast, the current state-of-the-art techniques fail to provide this capability. Consequently, the proposed approach achieves a higher degree of precision in estimating complex distributions.

4) **The proposed method can provide more meaningful interpretation of feature selection results.**

   The proposed feature selection methodology differs from the current state-of-the-art in terms of interpretability. Rather than relying on importance scores to interpret the feature selection results, the proposed method employs inclusion probabilities to assess the relevance of all input features.

To validate the effectiveness of this framework, three major parts are designed for this research, as described in the following:

- **Part I:** In the first part, the proposed framework is applied to the quantile linear regression model and tested with state-of-the-art techniques including three filter methods, two wrapper methods, an embedded method, as well as the origin linear model and a nonlinear model without feature selection using an open dataset. This dataset contains historical records for one region. This part focuses on examining the model performance without any recency effect.
- **Part II:** In the second part, the proposed framework is applied to the same predictive model used in Part I with another public dataset which contains historical records for 10 different zones. The main objective of this part is to examine the model performance with the consideration of recency effect. Besides, this part also validates that our conclusions are not restricted to one specific dataset or region.
- **Part III:** This part extends the proposed methodology and applies the probabilistic feature selection results to a nonlinear predictive model. The same dataset as in Part II is used in this part. The simulations are designed to validate if the proposed feature selection method could outperform the benchmark methods when applied to nonlinear predictive models.

1.5. Structure of the Thesis

This thesis consists of five chapters which is briefly described as follows:

Chapter 1 introduces basic concepts of electric load forecasting, PLF and feature selection. The literature on feature selection for electric load forecasting including PLF is briefly reviewed. The research objective and the main contribution of this research are lastly presented in this chapter.

Chapter 2 further discusses feature selection techniques. General background and development are presented. Formal definition and preliminary concepts are introduced. A comprehensive review of the state-of-art feature selection and PLF techniques is also given in this chapter.

Chapter 3 presents the proposed embedded feature selection method for PLF via Bayesian inference. The linear quantile regression model selected for forecasting is discussed first. Afterward, the proposed methodology is introduced in two steps following the Bayesian inference, specifying the priors, and sampling from the posteriors. The techniques that are used in the modeling and forecasting process are introduced. The whole structure of the proposed methodology is summarized at the end of this chapter.

Chapter 4 demonstrates the effectiveness of the proposed method by comparing with three filter methods, two wrapper methods and an embedded method using a public dataset that contains historical records for one zone. The simulations without considering any recency effect are implemented. A comprehensive evaluation criterion for PLF is introduced. Short-term PLF is carried out to evaluate the performance of the feature selection techniques. Feature selection interpretation is also discussed in this chapter.

Chapter 5 extends the simulations of Chapter 4 by considering recency effect, which comprehensively examines how recency effect impacts the performance of the proposed method. The tests are conducted on another public dataset that contains information for 10 different zones, which also further confirms that our conclusions are not restricted to one specific dataset.

Chapter 6 proposes a wrapper approach based on the proposed feature selection method. This chapter is designed to examine if the feature selection result of the proposed method could possibly improve the performance of nonlinear forecasting models. Thus, the proposed feature selection method plays as a wrapper, the result of which is then fed to a popular nonlinear probabilistic forecasting model for validation.

Chapter 7 summarizes the thesis.

2. Literature Review

2.1. Introduction

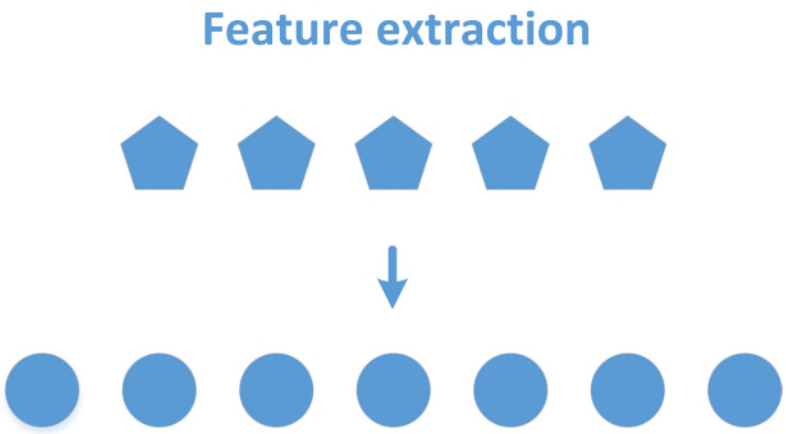
This chapter briefly introduces the basic concepts of feature selection and PLF and reviews the state-of-the-art literature for both topics.

2.2. Introduction of Feature Selection

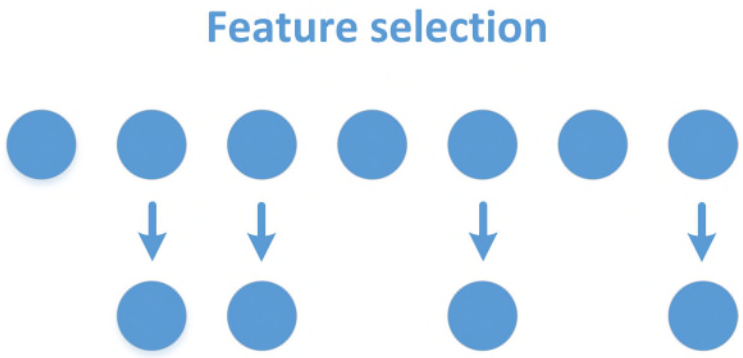
Over the past decades, the domain of features of the data involved in the applications of machine learning and data mining have expended explosively from tens to hundreds or even thousands of features. Serious challenges have been presented to existing learning methods due to the curse of dimensionality. The presence of a large set of features tend to increase the modeling complexity and the risk of overfitting of a learning model, thus resulting in reduced learning performance. To address this problem, feature selection methods have been extensively studied in the literature.

Feature selection, also known as variable selection, is defined as the process of selecting a subset of relevant features when constructing a predictive model. Feature selection is often raised up together with another term, feature extraction, which is defined as the process of transforming, combining, and reformatting raw data or existing features into new ones. Feature selection and feature extraction are both known as parts of feature engineering. The difference of these two terms is illustrated as Figure 2.1. In practice, feature extraction is often used to transform the raw data into feature that a particular algorithm can understand. Sometimes we can obtain some inherent features directly based on our prior knowledge of the problem to be solved. These inherent features can again be reformatted or transformed to new features that best fit the needs of the target

algorithm through feature extraction. The task of feature selection is clearer, which is to select an optimal subset of features that best improve the performance of the model and discard the rest.



(a) Feature extraction



(b) Feature selection

Figure 2.1 Schematic illustration of difference between feature extraction and feature selection

A typical data mining analysis pipeline is depicted in Figure 2.2, as described in [24]. First, the raw data is reformatted and transformed into a format in preparation for analysis. In this step, the data is split into different sets including training set, validation set and testing set. Thereafter, the preprocessed data goes through a feature engineering process. This step can include either feature extraction or feature selection alone, or these two combined together, with the features first being extracted and then selected. The obtained optimal subset of features is then fed into the training stage. In some cases, the performance of the training can be fed back to another round of feature

selection, such as a wrapper method which is introduced later. Finally, the trained model is validated and accessed, and the knowledge is obtained for the purpose of analysis. Feature selection plays an essential role in the stream of a successful data analysis process. Removing irrelevant and redundant features in the data space will ultimately improve the performance of a model, while improper feature selection will significantly deteriorate the model performance such as that relevant features are identified as irrelevant and removed from the feature space.

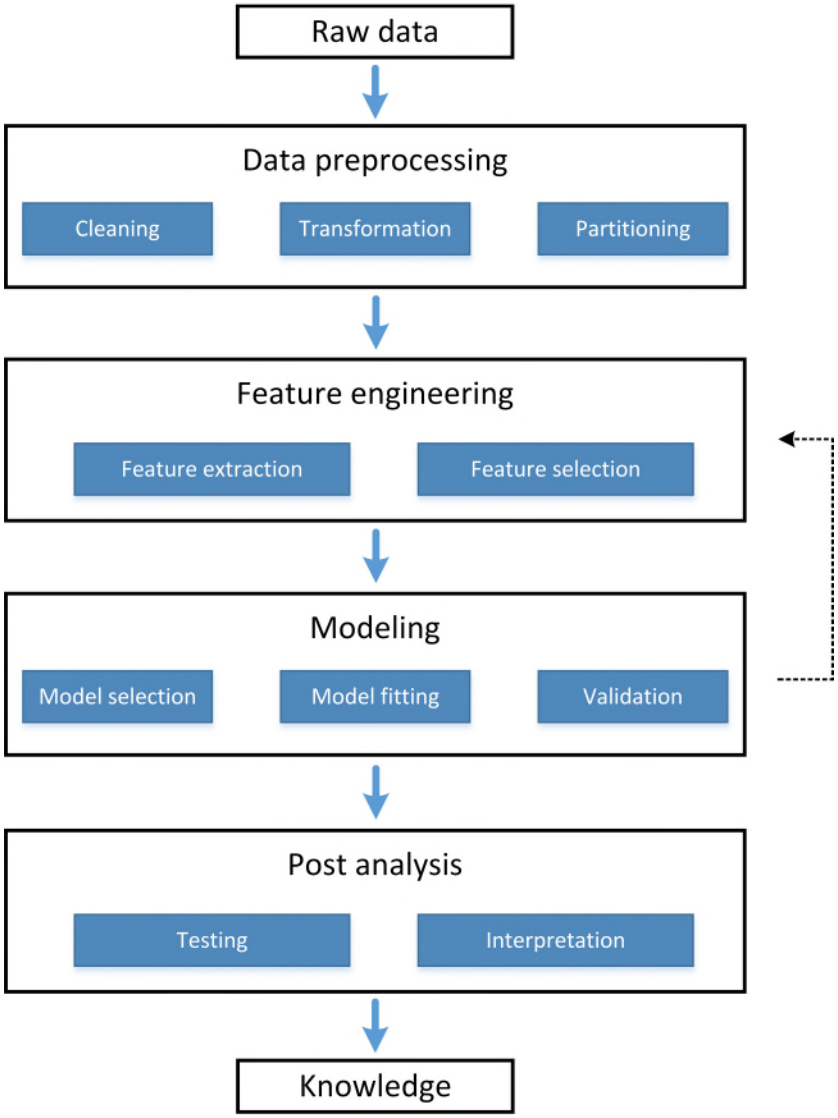


Figure 2.2 A typical data mining analysis pipeline

The objectives of feature selection are manifold, with the main ones being to avoid the curse of dimensionality, to reduce modeling complexity, to reduce the risk of over-fitting and to improve

the forecasting performance, by removing the irrelevancy and redundancy in the inputs. Features used to train a model are not a “the more, the merrier” thing. An apparent problem brought by high-dimensional data is that there is a positive correlation between the number of features and the training time. Besides of the dimensionality of the data, having redundant features in the input data will also dramatically slow down the training process of a learning model. For example, it may take too long for the gradient descent algorithm to oscillate and converge when having many redundant features in the training data. More iterations will be required by the algorithm, therefore resulting in much longer training time. Redundant features may also deteriorate the performance of the learning model. Taking the multiple linear regression model as an example, if there is redundancy in the training data matrix, the rank of the matrix will not be full. As a result, the optimal estimator cannot be obtained because the inverse calculation does not exist. Another example is that, when an algorithm has a predefined number of iterations, the algorithm may terminate too early and give a model with its performance lower than expected.

Feature selection algorithms can be basically classified into three categories, i.e., supervised, unsupervised and semi-supervised feature selection [25], based on whether the data is labelled or not. A majority of the literature fall in the scope of supervised feature selection because it is the earliest and most used practice. Supervised feature selection algorithms are used for labelled data. They identify relevant features for best achieving the goal of the supervised model by making use of the labeled outputs. Supervised feature selection methods can be further classified into filter methods, wrapper methods and embedded methods, which are introduced in detail in later sections. Unsupervised feature selection algorithms evaluate the features based on various criteria, such as entropy, variance, data structure, without needing label information. Semi-supervised feature selection algorithms integrate labeled data into unlabeled data as additional information to improve the learning performance when the data is partially labelled. A diagram of such classification is shown as Figure 2.3.

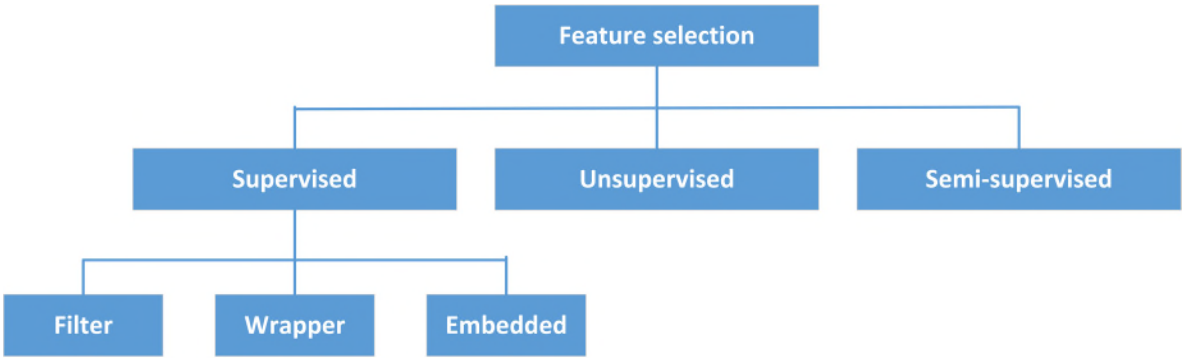


Figure 2.3 Feature selection categorization

Generally, feature selection is performed in four steps [26]:

1) Subset generation

   First, a search strategy is designed to search the feature space for a candidate subset of features. The search can be complete, sequential (forward or backward), random or based on certain algorithms.

2) Subset evaluation

   Thereafter, the candidate subset of features obtained in the first step is evaluated based on a certain evaluation criterion.

3) Stopping criterion

   A stopping criterion is predefined before the search. After going through all the generated subsets of features, the optimal subset is determined based on the evaluation criterion.

4) Validation

   The final step validates the optimal subset of features based on domain knowledge or a validation dataset.

Because load forecasting is a supervised learning problem, the rest of this chapter goes through the literature and briefly introduces the most used supervised feature selection methods.

2.3. Review of State-of-Art Feature Selection Techniques

In the following subsections, a brief description of the most used supervised feature selection

techniques and their methodological development are reviewed. A taxonomy of the methods in this category including filter methods, wrapper methods and embedded methods, is presented. The fundamentals, the main characteristics, and the advantages and disadvantages of these methods are reviewed.

2.3.1. Filter Methods

Filter methods measure the importance of each feature independently based on certain statistical criteria regardless of the forecasting algorithm. They examine the intrinsic characteristics of the features prior to the learning process. Representative filter methods include Fisher score [27], Relief [24], mutual information [28] and Pearson correlation coefficients [29], to name a few. In most cases, these methods perform the task of feature selection in a manner of two steps, as illustrated by Figure 2.4. First, a certain criterion is used to evaluate the features. The evaluation can be either univariate or multivariate. Thereafter, a threshold is chosen below which the features are neglected. A brief description for the previously mentioned filter methods is given below.

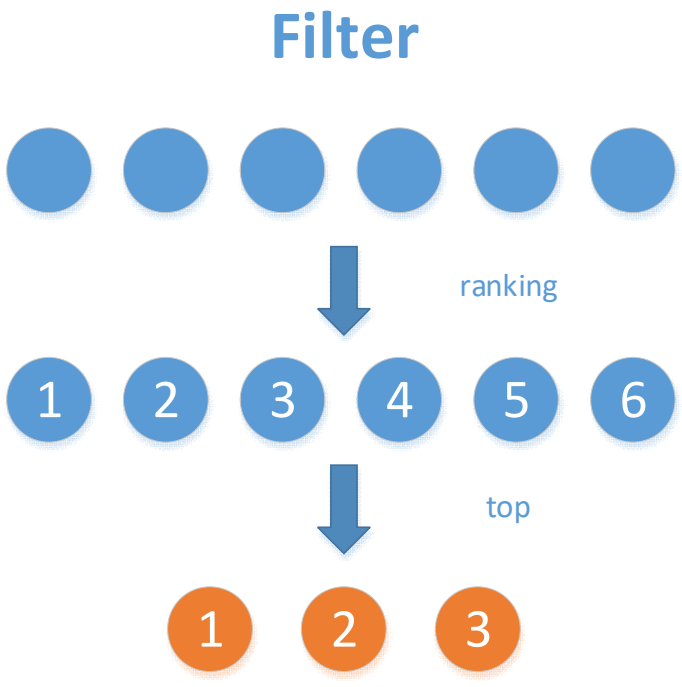


Figure 2.4 Schematic representation of filter method

Fisher score: This method selects each feature independently based on the Fisher criterion [27]. The Fisher score for the i^{th} feature is given by

$$F_i = \frac{\sum_{k=1}^{K} n_j(\mu_{ij} - \mu_i)^2}{\sum_{k=1}^{K} n_j \rho_{ij}^2} \tag{2.1}$$

where μ_{ij} and ρ_{ij} are the mean and the variance of the i^{th} feature in the k^{th} class respectively, n_k is the size of the k^{th} class, and μ_i is the mean of the i^{th} feature throughout the whole dataset. The objective of Fisher score method is to target those features that make the distances between data points in different classes as large as possible and the distances between data points in the same class as small as possible.

Relief: This method calculates a feature score based on the estimation of feature quality differences between nearest neighbor instance pairs. Relief-based algorithm penalizes the features that give different values to neighbors of the same class, while rewards predictors that give different values to neighbors of different classes. An introduction to the algorithm can be found in [24]. As an example, the original Relief algorithm developed by [30] is given by the following pseudo code as Table 2.1. In this table, n is the number of training instances. a is the number of features. m is a parameter that denotes the number of random training instances. $\boldsymbol{W}$ denotes the vector of the feature weights. A denotes one certain feature. $\boldsymbol{W}(A)$ denotes the weight of feature A. The nearest hit is defined as the nearest instance with the same class, and the nearest miss is defined as the nearest instance with the opposite class. The function $diff$ is defined as the difference between two instances I_1 and I_2 in the value of feature A. The definition of $diff$ is different for continuous and discrete features. For discrete features, $diff$ is given by

$$diff(A, I_1, I_2) = \begin{cases} 0 & if\ value(A, I_1) = value(A, I_2) \\ 1 & if\ otherwise \end{cases} \tag{2.2}$$

For continuous features, $diff$ is given by

$$diff(A, I_1, I_2) = \frac{|value(A, I_1) - value(A, I_2)|}{\max(A) - \min(A)} \tag{2.3}$$

Table 2.1 Pseudo code of the original Relief algorithm

Original Relief algorithm
for $i = 1, \dots, m$ repeat
randomly select an instance R_i
find a nearest hit H and miss M
for $A = 1, \dots, a$ repeat
$\boldsymbol{W}(A) = \boldsymbol{W}(A) - \frac{diff(A,R_i,H)}{m} + \frac{diff(A,R_i,M)}{m}$
end
end
return $\boldsymbol{W}$ which evaluates the quality of the features

Mutual information: The mutual information is a statistical index that measures dependence between variables. It has been extensively used in filter methods to evaluate the relevancy of a subset of features in predicting the response variable and identify redundant variables. The mutual information can be used as a score for filter methods. Given two discrete random variables X and Y, the mutual information between two random variables can be calculated by

$$I(X;Y) = \sum\sum p(x,y) log \frac{p(x,y)}{p(x)p(y)} \tag{2.4}$$

where p is the probability density function.

If the random variables are continuous, the calculation replaces the summations by integrals and gives the mutual information as

$$I(X;Y) = \int\int p(x,y) log \frac{p(x,y)}{p(x)p(y)} dxdy \tag{2.5}$$

Mutual information is closely related to entropy. Generally, entropy measures the disorder or randomness of a system. For a random variable, entropy measures the associated level of uncertainty in it. The entropy of a random variable X can be calculated by

$$E(X) = -\sum_{x_i \in X} P(X = x_i) \cdot \log\left(P(X = x_i)\right) \tag{2.6}$$

When it comes to consider two random variables together, the uncertainty is measured by the

joint entropy:

$$E(X,Y) = -\sum_{x_i \in X}\sum_{y_i \in Y} P(X = x_i, Y = y_i) \cdot \log\left(P(X = x_i, Y = y_i)\right) \tag{2.7}$$

Given that the random variable Y is known, the conditional entropy of X can be calculated by

$$E(X|Y) = -\sum_{x,y} P(x,y) \cdot \log\left(P(x,y)\right) \tag{2.8}$$

The relation between entropy and mutual information is given by the following formulations:

$$I(X;Y) = E(X) - E(X|Y) \tag{2.9}$$

$$I(X;Y) = E(Y) - E(Y|X) \tag{2.10}$$

$$I(X;Y) = E(X) + E(Y) - E(X,Y) \tag{2.11}$$

$$I(X;Y) = E(X,Y) - E(X|Y) - E(Y|X) \tag{2.12}$$

To give a more direct understanding, the relation between entropy and mutual information is illustrated by Figure 2.5.

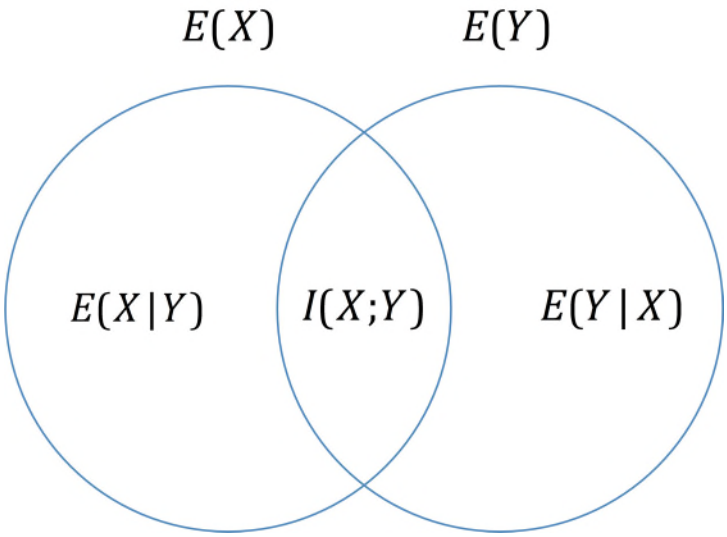


Figure 2.5 Relationship between entropy and mutual information

Pearson correlation coefficients: In statistics, Pearson correlation coefficient is used to measure the linear correlation between two variables. It is widely used in feature selection to identify highly correlated features. Highly correlated features show more linear dependency between them, hence showing close impact on the dependent variable. It is a common practice in dependency-based feature selection method to drop highly correlated features. Pearson correlation coefficient is defined as the covariance of the two variables divided by

the product of their standard deviations. Given two random variables X and Y, their corresponding Pearson correlation coefficient can be calculated by

$$\rho_{X,Y} = \frac{cov(X,Y)}{\sigma_X \sigma_Y} \tag{2.13}$$

where $cov(X,Y)$ is the covariance of the two random variables, σ_X and σ_Y are the standard deviation of X and Y respectively.

Pearson correlation coefficient ranges from -1 and 1. The value reflects the direction and strength of the correlation between two variables, as shown by Table 2.2 below.

Table 2.2 Interpretation of Pearson correlation coefficient

Pearson correlation coefficient	**Correlation type**	**Interpretation**
[-1,0)	Negative	When one variable changes, the other one changes in the same direction
0	None	The two variables have no correlation
(0,1]	Positive	When one variable changes, the other variable changes in the opposite direction

Filter methods are known to be very computational efficient compared to other methods and therefore they can be easily scale up to large dataset [25]. Filter methods are independent of any learning algorithms so that the bias in the feature selection process does not correlate with the bias in the learning process, hence preserving a better generalization property. The major disadvantages of the filter approaches are that they fail to consider the effect of feature dependencies. In other words, filter methods fail to consider the situation where some features may have little impact as an individual but big predictive power when they are combined together. More importantly, it ignores the biases of the forecasting models [18], leading to varied performance when the selected features are applied to different learning models.

2.3.2. Wrapper Methods

Wrapper methods use a wide range of search algorithms to search the feature space for the optimal subset of features by comparing the predictive performance of a specific learning model using different combinations of features. The search algorithms used by wrapper methods can be broadly classified into three types, i.e., exhaustive search, heuristic search and random search [31].

Exhaustive search, such as breath-first search, systematically enumerate all possible subsets of features. The time complexity for a size of m features is $O(2^m)$ [18]. Hence, using an exhaustive search strategy is prohibitive unless m is small.

Instead, wrapper methods often resort to heuristic methods, including sequential forward selection and sequential backward selection [32]. The sequential forward selection algorithm starts from an empty set and add the feature which best improves our model at a time until the required number of features are added, or the addition of a new feature does not improve the performance of the model. The sequential backward selection algorithm starts with the universal feature set and removes one feature at a time whose removal results in lowest decrease of the performance of the model until the required number of features are eliminated or the removal of a new feature does not degrade the performance of the model. It should be noted that both two methods are greedy algorithms that are likely to fall into local optimum [33]. Feature 2.6 depicts a diagram of the two sequential selection methods.

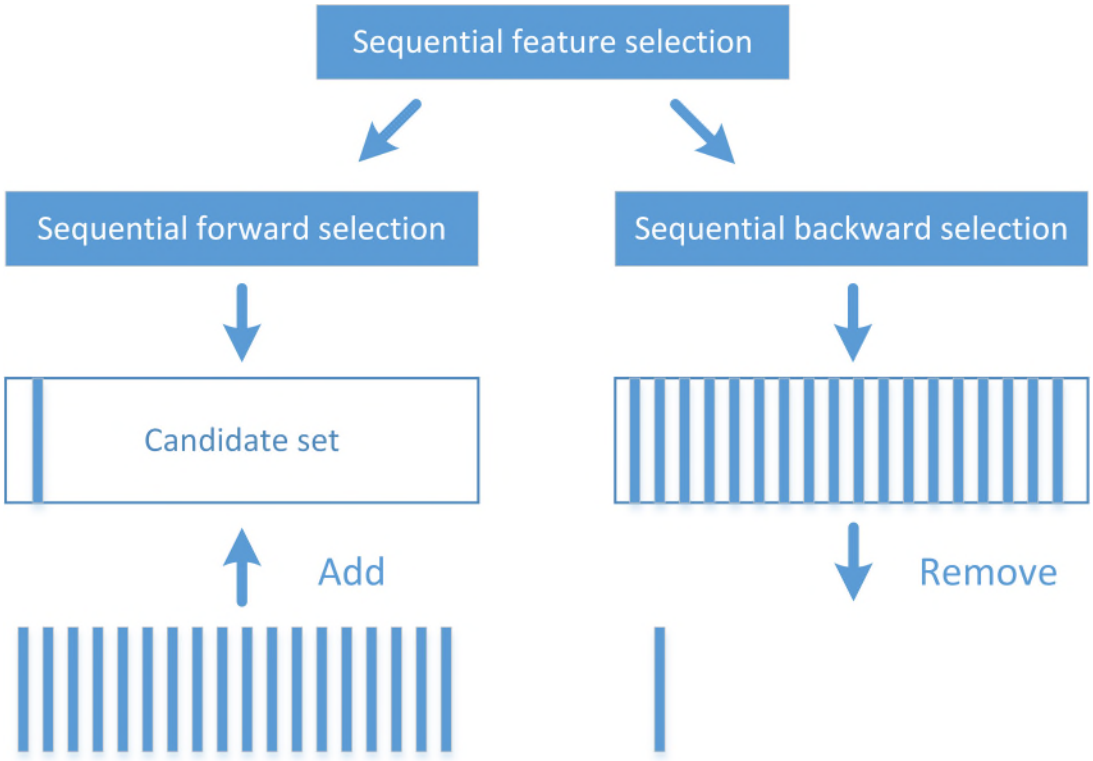


Figure 2.6 Diagram of sequential forward selection and sequential backward selection

Random search methods [34], [35] randomly select different subsets of features which are then fed to an induction algorithm to evaluate the performance. Random search methods can avoid being trapped in local optimum. However, random search methods are performed in a random manner and the results are difficult to reproduce. The following steps present the process of a general random search method.

1) Randomly generate an integer I between 1 and the total number of features
2) Randomly generate a sequence of I integers between 0 and $I - 1$ without repetition.
3) Use the generated sequence to select a subset of features. Train the model with this subset of features. Validate the model and save the value that represents the model performance.
4) Repeat the above steps based on the requirements of the algorithm.
5) Lastly, obtain the optimal subset of features that gives the best performance score.

Given a predefined learning model, the selection process of a wrapper method typically contains three parts:

1) searching the feature space to obtain a subset of features,
2) evaluating the performance of the learning model using the obtained subset of features,
3) repeating the above two steps until certain stopping criterion is met.

The feature selection process of a typical wrapper method is shown in Figure 2.7.

Although it has been empirically validated that wrapper methods generally outperform filter methods, they are criticized due to extremely high computation burden and can be intractable when coming across high-dimensional dataset.

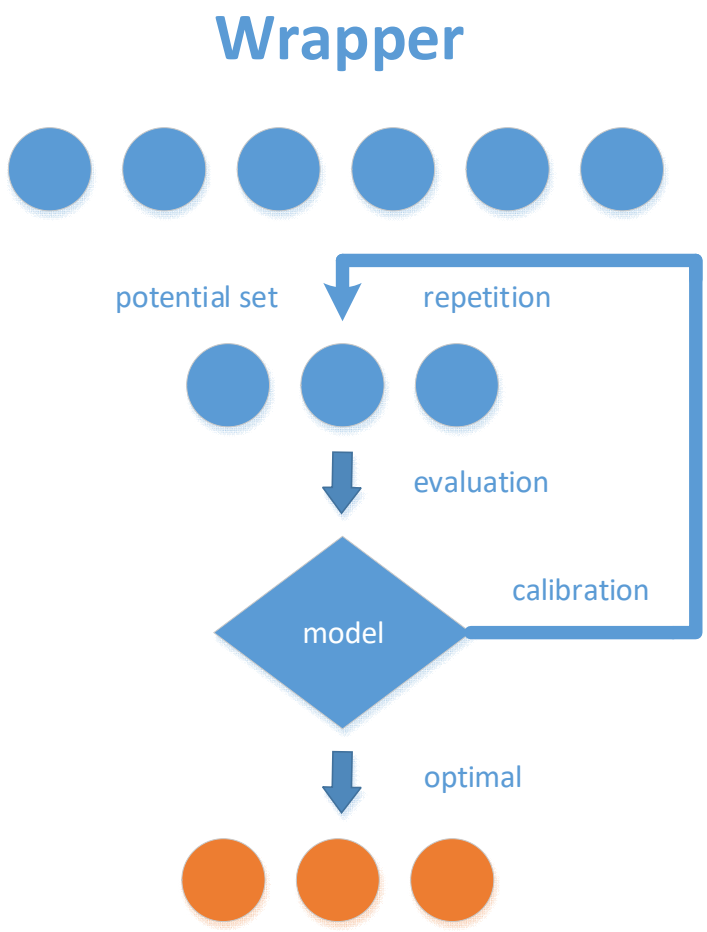


Figure 2.7 Schematic representation of wrapper method

2.3.3. Embedded Methods

Embedded methods incorporate the feature selection process as a part of the execution of the forecasting algorithm. The basic process of an embedded algorithm is depicted in Figure 2.8.

Embedded methods can basically be classified into three types, i.e., pruning methods, models with some built-in mechanisms and regularization methods [18]. Pruning methods attempt to remove features by setting the corresponding coefficients to zero while maintaining the performance of the model, such as recursive feature elimination for support vector machines [36]. This kind of technique begin by using the full set of features to establish a model and calculate a score for each feature. The model is then rebuilt after removing the least important feature and the feature scores are then recalculated. These steps are recursively performed until certain stopping criterion is met. This technique is quite efficient. However, not many models can be compatible with pruning methods. Also, pruning methods require that the model uses the full set of features at the beginning, which limits the scenarios where the methods can be used, such as when the number of features exceed the number of available samples. The most typical example of embedded method with a built-in mechanism for feature selection is decision tree-based algorithms [37]. During the induction of a decision tree, the algorithm calculates the feature importance and select a feature in each recursive step of the tree growth process. Hence, constructing a decision tree involves calculating the best predictive subset of features. An outstanding advantage of this type of methods is that they consider the interactions among the features. Usually, tree-based techniques allow the consideration of higher-order interactions [38]. However, the effectiveness and efficiency of these techniques significantly declines when the number of features grows [38]. Hence, many applications of tree-based algorithms focus on low-dimensional data. Another concern is that tree-based methods cannot automatically remove redundant features. For instance, the presence of redundant features can deteriorate the performance of a random forest [39]. Regularization methods optimize an objective function with a penalization term which forces the coefficients of several features to be very close to zero or exact zero. Then, the features whose corresponding coefficients are close or equal to zero are removed and the rest are selected. The most common examples of regularization methods are the LASSO regression [40] and the ridge regression [41]. LASSO regression technique

performs L1 regularization that adds penalty equivalent to the magnitude of coefficients. This algorithm sets the less relevant to zero or almost zero to respect the constraint. Similarly, ridge regression technique performs L2 regularization that adds penalty equivalent to square of the magnitude of coefficients. Unlike tree-based methods, these methods can eliminate redundant features. However, there is no built-in mechanism of detecting feature interactions. To solve this problem, interaction terms of the features are usually explicitly added in the analysis [42].

Embedded methods avoid the disadvantages of both the filters and wrappers as they not only consider the feature dependencies and the interaction with the forecasting algorithm, but also far more computational efficient than wrapper methods. Table 2.2 compares the commonly used feature selection methods.

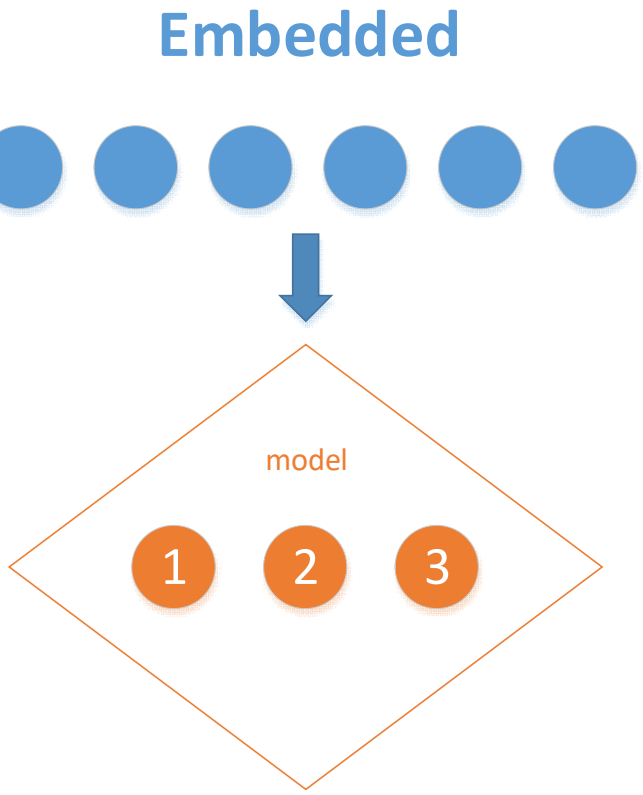


Figure 2.8 Schematic representation of embedded method

2.3.4. Hybrid Methods

Hybrid techniques combine two or more feature selection techniques to achieve optimal results and have been extensively applied for feature selection. For instance, the most common hybrid method is the combination of filter and wrapper methods which starts with an initial filtering of features followed by a wrapper method for selection. Hybrid method attempts to inherit the advantages of multiple methods by combining their complementary strengths. Different evaluation criteria are used in different search stages to improve the efficiency and prediction performance.

Table 2.3 Comparison of commonly used feature selection methods

	Advantages	**Disadvantages**
Filter	Model independent Computational efficient Better generalization	Ignore feature dependency Ignore biases of the model
Wrapper	Consider feature dependency More accurate than filter	Computational expensive Prone to overfitting Model specific
Embedded	Consider feature dependency More computational efficient than wrapper Less prone to overfitting than wrapper	Model specific

2.4. Introduction of PLF

The unique feature of electricity determines that it cannot efficiently stored in a large quantity. Hence, to ensure a secure and reliable power supply, the utilities must keep the generated power to meet the demand at every single time. Otherwise, the power system may be unstable which will have a large impact on the economic and social stability. Over decades, load forecasting has been widely used as a tool to help utilities to efficiently and effectively schedule and dispatch resources in power systems.

Basically, load forecasting is defined as the prediction of future load for a certain period ahead on a given system. The expected values given by load forecasting play an essential role in the decision-making process of the utilities. These values are usually point ones, which in other words means that one single value is generated at each time step. We call this kind of forecasting method point load forecasting, which gives an expected value at each forecasted time step. Point load forecasting has been widely studied and applied to many of the applications since the early time of power system. In recent years, the traditional power industry has been going through a significant transition process to serve the modern power grid, point load forecasting cannot meet the needs of operation and planning any longer. It is becoming more and more unreliable because a diversity of cutting-edge technologies are introduced to the power system. The installation of distributed energy resources and distributed energy storage systems brings significant uncertainties on the generation side, while the rapid increasing penetration of plugin electric vehicles, and dedicated demand response programs designed for active consumers introduce great variety and volatility on the demand side. This issue is address by PLF, which has grabbed increasing attention in recent years.

To manifest uncertainty, PLF methods give the prediction in the form of predictive intervals (PIs), quantiles or whole PDF, which are more informative than point forecasts and therefore can enhance the decision-making process in operation and planning of the power system. Among these three types of probabilistic outputs, traditional PI methods need to assume the shape of the predictive density in advance based on prior knowledge. This kind of method is unrobust because the result is affected by the seasonality and volatility of the load [43]. On the contrary, the methods that give quantiles as outputs do not make any assumptions on the shape of the predictive

distribution, which are classified as nonparametric estimation methods. On top of that, density forecasting methods are even more powerful because they can provide more holistic and flexible information compared to the former two methods by constructing the complete predictive distribution. Hence, density forecasting methods are considered as the most complete form of probabilistic forecasting method [44]. However, most density forecasting methods also need to predefine the shape of the predictive distribution, which, however, may lead to unreliable results due to improper distributional assumptions. Alternatively, the predictive quantiles given by quantile forecasting methods can be used to estimate the predictive distribution given that a large set of quantiles are calculated, or be transformed to predictive densities through nonparametric techniques, such as KDE, to provide more comprehensive information for the decision-making processes.

Generally, PLF methods can be classified into direct methods and indirect methods. Direct methods, such as quantile regression and KDE, can directly generate PFs. Indirect methods generate PFs from point forecasts by modeling and simulating the residuals of the underlying point forecasts [45] or by feeding temperature scenarios to point forecasting models [12]. In the following subsections, a comprehensive literature review on the most widely used techniques, including quantile regression methods, KDE, residual simulation methods and scenario generation methods, is presented.

2.5. Review of State-of-Art PLF Techniques

The existing body of literature related to PLF is relatively restricted. The subsequent subsections will present a concise overview of the most commonly used techniques including quantile regression, KDE, residual simulation and scenario generation, along with their methodological progression.

2.5.1. Quantile Regression

Quantiles are widely used in statistics to evaluate the performance of a group. Consider a

physical test of a group of students. We define that the score of a student is located at the p^{th} quantile of all the test scores if the score is higher than the proportion p of the whole test group and lower than the proportion $(1-p)$. For instance, the median, quartiles, quintiles, and deciles are some commonly used typical quantile values, which equally divide the population into two parts, four parts, five parts and 10 parts, respectively. As an example, Figure 2.9 plots a standard normal distribution showing the quartiles. The quartiles consist of three quantiles, which divide the distribution into four sections. Each section has the same area size. In other words, the area below the PDF curve is the same in the four intervals $(-\infty, q_1)$, (q_1, q_2), (q_2, q_3), and $(q_3, +\infty)$.

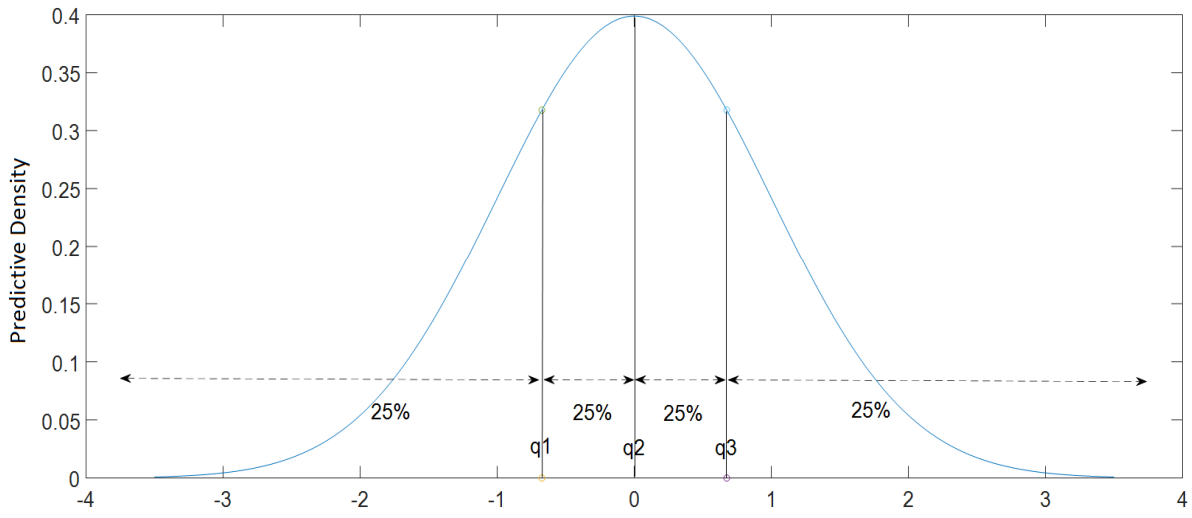


Figure 2.9 Quartiles of standard normal distribution

Quantile regression was first introduced by [46] in 1978 to extend the idea of quantiles to quantile regression. Quantile regression seeks to estimate the conditional quantile functions for models which map the relation of quantiles of the conditional distribution of the response variable and the predictors [47]. It has been extensively studied and applied in economics, and successfully introduced and applied in different time series forecasting problems, specifically load forecasting to quantify uncertainties.

To the best of the author's knowledge, the sharp rise in the attention to PLF in the literature happened in the Global Energy Forecasting Competition 2014 (GEFCom2014) [11], with quantile regression ranking in the top entries of PLF techniques. Since then, quantile regression has been

playing as the core methodology in developing various PLF techniques that are based on various statistical and artificial intelligent models. To highlight a few, [48] applies the quantile regression averaging method to a set of point forecasts generated from a statistical model, the naïve vanilla benchmark model [49] to obtain PFs. This method paves a way for bridging point load forecasting and PLF. Because the input of the quantile regression averaging model can be directly generated from point forecasting models, this method can be generalized to be combined with many point forecasting models and leverage the mature development in this area. Quantile regression can also be combined with machine learning algorithms. For example, [45] proposes to use the additive quantile regression model to forecast quantiles of the distribution of the future load. The model is estimated using the gradient boosting algorithm which is a popular machine learning approach that develops an accurate model by combining and converting a set of weak learners and is able to handle large and complex dataset. The gradient boosting algorithm is widely used because of its flexibility, accessibility, and robustness. However, it suffers from high computational burden, especially when training multiple models for a series of quantiles in quantile regression problems. To make the training cost affordable, [50] modifies the traditional quantile regression neural network by leveraging deep learning techniques. These techniques include batch training, early stopping, dropout, and noise layers, which can significantly reduce the computational burden when dealing with large dataset and can also improve the learning and generality of forecasting models. In the very recent years, deep neural networks have achieved significant advancement and researchers have been trying to add this new topic to the literature of load forecasting. A very recently published article [44] proposes to use a deep neural network to learn a fully parametrized quantile function. This method parametrizes both the quantiles and the associated probabilities with the proposed deep neural network to retrieve the full conditional distribution of the load.

The research proposed in this thesis is also based on quantile regression. A detailed discussion of the basics of quantile regression is given in Chapter 3.

2.5.2. Kernel Density Estimation

KDE gives a way of estimating an unknown PDF underlying a dataset. KDE belongs to the

family of non-parametric statistics. It is popular in density estimation because it can estimate the distribution of a continuous variable without relying on any parametric assumptions. As a non-parametric estimator, KDE does not rely on any fixed structure or functional form. The density shape can be automatically learned based on all the sample data.

Formally, let $X_1, \dots, X_n \in \mathbb{R}$ denote a set of random samples that are independently and identically drawn from an unknown distribution with a density function f. The PDF of f can be estimated by KDE and given as

$$\hat{f}_n(x) = \frac{1}{nh}\sum_{i=1}^{n} K(\frac{x - X_i}{h}) \tag{2.14}$$

where K denotes a smooth function which is also known as kernel function, and h is a positive real value which denotes the smoothing bandwidth. More intuitively, a kernel function is basically a weighting function, and the smoothing bandwidth is the width of the kernel function. The density at point x is estimated by taking the local density of the sample points within the distance h. A specific example is that the estimated PDF will look like a histogram if the data points within the distance are assigned with equal weight. The kernel function K is used to control the weights. when the distance between X_i and x increases, the associated weight will decrease towards zero. Therefore, the estimated PDF will have a large density value if the neighborhood has many observations, whereas a small density value is estimated if the neighborhood has only a small number of observations. To show the effect of the choice of kernel function on the estimation, we plot the shape of the estimated PDF for the same dataset using different kernel functions in Figure 2.10. It can be directly seen from Figure 2.10 that each density curve varies slightly but overall comparable, except that the curve given by box kernel function is not as smooth as others. The smoothness of the density curve is controlled by the bandwidth through changing the value of h. As an example, we plot three different density curves with different bandwidth values using a Gaussian kernel function, as shown in Figure 2.3. The optimal value for estimating a normal distribution is set as the default bandwidth. It can be inferred from the figure that a different bandwidth value can greatly affect the shape of the estimated density curve.

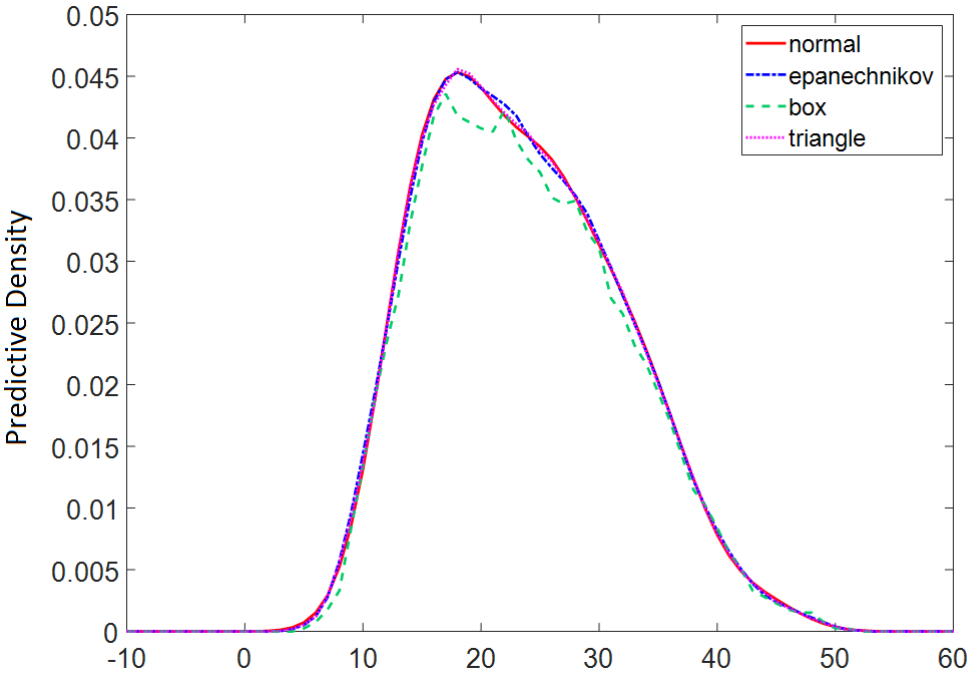


Figure 2.10 Density plot with different kernel functions

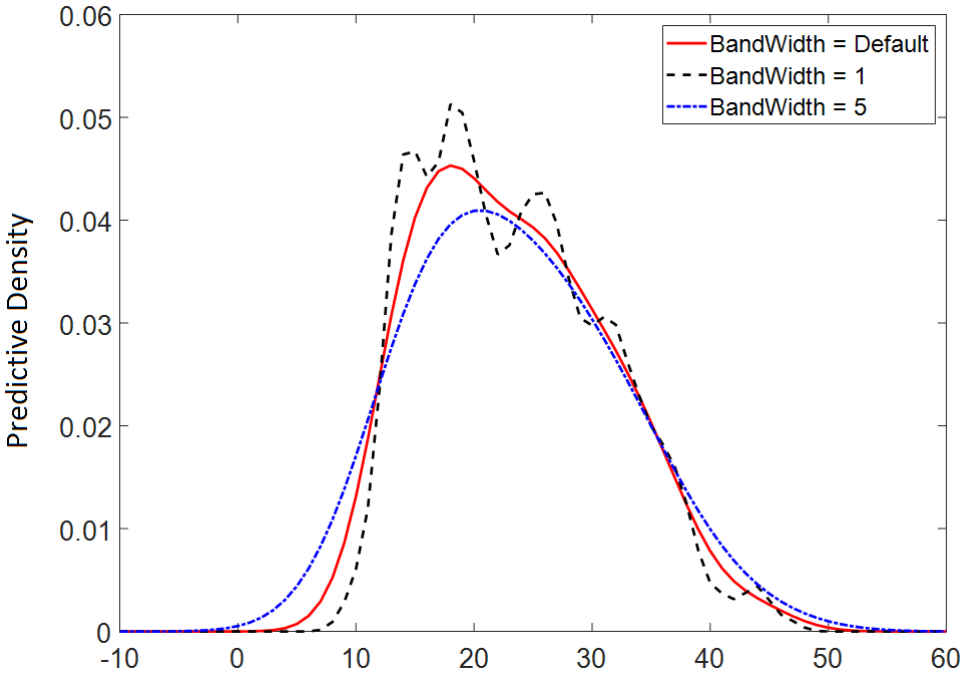


Figure 2.11 Density plot with different values of bandwidth

KDE is usually used as the final step to produce PFs given a set of conditional samples. In this case, KDE is only used as an auxiliary unconditional estimator. For example, the outputs of quantile forecasts are usually discrete, which can be used by KDE as inputs to recover continuous density curve of the forecasts [44], [51]. Applications of conditional KDE in PLF have also been studied in recent years. For instance, [14] makes a comprehensive comparison of the forecasting accuracy of KDE methods conditional on different input features including day-of-the-week, time-of-the-day, load of previous hours, etc. Similar works can also be found in [52], [53], which leverage conditional KDE to map the relation between exogenous predictors and the full predictive density of the load.

2.5.3. Residual Simulation

In the area of statistics, the residual is defined as the difference between the observed value and the estimated value of the response. Its definition can be easily confused with the definition of the term, error, which is defined as the difference between the deviation of the observed value from the real value, which is not necessarily observable.

Ideally, the residuals can be represented by a random noise, the distribution shape of which is close to a symmetrical bell with its peak centered at zero. This indicates that the given model is a good fit and unbiased without any unmodeled discernible patterns. However, the situation is quite different and complex in practical. There may exist trends, bias or even seasonality that the model fails to capture. Therefore, residual analysis has gained increased attention and been extensively used in statistics to facilitate the works of model validation. A typical practice is to directly model and forecast the temporal structure of the residuals to improve the model performance.

A majority of the research in the area of load forecasting assume the residual distribution of the PFs to be normally distributed. To answer a series of fundamental questions related to normality assumptions, [54] has carried out a comprehensive examination to investigate normality assumption and its implications in residual modeling for PLF. However, the simulation results show that none of the chosen residual series successfully passed the Kolmogorov–Smirnov test given certain significance level and critical value, indicating that such assumption is not reliable for PLF. It is also worth noting from the paper that the performance of deficient models can benefit

from adding residuals simulated from a normal distribution. However, this method is not suitable for models with more predictive power.

In fact, load forecasting residuals do not necessarily follow any well-defined parametric distribution because the load and the exogenous features are correlated in a very complex manner [55]. Many efforts have been made in the literature to avoid relying on unverified distribution assumption on load forecasting residuals. A typical example of these non-parametric methods is using quantile regression to model the residuals. Thereafter, these residuals can be further integrated with point forecasts to produce PFs. A typical relevant work is [15], which leverages quantile regression to model the PLF residuals with the point forecasts used as an input, and then combines the point forecasts with the conditional distribution of the residuals together to generate the final PFs.

2.5.4. Scenario Generation

In comparison with other PLF methods, scenario generation is more commonly accepted and widely applied in practice due to its simplicity and interpretability. This method is basically implemented in a manner of two steps. First, the input variables are simulated to generate a series of different scenarios. Thereafter, the generated scenarios are used as inputs by a point forecasting model to produce several point forecasts which are then used to estimate the final PFs. A schematic view of this approach is illustrated in Figure 2.12. Because calendar variables (month-of-the-year, day-of-the-week, hour-of-the-day) are fixed and the load demand is mainly driven by weather, many literatures of this topic make efforts in simulating temperature information to generate different input scenarios. There have been many techniques in the literature which can be basically classified into four typical categories: fixed-date, shifted-date, bootstrap and surrogate methods [17], as introduced below.

1) Fixed-date method: this method assigns the temperature profile of a past period date by date and then creates a temperature scenario of a future period.
2) Shifted-date method: this method generates scenarios by shifting the temperature profile of a past period forward or backward by one or more days.
3) Bootstrap method: this method divides the temperature profile of each past year into equal

length, and then drawn with replacement repeatedly to produce a new profile.

4) Surrogate method: this method generates temperature profiles by taking the Fourier transform of the past temperature series. This method can keep the information of the distribution and autocorrelation of the original temperature series.

These methods are comprehensively compared and evaluated in [17] and a practical guideline for model selection when using these methods is also proposed.

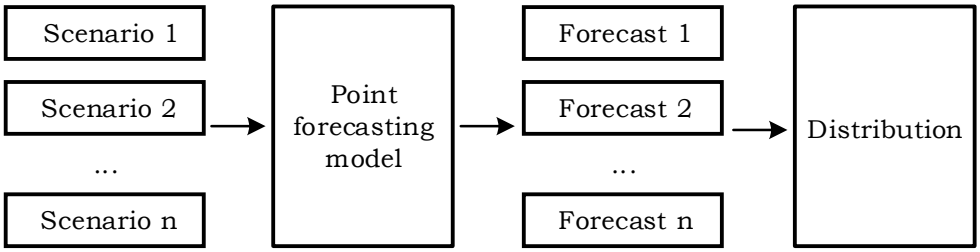


Figure 2.12 Schematic view of a typical scenario generation method

2.5.5. Other Techniques

PLF methods can generally be classified into two strands. One strand, such as quantile regression and KDE, can directly estimate PFs, while another strand, including residual simulation and scenario generation methods, extends point forecasts to PFs.

Research is not simply constrained within these two categories. To achieve better performance, more efforts have been made to enlarge the PLF literature, including but not limited to developing hybrid models, combining multiple forecasts, or adopting new techniques. [56] develops a hybrid method by using residual simulation as a post-processing step to improve performance of a scenario generation method. The author of this thesis has also published a novel hybrid method that combines Bayesian inference, MCMC and ensemble learning together to generate PFs in distribution networks [55]. Instead of using quantile regression for one single model, [48] applies an averaging technique to a set of point forecasts generated from the same family of quantile regression models, which are obtained by varying input features via feature selection. Such averaging mechanism can significantly reduce the risk of making poor decisions over model selection, and therefore can enhance the performance of quantile forecasts. Stochastic processes are

well documented in the literature to address problems related to uncertainties. [57] establishes a nonlinear quantile regression model by integrating Gaussian process into quantile regression and has reached satisfactory forecasting performance. However, this method suffers from high computation burden brought by the random process. To reduce computation complexity, [58] gives a sparser solution by introducing a heteroscedastic Gaussian process model using $l_{1/2}$ regularization. In recent years, continuous and consistent efforts have been made to enrich the limited literature on PLF. It is believed that PLF is a timely topic for the time being and will still be in the future.

2.6. Discussion

The current literature adequately covers the individual topics of feature selection and PLF. However, the literature on the combination of these two topics, feature selection for PLF, is limited. The existing practices in this field typically rely on heuristic methods such as filter or wrapper methods, which use a point error measure for variable selection and may not be suitable for probabilistic models. To inherently capture the uncertainty while doing feature selection, the feature selection process should rely on a probabilistic error measure that is consistent with the final probabilistic error measure of the forecasting model. To address this gap in the literature, we propose an embedded feature selection method for PLF that overcomes the limitations of existing methods. Our approach facilitates the modeling of complex uncertainties, handles sparse feature spaces, and yields more interpretable feature selection outcomes. Detailed information about the capabilities of our method is presented in the following chapter.

2.7. Summary

In the first half, this chapter first briefly introduces the general concept of feature selection. Then, a brief review of the state-of-art feature selection techniques is provided. More efforts are given to the supervised feature selection methods, as load forecasting is typically a supervised

learning problem. Supervised feature selection methods include filter methods, wrapper methods, embedded methods, and hybrid methods, each of which has been discussed in detail in this chapter. An overall comparison between these methods is given at the end.

In the second half, a comprehensive review of the state-of-art PLF techniques is presented. More efforts are made to the most widely used methods in this chapter, i.e., quantile regression, KDE, residual simulation, and scenario generation methods. Other practices including developing hybrid models, combining multiple forecasts, or adopting new techniques are also introduced with a few typical works highlighted.

3. Proposed Embedded Feature Selection Method for Probabilistic Load Forecasting

3.1. Introduction

In this chapter, the proposed embedded feature selection method via Bayesian quantile regression is presented. Firstly, Chapter 3.2 introduces the technical background including fundamentals for quantile regression, the linear model used for quantile regression, the corresponding features, as well as the evaluation criteria for PLF. Then, Chapter 3.3 specifies the proposed embedded feature selection method following the framework of Bayesian inference, including prior specification and posterior inference by Gibbs sampling, which is an MCMC technique.

3.2. Predictive Model and Evaluation Criteria

In the proposed method, quantile regression is adopted as the base predictive model, as it has been the approach of great theoretical interest as well as plenty of practical applications in the context of PLF, with competitive forecasting performance well documented in the literature. To set the scene for the following chapters, this section gives a brief description of quantile regression, the adopted linear model, as well as the evaluation criteria for PLF.

3.2.1. Quantile Regression

Load forecasting is basically a regression problem based on historical load records and relevant

variables (weather conditions, calendar effects, etc.). In general, the problem can be formulated by

$$y_t = g(\boldsymbol{x}_t) + \varepsilon_t \tag{3.1}$$

where y_t denotes the load at time t, $\boldsymbol{x}_t$ is a vector of features for the observation at time t, and ε_t is the corresponding random error term at time t with mean zeros and constant variance. In the case that mean regression is applied, the problem reduces to the estimation of the conditional expectation of the response given the assumption on the error term. The problem of PLF is basically the estimation of the conditional distribution of the response, which can be achieved by moving from mean regression to quantile regression.

In this paper, the linear quantile regression is used as the probabilistic forecasting model to estimate the conditional quantiles of the load. Quantile regression has emerged as a prevalent technique in developing various PLF methodologies using statistical and artificial intelligence models. A noteworthy advantage of quantile regression is its nonparametric nature, allowing it to make no assumptions about the shape of the predictive distribution. Conversely, parametric methods rely on predefined distributions, which may not be a good fit of the data, and improper distributional assumptions may deteriorate the accuracy of the result. Additionally, the predicted quantiles can be used to retrieve the full predictive distribution through some estimation techniques such as kernel density estimation, enabling more comprehensive information to be available for the decision-making process.

Quantiles are defined as points dividing a sample into equal-sized groups, for example, the median is the 0.5$^{\text{th}}$ quantile showing the central location of the entire sample. Formally, the p^{th} quantile denotes the value below which the proportion of the data points to the entire population is q. A quantile is a continuous value between the range [0,1], so any position of a distribution can be calculated given the sample and a predefined quantile. Formally, given the linear regression model by letting $g(\boldsymbol{x_t}) = \boldsymbol{x_t^T \beta}$ in (1) where $\boldsymbol{\beta} = (\beta_0, \beta_1, \dots, \beta_D)$ is a vector of coefficients with D denoting the dimension of the input features and β_0 being the coefficient for the intercept term, i.e.,

$$y_t = \boldsymbol{x}_t^T \boldsymbol{\beta} + \varepsilon_t \tag{3.2}$$

the p^{th} conditional quantile of y_t given x_t can be expressed by

$$Q_p(y_t|\boldsymbol{x}_t) = \boldsymbol{x}_t^T \boldsymbol{\beta}_p \tag{3.3}$$

where $\boldsymbol{\beta}_p = (\beta_{0p}, \beta_{1p}, \dots, \beta_{Dp})$ is a vector of coefficients dependent on the p^{th} quantile of the random error term ε_t. The estimation of $\boldsymbol{\beta}_p$ can be obtained by solving the following minimization

problem,

$$\widehat{\boldsymbol{\beta}}_p = \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \sum_t \rho_p(y_t - \boldsymbol{x}_t^T \boldsymbol{\beta}_p) \tag{3.4}$$

where the loss function is given by

$$\rho_p(\theta) = \theta(p - I(\theta < 0)) \tag{3.5}$$

where $I(\cdot)$ denotes the indicator function. The function $\rho_p(\cdot)$ is the tilted absolute value function which is plot in Figure 3.1.

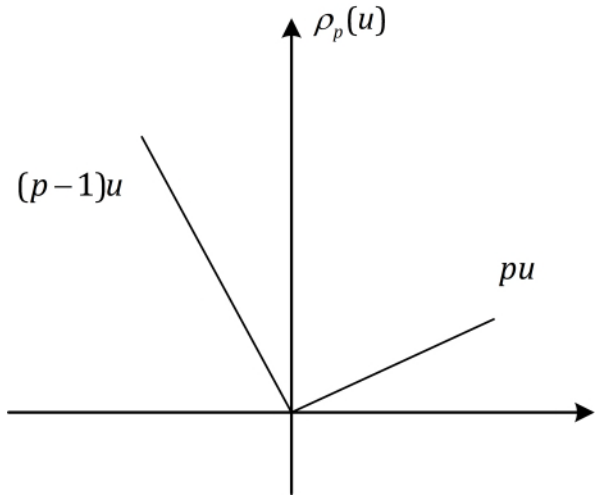


Figure 3.1 Tilted absolute value function

Instead of solving the above linear programming problem, this paper utilizes Bayesian inference as the model estimation method through which the proposed embedded feature selection technique is developed. The selection of Bayesian inference as the method for estimating model parameters is motivated by several factors. Firstly, it is through the use of Bayesian inference that our proposed method can integrate a feature selection process into the PLF model. This allows the selected features to vary across different quantiles and enables all features to be considered for their impact. Additionally, the selection of appropriate prior distributions removes the assumption of normality on the error term and provides a solution for handling sparse input feature spaces. The details are discussed in Chapter 3.3.

3.2.2. The Naïve Vanilla Benchmark Model

This research uses a multiple linear regression model with the consideration of recency effect as the base predictive model. This model is an extension of naïve vanilla benchmark model, which was first proposed by [59] and was ranked in the top 25% among over 100 teams in the GEFCom2012 where it was used to generate benchmark scores. Naïve vanilla benchmark model takes into account the impact of local trend, temperature, calendar variables and interaction effects between the temperature and the calendar variables, which can be formulated as

$$y_t = \beta_0 + \beta_1 Trend_t + \beta_2 M_t + \beta_3 W_t + \beta_4 H_t + \beta_5 W_t H_t + f(T_t) \tag{3.6}$$

$$f(T_t) = \beta_6 T_t + \beta_7 T_t^2 + \beta_8 T_t^3 + \beta_9 T_t M_t + \beta_{10} T_t^2 M_t + \beta_{11} T_t^3 M_t + \beta_9 T_t H_t + \beta_{10} T_t^2 H_t + \beta_{11} T_t^3 H_t \tag{3.7}$$

where M_t, W_t and H_t denote the month-of-the-year, day-of-the-week and hour-of-the-day categorical variables corresponding to time t, respectively. The trend variable is defined by assigning a natural number to each hour in ascending order. Each component of the model is added for a reason, which is explained in turn as follows.

1) Trend

   This quantitative variable is defined for the entire dataset to capture the increasing trend beneath it [33]. Each data point is assigned by a natural number in a natural order. For example, the value of the trend variable of the first hour of the whole dataset is 1, followed by the second hour assigned by 2, and so forth. It is usually true that when the local economy of a territory is in good health and the electricity service of the local utility is stable, there will be a mild increasing trend behind the electricity consumption of this territory. However, this assumption is not valid when there is a significant change in the economic pattern during the period, such as a great recession or a big boom in the local economy. Service changes, such as the merge of two utilities or the split of a utility, will also make this assumption invalid. In this thesis, we consider only the scenarios where the assumption of an increasing trend is valid.

2) Hour-of-the-day, day-of-the-week, month-of-the-year

   Day, week and year are the three main seasonal blocks in a load series [49]. For each block, the treatment can be different based on the consumption behavior of a certain service region. Take the modeling of week as an example. One week is composed of 7

days, which can usually be divided into weekdays and weekends. A more precise practice is to have three categories, weekdays, Saturday, and Sunday, while the highest resolution of modeling a week is to clearly define each day of the 7 days, i.e., Monday to Sunday. When dealing with datasets from different countries, we need to pay attention to the local customs because some countries take days other than Saturday and Sunday as weekends. Naïve vanilla benchmark model treats all the seasonal blocks using their highest resolution, i.e., day being modeled by 24 hours, week being modeled by 7 days, and year being modeled by 12 months.

3) Temperature

   It is well known that temperature has a large impact on electric load consumption patterns. Apparently, the load that is mostly affected by temperature is air conditioning system, which accounts for one of the largest electricity consumptions in a residential household. In different seasons, an air conditioning system behaves different because they work in different modes, heating, cooling, drying, etc., resulting in different consumption patterns. Besides of the direct impacts of temperature on different loads, temperature also implicitly changes our lifestyles. For example, in Canada, people go for outdoor activities in summer, while indoor activities take most of Canadians' wintertime. This also determines the load profile of a service territory. In naïve vanilla benchmark model, temperature is modeled by 3^{rd} ordered polynomial function.

4) Interaction effects

   Interaction effects between the above-mentioned terms are also important factors that affect the load patterns. It is commonly known that an afternoon in June is much warmer than an afternoon in December. Also, the consumption activity during afternoons at weekends would be different with that during weekdays. Hence, it is important to consider the interaction effects among the terms, including the interaction effect between temperature and the calendar variables, and the interaction effect among these calendar variables, which are added in the naïve vanilla benchmark model.

Illustrative plots of a public dataset are given in Chapter 4 to give an intuitive understanding of each component of the model.

3.2.3. Multiple Linear Regression Considering Recency Effect

However, it has been validated that the features included in the naïve vanilla benchmark model does not suffice for a load forecasting problem. The model does not consider the impact of temperatures from preceding hours, the absence of which can cause the following discrepancies between the forecasts and the actual loads [59]:

1) the model over/under-forecasts the peak loads for consecutive days in different seasons;
2) the forecast leads/lags the actual load for consecutive hours on several days.

In this regard, such effect is captured by including the temperatures of preceding hours in the naïve vanilla benchmark model, which gives a multiple linear regression model with the following formulation:

$$y_t = \beta_0 + \beta_1 Trend_t + \beta_2 M_t + \beta_3 W_t + \beta_4 H_t + \beta_5 W_t H_t + f(T_t) + \sum_{d=1}^{N_D} f_r(\tilde{T}_{t,d}) + \sum_{h=1}^{N_H} f_r(T_{t-h}) \tag{3.8}$$

$$f_r(T_t) = \beta_6 T_t + \beta_7 T_t^2 + \beta_8 T_t^3 + \beta_9 T_t M_t + \beta_{10} T_t^2 M_t + \beta_{11} T_t^3 M_t + \beta_9 T_t H_t + \beta_{10} T_t^2 H_t + \beta_{11} T_t^3 H_t \tag{3.9}$$

$$\tilde{T}_{t,d} = \frac{1}{24} \sum_{h=24d-23}^{24d} T_{t-h} \qquad d = 1,2,\dots,N_D \tag{3.10}$$

where N_D and N_H denote the number of days and hours of the lagged temperature.

This effect is referred to as recency effect, which is a cognitive concept that recent events, facts, information, impressions, or other items, are more favored than historical ones. This concept can also be introduced to electric load forecasting. Similarly, a load would probably tend to memorize recent temperatures. In other words, the temperature of preceding hours can have impacts on the current load. This is true because people may need time to react to temperature changes, resulting in a lagging between the temperature and the consumption activities.

3.2.4. Converting Categorical Variables into Numerical Features

In the above-mentioned model, hour-of-the-day, day-of-the-week and month-of-the-year are basically categorical variables. Although categorical variables are represented by numbers,

however, unlike numerical variables, they cannot be entered into the regression model directly and must be recoded. The number assigned to each variable only refers to the category it belongs to. There are a variety of coding methods that can be used for numerical coding of categorical variables, such as dummy coding, sum coding, deviation coding, etc. In the case of load forecasting, the calendar effects are basically nominal variables that can be well coded by the dummy coding scheme. Hence, the dummy coding method [60] is utilized to convert the calendar variables into a series of numerical features that the model can understand.

As the simplest and most frequently used coding scheme, dummy coding method recodes a categorical variable into a series of dichotomous variables that only take the value of 1 or 0. Hence, dummy variables are also called "binary flag variables". Before we formally introduce the working mechanism of dummy coding method, we should first be aware of the dummy variable trap, which is also called the situation of perfect multicollinearity.

Multicollinearity happens when the model contains some predictors are correlated not only to the response but also to other predictors, resulting in redundancy in the predictors. Multicollinearity will lead to incorrect coefficients of the regression model and hence the results are not acceptable. Consider a multiple linear regression model that take the categorical variable gender as the explanatory variables which are coded with two dimensions:

$$y = \beta_0 + \beta_1 \cdot x_{male} + \beta_2 \cdot x_{female} + \epsilon \tag{3.11}$$

where y is the response variable, x_{male} and x_{female} are the explanatory variables, β_0 is the intercept, β_1 and β_2 are the coefficients, and ϵ denotes the error term. It is obvious that the two dimensions x_{male} and x_{female} are perfected correlated because one would either be male or female from the biological aspect. Hence, we can replace x_{female} with $(1 - x_{male})$ in equation 3.11, yielding

$$\begin{aligned} y &= \beta_0 + \beta_1 \cdot x_{male} + \beta_2 \cdot (1 - x_{male}) + \epsilon \\ &= (\beta_0+\beta_2) + (\beta_1 - \beta_2) \cdot x_{male} + \epsilon \end{aligned} \tag{3.12}$$

In this way, equation 3.11 is rewritten using only one variable x_{male}, as shown by equation 3.12. The new regression coefficients to be estimated are now $(\beta_0+\beta_2)$ and $(\beta_1 - \beta_2)$, which can be replaced by two new coefficients. This simple example shows the scenario of a categorical variable with only two categories. In the scenario of more than two categories, we can execute

the following steps to validate if a given dataset suffers from dummy variable trap or not.

1) Define the given dataset as a matrix X;
2) Obtain the transpose of X as X^T;
3) Calculate the dot product of X and X^T, i.e., $X^T X$;
4) Calculate the determinant of $X^T X$, $|X^T X|$;
5) If the determinant $|X^T X|$ is zero, then the dataset will have the dummy variable trap, otherwise there will be no dummy variable trap in this dataset.

Take the variable day-of-the-week as an example. Consider the following coding scheme that we use 7 dimensions to represent a day in a week with its associated dimension set to 1 and other dimensions set to 0, as shown in Table 3.1. Assume that the regression model includes an intercept term, and we have a dataset with 7 instances, Monday, Tuesday, Wednesday, Thursday, Friday, Saturday and Sunday. Performing the validation steps above yields

$$X = \begin{matrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{matrix} \tag{3.13}$$

$$X^T = \begin{matrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{matrix} \tag{3.14}$$

$$X^T X = \begin{matrix} 7 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{matrix} \tag{3.15}$$

$$|X^T X| = 0 \tag{3.16}$$

The determinant is zero and the matrix is singular. Hence, the given dataset suffers from the dummy variable trap.

Table 3.1 A coding scheme for day-of-the-week that will have dummy variable trap problem

	Feature 1	Feature 2	Feature 3	Feature 4	Feature 5	Feature 6	Feature 7
Monday	1	0	0	0	0	0	0
Tuesday	0	1	0	0	0	0	0
Wednesday	0	0	1	0	0	0	0
Thursday	0	0	0	1	0	0	0
Friday	0	0	0	0	1	0	0
Saturday	0	0	0	0	0	1	0
Sunday	0	0	0	0	0	0	1

To address this problem, we drop one of the categories to avoid dummy variable trap. Here is a simple example to show how dummy coding works in practice. Assume that a categorical variable has only two level of categories, say, gender with the categories of male and female. We can easily create a single dummy variable to represent the two categories, with the male set to 1 and female set to 0. When a categorical variable has three or more categories, two or more dummy variables are required to code these categories. Generally, a categorical variable with K category levels is presented by $K - 1$ features. The variable at the reference level is coded as all 0s. For all the variables that are not at the reference level, each of the variables will be replaced by a new recoded variable that has a value of 1 at that level and 0 for other levels. The eliminated category that is assigned with no dummy variable is the reference category. All comparisons are made in reference to this category. If the intercept is not chosen as the reference category, then the value of the coefficient of the intercept will reflect the mean value of the reference category. In practice, the reference category is strictly up to the choice of the researcher. The regression coefficients associated to the dummy variables are the differential intercept coefficients which reflect how much the value of the category with a value of 1 differs from the reference category. For instance, the day-of-the-week variable is encoded as shown in Table 3.2.

Table 3.2 Encoding scheme for day-of-the-week variable using dummy coding method

	Feature 1	Feature 2	Feature 3	Feature 4	Feature 5	Feature 6
Monday	1	0	0	0	0	0
Tuesday	0	1	0	0	0	0
Wednesday	0	0	1	0	0	0
Thursday	0	0	0	1	0	0
Friday	0	0	0	0	1	0
Saturday	0	0	0	0	0	1
Sunday	0	0	0	0	0	0

However, such encoding scheme introduces great sparsity to the input data when many categorical variables are included. The massive increase in the number of dimensions may deteriorate the overall performance of the model. In this regard, a sparse feature selection method based on Bayesian quantile regression is proposed to address the problem and will be discussed in Section 3.3.

3.2.5. Evaluation Criteria

Evaluating the forecasting accuracy of PLF requires specific numerical measures. Comprehensive measures include Brier score [61], Winkler score [62], ranked probability score (RPS) [63], continuous ranked probability score (CRPS) [64], and quantile score [65], etc.

1) Brier score

   The Brier score is a metric that is used to assess the accuracy of PFs. It is defined as the mean squared difference between the forecasted probability and the real outcome. However, the Brier score can only be used for scenarios where the outcomes are binary and categorical and can be identified as true or false. It cannot be used for variables that can take on three or more values. Moreover, the outcomes must be mutually exclusive and assigned with probabilities which must sum to 1. A common formula of the brier score can be calculated as the mean squared error:

$$Brier\ score = \frac{1}{N}\sum_{t=1}^{N}(f_t - o_t)^2 \tag{3.16}$$

where f_t is the probability of the forecast at time step t, o_t is the outcome at time step t, and N is the number of forecasting instances. It can be easily inferred from the formula that a smaller Brier score indicates a better result. We can also infer that the value of a Brier score is limited within the range of [0,1]. Here is a simple example that shows how the Brier score works. Imagine that we are forecasting the probability P of that it will snow in October in Saskatoon, Canada. Based on equation 3.16, we can calculate the Brier score for the following scenarios:

- If the prediction is $P = 1$ and it snows, the Brier score is 0, which is the best score.
- If the prediction is $P = 1$ and it does not snow, the Brier score is 1, which is the worst score.
- If the prediction is $P = 0.5$, then no matter it snows or not, the Brier score is 0.25.
- If the prediction is $P = 0.8$ and it snows, the Brier score is 0.04.

When a scenario requires the evaluation of multi-category prediction, the above-mentioned Brier score cannot work anymore. Instead, we need to use the original definition given by Brier as shown below:

$$original\ Brier\ score = \frac{1}{N}\sum_{t=1}^{N}\sum_{i=1}^{Q}(f_{ti} - o_{ti})^2 \tag{3.17}$$

where f_{ti} is predicted probability of category i at time step t, o_t is the outcome of category i at time step t, Q is the number of total categories, N is the number of instances of all categories. It should be noted that for binary forecasting, the value given by the original Brier score is twice of the known Brier score.

2) Winkler score

Sometimes a probabilistic forecast can be given in the form of predictive intervals. The Winkler score is designed to evaluate this kind of outcome, which allows a joint assessment of the unconditional coverage and the width of the interval [10]. Formally, let $[l_{\alpha t}, u_{\alpha t}]$ denote the $(1-\alpha)\times 100\%$ prediction interval. The Winkler score is defined as

$$W_{\alpha,t} = \begin{cases} (u_{\alpha t} - l_{\alpha t}) + \frac{2}{\alpha}(l_{\alpha t} - y_t) & if\ y_t < l_{\alpha t} \\ (u_{\alpha t} - l_{\alpha t}) & if\ l_{\alpha t} \leq y_t \leq u_{\alpha t} \\ (u_{\alpha t} - l_{\alpha t}) + \frac{2}{\alpha}(y_t - u_{\alpha t}) & if\ y_t > u_{\alpha t} \end{cases} \quad (3.18)$$

The score can be interpreted as follows. It is exactly the length of the interval if the observation falls inside the interval. If the observation falls outside the interval, the score is defined as the length of the interval plus a penalty term which is proportional to the distance between the observation and the nearest edge of the interval. It can be easily inferred that the smaller the Winkler score, the better the prediction.

3) RPS

The RPS is a discrete metric that measures the accuracy of a probabilistic forecast of a categorical variable which is ranked or ordered. Hence, the categories that are measured have a discrete nature. Formally, the RPS can be calculated by the following formula:

$$RPS = \frac{1}{r-1}\sum_{i=1}^{r}(\sum_{j=1}^{i} p_j - \sum_{j=1}^{i} e_j)^2 \quad (3.19)$$

where r denotes the number of total outcomes, p_j is the predicted probability of the j^{th} outcome, and e_j is the actual probability of the j^{th} outcome. When $r = 2$, the RPS gives the Brier score. It can be easily inferred from the formula that the value of RPS lies in the interval of [0,1], and the smaller the value, the better the prediction.

4) CRPS

CRPS is used in the scenario where the observation is a scalar, and the prediction is a cumulative distribution function. It can be considered as a generalization of the RPS where the outcomes are continuous rather than discrete. Formally, the CRPS is defined as the integral of the difference between the cumulative distribution function $F(y)$ of the predicted density and the outcome y^*, as expressed by the following formula:

$$CRPS(F, y^*) = \int_{-\infty}^{+\infty} \big(F(y) - \mathbb{1}(y - y^*)\big)^2 dy \quad (3.20)$$

where $\mathbb{1}$ is the Heaviside step function. The value gives by the function is 1 if the real argument is non-negative, otherwise the value is 0, as expressed by equation 3.21.

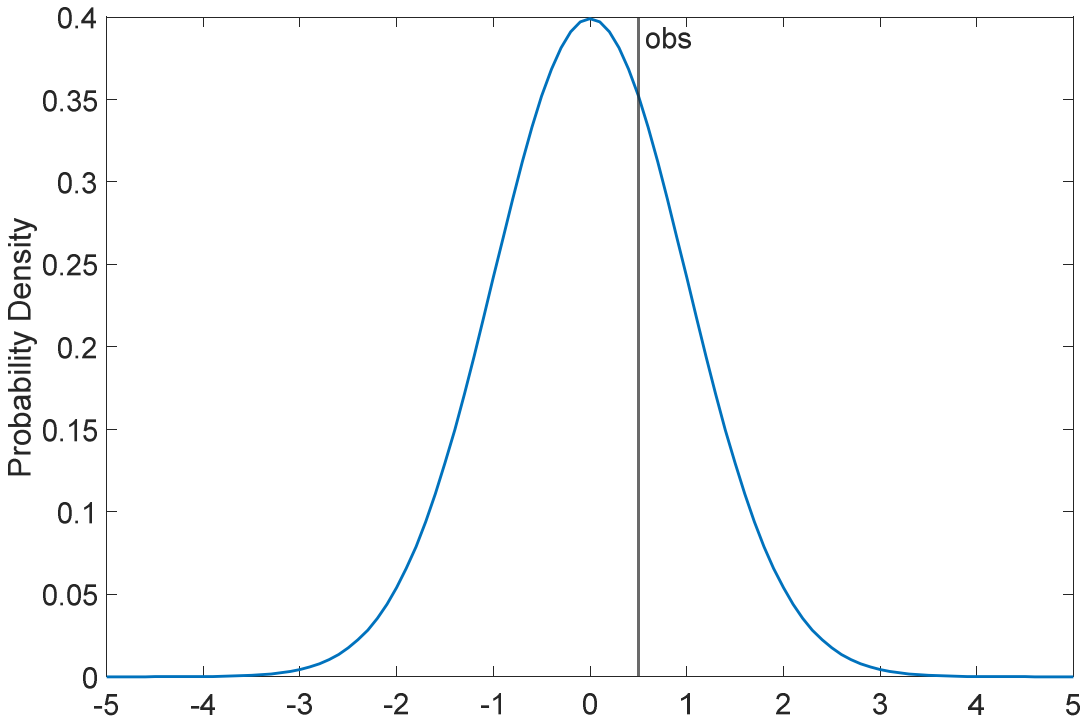
$$\mathbb{1}(x) = \begin{cases} 1 & , x \geq 0 \\ 0 & , x < 0 \end{cases} \quad (3.20)$$

A graphical illustration of the CRPS is plotted in Figure 3.2. The value of CRPS is the area of the shaded region. An alternative representation can be expressed by

$$CRPS(F, y^*) = E_F|Y - y^*| - \frac{1}{2}E_F|Y - Y'| \tag{3.21}$$

where Y and Y' are two independent random variables which have the same cumulative distribution function F. As shown by equation 3.21, the CRPS generalizes the mean absolute error to the case of PFs. In contrast with other probabilistic forecast measures, the CRPS considers the PFs as a whole, rather than just focus on certain points of the PFs. It quantifies both the calibration and sharpness [66] of the predictive distribution and hence provides a comprehensive evaluation of the result. By representing F through an L-ensemble $y_{i=1,\dots,L}$, equation 3.21 leads to the following estimator

$$\widehat{CRPS}_{NRG}(F, y^*) = \frac{1}{L}\sum_{i=1}^{L}|y_i - y^*| - \frac{1}{2L^2}\sum_{i,j=1}^{L}|y_i - y_j| \tag{3.22}$$



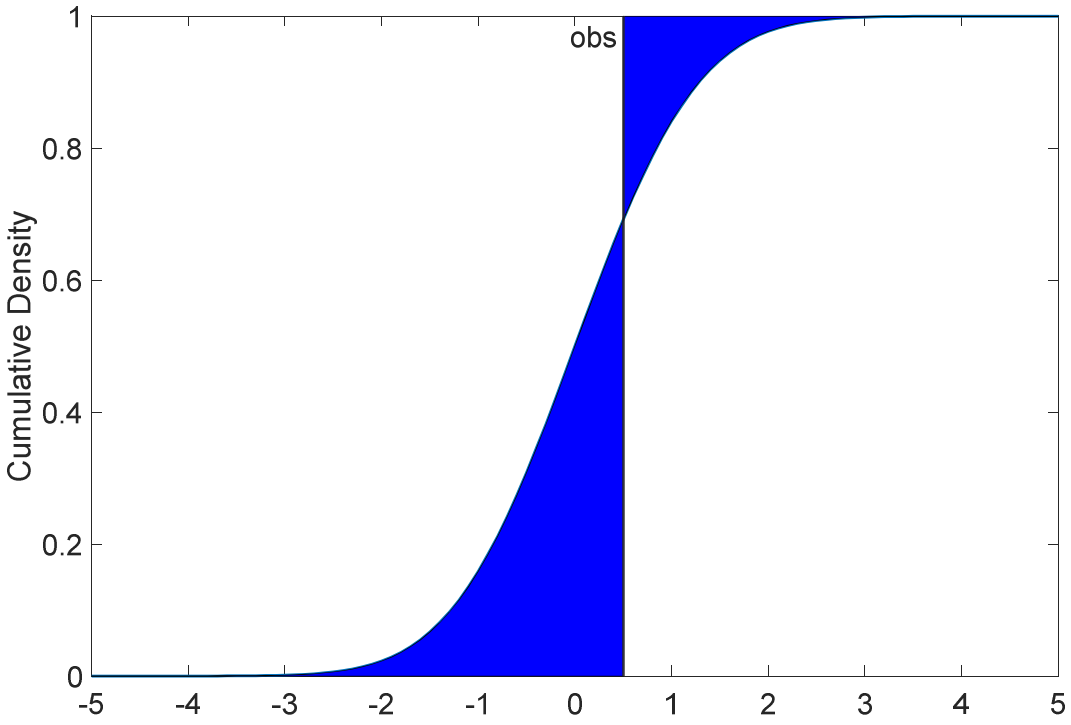


Figure 3.2 Graphic illustration of the calculation of CRPS

5) Quantile score

Because our predictive model gives the results in the form of quantiles, quantile score is used as the evaluation criteria for PLF. Quantile score was first introduced in the GEFCom2014 [65], since when it has been widely used by the load forecasting community because it considers both the sharpness and resolution in the evaluation. Formally, the quantile score is defined as the mean of the pinball losses across all the quantiles and all the forecasting horizon, where the pinball loss for each quantile and each time step is calculated by

$$Pinball(\hat{y}_{t,p}, y_t, p) = \begin{cases} \left(1 - \frac{p}{100}\right)(\hat{y}_{t,p} - y_t) & y_t < \hat{y}_{t,p} \\ \frac{p}{100}(y_t - \hat{y}_{t,p}) & y_t \geq \hat{y}_{t,p} \end{cases} \tag{3.23}$$

where $\hat{y}_{t,p}$ is the p^{th} quantile of the forecasted load at time t. A lower quantile score indicates better performance.

3.3. Feature Selection via Bayesian Quantile Regression

The proposed predictive model with embedded feature selection for PLF is constructed under the Bayesian framework which is introduced as the following subsections.

3.3.1. Bayesian Inference

To better understand the proposed framework, we first briefly introduce the Bayesian inference [67], which is a powerful statistical method to model random variables. Bayesian inference interprets probabilities as subjective believes. It aims to specify a procedure of updating one's belief upon seeing the data. When estimating a model, Bayesian statistics consider the model parameters as uncertain and drawn from some probability distributions. The essence of Bayesian inference is encapsulated by Bayes' Theorem of conditional probabilities:

$$p(\boldsymbol{\mu}|\boldsymbol{D}) = p(\boldsymbol{\mu})p(\boldsymbol{D}|\boldsymbol{\mu})/p(\boldsymbol{D}) \tag{3.24}$$

- D denotes the data.
- $\boldsymbol{\mu}$ denotes the parameter vector.
- $p(\boldsymbol{\mu})$ is the probability of the parameters without considering the data.
- $p(\boldsymbol{\mu}|\boldsymbol{D})$ is the probability of the parameters given the data.
- $p(\boldsymbol{D}|\boldsymbol{\mu})$ is the probability of the data given the parameters.
- $p(\boldsymbol{D})$ is the probability of data given any parameters.

Formally in Bayesian statistics, $p(\boldsymbol{\mu})$ is called prior distribution, $p(\boldsymbol{\mu}|\boldsymbol{D})$ is called posterior distribution, $p(\boldsymbol{D}|\boldsymbol{\mu})$ is called likelihood function and $p(\boldsymbol{D})$ is considered as a normalizing constant, which can also be called the evidence. The prior distribution is chosen based on our domain-knowledge of the problem to be solved and of the parameter to be estimated. This process is done without the knowledge of any sample data. The likelihood is calculated as the probability of observing the data given the prior hypothesis. The normalizing constant, $p(\boldsymbol{D})$, is used to ensure that the integral of the posterior distribution equals to one. However, the computation of this constant shows extremely high complexity. In practice, equation 3.24 is expressed in the following formula:

$$p(\boldsymbol{\mu}|\boldsymbol{D}) \propto p(\boldsymbol{\mu})p(\boldsymbol{D}|\boldsymbol{\mu}) \tag{3.25}$$

It can be easily seen that the posterior is proportional to the likelihood times the prior.

A common problem that we usually need to solve is to infer an unknown distribution from a given observed dataset. To address this problem, the Bayesian inference first places a prior over the unknown distribution based on some prior knowledge, and then computes the posterior following the Bayes' Theorem. A general example given below shows the procedure of doing Bayesian inference:

1) First, we choose a prior distribution, say $p(\theta)$, based on our domain knowledge about the parameter θ that is to be estimated.
2) Then, we choose a statistical model, $p(x|\theta)$, based on our domain knowledge about the data given the parameters.
3) The posterior distribution $p(\theta|\boldsymbol{D})$ is obtained by updating the prior with the likelihood given the observed dataset $\boldsymbol{D} = \{X_1, \dots, X_n\}$.

The posterior distribution can be expressed as below according to Bayes' Theorem:

$$p(\theta|X_1, \dots, X_n) = \frac{p(X_1, \dots, X_n|\theta)p(\theta)}{p(X_1, \dots, X_n)} = \frac{L_n(\theta)p(\theta)}{p(\boldsymbol{D})} \propto L_n(\theta)p(\theta) \tag{3.26}$$

where $L_n(\theta)$ denotes the likelihood. The normalizing constant $p(\boldsymbol{D})$ can be calculated by

$$p(\boldsymbol{D}) = p(X_1, \dots, X_n) = \int p(X_1, \dots, X_n|\theta)p(\theta)d\theta = \int L_n(\theta)p(\theta)d\theta \tag{3.27}$$

Furthermore, we can use mean value or mode of the posterior to give a point estimation:

$$\widehat{\theta_n} = \int \theta p(\theta|\boldsymbol{D})d\theta = \frac{\int \theta L_n(\theta)p(\theta)d\theta}{\int L_n(\theta)p(\theta)d\theta} \tag{3.28}$$

We can also estimate an interval. Given $\alpha \in (0,1)$, we can find u and v such that

$$\int_{-\infty}^{u} p(\theta|\boldsymbol{D})\, d\theta = \int_{v}^{\infty} p(\theta|\boldsymbol{D})\, d\theta = \frac{\alpha}{2} \tag{3.29}$$

Thereafter,

$$P(\theta \in (u, v)|\boldsymbol{D}) = \int_{u}^{v} p(\theta|\boldsymbol{D})d\theta = 1 - \alpha \tag{3.30}$$

(u, v) is a $(1 - \alpha)$ credible interval.

3.3.2. Prior Specification

Following the Bayesian framework, the proposed model is specified as follows. To ease the latter Gibbs sampling procedure, a mixture representation of asymmetric Laplace distribution based on exponential and normal distributions [68] is used as the prior for the random error term ε, which can be expressed as

$$\varepsilon = \theta z + \tau\sqrt{z\alpha}u \tag{3.31}$$

where α is a scale parameter, z is a standard exponential variable and u is a standard normal variable. For a given quantile $q \in [0,1]$, it holds that $\theta = (1-2q)/q(1-q)$ and $\tau^2 = 2/q(1-q)$. To incorporate the embedded feature selection structure, an inclusion indicator variable γ is introduced. Letting γ_{jp} be the inclusion indicator for the j^{th} feature in the p^{th} quantile model, the proposed feature selection structure can be hierarchically specified by

$$\beta_{jp} \sim \gamma_{jp} N(\beta_0, \sigma_{jp}^2) + (1-\gamma_{jp})\delta_0 \tag{3.32}$$

$$\sigma_{jp}^{-2} \sim Gamma(a_\sigma, b_\sigma) \tag{3.33}$$

$$\gamma_{jp} \sim Bernoulli(\pi_{jp}) \tag{3.34}$$

where β_0 and σ_{jp}^2 are the prior mean and variance respectively, δ_0 is a point mass at 0, π_{jp} is the prior probability of $\gamma_{jp} = 1$. Note that γ_{jp} controls the inclusion of the j^{th} feature in the p^{th} quantile model, with $\gamma_{jp} = 0$ implying $\beta_{jp} = 0$. A value of $\beta_{jp} = 0$ means that the j^{th} feature is assigned a zero coefficient, resulting in the exclusion of this feature from the p^{th} quantile model. To address the problem of sparseness, π_{jp} is considered random and endowed with a sparseness-favoring prior [69], i.e.,

$$\pi_{jp} \sim \rho_{jp} Beta(a_\pi, b_\pi) + (1-\rho_{jp})\delta_0 \tag{3.35}$$

$$\rho_{jp} \sim Bernoulli(0.5) \tag{3.36}$$

3.3.3. Posterior Inference by MCMC – Gibbs Sampling

Given the above prior specifications, it is almost intractable to maintain the full posterior over the random variables. In practice, a typical way is draw sufficient samples from the distribution to approximate the target distribution. This kind of method is referred to as Monte Carlo sampling. Interestingly, the name Monte Carlo is named after a city in Monaco which owns a lot of casinos where many random stuffs happen every day.

However, Monte Carlo sampling is commonly used in low-dimensional scenarios and does not work well with high-dimensional dataset due to the curse of dimensionality. When the number of parameters increases, the sample space will increase exponentially. Another critical reason is that Monte Carlo sampling can only be strictly used in scenarios where the samples drawn from the target distribution can only be independent. Therefore, it cannot be applied to probabilistic models where the samples drawn depends on each other. To solve this problem, MCMC is introduced.

Basically, MCMC can be divided into two parts: Markov chain and Monte Carlo. Monte Carlo is a stochastic technique that takes random samples from a probabilistic distribution and estimates a target quantity. On top of that, a Markov chain is a stochastic model that generates a random sequence of states between which the transaction follows certain probabilistic rules. MCMC combines these two techniques by constructing a Markov chain to draw random samples from the target distribution. The obtained random samples are then averaged to approximate the expected quantities.

In the proposed research, the Gibbs sampling [70], which is an MCMC technique, is adopted to sample from the posteriors. The samples can then be used to approximate the posterior distribution. This is achieved by using discrete formulas applied to these samples to approximate the integrals of interest. The basic idea of MCMC is to do independent and identically distributed sampling from a target distribution Ω via a Markov chain mechanism. After N samples, $\{\boldsymbol{s}^{(i)}\}_{i=1}^{N}$ is obtained from the sampling procedure and the target distribution can be approximated by the following empirical point-mass function:

$$\Omega_N(\boldsymbol{s}) = \frac{1}{N} \cdot \sum_{i=1}^{N} \delta_{\boldsymbol{s}^{(i)}}(\boldsymbol{s}) \tag{3.37}$$

and any description of the target distribution (some expected value of a function f) can be computed by

$$E[f(\boldsymbol{s})]_{\Omega} \approx \frac{1}{N} \cdot \sum_{i=1}^{N} f\left(\boldsymbol{s}^{(i)}\right) \tag{3.38}$$

As an MCMC sampling algorithm, the Gibbs sampler updates each variable in turn by sampling from its posterior conditional on other variables. It constructs a Markov chain where the next sample is drawn according to the conditional probability given the previous sample.

Formally, given a D-dimensional variable vector $\boldsymbol{\mu} = (\mu_1, \dots, \mu_D)$ and a prior distribution q_0, a generic Gibbs sampler can be described in Table 3.3.

Table 3.3 The Gibbs sampler

Gibbs sampler
set $t = 0$ and initialize $\boldsymbol{\mu}^{(0)} \sim q_0$
for $t = 1, \dots, T$ repeat
for each dimension $i = 1, \dots, D$
draw $\boldsymbol{\mu}_i^{(t)} \sim P(\boldsymbol{\mu}_i \mid \boldsymbol{\mu}_1^{(t)}, \dots, \boldsymbol{\mu}_{i-1}^{(t)}, \boldsymbol{\mu}_{i+1}^{(t)}, \dots, \boldsymbol{\mu}_D^{(t)})$
end
end

Note that the sampler is initialized with random values; under this circumstance, the first few samples should be discarded because they may not represent the actual posterior distribution. Such discarded iterations are known as the burn-in period. All the rest effective samples will then be used to estimate the target distribution and its descriptions given by equation 3.37 and 3.38. The samples are updated sequentially from the following conditional posterior distributions.

1) Update $\boldsymbol{\beta}_p$ with $\boldsymbol{\beta}_p = (\boldsymbol{\beta}_{p\bar{\gamma}}, \boldsymbol{\beta}_{p\gamma})$

   $\boldsymbol{\beta}_{p\bar{\gamma}} = 0$;

   for $\boldsymbol{\beta}_{p\gamma}$,

$$\boldsymbol{\beta}_p \sim N(\overline{\boldsymbol{\beta}}, \overline{\boldsymbol{\Sigma}_\beta}) \tag{3.39}$$

$$\overline{\boldsymbol{\Sigma}_\beta} = (\sum_{t=1}^{T} \frac{\boldsymbol{x}_{t\gamma}^T \boldsymbol{x}_{t\gamma}}{\tau^2 \alpha z_t} + \Sigma_\sigma^{-1})^{-1} \tag{3.40}$$

$$\overline{\boldsymbol{\beta}} = \overline{\boldsymbol{\Sigma}_\beta}(\sum_{t=1}^{T} \frac{\boldsymbol{x}_{t\gamma}(y_t - \theta z_t)}{\tau^2 \alpha z_t}) \tag{3.41}$$

   where $\boldsymbol{\beta}_{p\gamma}$ is the vector of regression coefficients corresponding to $\gamma = 1$ including the intercept, $\boldsymbol{\beta}_{p\bar{\gamma}}$ is the coefficient vector with $\gamma = 0$, $\boldsymbol{x}_{t\gamma}$ is the feature vector at time t corresponding to $\gamma = 1$, and Σ_δ is the diagonal prior variance matrix with diagonal element σ_{jp}^2 if $\gamma_{jp} = 1$ and 0 if $\gamma_{jp} = 0$.

2) Update σ_{jp}^2

$$\sigma_{jp}^{-2} \sim Gamma(a_\sigma + \frac{1}{2}, b_\sigma + \frac{\beta_{jp}^2}{2}) \tag{3.42}$$

3) Update π_{jp}

If $\rho_{jp} = 0$, $\pi_{jp} = 0$;

If $\rho_{jp} = 0$,

$$\pi_{jp} \sim Beta(a_\pi + \gamma_{jp}, b_\pi + 1 - \gamma_{jp}) \tag{3.43}$$

4) Update ρ_{jp}

If $\gamma_{jp} = 1$, $\rho_{jp} = 1$;

If $\gamma_{jp} = 0$,

$$P(\rho_{jp} = 1) = \frac{\frac{\Gamma(a_\pi + b_\pi)\Gamma(b_\pi + 1)}{\Gamma(b_\pi)\Gamma(a_\pi + b_\pi + 1)}}{1 + \frac{\Gamma(a_\pi + b_\pi)\Gamma(b_\pi + 1)}{\Gamma(b_\pi)\Gamma(a_\pi + b_\pi + 1)}} \tag{3.44}$$

5) Update γ_{jp}

$$\gamma_{jp} \sim Bernoulli(p_\gamma) \tag{3.45}$$

$$p_\gamma = \frac{\pi_{jp} L(y; \boldsymbol{x}, \gamma_{jp} = 1, \gamma_{(-j)p})}{\pi_{jp} L(y; \boldsymbol{x}, \gamma_{jp} = 1, \boldsymbol{\gamma}_{(-j)p}) + (1 - \pi_{jp}) L(y; \boldsymbol{x}, \gamma_{jp} = 0, \boldsymbol{\gamma}_{(-j)p})} \tag{3.46}$$

where L is the likelihood of y given other parameters and data, and $\boldsymbol{\gamma}_{(-j)p}$ denotes the inclusion variable vector $\boldsymbol{\gamma}_p$ with its j^{th} element removed.

6) Update $\boldsymbol{z}$

$$\boldsymbol{z} \sim GIG(\frac{1}{2}, a_z, b_z) \tag{3.47}$$

$$a_z = \frac{y_t - \boldsymbol{x}_t^T \boldsymbol{\beta}_p}{\tau\sqrt{\alpha}} \tag{3.48}$$

$$b_z = \sqrt{\frac{2}{\alpha} + \frac{\theta^2}{\tau^2 \alpha}} \tag{3.49}$$

where $GIG(v, a, b)$ denotes the generalized inverse Gaussian distribution [71] with a PDF in the form of

$$f(x) = \frac{(b/a)^{v}}{2Kab} x^{v-1} e^{-\frac{1}{2}(\frac{a^2}{x} + b^2 x)} \tag{3.50}$$

with $x > 0, a, b \geq 0$.

3.4. Summary

In this chapter, the proposed embedded feature selection method for PLF along with the corresponding fundamentals are presented. The quantile linear regression based on the naïve vanilla benchmark model considering recency effect is introduced as the predictive model for PLF. To recode the categorical variables into numerical ones for the use of the proposed regression model, the dummy coding method is also introduced. Thereafter, the proposed feature selection framework following the steps of Bayesian inference is discussed in detail, including prior specification and posterior inference by an MCMC sampling technique, the Gibbs sampling. In the first step of prior specification, a mixture representation of asymmetric Laplace distribution based on exponential and normal distributions is used as the prior distribution for the random error term to ease the Gibbs sampling procedure. Besides, a sparseness-favoring prior is associated with the inclusion indicator variable to handle the sparsity of the feature space. In the step of posterior inference, the Gibbs sampler updates each variable in turn by sampling from its posterior distribution conditional on other variables, and the final results are given through using discrete formulas applied to the samples from the posterior distribution to summarize our knowledge of the parameters.

4. Case Study I: Test on One Region without Considering Recency Effect

4.1. Introduction

In Chapters 4, 5, and 6, comprehensive simulations are carried out to evaluate the effectiveness of the feature selection techniques based on two public datasets from the GEFCom2012 and GEFCom2014 respectively. This chapter focuses on examining the model performance on short-term PLF without considering recency effect.

4.2. Data Description and Test Settings

This subsection discusses the data and the test settings used in this chapter. An illustrative and interpretative description of the data used in the case study is given in detail. The training set and test set, along with the error measure are also discussed.

4.2.1. Data Description

The data used for this case study contains historical records for one region provided by the GEFCom2014. The GEFCom2014 dataset contains 69 months of hourly load data from January 2005 to September 2010 and 117 months of hourly temperature data recorded from 25 weather stations from January 2001 to September 2010. This dataset is used to examine the performance of the selected methods for one region without considering recency effect. Figure 4.1 plots the overall load profile from January 2005 to September 2010. Figure 4.2 plots the corresponding temperature. We can see a clear increasing trend in the load which might be caused by

economic and population growth and a periodic relativity between the load and the temperature. The scatter plot given by Figure 4.3 illustrates the overall load-temperature relationship of the whole dataset from January 2005 to September 2010. These illustrations figuratively explain the reasonability of adding a trend term and including temperature and calendar related features in the forecasting model. Further, Figure 4.4 and Figure 4.5 illustrate the scatter plot of hourly load and temperature for 12 months and 24 hours. It can be inferred from the plots that linear piecewise functions or polynomials of the temperature can be applied to model the relationship. It can also be seen that each subplot shows some differences, major or slight, compared to the others, indicating that the interaction between the polynomials of temperature and the calendar effects, the month, and the hour, should be modeled respectively.

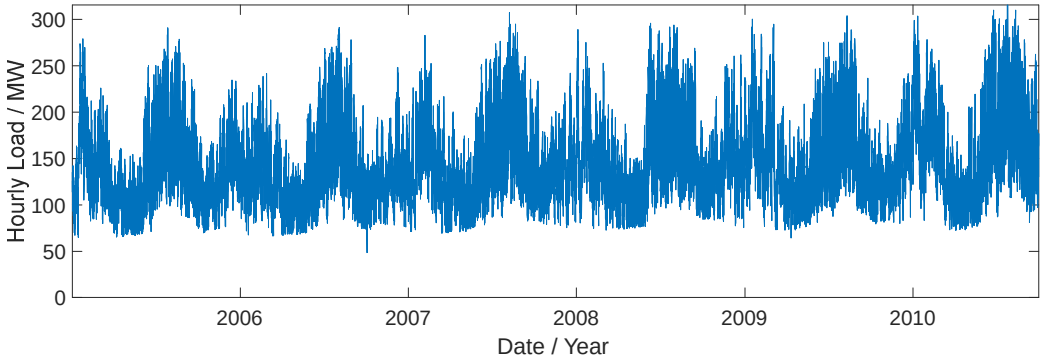


Figure 4.1 Overall load profile year by year from January 2005 to September 2010

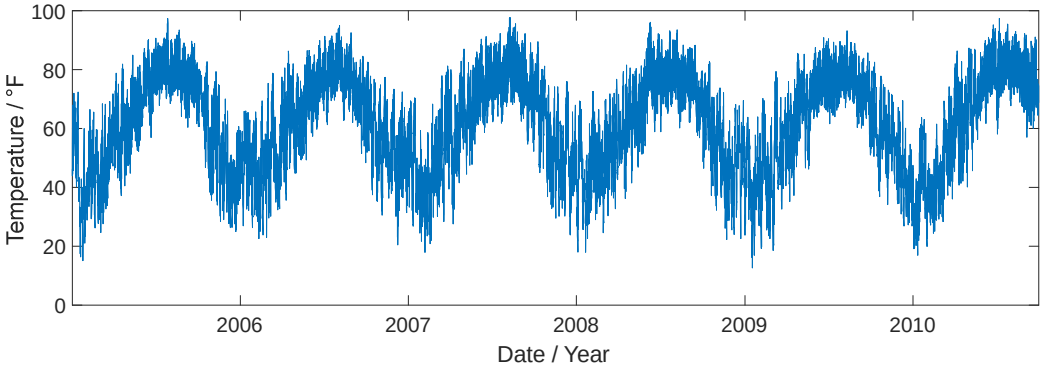


Figure 4.2 Overall temperature profile year by year from January 2005 to September 2010

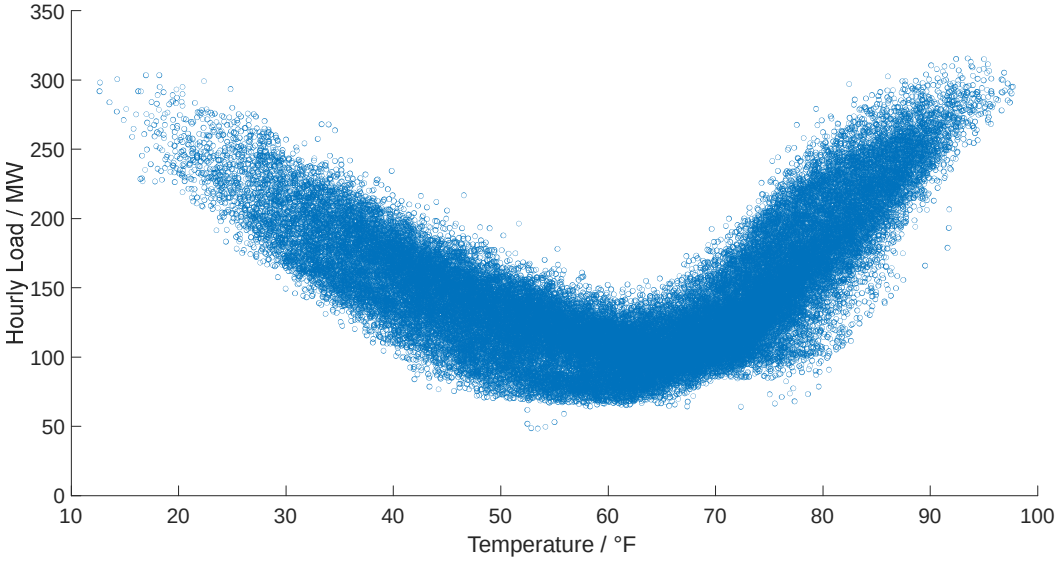
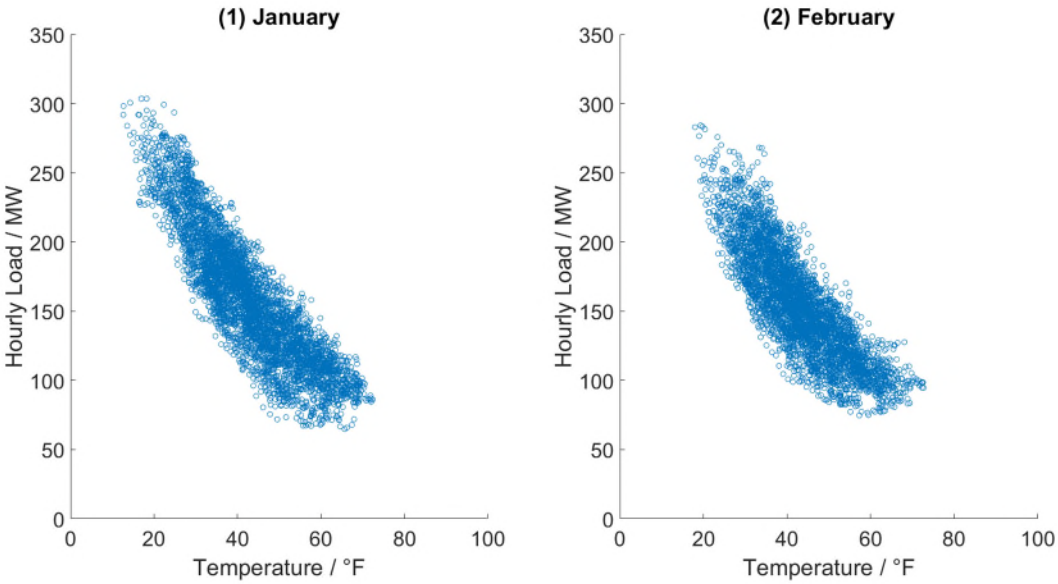
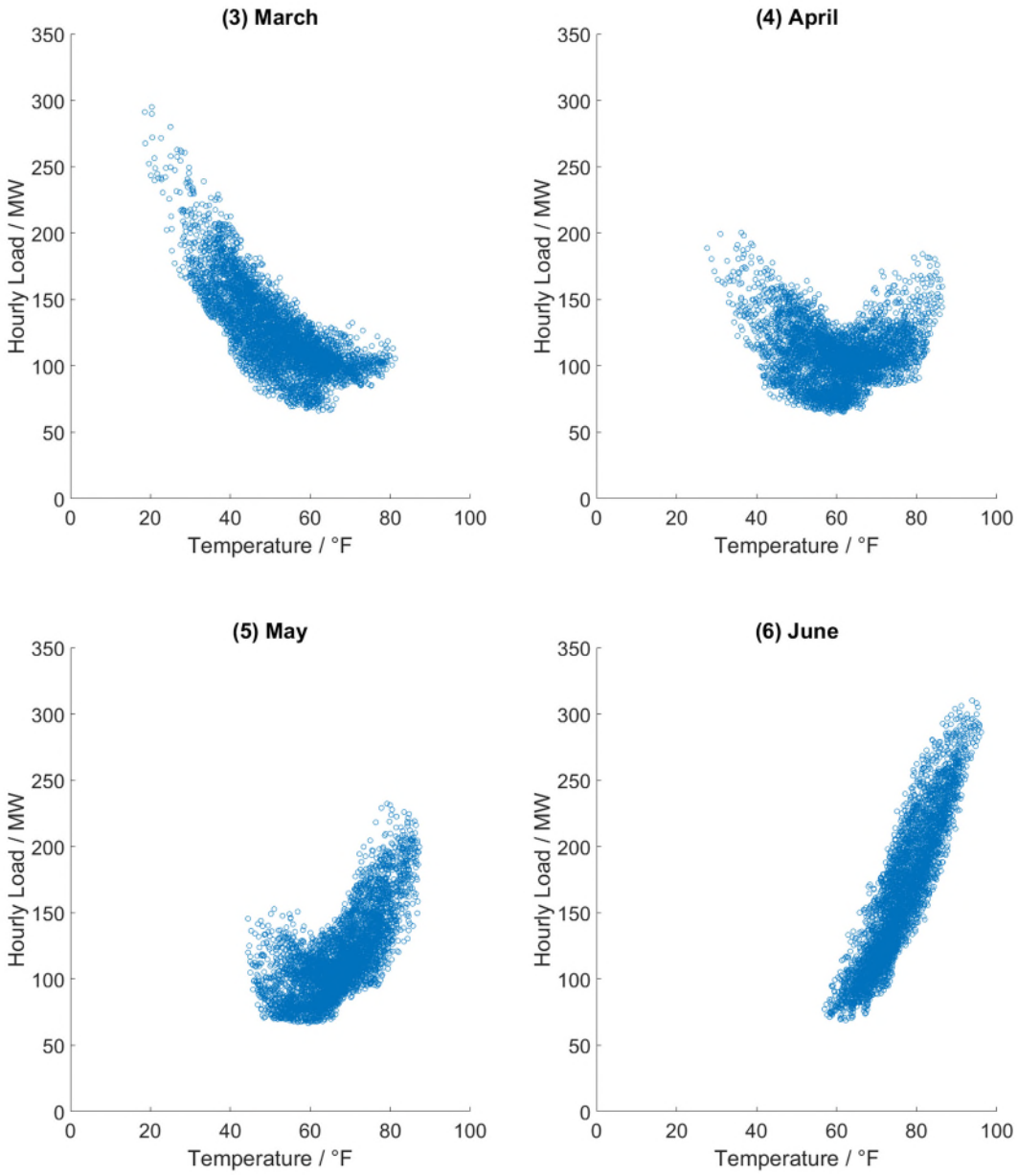
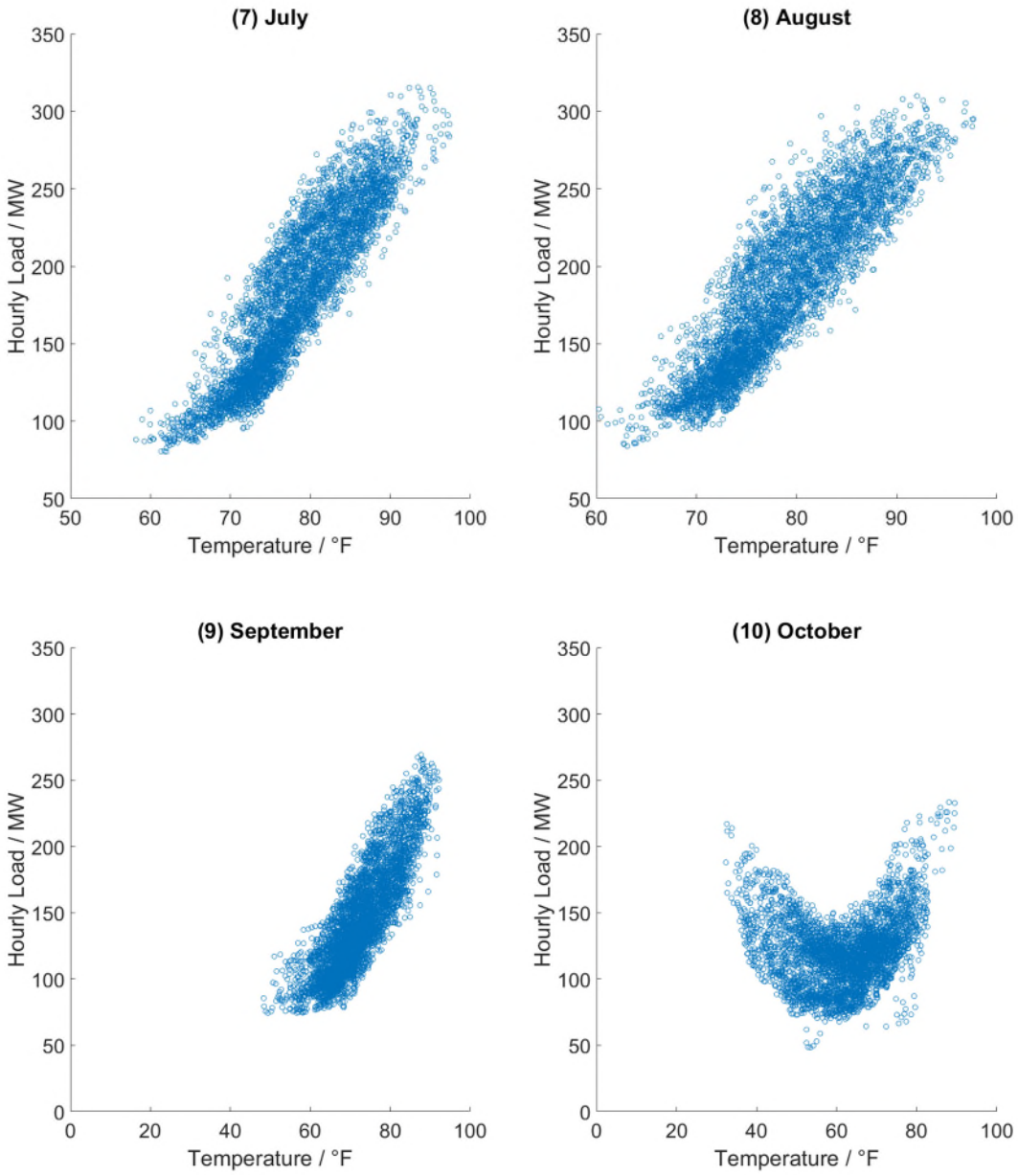


Figure 4.3 Scatter plot of hourly load and temperature for the whole dataset




(3) March
(4) April
(5) May
(6) June
Hourly Load / MW
Temperature / °F
0
50
100
150
200
250
300
350
0
20
40
60
80
100


(7) July
(8) August
(9) September
(10) October
Hourly Load / MW
Temperature / °F

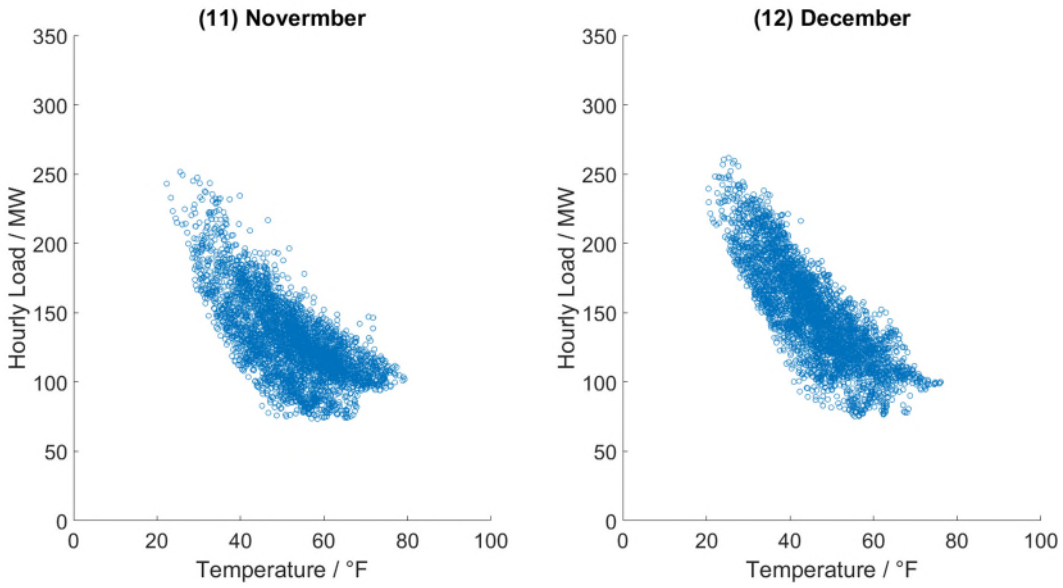
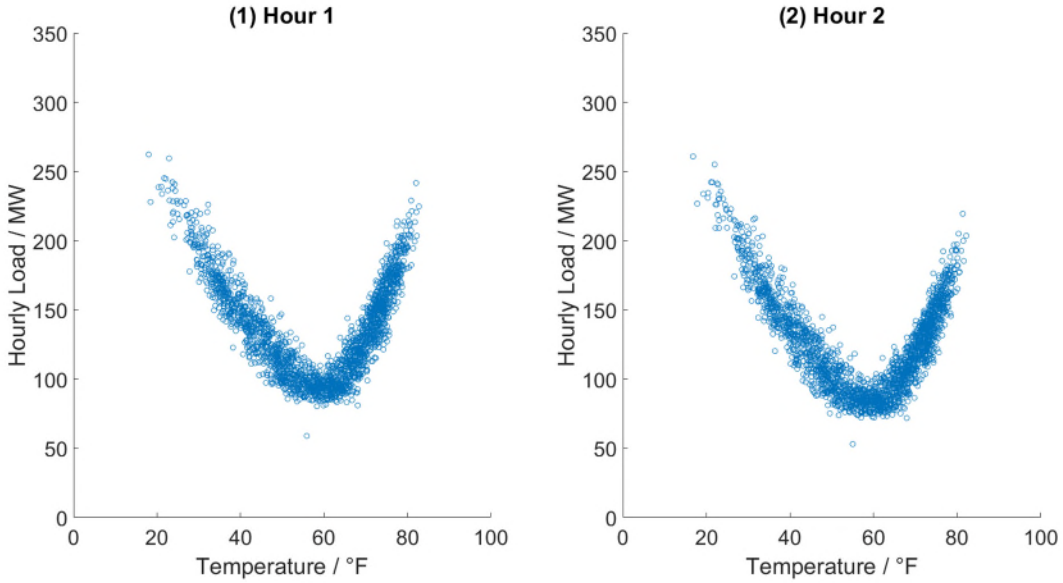
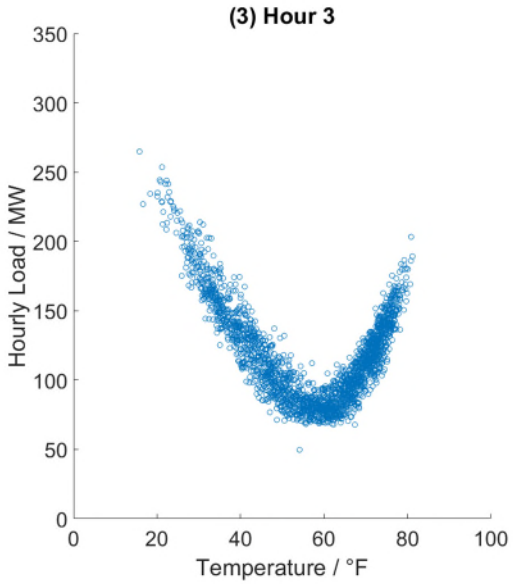
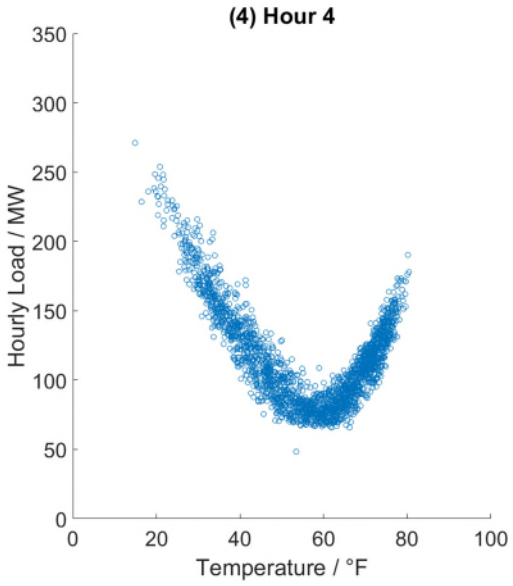
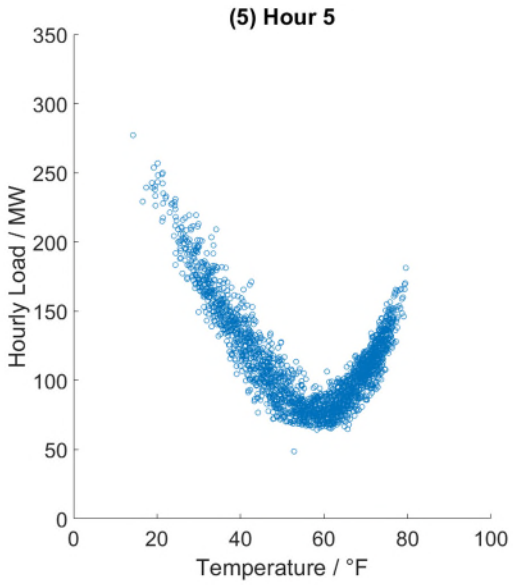


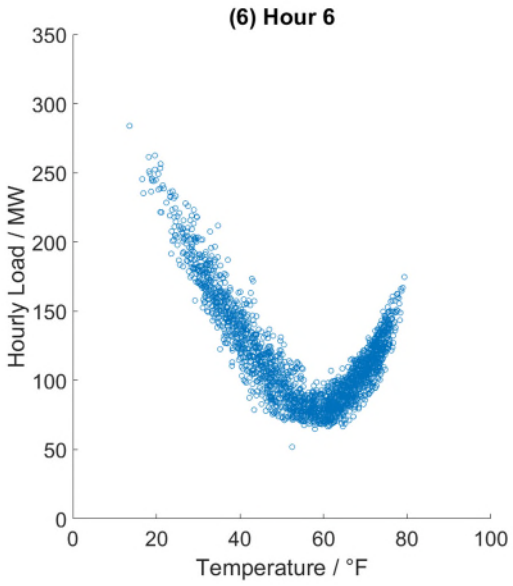
Figure 4.4 Scatter plot of hourly load and temperature for 12 months

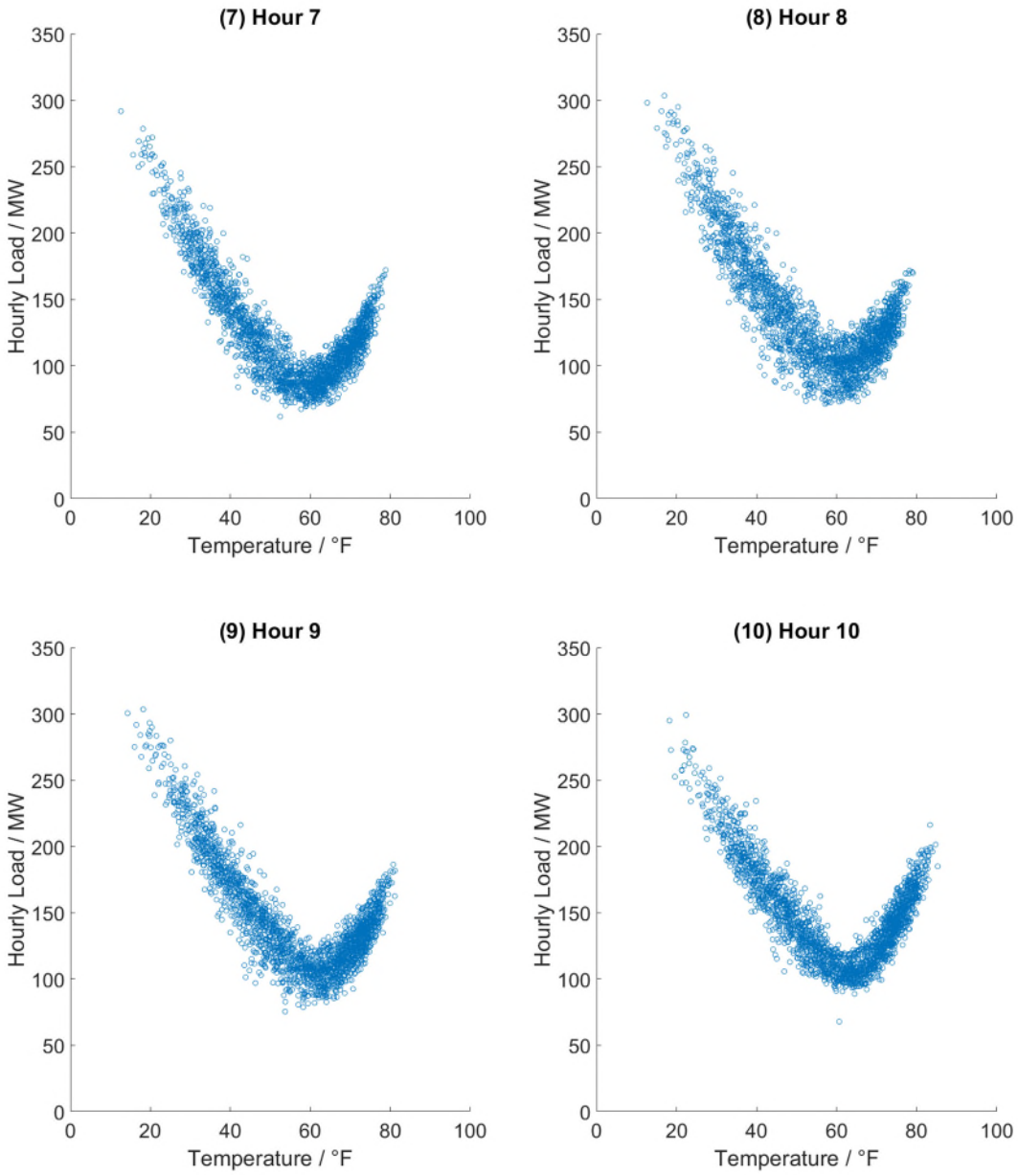


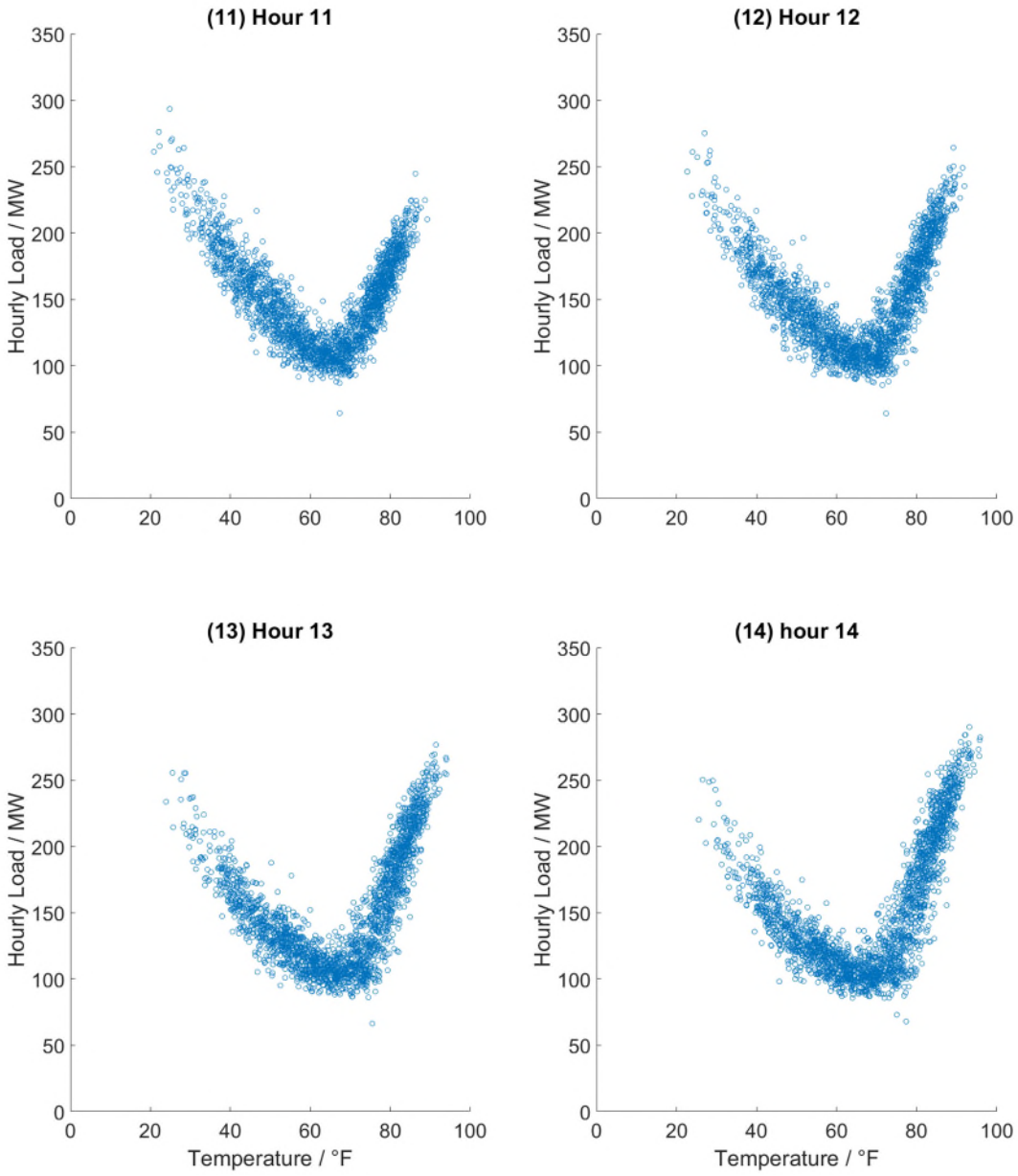

(3) Hour 3
Hourly Load / MW
Temperature / °F
0
50
100
150
200
250
300
350
0
20
40
60
80
100

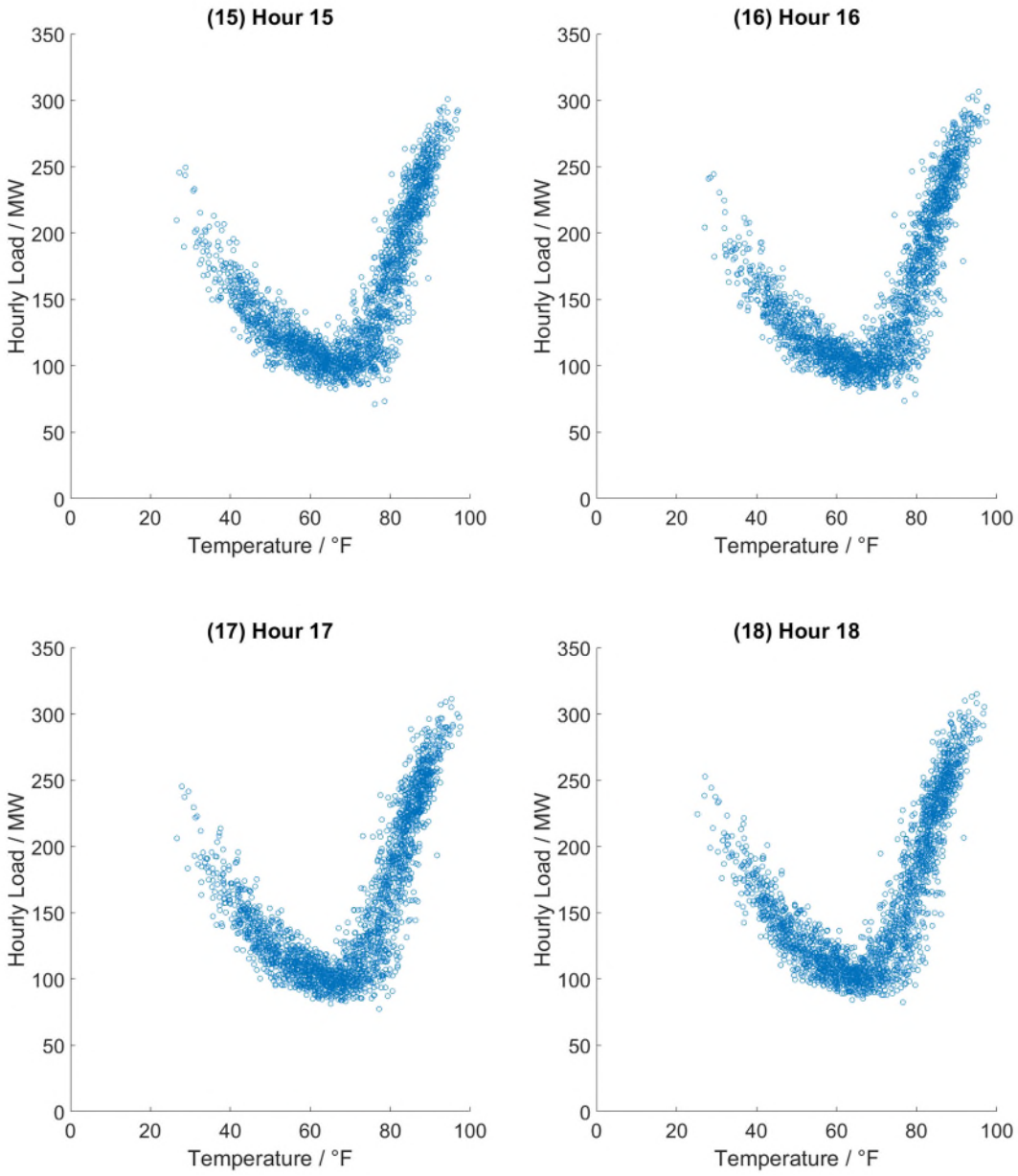

(4) Hour 4
Hourly Load / MW
Temperature / °F
0
50
100
150
200
250
300
350
0
20
40
60
80
100

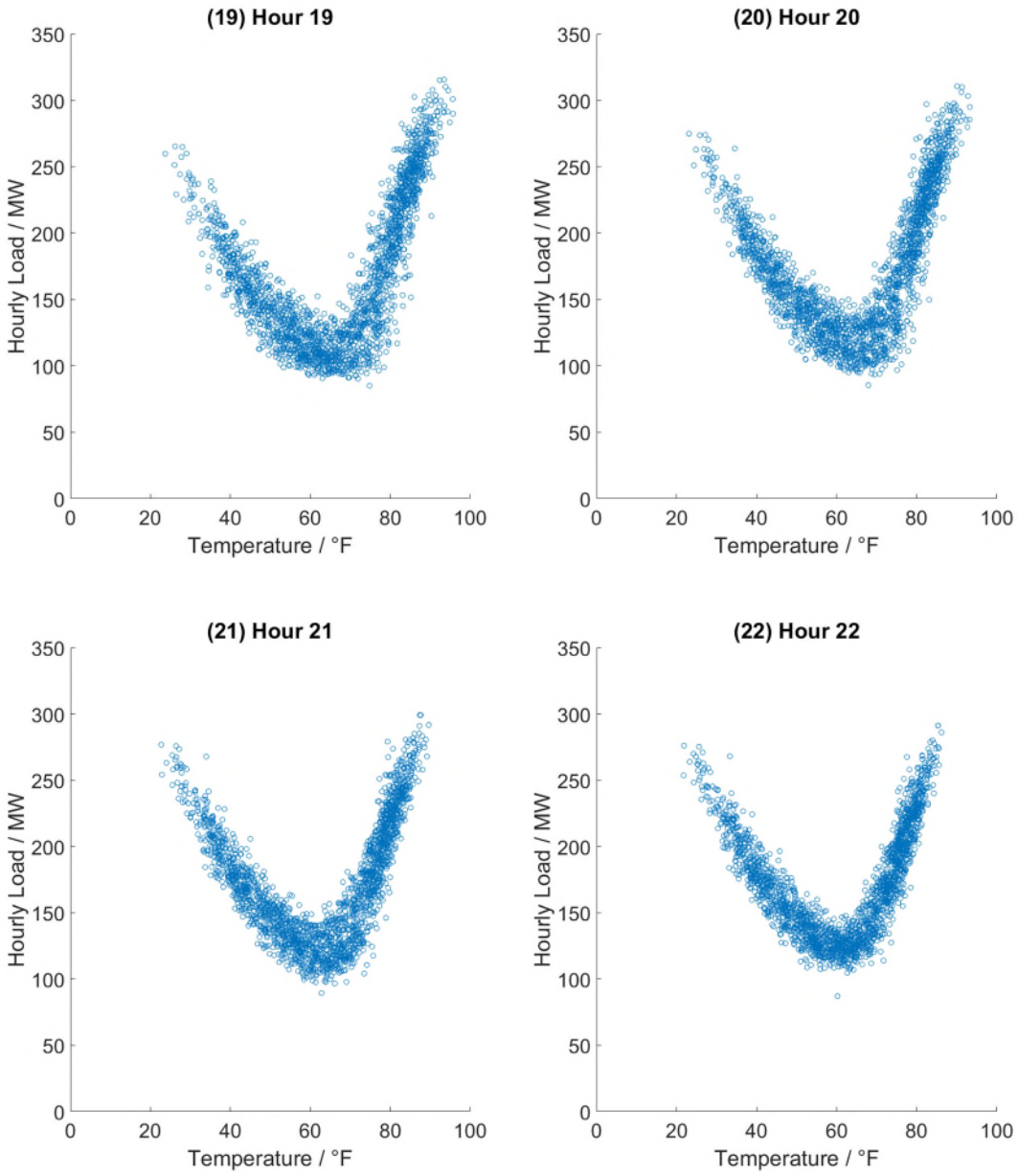

(5) Hour 5
Hourly Load / MW
Temperature / °F
0
50
100
150
200
250
300
350
0
20
40
60
80
100


(6) Hour 6
Hourly Load / MW
Temperature / °F
0
50
100
150
200
250
300
350
0
20
40
60
80
100


(7) Hour 7
(8) Hour 8
(9) Hour 9
(10) Hour 10
Hourly Load / MW
Temperature / °F
0
50
100
150
200
250
300
350
0
20
40
60
80
100


(11) Hour 11
(12) Hour 12
(13) Hour 13
(14) hour 14
Hourly Load / MW
Temperature / °F
0
50
100
150
200
250
300
350
0
20
40
60
80
100


(15) Hour 15
(16) Hour 16
(17) Hour 17
(18) Hour 18
Hourly Load / MW
Temperature / °F
0
50
100
150
200
250
300
350
0
20
40
60
80
100


(19) Hour 19
(20) Hour 20
(21) Hour 21
(22) Hour 22
Hourly Load / MW
Temperature / °F
0
50
100
150
200
250
300
350
0
20
40
60
80
100

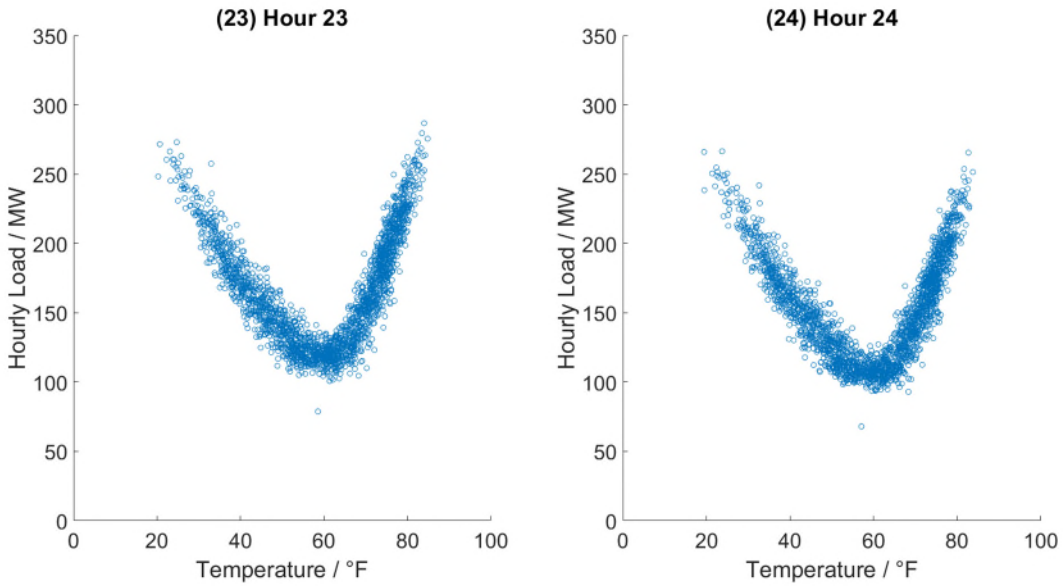


Figure 4.5 Scatter plot of hourly load and temperature for 24 hours of the day

4.2.2. Test Settings

From the GEFCom2014 dataset, a two-year period hourly load and temperature data from 1 November 2006 to 30 October 2008 are chosen as the training set. We opted to use a two-year period based on several considerations. Firstly, this period can effectively encompass a calendar year, comprising 12 months. Secondly, the two-year duration provides a twofold increase in the amount of information that can be learned by the model from this seasonal block when viewed annually, which should suffice. We avoid using longer periods, such as three years or more, due to the significant computational burden that they impose on several methods, especially the wrapper method, which makes the comparison unfeasible. The following two-week data from 1 November 2008 to 14 November 2008 are used to determine the optimal number of features for filter methods and evaluate the best subset of features for wrapper method. Thereafter, the two-week data from 15 November 2008 to 28 November 2008 are used for validating the performance of all the tested methods. Temperature is assumed to stay fixed within each hour. Encoding all the calendar variables by the proposed dummy encoding method, the total length of the features included in the linear model considering recency effect is $N_F = 1 + 1 + 11 +$

$6 + 23 + 23 \times 6 + (3 + 3 \times 11 + 3 \times 23)(1 + N_D + N_H) = 285 + 105(N_D + N_H)$.

A total of $Q = 19$ quantiles for a set of probabilities $\boldsymbol{\kappa} = \{0.05, 0.1, 0.15, \dots, 0.9, 0.95\}$ are used to form the PLF. Under these settings, the quantile score defined in Chapter 3 can be computed by

$$S_{QS} = \frac{1}{QH}\sum_{q=1}^{Q}\sum_{h=1}^{H} Pinball\left(\hat{y}_{t,p_q}, y_t, p_q\right) = \begin{cases} \frac{1}{QH}\sum_{q=1}^{Q}\sum_{h=1}^{H}\left(1 - \frac{p_q}{100}\right)\left(\hat{y}_{t,p_q} - y_t\right) & y_t < \hat{y}_{t,p_q} \\ \frac{1}{QH}\sum_{q=1}^{Q}\sum_{h=1}^{H}\frac{p_q}{100}\left(y_t - \hat{y}_{t,p_q}\right) & y_t \geq \hat{y}_{t,p_q} \end{cases} \tag{4.1}$$

where H is the length of the forecasting horizon with $H = 24 * 14$ and $p_q = 100\kappa_q$.

4.3. Benchmarks

The simulation starts with the original quantile linear regression model. To deliver a comprehensive comparison, three filter methods, one wrapper methods, an embedded method, and the original quantile linear regression as well as a nonlinear predictive model without feature selection are added as benchmarks. The proposed method is denoted by BQLRFS.

1) The first filter algorithm adopted here examines the importance of each feature individually using an ***F***-test and ranks the features using the ***p***-values of the ***F***-test statistics. The method is denoted by FTEST. The second filter algorithm calculates the feature weights using a diagonal adaptation of neighborhood component analysis, which is denoted by NCA. The third filter method uses the RReliefF algorithm [72] and is denoted by RRF. The implementation of these three methods is carried out in MATLAB, utilizing the functions provided by the Statistics and Machine Learning Toolbox.

   FTEST: a statistical test is a method that is used to infer if the given data support a specific hypothesis sufficiently. Basically, it indicates whether the difference between models is significant or not. For the problem of feature selection, ***F***-test is a statistical test that is used to evaluate the importance of individual feature. ***F***-test examines the hypothesis that the load values grouped by features are drawn from populations with the same mean against the alternative hypothesis that the population means are not all the

same. A p-value can be calculated from the test. This value describes how likely a particular set of observations are to be found if the null hypothesis were true. A small p-value indicates a high importance of the corresponding feature.

NCA: NCA is a supervised learning algorithm based on certain distance metrics over the data. It can be used for feature selection when solving a classification/regression problem by optimizing the modeling accuracy. Formally, given a training set with N observations $\{(\boldsymbol{x_i}, y_i), i = 1,2, \dots, N\}$ where $\boldsymbol{x_i}$ is a feature vector with D dimensions and y_i is the corresponding response value, the objective of the algorithm is to find a vector of feature weights that is adapted select the subset of features that optimizing the classification/regression model. Denoting the vector of feature weights $\boldsymbol{w} = (w_1, \dots, w_D)$, the function measuring the distance between two feature vector samples $\boldsymbol{x_i}$ and $\boldsymbol{x_j}$ is given by

$$D_{\boldsymbol{w}}(\boldsymbol{x_i}, \boldsymbol{x_j}) = \sum_{d=1}^{D} w_d^2 |\boldsymbol{x_{id}} - \boldsymbol{x_{jd}}| \tag{4.2}$$

Then, the probability that $\boldsymbol{x_i}$ is picked as the reference point for $\boldsymbol{x_j}$ is calculated by

$$p_{ij} = \frac{k(D_{\boldsymbol{w}}(\boldsymbol{x_i}, \boldsymbol{x_j}))}{\sum_{j=1, j\neq i}^{N} k(D_{\boldsymbol{w}}(\boldsymbol{x_i}, \boldsymbol{x_j}))} \tag{4.3}$$

Considering a regression problem such as load forecasting, let $\widehat{y_i}$ be the predicted value for $\boldsymbol{x_i}$ and l be a loss function quantifying the difference between $\widehat{y_i}$ and y_i. Then the expected value of $l(\widehat{y_i}, y_i)$ can be given by

$$l_i = \sum_{j=1, j\neq i}^{N} p_{ij} l(y_i, y_j) \tag{4.3}$$

Further, the feature selection result is given by the feature weights that minimize the objective function below with a regularization term added:

$$f(\boldsymbol{w}) = \frac{1}{N} \sum_{i=1}^{N} l_i + \lambda \sum_{d=1}^{D} w_d^2 \tag{4.4}$$

For more information, see [73].

RRF: RRF method works with continuous features and response. It calculates the feature weights by rewarding features that give different values to neighbours with different response values and penalizing features that give different values to neighbours

with the same response values. This algorithm is explained in detail as follows.

Given two nearest neighbours and the following denotations, the pseudo code for the RRF algorithm is described in Table 4.1 below:

- Let w_{dy} denotes the weight of having different values for the response y.
- Let w_{dj} denotes the weight of having different values for the j^{th} feature F_j.
- Let $w_{dy\&dj}$ denotes the weight of having different values for the response y and different values for the j^{th} feature F_j.
- Let w_j denotes the weight for the j^{th} feature F_j.
- Let y_r denote the response for observation $\boldsymbol{x_r}$ and y_q denote the response for observation $\boldsymbol{x_q}$. Then, the difference in the response is denoted by $\Delta_y(\boldsymbol{x_r}, \boldsymbol{x_q})$ and the difference in the value of the j^{th} feature is denoted by $\Delta_j(\boldsymbol{x_r}, \boldsymbol{x_q})$.
- Let d_{rq} denotes the distance function measuring the distance between two instances $\boldsymbol{x_r}$ and $\boldsymbol{x_q}$.

Table 4.1 Pseudo code for the RRF algorithm

Algorithm: RRF

set all w_{dy}, w_{dj}, $w_{dy\&dj}$ and w_j to 0

for $i = 1, \ldots, m$ repeat

randomly select an observation $\boldsymbol{x_r}$

find the k-nearest observations to $\boldsymbol{x_r}$

for each nearest neighbour $\boldsymbol{x_q}$, update the following weights

$$w_{dy}^{i} = w_{dy}^{i-1} + \Delta_y(\boldsymbol{x_r}, \boldsymbol{x_q}) \cdot d_{rq}$$

$$w_{dj}^{i} = w_{dj}^{i-1} + \Delta_j(\boldsymbol{x_r}, \boldsymbol{x_q}) \cdot d_{rq}$$

$$w_{dy\&dj}^{i} = w_{dy\&dj}^{i-1} + \Delta_y(\boldsymbol{x_r}, \boldsymbol{x_q}) \cdot \Delta_j(\boldsymbol{x_r}, \boldsymbol{x_q}) \cdot d_{rq}$$

$$\Delta_y(\boldsymbol{x_r}, \boldsymbol{x_q}) = \frac{|y_r - y_q|}{\max(y) - \min(y)}$$

$$\Delta_j(\boldsymbol{x_r}, \boldsymbol{x_q}) = \frac{|x_{rj} - x_{qj}|}{\max(F_j) - \min(F_j)}$$

$$d_{rq} = \frac{e^{-(rank(r,q)/sigma)^2}}{\sum_{l=1}^{k} e^{-(rank(r,l)/sigma)^2}}$$

end

After fully updating the weights above, the feature weights w_j can be calculated by

$$w_j = \frac{w_{dy\&dj}}{w_{dy}} - \frac{w_{dj} - w_{dy\&dj}}{m - w_{dy}}$$

2) For the wrapper method, one greedy search algorithm, the sequential forward selection, which has been discussed in Section 2.2.2, is adopted, and denoted by SFS. This method is implemented in Python following the steps below:

   a). Initialize an empty set of features and a set of candidate features to be added to the feature set.

   b). Train the multiple linear regression model as mentioned in Chapter 3.2.3 using the training data with the empty feature set and evaluate the performance on the data given for evaluation.

   c). For each candidate feature not in the feature set, add it to the feature set, train a model with the augmented feature set, and evaluate the performance on the same data given for evaluation.

   d). Select the best candidate feature that improves the performance the most and add it to the feature set.

   e). Repeat the above steps until the desired number of features or optimal performance is achieved.

   f). Train the final model using the selected feature set and evaluate its performance on the validation dataset.

3) To show the superiority of the proposed method, the method proposed by [23], which is the most recently published research on embedded feature selection for PLF, is used as the embedded method benchmark, denoted by QRLASSO. The full name of the method is least absolute shrinkage and selection operator (LASSO) based on quantile regression. This method estimates the parameter $\boldsymbol{\beta}_p$ by minimizing the objective function of the quantile regression with an L_1-norm penalty, i.e.,

$$\widehat{\boldsymbol{\beta}_p} = arg\min_{\boldsymbol{\beta}_p} \sum_{t=1}^{T} l_p\left(y_t - \boldsymbol{x}_t^T \boldsymbol{\beta}_p\right) + \lambda_p ||\boldsymbol{\beta}_p||_1 \tag{4.5}$$

where l_p is the loss function and λ_p is the weight of sparse penalty of the p^{th} quantile,

respectively. The QRLASSO is similar to the strategy of the conventional LASSO method, except that the optimal λ_p is different for each quantile. In order to attain an optimum level of performance and a fair comparison, we undertake the same model selection procedure to explore the most suitable adjustment parameter λ_p for every quantile, as detailed in reference [23].

4) To prove the competitiveness of the quantile linear regression model for PLF, a nonlinear model, the quantile regression neural network (QRNN) with default setting is also included in the testing. The original quantile linear regression model is denoted by QLR, which has been introduced in Chapter 3.2.1. The QRNN algorithm is implemented using the QRNN package [74] designed in the R programming language and is briefly discussed as follows.

   QRNN simply combines quantile regression and neural network together to represent the nonlinear relation between features and quantiles of the response variable. A commonly used neural network model for time series forecasting is feed forward neural network, the general architecture of which is depicted by Figure 4.6.

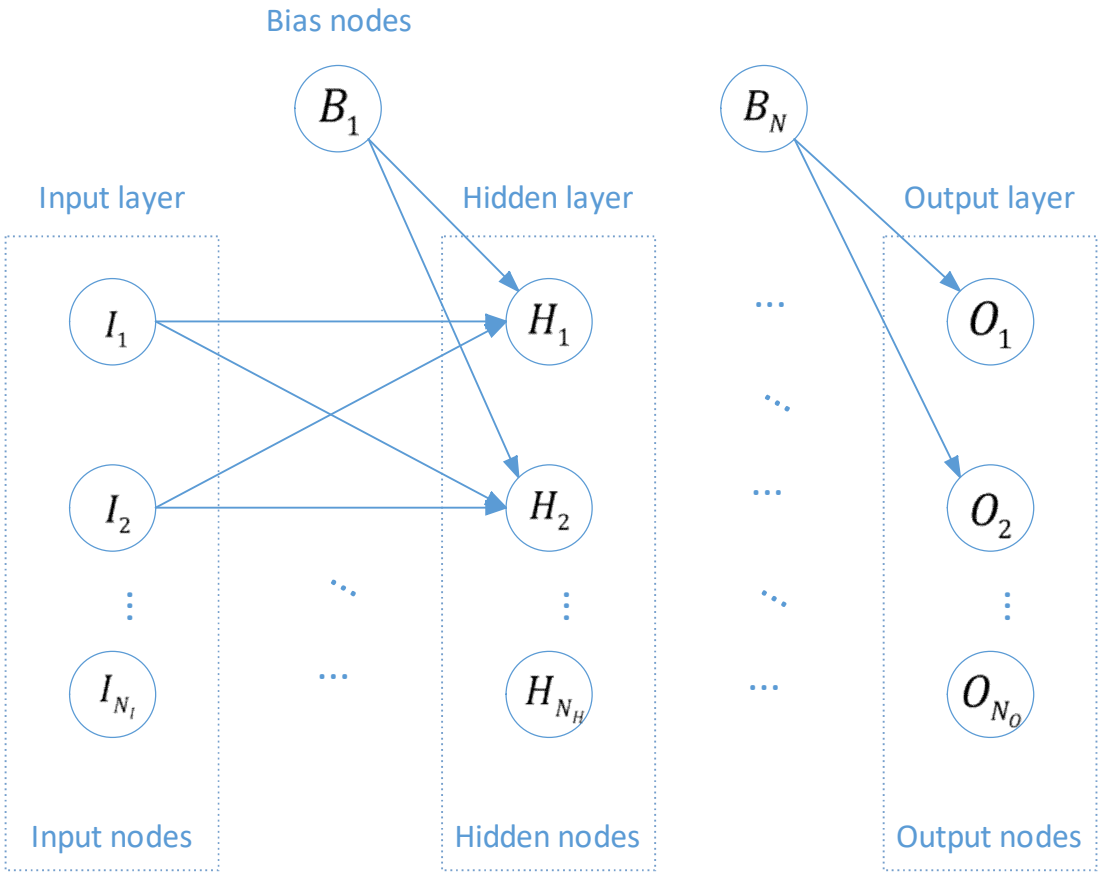


Figure 4.6 General architecture of a feed forward neural network

The following introduction to the algorithm supposes a network with two hidden layers and D input variables $x_1, \ldots, x_D$. Under this setting, the output from the k^{th} hidden node in the first layer can be expressed as

$$g_k = f_1\left(\sum_{d=1}^{D} x_d w_{dk}^{(h)} + b_k^{(h)}\right) \tag{4.6}$$

and similarly, the output from the l^{th} hidden node in the second layer can be expressed as

$$h_l = f_2\left(\sum_{k=1}^{K} g_k w_{kl}^{(h)} + b_l^{(h)}\right) \tag{4.7}$$

where $f_1(\cdot)$ and $f_2(\cdot)$ are the activation functions for the first and the second hidden layer respectively, and $w^{(h)}$ denotes the weights for the hidden layers, and $b^{(h)}$ represents the bias of associated hidden layers. Supposing that the output layer of the network consists of

only one single node, the estimated p^{th} conditional quantile for the t^{th} instance can be given by

$$\widehat{Q_t^{(p)}} = f_o\left(\sum_{l=1}^{L} h_l w_l^{(o)} + b^{(o)}\right) \tag{4.8}$$

where $w^{(o)}$ denotes the weights for the output layer, and $b^{(o)}$ is the associated bias, and $f_o(\cdot)$ is the activation function of the output layer. Similar to the error function of linear quantile regression given as (3.4), the error function to be minimized for QRNN is

$$\sum_t \rho_p(y_t - \widehat{Q_t^{(p)}}) \tag{4.9}$$

where the loss function $\rho_p(\cdot)$ here is different from the one given by (3.5) and is defined as

$$\rho_p(u) = \begin{cases} ph(u) & if\ \ u \geq 0 \\ (p-1)h(u) & if\ \ u < 0 \end{cases} \tag{4.10}$$

$$h(u) = \begin{cases} \dfrac{u^2}{\xi} & if\ \ 0 \leq |u| \leq \xi \\ |u| - \dfrac{\xi}{2} & if\ \ u < 0 \end{cases} \tag{4.11}$$

where $h(u)$ is the Huber function [75]. The reason for such change is that (3.5) is not defined at the origin and thus not differentiable everywhere. Hence, the Huber function is used to smooth the loss function.

4.4. Technical Specification

All simulations Chapter 4, Chapter 5 and Chapter 6 are run on a Linux-based, heterogeneous, high-performance computing cluster at the University of Saskatchewan. Plato has a total of 120 compute nodes with an aggregate 2 000 CPU cores and 7.4 TB RAM, yielding a theoretical 64 tera floating-point operations per second. There are 94 general-purpose nodes, 2 GPU nodes, and 2 large-memory nodes. The software used and the technical specification for each method is introduced in detail as below.

1) The proposed method, the three filter methods, and the original linear quantile regression model are tested with MATLAB scripts, using the Penguin high-density nodes:
   - 2 x twenty-core Intel(R) Xeon(R) Gold 6148 CPU @ 2.40GHz (AVX512, AVX2,

AVX)

- 192 GB RAM
- 2 x 1 GB Ethernet to Cluster network
- 781 GB local storage drive (/local)

2) The wrapper method is tested with Python script, using both the Penguin high-density nodes and the Dell PowerEdge R920 big memory node:
   - 4 x twelve-core Intel(R) Xeon(R) CPU E7-4850 v2 @ 2.30GHz (AVX)
   - 2TB RAM
   - 2 x 10 GB Ethernet to Cluster network
   - 1.5 TB local storage drive (/local)
3) The QRLASSO method is tested with MATLAB script, using both the Penguin high-density nodes and the Dell PowerEdge R920 big memory node.
4) The QRNN method is tested with R script, using the Penguin high-density nodes.

4.5. Case Studies and Results of the GEFCom2014 Dataset

To make a comprehensive comparison and show the effectiveness of the proposed method, the overall performance, and the performance of the tested methods over a set of quantiles are examined in this simulation. The computational costs of the tested methods are reported. The interpretability of the feature selection methods is also discussed in detail.

4.5.1. Result without Considering Recency Effect

Table 4.2 presents the quantile score of the proposed method and all the benchmarks without considering recency effect. The proposed method BQLRFS has the lowest overall quantile score among all the tested methods and is significantly lower than the other methods. The feature selection benchmarks including three filter methods, one wrapper method and an embedded method. Compared with QLR, which is the naïve benchmark without feature selection, FTEST, NCA and SFS show little improvement in PLF performance. NCA has the same accuracy as

QLR, indicating that NCA select all features in this case. Only RRF, QRLASSO and the nonlinear naïve benchmark QRNN show comparable improvement, while they are still beaten by the proposed method.

It is worth emphasizing that while QRLASSO is the only recently proposed method that integrates feature selection into a PLF model, its performance falls short of expectations and is outperformed by the proposed approach. This disparity can be attributed to several factors. Firstly, while both QRLASSO and the proposed method permit the selection of features to vary across quantiles, the latter approach surpasses the former by allowing all input features to demonstrate their impact through the utilization of Bayesian inference. By adopting Bayesian inference, relevant features are assigned higher inclusion probabilities, whereas less relevant features are less likely to be selected. Thus, all input features can potentially influence the load forecasting results. On the contrary, the LASSO method removes less relevant features by shrinking their coefficients to zero, completely erasing their impact. Another crucial factor is that the proposed method is better equipped to handle sparse feature spaces than QRLASSO due to the integration of a sparse-favoring prior over the inclusion indicator variables via Bayesian inference. In contrast, QRLASSO does not possess a design tailored to handle sparsity. Furthermore, QRLASSO requires a model selection process to determine the optimal adjustment parameter for each quantile, whereas the proposed method leverages prior knowledge to select hyperparameters for the prior distribution in Bayesian modeling, eliminating the need for a model selection process.

Table 4.2 QS of all tested methods without considering recency effect

	BQLRFS	FTEST	NCA	RRF	SFS	QRLASSO	QLR	QRNN
QS	3.645	4.464	4.451	3.921	4.403	4.133	4.451	4.083

The pinball loss of each quantile averaged over the forecasted horizon for each method is reported in Table 4.3, with the lowest value of each row bolded. Conditional formatting is applied to the table cells to enhance the visualization of the data. Specifically, a color gradient is employed to represent the values assigned to each cell, with the degree of green or red saturation corresponding to the magnitude of the value. A smaller value is depicted in a greener hue, whereas a larger value is represented in a redder hue. BQLRFS has the lowest average

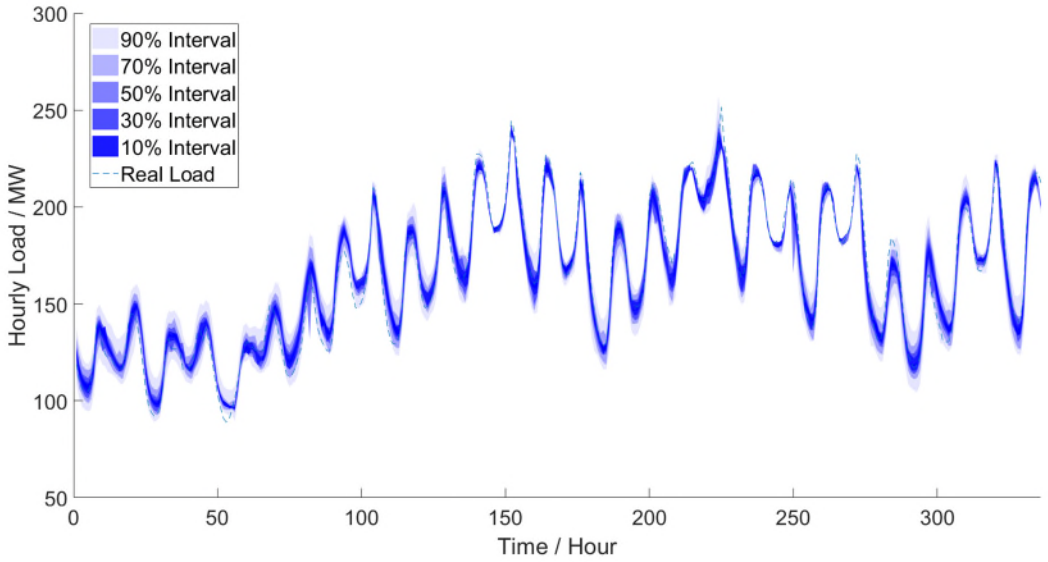
pinball loss across the forecasted horizon for 13 out of 19 total quantiles, while the loss of BQLRFS can still be considered relatively low for the rest of the quantiles compared to the benchmarks. It can be clearly seen from the table that the average pinball loss tends to become larger towards the middle range of the quantiles. Though all the methods have close and relatively low average pinball loss for small and large quantiles, the loss of BQLRFS is significantly lower and less fluctuating than that of the benchmarks regarding the middle range of the quantiles, indicating robust performance across all forecasted quantiles.

Table 4.3 Pinball loss of each quantile averaged over the forecasted horizon for all methods without considering recency effect

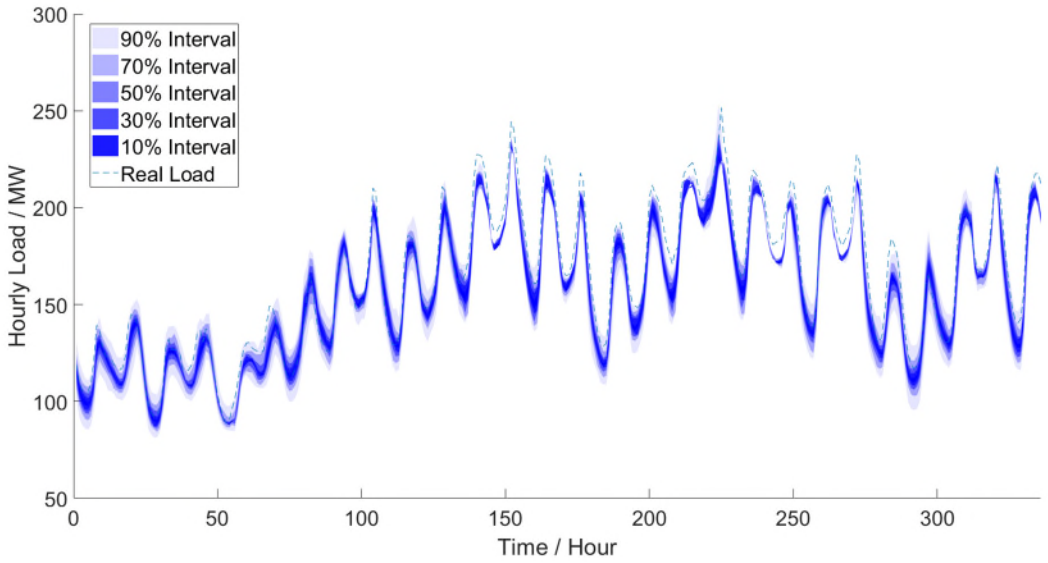
Quantile	BQLRFS	FTEST	NCA	RELIEFF	SFS	QRLASSO	QRMLR	QRNN
0.05	1.77	1.24	1.23	1.36	1.15	1.34	1.23	1.19
0.10	2.43	2.05	2.05	2.31	2.01	2.14	2.05	1.95
0.15	2.98	2.74	2.75	3.04	2.70	2.78	2.75	2.73
0.20	3.14	3.30	3.31	3.67	3.29	3.35	3.31	3.19
0.25	3.93	3.76	3.75	4.15	3.81	3.82	3.75	3.69
0.30	4.08	4.28	4.23	4.55	4.29	4.26	4.23	4.19
0.35	4.27	4.65	4.64	4.76	4.69	4.59	4.64	4.65
0.40	4.47	5.07	4.99	4.94	5.05	4.83	4.99	4.83
0.45	4.48	5.37	5.29	5.11	5.31	5.07	5.29	5.17
0.50	4.52	5.57	5.45	5.19	5.50	5.23	5.45	5.03
0.55	4.34	5.71	5.57	5.10	5.56	5.22	5.57	5.56
0.60	4.37	5.63	5.55	5.10	5.47	5.18	5.55	5.61
0.65	4.24	5.54	5.55	4.92	5.47	5.01	5.55	5.20
0.70	4.02	5.49	5.46	4.70	5.40	4.82	5.46	4.87
0.75	3.95	5.46	5.50	4.41	5.33	4.68	5.50	4.42
0.80	3.08	5.22	5.27	3.85	4.97	4.57	5.27	4.95
0.85	3.45	4.94	4.97	3.29	4.83	4.33	4.97	4.60
0.90	2.96	4.50	4.65	2.51	4.66	3.96	4.65	3.30
0.95	2.76	4.29	4.37	1.43	4.19	3.34	4.37	2.47

To give an intuitive comparison, the forecasted quantiles of all tested methods and the real load over the forecasted horizon from 15 November 2008 to 28 November 2008 is depicted in

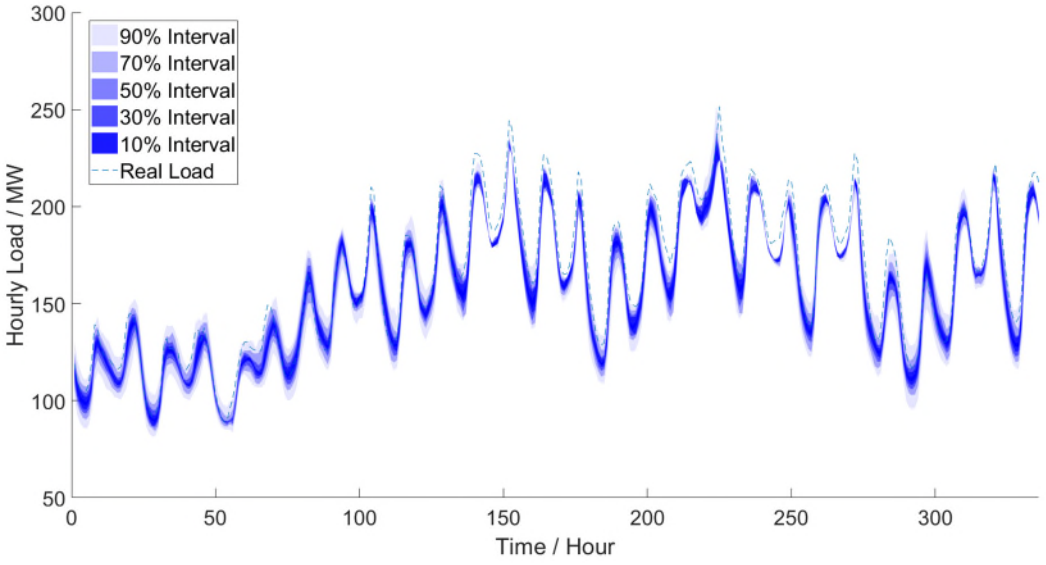
separate sub figures of Figure 4.7. For all plots in Figure 4.7, 5 typical predictive intervals, the 10%, 30%, 50%, 70% and 90% intervals are selected and plotted for better visualization, because the shaded area would be too dense to be distinguished if all quantiles given by predictive intervals are depicted.



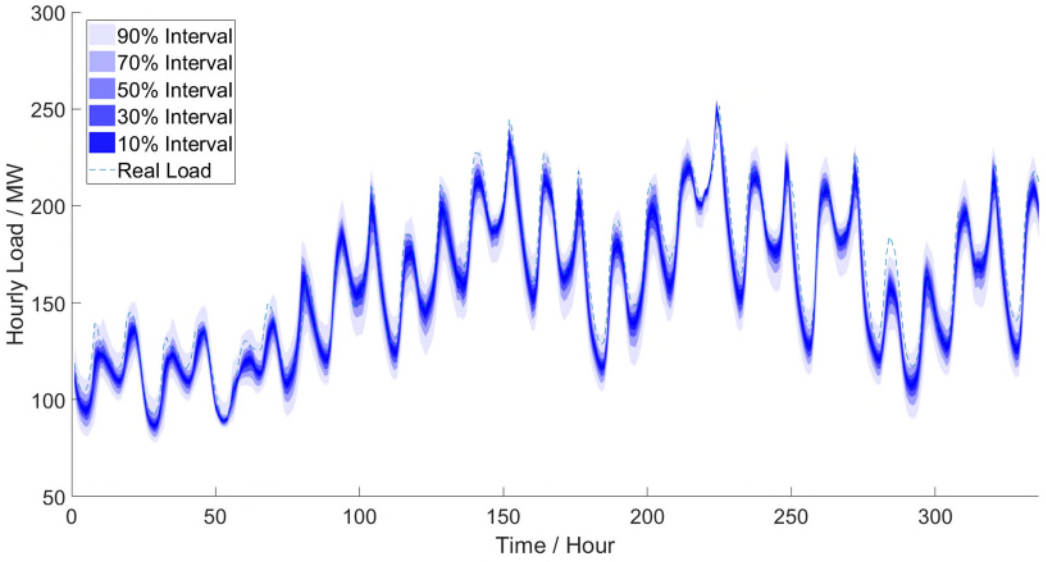
(1) BQLRFS



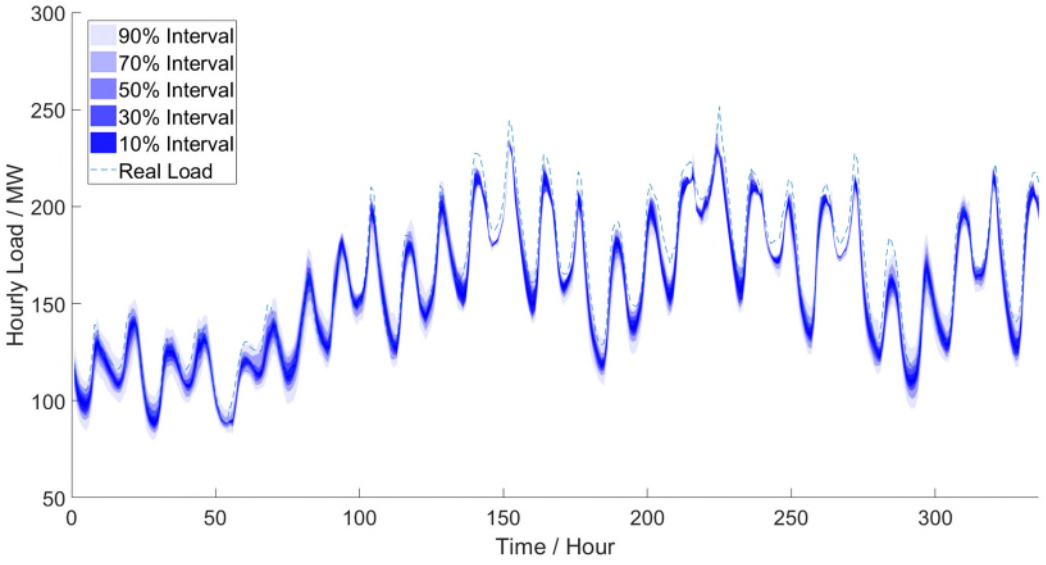
(2) FTEST



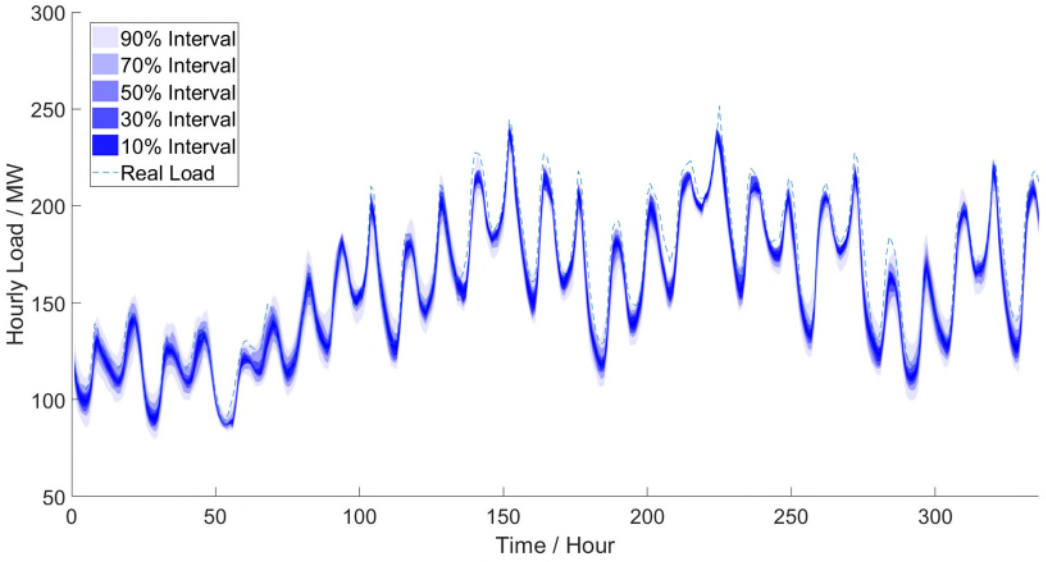
(3) NCA



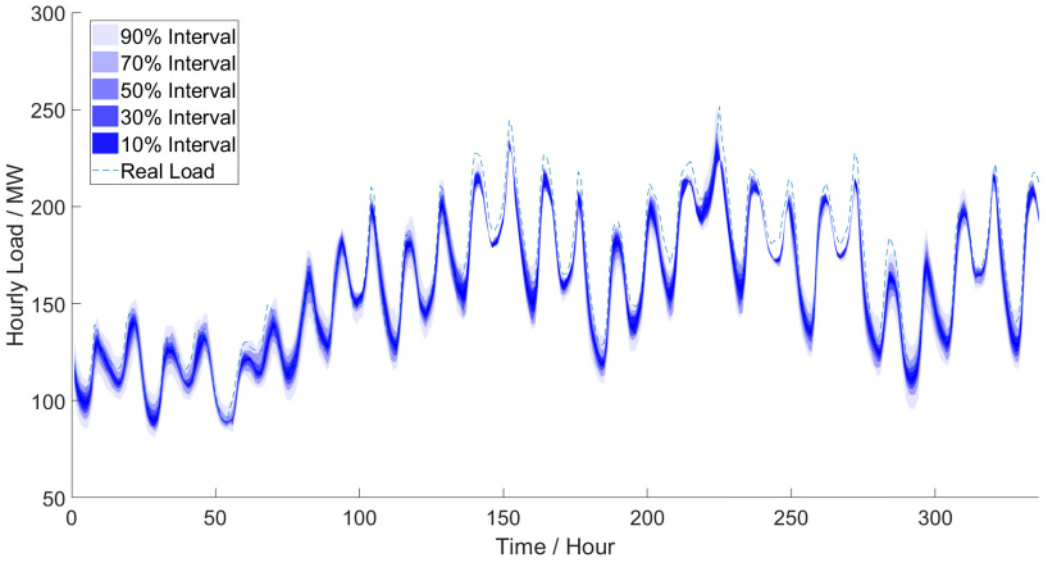
(4) RELIEFF



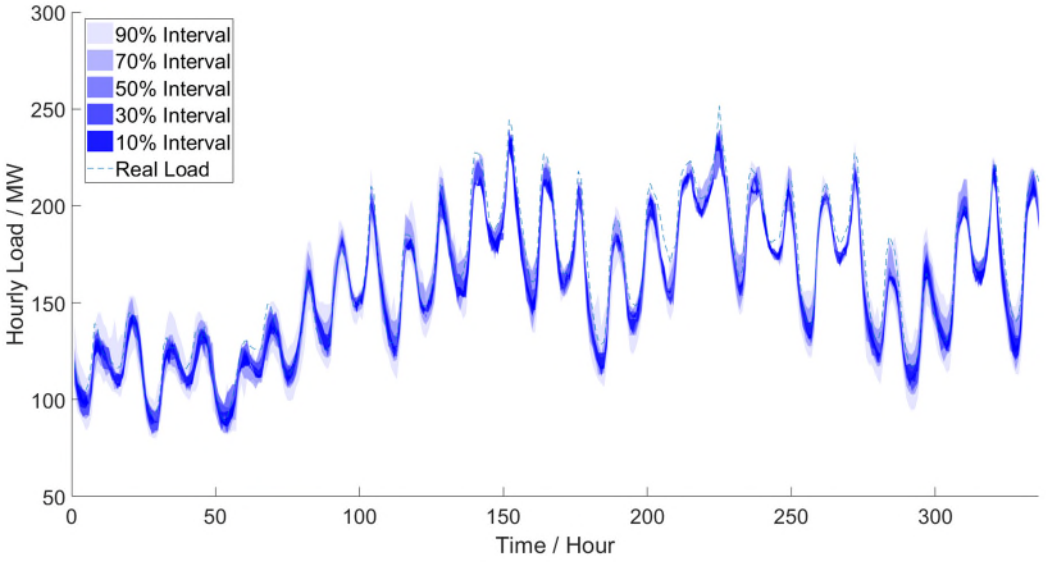
(5) SFS



(6) QRLASSO



(7) QRMLR



(8) QRNN

Figure 4.7 Predictive intervals of all tested methods and the real load over the forecasted horizon

4.5.2. Computational Cost

Table 4.4 presents the computation times of all tested methods. It should be noted that the computation times of all filter methods include a model selection process to determine the optimal number of features included for prediction. The running times of this process for FTEST, NCA and RRF are 21224s, 22161s and 22440s, respectively, which dominate the time consumption. Although filter methods are the most efficient feature selection methods themselves, a significant amount of time needs to be spent in a model selection process unless subjectively predefined. The computation burden for the rest of the methods is within acceptable range for off-line training.

Table 4.4 Computation time of all tested methods

Method	Computation Time
BQLRFS	4746 s
FTEST	21224 s
NCA	22161 s
RRF	22440 s
SFS	1055 s
QRLASSO	3847 s
QLR	336 s
QRNN	2313 s

4.5.3. Feature Selection Interpretation

The way that how feature selection is interpreted by each method is discussed in this subsection.

Filter methods measure the importance of each feature independently based on certain statistical criteria regardless of the forecasting algorithm. For each of the filter methods tested in the simulation, a bar plot of feature importance scores is created and shown as Figure 4.8, Figure 4.9 and Figure 4.10 for FTEST, NCA and RRF, respectively. Please be noted that the

scores given by FTEST are the negative logs of the p-values. In MATLAB, the corresponding score value is set to be Inf if a p-value is smaller than eps(0), where eps calculates the floating-point relative accuracy and eps(0) is equal to 4.9407×10^{-324}. Hence, to better visualize the score plot, the bar plot assigns Inf values the same length as the largest finite score. The RRF method may generate negative feature weights, indicating that these features are not good predictors. The plots of the feature importance scores for the filter methods are depicted in the following figures.

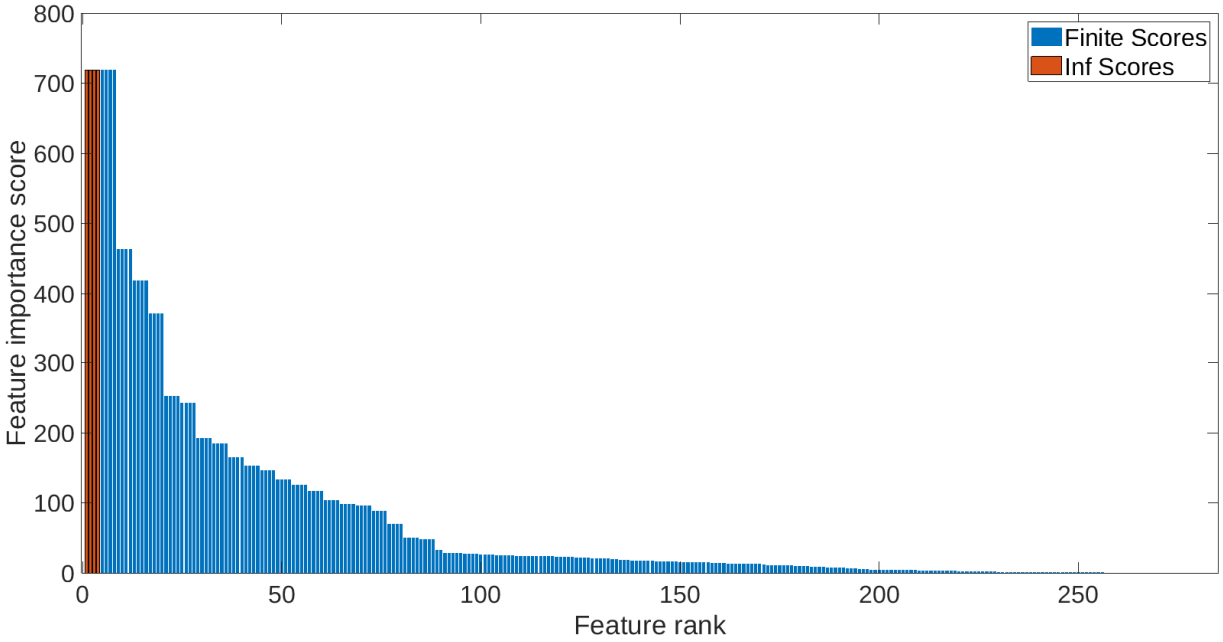


Figure 4.8 Feature importance scores given by method FTEST

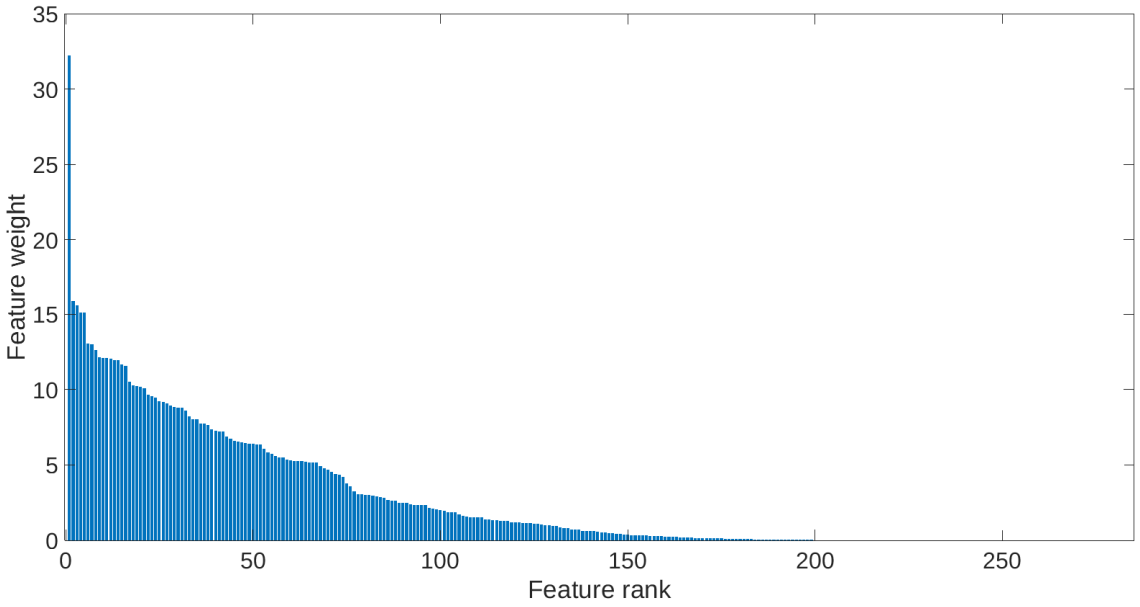


Figure 4.9 Feature weights given by method NCA

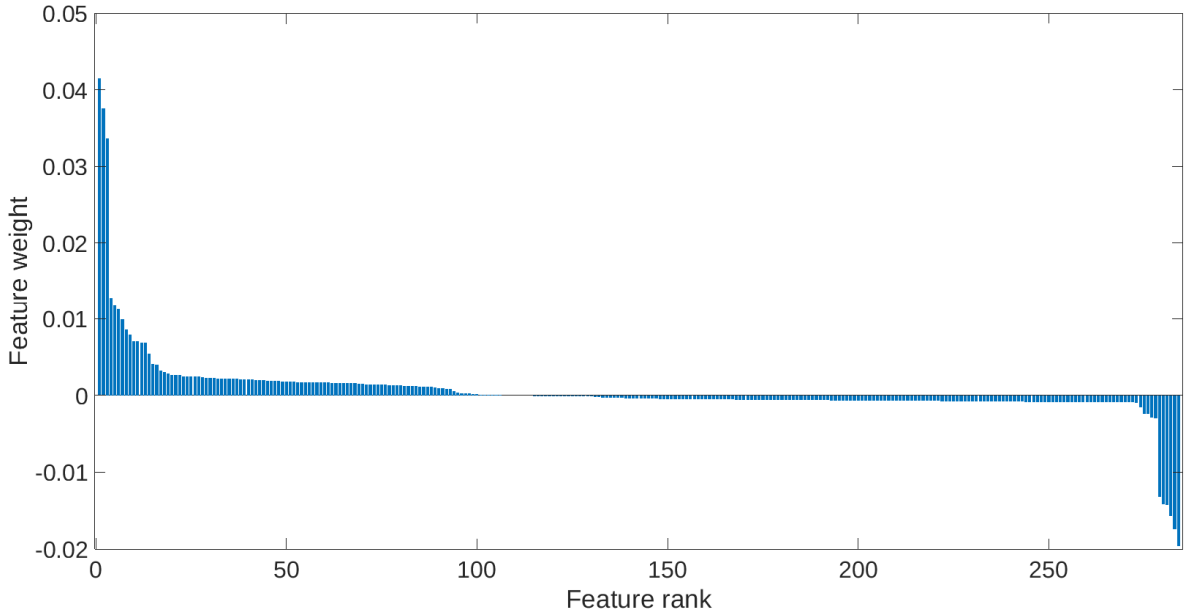


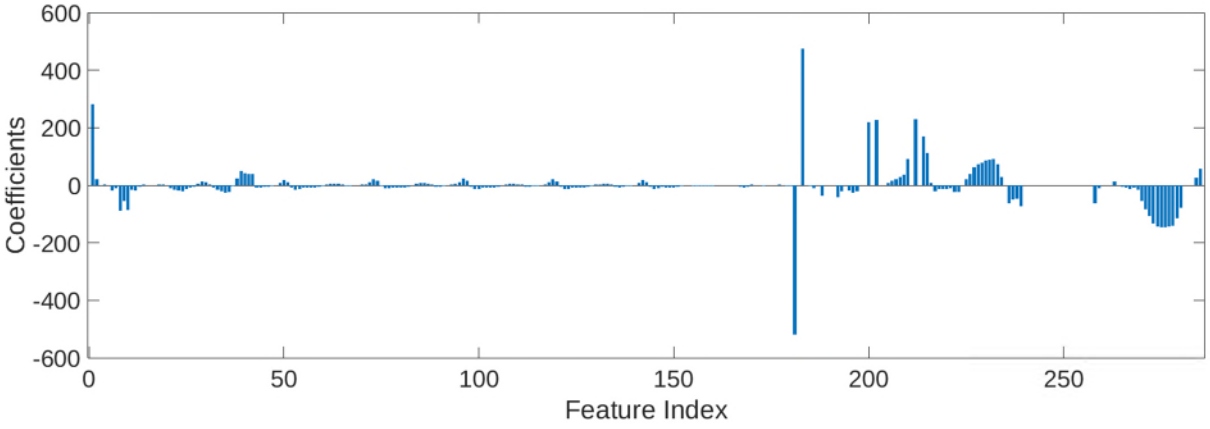
Figure 4.10 Feature weights given by method RRF

Because the filter methods only evaluate the feature importance or weight, a predetermined threshold is needed to select a certain number of features. However, manually determining such threshold is so subjective that it may deteriorate the model performance if it is not carefully selected. A more convincing way is to add a model selection process to determine the optimal

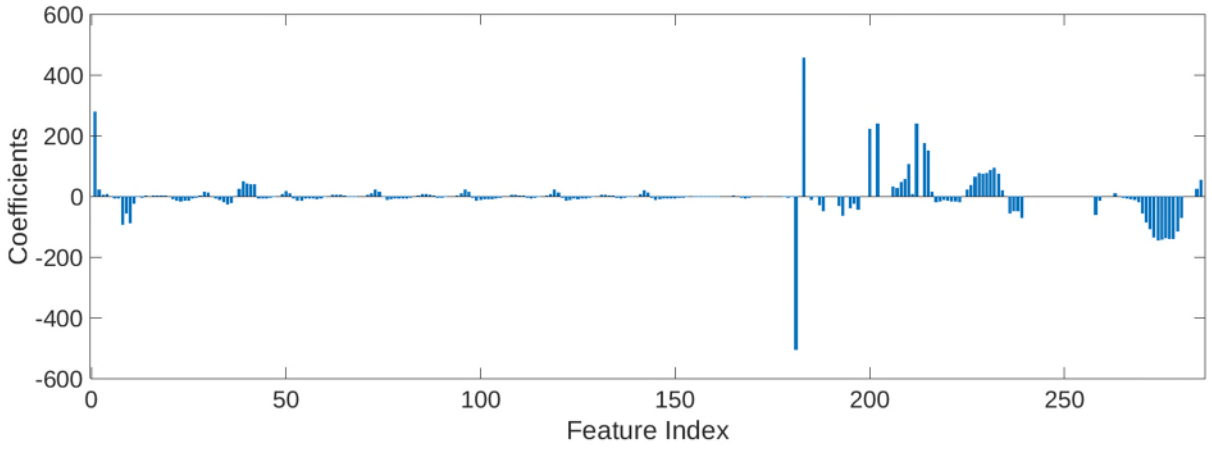
number of features, which however, requires extra data and is quite time-consuming, as discussed in the previous subsection.

The wrapper method, SFS, sequentially searches for the subset of features that best improves the model performance. Hence, the optimal subset of features is exactly the selected features.

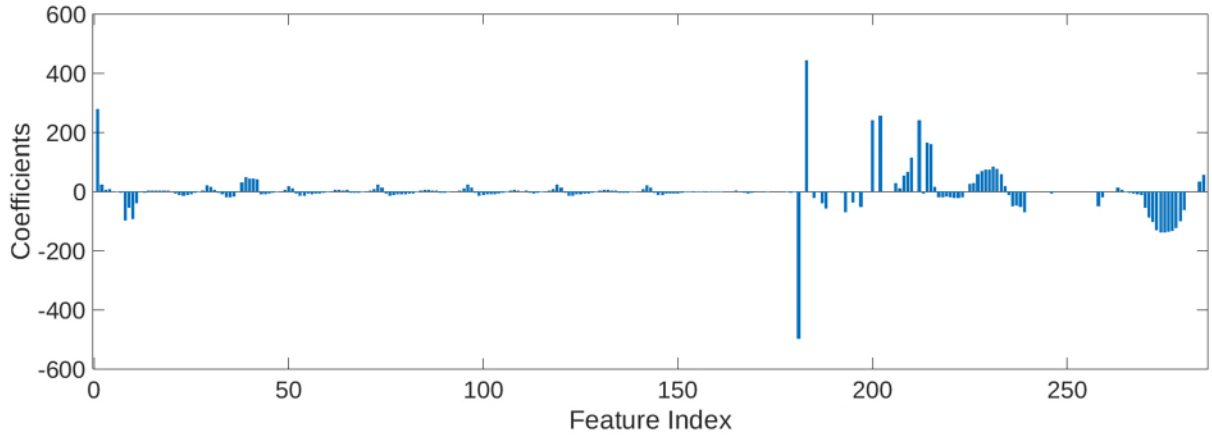
QRLASSO method introduces an L_1-norm penalty term into the quantile regression model to penalize the unimportant features. As a result, the associated coefficients of features with low importance are assigned with a very small absolute value. Filters and wrappers select features based on point forecasting and use the feature selection result to fit a PLF model, which, however, is unreasonable. It indeed saves a lot of work by simply using the same selected features for all quantiles, but in reality, the impact of each feature on each quantile could be different. A feature that has little impact on median value could be important when predicting extreme quantiles. QRLASSO allows selected features to vary among different quantiles. Figure 4.11 illustrates the estimated coefficients for three selected quantiles (0.6, 0.7 and 0.8) as an example.



(a)



(b)

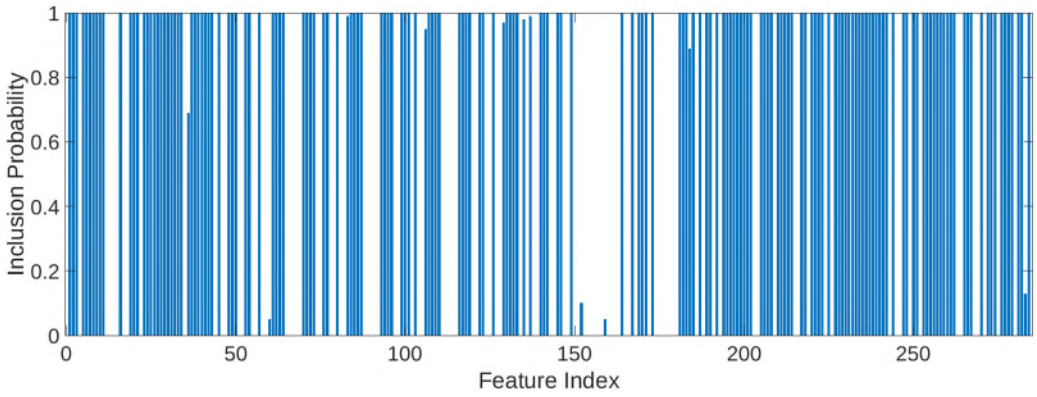


(c)

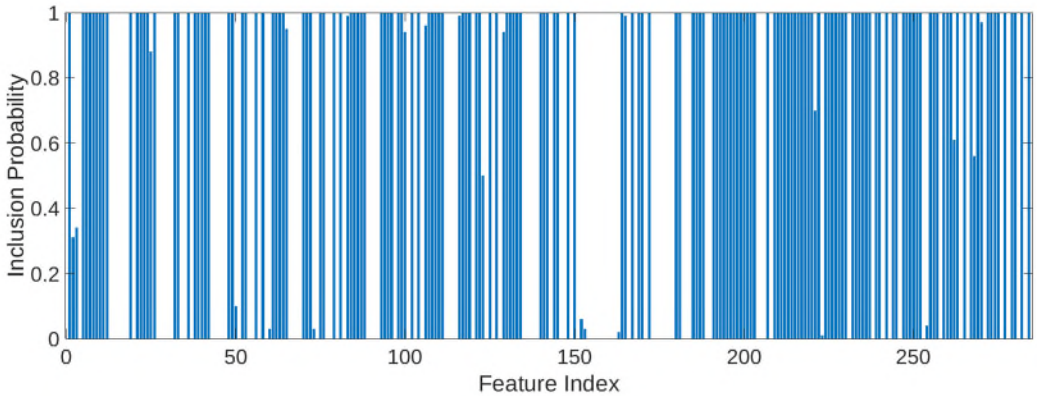
Figure 4.11 Estimated coefficients for three selected quantiles: (a) 0.6, (b) 0.7, (c) 0.8

Unlike all of the methods above, the proposed method introduces a new feature selection scheme that not only allows selected features to vary among quantiles, but also encourages all features to participate in the forecasting model with a certain probability. The results indicate that each quantile of the PFs is affected by different set of selected features, while the features selected by the filter and wrapper methods stay the same for all quantiles. For instance, an important feature would be selected with a high probability, while an unimportant feature might not be totally excluded from the model. It still could be selected but with a low probability. In other methods, the features are treated equally after they are selected. In the proposed method, the impact of each feature is controlled by the associated probability. Moreover, due to the mechanism of Gibbs sampling which updates each variable in turn by sampling from its

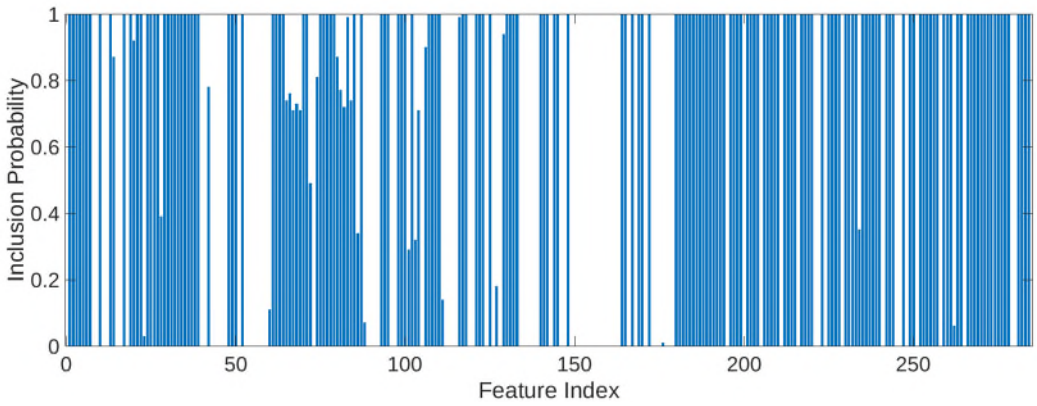
posterior conditional on other variables, the probability of a feature being selected is affected by the value of other features. This is quite true in real-world applications. For example, in extreme scenarios where the temperature is very high or very low, the customers tend to always keep their air conditioners on regardless of other conditions, which, in other words, means that the impact of other variables on the load would be less under this situation. Such feature selection scheme makes it possible to estimate more complex predictive distributions in PLF models in a practical perspective. As an example, Figure 4.12 illustrates the inclusion probabilities for all input features for three selected quantiles (0.6, 0.7 and 0.8).



(a)



(b)



(c)

Figure 4.12 Inclusion probabilities for all input features for three selected quantiles:

(a) 0.6, (b) 0.7, (c) 0.8

4.6. Summary

This chapter examines the performance of the proposed method without considering recency effect. An open dataset, the GEFCom2014 dataset, is used to compare the performance of the proposed method with several benchmarks including three filter methods, two wrapper methods, an embedded methods and two naïve methods without feature selection. The results of the first case study have been discussed, including the forecasting performance, the computation burden, and the feature selection interpretation of each tested method.

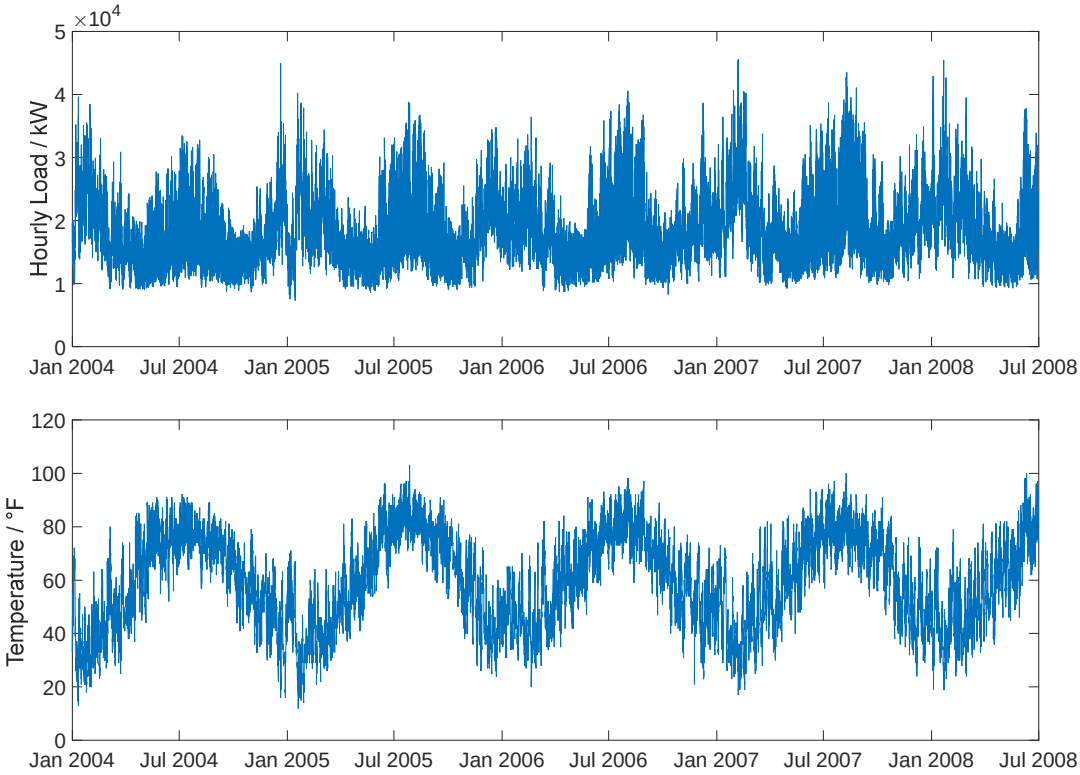
5. Case Study II: Test on Multiple Regions Considering Recency Effect

5.1. Introduction

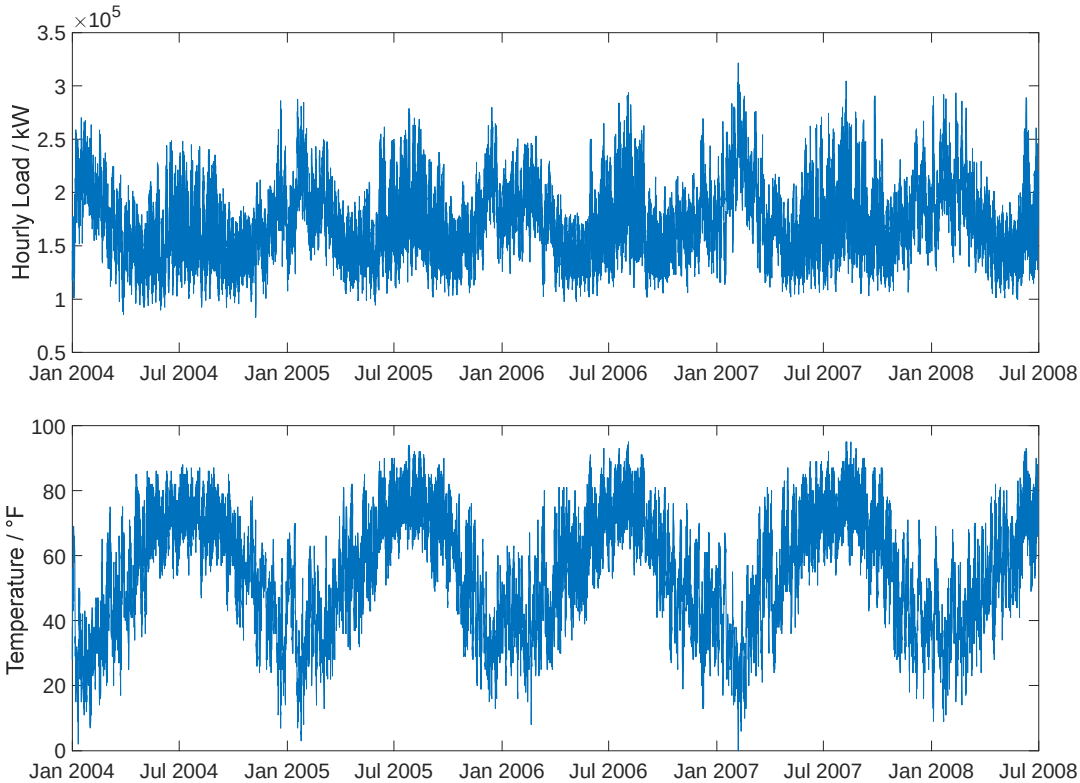
The case study in this chapter is designed to examine the model performance considering recency effect which brings in more features. As the data used in this chapter are collected from multiple regions with different weather scenarios, this case study is also aimed to further confirm that our conclusions are not restricted to one specific load zone or dataset.

5.2. Data Description and Test Settings

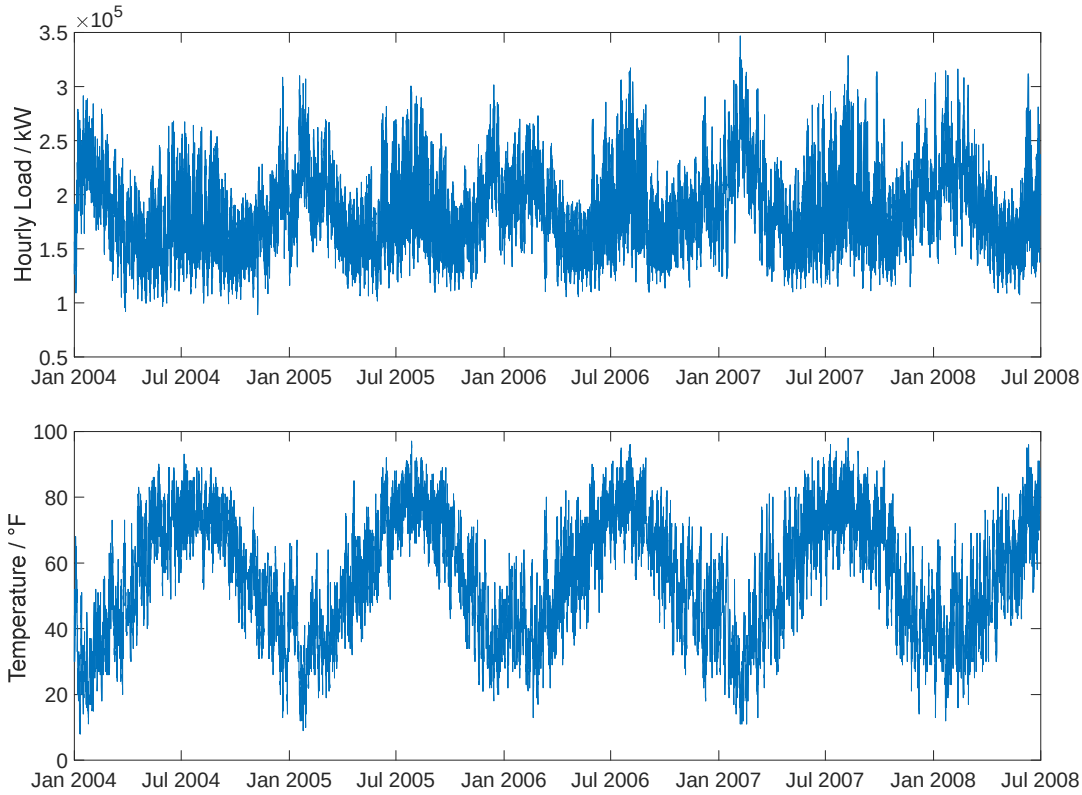
The data used in this case study is provided by the GEFCom2012 dataset. This dataset contains 54 months of hourly load data of 20 regions of North California since 1 January 2004 along with corresponding hourly temperature data from 11 weather stations. Among the 11 regions that have weather information, one region experienced a system reconfiguration during the recording period. Hence, 10 regions with weather information are selected. Figure 5.1 plots the load profile and corresponding temperature year by year for the selected regions. It can be seen from the plots that the loads of these selected zones show a clear periodic pattern correlated to temperature and calendar effects, without significant system reconfigurations. Some regions show a clear slowly increasing trend, and some keep a steady load level. It can also be seen from the figures that there is occasionally a very few of extreme or zero values in the loads which could be error readings. In this case, the wrong records are replaced by the average value of the adjacent data points.



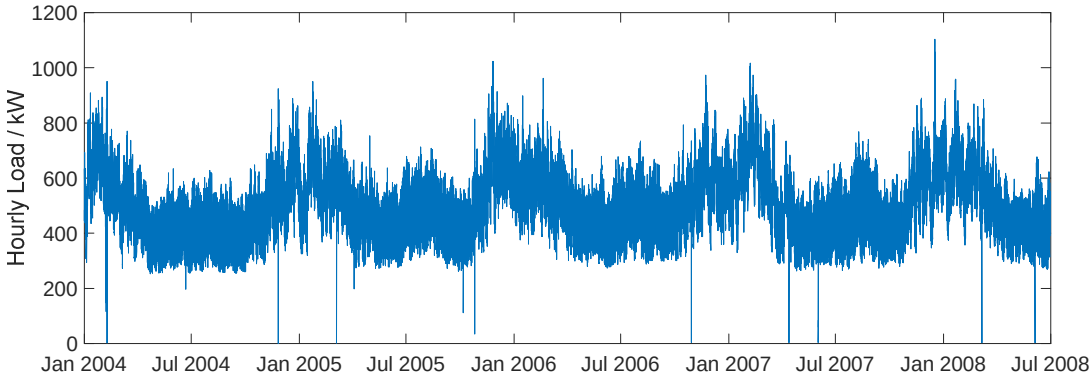
(1) Load and temperature profile of Zone 1

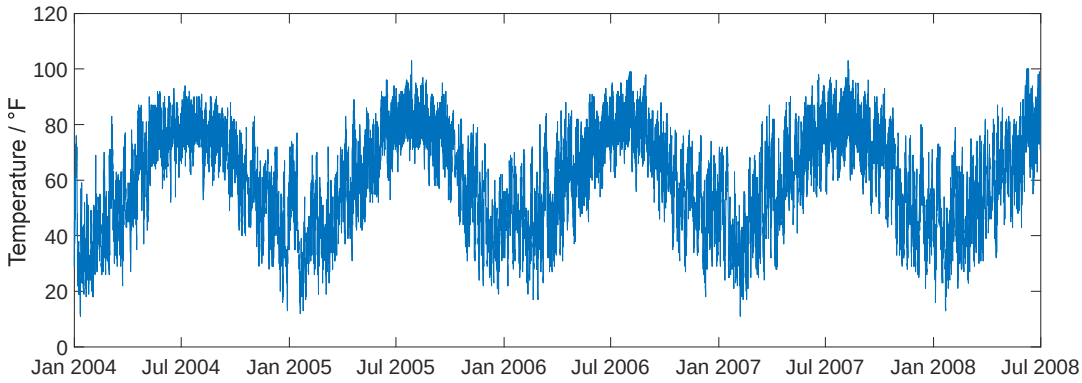


(2) Load and temperature profile of Zone 2

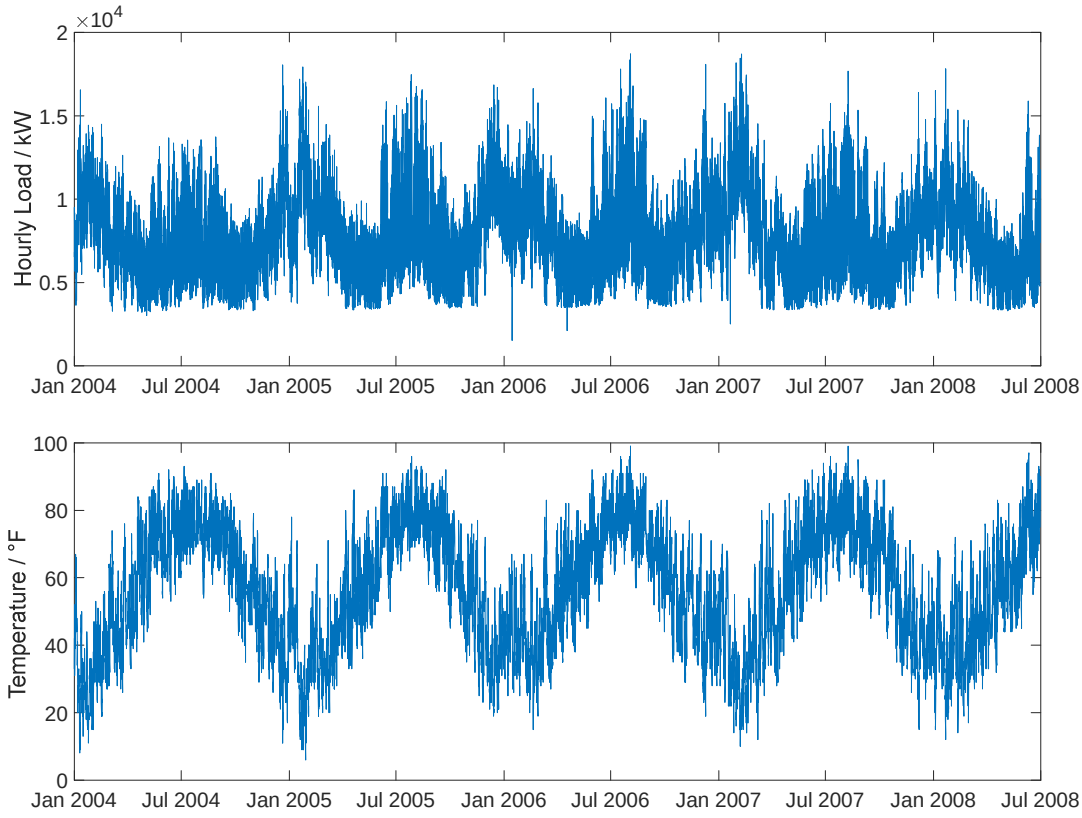


(3) Load and temperature profile of Zone 3

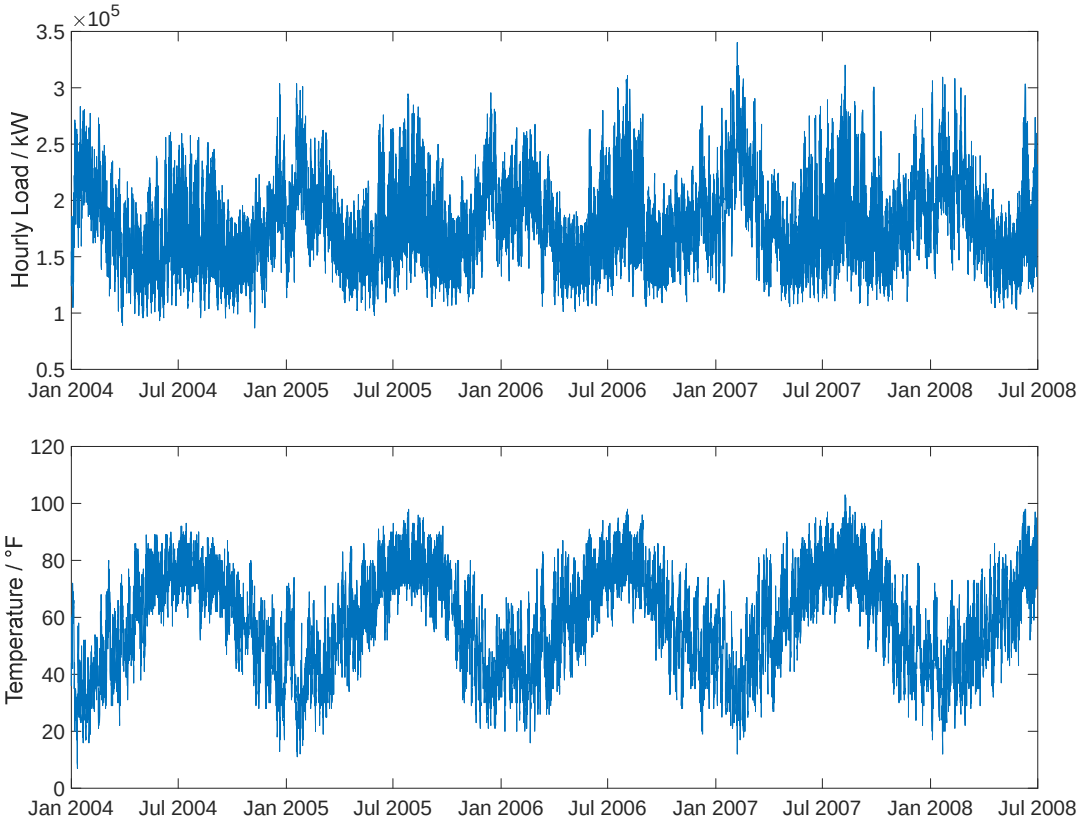




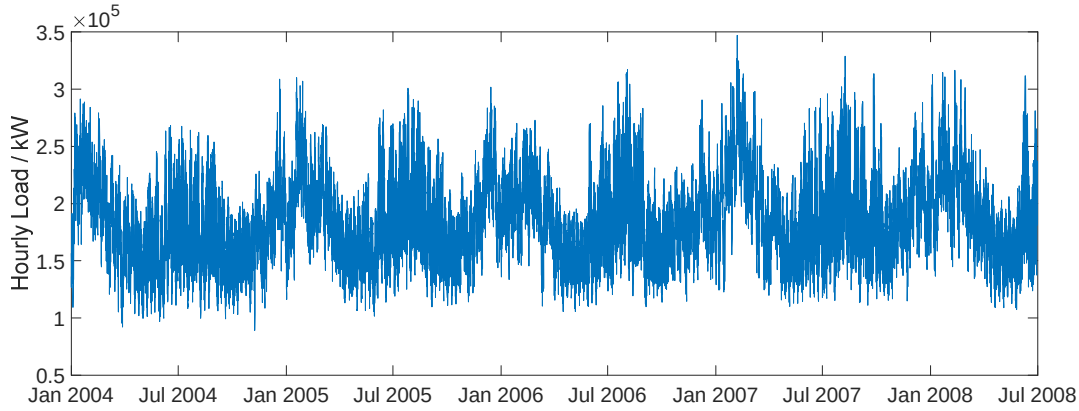
(4) Load and temperature profile of Zone 4

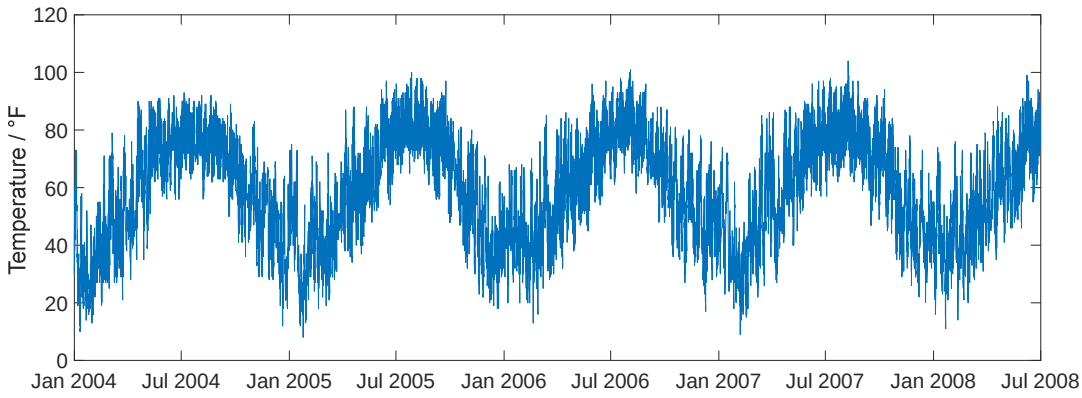


(5) Load and temperature profile of Zone 5

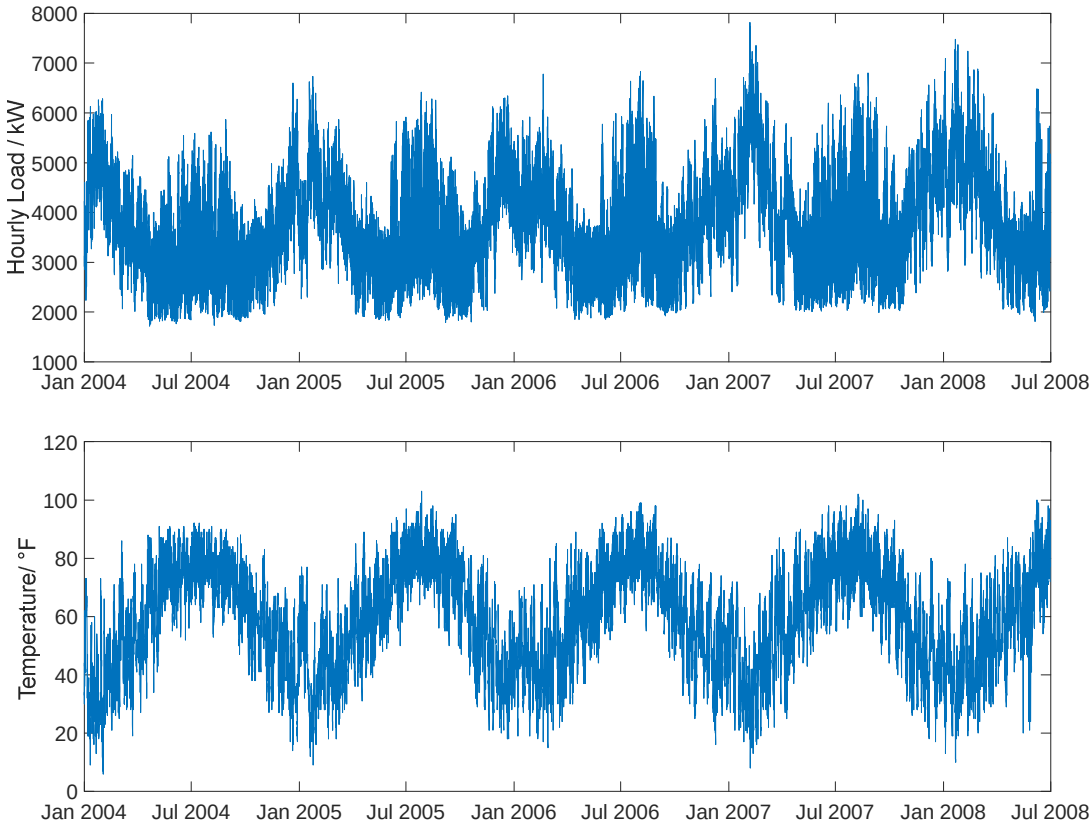


(6) Load and temperature profile of Zone 6

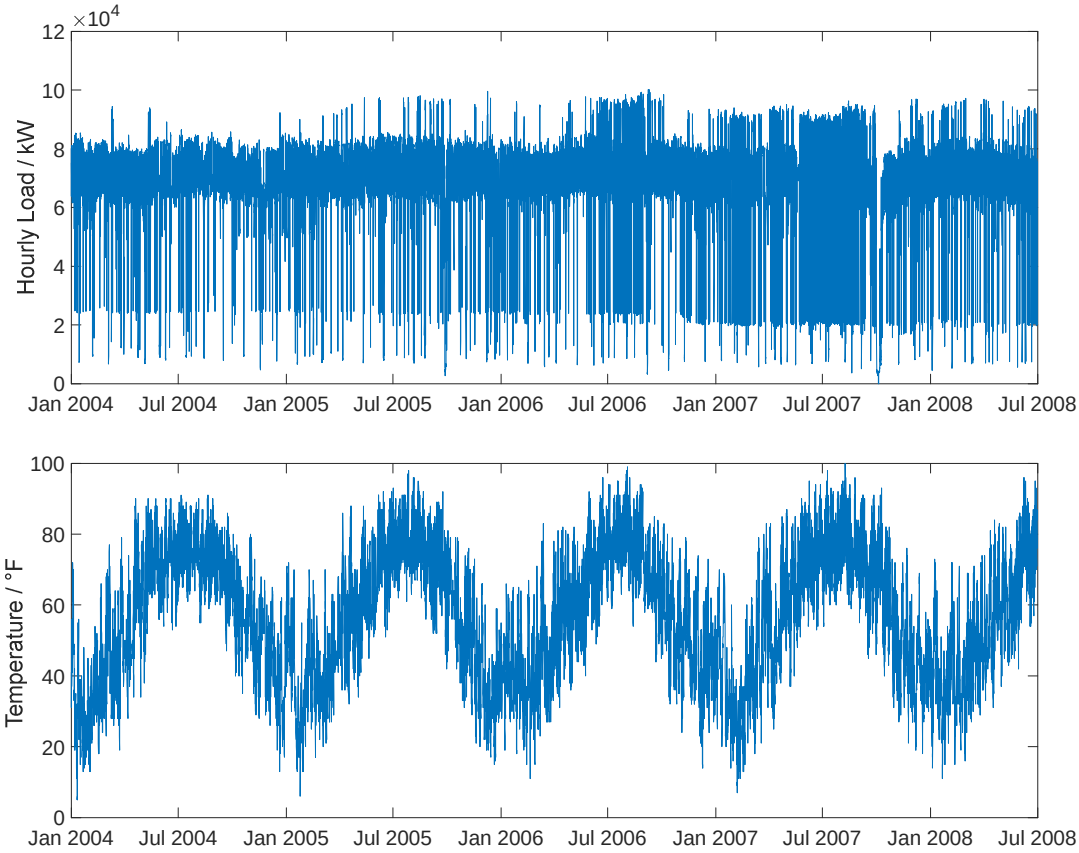




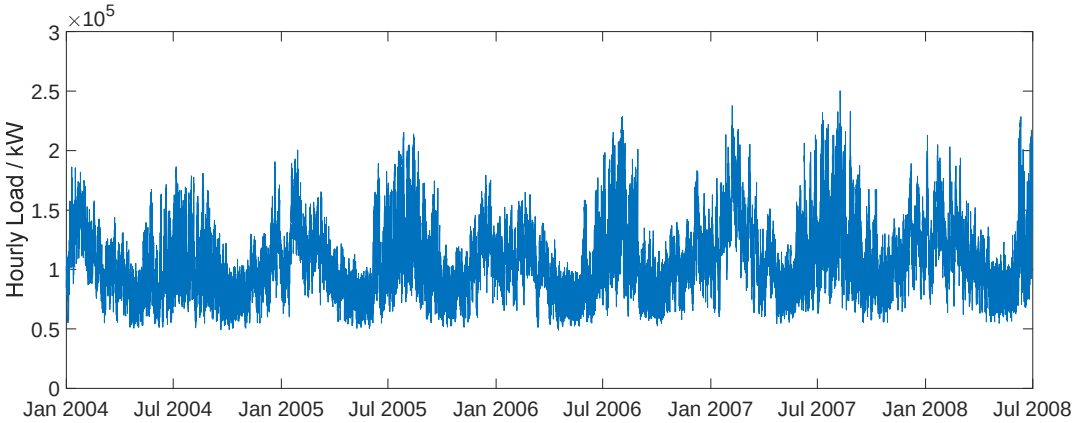
(7) Load and temperature profile of Zone 7

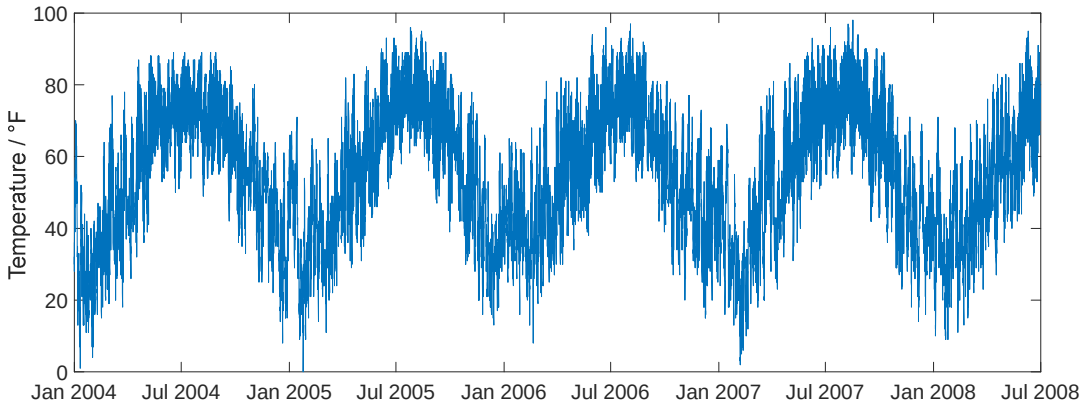


(8) Load and temperature profile of Zone 8



(9) Load and temperature profile of Zone 9





(10) Load and temperature profile of Zone 10

Figure 5.1 Overall load and temperature profile of 10 selected zones year by year from January 2004 to July 2008

From the GEFCom2012 dataset, a two-year period hourly load and temperature data from 1 May 2006 to 30 April 2008 are chosen as the training set. The following two-week data from 1 May 2008 to 14 May 2008 are used to determine the optimal number of features for filter methods. Thereafter, the two-week data from 15 May 2007 to 28 May 2007 are used for validating the performance of all the tested methods. Temperature is assumed to stay fixed within each hour. As has discussed in the previous chapter, encoding all the calendar variables by the proposed dummy encoding method, the total length of the features included in the linear model considering recency effect is $N_F = 285 + 105(N_D + N_H)$. In the case of $D = 7$, $H = 12$, the total number of features will be $N_F = 2280$. In this case study, we consider 7 different scenarios with D varying from 1 to 7 and H varying from 0 to 12 with an increment size of 2. That is $(D, H) = \{(1,0), (2,2), (3,4), (4,6), (5,8), (6,10), (7,12)\}$. Table 5.1 shows the mapping relation between the feature length and the value of (D, H).

Same as the test settings in Chapter 4, a total of $Q = 19$ quantiles for a set of probabilities $\boldsymbol{\kappa} = \{0.05, 0.1, 0.15, \ldots, 0.9, 0.95\}$ are used to form the PLF. The error measurement stays the same which is given by quantile score.

Table 5.1 Mapping relation between the feature length and the value of (D, H)

$(\boldsymbol{D}, \boldsymbol{H})$	**Num. of features**
$(\mathbf{1}, \mathbf{0})$	390
$(\mathbf{2}, \mathbf{2})$	705
$(\mathbf{3}, \mathbf{4})$	1020
$(\mathbf{4}, \mathbf{6})$	1335
$(\mathbf{5}, \mathbf{8})$	1650
$(\mathbf{6}, \mathbf{10})$	1965
$(\mathbf{7}, \mathbf{12})$	2280

5.3. Benchmarks

For consistency, this chapter uses the same benchmarks as Chapter 4, including three filter methods, one wrapper method, one embedded method and two naïve benchmarks without feature selection. The denotations of these methods stay the same as those in the previous chapter. Due to the heavy computation burden of the process of determining optimal number of features for filters methods, the number of features included is not increased by 1 but at a step of 50 to reduce the computation time. Otherwise, the process may take over months.

5.4. Case Studies and Results of the GEFCom2012 Dataset

In this simulation, data from 10 different zones with different temperature conditions are tested. The overall performance, and the performance of the tested methods over a set of quantiles are examined, with the consideration of recency effect. The objective of this case study is mainly to answer the following questions:

- Will adding recency effect to the tested models improve the forecasting performance?
- Is the included recency effect the longer the better?
- What is the performance difference among the tested methods?

- Does the conclusion drawn in the previous chapter still hold with different datasets?

Table 5.2 – 5.11 present the simulation results of the 10 different zones, showing the overall quantile scores for scenarios of different recency effects.

Table 5.2 Overall quantile scores of all methods for Zone 1 considering different recency effect

	BQLRFS	FTEST	NCA	RRF	SFS	QRLASSO	QLR	QRNN
D0H0	499.94	733.47	515.66	733.47	510.01	753.71	733.47	751.75
D1H0	**419.26**	616.61	616.62	617.36	420.83	655.42	616.62	736.48
D2H2	475.37	614.04	615.37	603.79	506.82	667.11	615.37	729.10
D3H4	481.32	596.64	603.40	588.88	521.05	656.70	603.40	731.51
D4H6	485.92	600.43	450.49	572.72	522.92	646.81	592.78	650.94
D5H8	481.23	595.43	534.57	553.47	539.97	641.20	595.32	680.60
D6H10	509.81	613.47	544.91	538.84	558.86	662.72	633.41	709.57
D7H12	532.64	684.25	581.55	544.19	587.54	695.47	653.52	738.74

Table 5.3 Overall quantile scores of all methods for Zone 2 considering different recency effect

	BQLRFS	FTEST	NCA	RRF	SFS	QRLASSO	QLR	QRNN
D0H0	2409.44	2414.75	2414.74	2348.67	2165.77	2552.53	2414.75	2607.12
D1H0	**1987.94**	2002.47	2002.47	2191.71	2225.07	2088.35	2002.49	2068.41
D2H2	2009.07	2010.37	2018.59	2321.03	2268.45	2071.89	2010.36	2155.74
D3H4	2010.45	2016.97	2050.23	2486.45	2136.60	2085.02	2016.97	2252.23
D4H6	2056.74	2058.12	2058.12	2058.12	2030.03	2096.16	2058.12	2299.32
D5H8	2082.20	2067.22	2067.22	3098.53	2457.33	2082.53	2067.22	2361.98
D6H10	2145.45	2130.05	2156.24	3043.32	2566.87	2101.63	2130.05	2592.03
D7H12	2201.46	2347.13	2268.49	3099.47	2636.11	2204.40	2243.21	2705.78

Table 5.4 Overall quantile scores of all methods for Zone 3 considering different recency effect

	BQLRFS	FTEST	NCA	RRF	SFS	QRLASSO	QLR	QRNN
D0H0	2312.29	2679.81	2679.81	2592.41	2345.50	2839.53	2679.80	2933.84
D1H0	**2003.33**	2519.62	2519.62	2124.41	2109.27	2526.40	2519.62	2531.97
D2H2	2091.25	2412.09	2412.09	2595.21	2125.08	2450.52	2412.09	2586.78
D3H4	2119.67	2431.79	2195.87	2431.79	2143.73	2344.55	2431.79	2556.32
D4H6	2108.01	2413.40	2413.40	2413.40	2149.50	2317.11	2413.40	2609.78
D5H8	2130.52	2236.98	2236.98	2236.98	2192.62	2201.71	2236.98	2757.43
D6H10	2313.12	2647.41	2670.65	2647.41	2387.36	2287.94	2647.41	2809.88
D7H12	2428.02	2777.12	2777.12	2777.12	2454.78	2513.11	2777.12	2919.82

Table 5.5 Overall quantile scores of all methods for Zone 4 considering different recency effect

	BQLRFS	FTEST	NCA	RRF	SFS	QRLASSO	QLR	QRNN
D0H0	8.08	10.71	10.71	10.83	9.83	10.77	10.71	11.10
D1H0	7.94	10.46	10.33	10.44	8.12	10.25	10.32	10.88
D2H2	**7.86**	10.47	10.14	10.42	9.46	9.99	10.36	11.10
D3H4	7.93	11.35	11.28	9.86	9.43	10.32	11.29	10.53
D4H6	8.21	11.47	9.33	9.66	9.21	10.27	11.33	11.02
D5H8	8.23	11.56	11.60	9.55	9.72	10.10	11.43	10.89
D6H10	8.39	11.52	11.46	9.30	9.96	10.12	11.33	11.19
D7H12	8.50	11.62	11.61	9.09	10.37	10.44	11.19	11.65

Table 5.6 Overall quantile scores of all methods for Zone 5 considering different recency effect

	BQLRFS	FTEST	NCA	RRF	SFS	QRLASSO	QLR	QRNN
D0H0	140.49	162.65	163.03	162.65	168.05	166.87	162.65	138.23
D1H0	136.16	168.88	168.88	165.34	156.86	168.98	168.88	139.33
D2H2	**133.20**	164.60	162.74	139.95	186.52	166.21	165.27	139.15
D3H4	141.52	178.16	178.16	159.39	185.23	173.50	178.16	142.94
D4H6	143.16	185.05	183.34	161.27	200.20	176.50	185.05	145.06
D5H8	154.46	189.03	189.03	160.23	183.68	176.11	189.03	167.42
D6H10	155.76	192.56	188.62	157.58	188.69	173.75	192.56	163.06
D7H12	160.85	192.36	192.36	157.23	192.14	174.45	192.36	166.58

Table 5.7 Overall quantile scores of all methods for Zone 6 considering different recency effect

	BQLRFS	FTEST	NCA	RRF	SFS	QRLASSO	QLR	QRNN
D0H0	2372.77	2404.42	2404.42	2623.13	2351.37	2574.47	2404.43	2535.98
D1H0	**2139.71**	2248.42	2248.43	2458.09	2195.46	2320.53	2248.43	2290.21
D2H2	2173.20	2272.86	2269.13	2627.13	2262.49	2289.12	2269.13	2631.40
D3H4	2208.48	2288.86	2271.66	3089.50	2314.88	2268.02	2288.86	2461.44
D4H6	2287.97	2265.94	2600.97	2966.41	2376.75	2262.76	2265.94	2473.99
D5H8	2296.58	2262.71	2242.54	2657.96	2494.94	2231.84	2262.71	2689.64
D6H10	2310.33	2288.76	2709.90	2610.20	2558.57	2241.80	2288.76	2838.48
D7H12	2321.13	2321.80	2776.62	2851.60	2687.14	2331.42	2321.80	2998.44

Table 5.8 Overall quantile scores of all methods for Zone 7 considering different recency effect

	BQLRFS	FTEST	NCA	RRF	SFS	QRLASSO	QLR	QRNN
D0H0	2453.56	2929.51	2929.51	2745.63	2413.07	3079.82	2929.51	2975.72
D1H0	**2171.04**	2493.42	2493.41	2384.37	2326.77	2659.54	2493.12	2638.81
D2H2	2218.24	2517.98	2434.23	2517.98	2336.08	2650.92	2517.98	2812.34
D3H4	2455.23	2560.89	2560.90	2560.90	2444.92	2611.61	2560.89	2828.40
D4H6	2308.94	2576.78	2716.28	2576.78	2499.74	2622.54	2576.78	2955.01
D5H8	2340.00	2599.69	2473.69	2599.69	2483.69	2610.00	2599.69	3161.49
D6H10	2646.78	2887.91	2825.81	2887.91	2612.55	2661.47	2887.91	3199.09
D7H12	2546.38	2901.24	2845.90	2901.24	2766.87	2858.11	2901.24	3225.91

Table 5.9 Overall quantile scores of all methods for Zone 8 considering different recency effect

	BQLRFS	FTEST	NCA	RRF	SFS	QRLASSO	QLR	QRNN
D0H0	83.22	88.03	84.68	84.70	103.22	98.69	84.68	101.69
D1H0	**81.59**	82.35	82.09	87.22	103.50	91.23	82.38	98.21
D2H2	102.40	118.87	119.29	105.92	105.06	85.53	119.35	94.63
D3H4	114.80	140.14	140.14	132.67	124.07	83.86	140.14	96.96
D4H6	121.82	156.17	121.00	152.31	125.63	85.74	156.17	99.10
D5H8	131.87	178.40	183.24	167.39	140.72	89.37	178.40	103.42
D6H10	149.22	200.04	158.32	174.05	155.14	92.47	197.92	111.45
D7H12	150.19	194.93	190.44	187.82	168.55	104.47	191.40	122.37

Table 5.10 Overall quantile scores of all methods for Zone 9 considering different recency effect

	BQLRFS	FTEST	NCA	RRF	SFS	QRLASSO	QLR	QRNN
D0H0	3140.45	3271.72	3346.69	3198.70	3800.67	3301.31	3173.79	3703.56
D1H0	**3095.20**	3591.89	3277.18	3273.91	3626.97	3250.63	3123.95	3608.95
D2H2	3100.32	3505.99	3261.75	3302.13	3875.40	3277.74	3141.21	3579.51
D3H4	3166.50	3413.46	3305.25	3353.86	3951.73	3275.55	3175.82	3607.67
D4H6	3177.68	3423.19	3292.76	3365.95	3910.90	3330.19	3259.22	3702.55
D5H8	3263.98	3364.21	3282.85	3311.75	3864.16	3341.02	3374.85	3521.81
D6H10	3414.73	3326.88	3308.08	3347.12	3897.66	3345.38	3599.14	3626.44
D7H12	3620.36	3327.86	3271.53	3356.78	3962.41	3411.20	4010.33	3717.88

Table 5.11 Overall quantile scores of all methods for Zone 10 considering different recency effect

	BQLRFS	FTEST	NCA	RRF	SFS	QRLASSO	QLR	QRNN
D0H0	1765.34	1911.03	1911.03	1911.03	1861.57	1973.05	1911.03	2477.21
D1H0	1170.60	1274.42	1274.40	1262.52	1668.87	1365.55	1274.42	1694.75
D2H2	**1140.22**	1194.84	1194.14	1225.02	1856.45	1278.42	1194.14	1491.92
D3H4	1161.41	1203.60	1221.87	1239.41	2038.89	1278.13	1206.27	1533.22
D4H6	1166.61	1202.37	1213.25	1300.34	2061.15	1292.48	1202.09	1745.08
D5H8	1203.73	1255.04	1258.72	1294.19	1963.64	1232.55	1255.04	1870.89
D6H10	1213.40	1277.41	1297.61	1266.73	2010.83	1284.82	1277.41	1754.98
D7H12	1319.97	1397.18	1443.50	1315.72	2112.36	1389.77	1389.03	1889.92

Conditional formatting is applied to the table cells for better visualization. A color gradient is applied to all cells of each table based on their values. The smaller/larger the value is, the greener/redder the color is. As a reference, the result without considering recency effect is

added to the first row of each table.

At a first glance, we can see from the tables that the lowest quantile score in each column of each table does not locate in the first row. This result conveys the information that considering recency effect does improve the performance of the tested models. It can also be noticed that the lowest quantile score in each column does not happen in the last row, indicating that longer recency effect does not usually lead to more accurate results. Additionally, Table 5.12 summarizes where the minimum quantile score happens for each zone. It is worth noting that D1H0 occupies 7 out of 10 minimum values and D2H2 takes the rest, indicating that in our case, the PLF performance could be significantly improved simply by adding a relatively short length of recency effect. Hence, in real practices, it is suggested that we go through a model selection process that tests multiple models with different length of recency effect and select the one with the best performance.

Table 5.12 Where the minimum quantile score happens for each zone

Zone ID	*D0H0*	*D1H0*	*D2H2*	*D3H4*	*D4H6*	*D5H8*	*D6H10*	*D7H12*
1		√						
2		√						
3		√						
4			√					
5			√					
6		√						
7		√						
8		√						
9		√						
10			√					

To give a deeper and comprehensive discussion on the performance of all tested methods, the average quantile score and the lowest quantile score given by each method for every zone is calculated and shown in Table 5.13 and Table 5.14. The average quantile score reflects the

overall model performance when handling different length of recency effect. From Table 5.13, we can see that the proposed method outperforms the benchmark methods in most scenarios (9 out of 10 zones) in terms of average quantile score, while the benchmark methods show unstable performance when dealing with different zones. Among the benchmark methods with feature selection, QRLASSO is relatively better than the others. For the two naïve benchmark methods without feature selection, it can be noted that the nonlinear method QRNN has even worse performance compared to the linear method QLR. Further, it can be clearly seen from Table 5.14 that, in each scenario, the smallest quantile score is always given by the proposed method compared to other benchmarks, although the proposed method may not have the best performance when no recency effect is included and when other length of recency effect is considered in some cases. Besides, it can be noticed that even though QRLASSO has relatively better performance in terms of average quantile score among the benchmark methods, it shows worse performance when it comes to the minimum quantile score. The results suggest that while QRLASSO maintains relatively stable performance across various recency effect scenarios, it exhibits limited sensitivity to the impact of changes in feature length stemming from the presence of differing recency effects. This may be attributed to QRLASSO's limitations in addressing sparse and high-dimensional feature inputs arising from the incorporation of recency effects. On the contrary, the proposed method shows the best performance in terms of both average quantile score and minimum quantile score, confirming the effectiveness of the feature selection process.

Furthermore, this result also confirms that the proposed method maintains a robust performance when applied to different datasets.

Table 5.13 Average quantile score given by each method for Zone 1 ~ 10

Zond ID	BQLRFS	FTEST	NCA	RRF	SFS	QRLASSO	QLR	QRNN
1	485.69	631.79	557.82	594.09	521.00	672.39	630.49	716.09
2	2112.84	2130.89	2129.51	2580.91	2310.78	2160.31	2117.90	2380.33
3	2188.28	2514.78	2488.19	2477.34	2238.48	2435.11	2514.78	2713.23
4	8.14	11.15	10.81	9.89	9.51	10.28	11.00	11.05
5	145.70	179.16	178.27	157.96	182.67	172.05	179.25	150.22
6	2263.77	2294.22	2440.46	2735.50	2405.20	2315.00	2293.76	2614.95
7	2392.52	2683.43	2659.97	2646.81	2485.46	2719.25	2683.39	2974.60
8	116.89	144.87	134.90	136.51	128.24	91.42	143.81	103.48
9	3247.40	3403.15	3293.26	3313.78	3861.24	3316.63	3357.29	3633.55
10	1267.66	1339.49	1351.82	1351.87	1946.72	1386.85	1338.68	1807.25

Table 5.14 Minimum quantile score given by each method for Zone 1 ~ 10

Zond ID	BQLRFS	FTEST	NCA	RRF	SFS	QRLASSO	QLR	QRNN
1	419.26	595.43	450.49	538.84	420.83	641.2	592.78	650.94
2	1987.94	2002.47	2002.47	2058.12	2030.03	2071.89	2002.49	2068.41
3	2003.33	2236.98	2195.87	2124.41	2109.27	2201.71	2236.98	2531.97
4	7.86	10.46	9.33	9.09	8.12	9.99	10.32	10.53
5	133.20	162.65	162.74	139.95	156.86	166.21	162.65	138.23
6	2139.71	2248.42	2242.54	2458.09	2195.46	2231.84	2248.43	2290.21
7	2171.04	2493.42	2434.23	2384.37	2326.77	2610.00	2493.12	2638.81
8	81.59	82.35	82.09	84.70	103.22	83.86	82.38	94.63
9	3095.20	3271.72	3261.75	3198.70	3626.97	3250.63	3123.95	3521.81
10	1140.22	1194.84	1194.14	1225.02	1668.87	1232.55	1194.14	1491.92

5.5. Summary

This chapter examines the model performance considering recency effect. The historical records containing different load levels and temperature scenarios of 10 different zones are used to validate the effectiveness of the methods. For each zone, 7 different length of recency effects are tested, and the outcome without considering any recency effect is used as a reference. The average quantile score and the minimum quantile score are calculated and shown for a comprehensive comparison. This case study shows the effectiveness of the proposed method when recency effect is included and further confirms the robustness of the proposed method when it is applied to a variety of different datasets.

6. Case Study III: A Wrapper Approach - Applying the Feature Selection Result to Nonlinear Forecasting Models

6.1. Introduction

Rich literature exists on load forecasting, among which nonlinear forecasting models are more attractive as they can map the nonlinear relation between the predictors and the response. However, nonlinear models for PLF have a much more complex mechanism compared to linear models, especially those artificial intelligent methods which lack proper interpretability. This makes it quite difficult to design an efficient and effective feature selection process for nonlinear PLF models.

In this regard, the case study in this chapter uses the proposed method as a wrapper approach, which is thereafter applied to a nonlinear model to validate if it could improve the performance of such model.

6.2. Data Description and Test Settings

This case study uses the same data provided by the GEFCom2012 as Chapter 5. The same 10 zones and the historical records including load and temperature with the same start and end date are used in this chapter. Same as the test settings in Chapter 4 and Chapter 5, a total of $Q = 19$ quantiles for a set of probabilities $\boldsymbol{\kappa} = \{0.05, 0.1, 0.15, \dots, 0.9, 0.95\}$ are used to form the PLF. The error measurement stays the same which is given by quantile score.

6.3. Nonlinear PLF Model

To make the results comparable, a popular quantile regression based nonlinear PLF model is selected for simulation. This model is a widely used tree-based method, the quantile random forest [76], which is introduced as follows, with a similar notation used in [77].

A decision tree (classification or regression tree) goes through the decisions from the top root down to the bottom leaf node to predict a response. A random forest uses the training data to grow an ensemble of decision trees. The grown of trees of a random forest is endowed with randomness and the final prediction is obtained by averaging over the responses of all grown trees. Formally,

let ϑ denote the random parameters that determine the grown of a tree;

let $T(\vartheta)$ denote the tree that corresponds to ϑ;

let $\omega(x, \vartheta)$ be the weight vector that is associated with ϑ and data point $(X = x, Y = y)$;

let k denote the number of trees in the random forest ensemble;

let n denote the number of observations used in the training stage;

Under these settings, the process of estimating the empirical conditional distribution of the response can be summarized as below. We do not go over the mathematical formulations of how the weights are calculated in detail, which can be easily found in [76].

1) Establish k trees following the mechanism of growing a random forest.
2) Drop the training observations down through all the decision trees in the ensemble. Compute the weight $\omega_i(x_i, \vartheta_t)$ for every observation i. Then, compute the weight averaged over the whole collection of trees:

$$\omega_i(x) = \frac{1}{k}\sum_{t=1}^{k} \omega_i(x_i, \vartheta_t) \tag{6.1}$$

3) Then, compute the conditional distribution for the i^{th} observation:

$$\hat{F}(y|X = x_i) = \sum_{j=1}^{n} \omega_j(x_i)\, I\{Y_j \le y\} \tag{6.2}$$

4) Thereafter, the p^{th} conditional quantile of the observation y can be calculated by

$$Q_p(x_i) = \inf\{y: \hat{F}(y|X = x_i) \geq p\} \tag{6.3}$$

Lastly, it is worth mentioning that random forests only keep the mean value of the observations, while quantile random forests evaluate the empirical conditional distribution of the response based on the value of all observations, which mines the information of the data to a larger extent.

6.3. Case Studies and Results

The case studies in this chapter focus on two comparisons. First, we compare the performance of the proposed method with that of the nonlinear model. The features that are inputted into the nonlinear model are the feature selection result given by the proposed method. Second, we make efforts to find out if such a wrapper approach could improve the performance of the nonlinear model. For each zone, we run the simulation based on the length of recency effect that is determined by where the minimum quantile score happens in the previous chapter, as given by Table 5.12. Note that in this table, random forest is denoted by RF, and feature selection is denoted by FS.

It can be clearly seen from the table that, the proposed method BQLRFS still outperforms the nonlinear model in the cases of all 10 zones, even when feature selection is considered, confirming the superiority of the proposed method. This result also indicates that we might not always pursue the complexity of a model. Sometimes a simpler model such as a linear model could provide a more efficient and effective outcome. By comparing the results obtained from the random forest-based model with and without feature selection, we can see a slight improvement in the quantile scores after the feature selection method is applied. Hence, it can be easily concluded that the proposed feature selection method could be used as a wrapper approach to improve the forecasting performance of other forecasting models.

Table 6.1 Forecasting performance of the tested nonlinear model

Zone ID	Recency effect	BQLRFS	RF without FS	RF with FS
1	D1H0	419.26	520.80	514.73
2	D1H0	1987.94	2992.81	2915.71
3	D1H0	2003.33	3775.80	3686.33
4	D2H2	7.86	10.34	10.28
5	D2H2	133.20	170.86	170.49
6	D1H0	2139.71	3909.07	3865.14
7	D1H0	2171.04	3379.70	3293.95
8	D1H0	81.59	140.04	141.74
9	D1H0	3095.20	3077.83	2974.53
10	D2H2	1140.22	1706.57	1703.69

6.4. Summary

In this chapter, the proposed method is used as a wrapper approach and applied to a nonlinear model, the quantile random forest. Two comparisons are made in the simulation. First, we compare the performance of the proposed method with the nonlinear model, the input feature of which are the feature selection result of the proposed method. Second, we also compare the performance of the nonlinear model with and without feature selection. This chapter explores a further application of the proposed method that it can be used as a wrapper approach. The case study validates the effectiveness of this approach.

7. Summary and Conclusions

It is believed that in future power grids much more variability and volatility in system load will be seen, both temporarily and spatially. To handle and consider the corresponding high uncertainty, there has been increased recent interest in PLF. Most of the relevant works focus on establishing and optimizing the predictive model, however, with very few attentions paid to the feature selection phase. In fact, feature selection is quite essential to the area of forecasting as it is aimed to avoid the curse of dimensionality, reduce modeling complexity, reduce the risk of over-fitting and improve the forecasting performance. In this regard, this research intends to develop a novel embedded feature selection framework for PLF, which is applicable to both linear and nonlinear predictive models.

Chapter 1 gives a brief introduction to electric load forecasting and the transition from point load forecasting to PLF. The literature on the topic of feature selection for load forecasting are briefly reviewed. Chapter 2 introduces the background of feature selection and briefly reviews the state-of-art techniques. Both advantages and limitations of these techniques, including filter methods, wrapper methods, embedded methods and hybrid methods, are discussed.

Chapter 3 presents the proposed framework for feature selection and the methodology of its application to a linear model, the quantile linear regression, is introduced. The methodology for a nonlinear model, the QRNN, is still under development and will be discussed in future steps of this research. For the linear model, an embedded feature selection structure is incorporated to identify and select subsets of input features by introducing an inclusion indicator variable for each feature. Then, Bayesian inference is applied to the model with a sparseness favoring prior endowed over the inclusion indicator variables. To tackle with the problem of the computation of posteriors which is almost intractable, an MCMC approach is adopted to sample the parameters from the posteriors. Finally, we use discrete formulas applied to the samples from the posterior distribution to summarize our knowledge of the parameters. Bayesian inference

allows each quantile of the distribution of the dependent load to be affected by different sets of features, and therefore allows us to estimate complex predictive densities more accurately. Besides, it also allows the estimation of the inclusion probabilities for all input features.

Comprehensive case studies are carried out in Chapter 4 to validate the performance of the proposed framework on the quantile linear regression model without considering recency effect. A case study is designed based on an open dataset for one supply zone. In the case study, the proposed method is compared with a wide range of state-of-the-art benchmarks including three filter methods, one wrapper methods, an embedded method and two origin models without feature selection. The results show that the proposed method BQLRFS has the best overall performance among all the tested methods. The computation complexity of each method is also provided. The interpretability of the tested methods is discussed at the end of this chapter.

Chapter 5 further examines the performance of the proposed method considering recency effect. The case study is carried out based on another open dataset containing different load levels and temperature scenarios of 10 different zones. Each zone is tested with 7 different length of recency effects, and the outcome without considering any recency effect is used as a reference. The average quantile score and the minimum quantile score are calculated and shown for a comprehensive comparison. The result shows that the proposed method shows the best performance in most scenarios in terms of both average quantile score and minimum quantile score. This case study proves the effectiveness of the proposed method when recency effect is included and further confirms the robustness of the proposed method when it is applied to a variety of different datasets.

Chapter 6 extends the proposed method by applying it as a wrapper approach. The output of the proposed method is applied to a nonlinear model to validate if it could improve its performance. The same data as Chapter 5 is used, and the same 10 zones and the historical records (load and temperature) of the same period are used in this chapter. The result shows that the proposed method BQLRFS still outperforms the nonlinear model in the cases of all 10 zones, even when feature selection is considered. However, only a slight improvement in the quantile scores is observed after the proposed wrapper method is applied. This indicates that although we could use the proposed wrapper approach to improve the performance of other models, the improvement may not be significant.

There is still plenty of room for extending our method under the background of Bayesian theory. Future research will mainly focus on two directions. First, the proposed methodology is developed based on a linear model which shows competitively outstanding performance. It is expected that the proposed Bayesian framework can be further extended to be integrated into nonlinear models to develop a new embedded feature selection method. Another prospective direction is to combine feature selection with deep learning techniques. The rapid development in deep learning over the past a few years has facilitated a wide range of research, also in the energy domain. The following two possible topics are assumed for future research. A convolutional neural network (CNN) is a popular deep learning model which consists of two components, feature extraction and classification. Hence, CNN has the ability of automatically extracting features from raw data. The first expected research under this scheme is to develop a proper feature selection process to select the extracted features. Another widely used deep learning model is long short-term memory (LSTM) networks. LSTM networks are predominately used for handling sequential data. This model is capable of learning varying impacts of the features over different time. It is expected that a novel feature selection scheme that considering time dependencies among features.

REFERENCES

[1] W.C. Hong, “Intelligent energy demand forecasting,” London, UK: Springer, 2013

[2] A. S. Khwaja, M. Naeem, A. Anpalagan, A. Venetsanopoulos, and B. Venkatesh, “Improved short-term load forecasting using bagged neural networks,” *Electr. Power Syst. Res.*, vol. 125, pp. 109–115, Aug. 2015.

[3] D. W. Bunn and E. D. Farmer, “Comparative Models for Electrical Load Forecasting,” New York, NY, USA: Wiley, 1985, Chichester.

[4] B. F. Hobbs, S. Jitprapaikulsarn, S. Konda, V. Chankong, K. A. Loparo, and D. J. Maratukulam, “Analysis of the value for unit commitment of improved load forecasts,” *IEEE Trans. Power Syst.*, vol. 14, no. 4, pp. 1342–1348, Nov. 1999.

[5] J. Mulvaney-Kemp, S. Fattahi, and J. Lavaei, “Smoothing property of load variation promotes finding global solutions of time-varying optimal power flow,” *IEEE Trans. Control Netw. Syst.*, vol. 8, no. 3, pp. 1552–1564, Sept. 2021.

[6] R. A. Jabr, “Power Flow Based Volt/var Optimization Under Uncertainty,” *J. Mod. Power Syst. Clean Energy*, vol. 9, no. 5, pp. 1000–1006, Sept. 2021.

[7] Z. Li, W. Wu, B. Zhang, and X. Tai, “Analytical reliability assessment method for complex distribution networks considering post-fault network reconfiguration,” *IEEE Trans. Power Syst.*, vol. 35, no. 2, pp. 1457–1467, March 2020.

[8] Y. Guo, S. Li, C. Li, and H. Peng, “Short-term reliability assessment for islanded microgrid based on time-varying probability ordered tree screening algorithm,” *IEEE Access*, vol. 7, pp. 37324–37333, 2019.

[9] D. Sáez, F. Ávila, D. Olivares, C. Cañizares, and L. Marín, “Fuzzy prediction interval models for forecasting renewable resources and loads in microgrids,” *IEEE Trans. Smart Grid*, vol. 6, no. 2, pp. 548–556, Mar. 2015.

[10] T. Hong and S. Fan, “Probabilistic electric load forecasting: A tutorial review,” *Int. J. Forecast.*, vol. 32, no. 3, pp. 914–938, Jul. 2016.

[11] W. Zhang, H. Quan, and D. Srinivasan, “An Improved Quantile Regression Neural Network for Probabilistic Load Forecasting,” *IEEE Trans. Smart Grid*, vol. 10, no. 4, pp. 4425–4434, Jul. 2019.

[12] W. Zhang, H. Quan, O. Gandhi, R. Rajagopal, C. W. Tan, and D. Srinivasan, “Improving Probabilistic Load Forecasting Using Quantile Regression NN with Skip Connections,” *IEEE Trans. Smart Grid*, vol. 11, no. 6, pp. 5442–5450, Nov. 2020.

[13] A. Bracale, P. Caramia, P. De Falco, and T. Hong, “Multivariate Quantile Regression for Short-Term Probabilistic Load Forecasting,” *IEEE Trans. Power Syst.*, vol. 35, no. 1, pp. 628–638, Jan. 2020.

[14] S. Arora and J. W. Taylor, “Forecasting electricity smart meter data using conditional kernel density estimation,” *Omega (United Kingdom)*, Mar. 2016.

[15] Y. Wang, Q. Chen, N. Zhang, and Y. Wang, “Conditional residual modeling for probabilistic load forecasting,” *IEEE Trans. Power Syst.*, vol. 33, no. 6, pp. 7327–7330, Nov. 2018.

[16] Z. Cao, C. Wan, Z. Zhang, F. Li, and Y. Song, “Hybrid Ensemble Deep Learning for Deterministic and Probabilistic Low-Voltage Load Forecasting,” *IEEE Trans. Power Syst.*, vol. 35, no. 3, pp. 1881–1897, May 2020.

[17] J. Xie and T. Hong, “Temperature scenario generation for probabilistic load forecasting,” *IEEE Trans. Smart Grid*, vol. 9, no. 3, pp. 1680–1687, May 2018.

[18] C. C. Aggarwal, X. Kong, Q. Gu, J. Han, and P. S. Yu, “Feature Selection for Classification: A survey,” *Data Classif. Algorithms Appl.*, pp. 571–605, Jan. 2014

[19] O. Abedinia, N. Amjady, and H. Zareipour, “A New Feature Selection Technique for Load and Price Forecast of Electrical Power Systems,” *IEEE Trans. Power Syst.*, vol. 32, no. 1, pp. 62–74, Jan. 2017.

[20] S. Li, P. Wang, and L. Goel, “A novel wavelet-based ensemble method for short-term load forecasting with hybrid neural networks and feature selection,” *IEEE Trans. Power Syst.*, vol. 31, no. 3, pp. 1788–1798, May 2016.

[21] M. Tan, S. Yuan, S. Li, Y. Su, H. Li, and F. H. He, “Ultra-Short-Term Industrial Power Demand Forecasting Using LSTM Based Hybrid Ensemble Learning,” *IEEE Trans. Power Syst.*, vol. 35, no. 4, pp. 2937–2948, Jul. 2020.

[22] J. Xie and T. Hong, “Variable selection methods for probabilistic load forecasting: Empirical evidence from seven states of the United States,” *IEEE Trans. Smart Grid*, vol. 9, no. 6, pp. 6039–6046, Nov. 2018.

[23] Y. Wang, D. Gan, N. Zhang, L. Xie, and C. Kang, “Feature selection for probabilistic load forecasting via sparse penalized quantile regression,” *J. Mod. Power Syst. Clean Energy*, vol. 7, no. 5, pp. 1200–1209, Apr. 2019.

[24] R. J. Urbanowicz, M. Meeker, W. La Cava, R. S. Olson, and J. H. Moore, “Relief-based feature selection: Introduction and review,” *J. Biomed. Inform.*, vol. 85, no. January, pp. 189–203, Jan. 2018.

[25] J. C. Ang, A. Mirzal, H. Haron, and H. N. A. Hamed, “Supervised, unsupervised, and semi-supervised feature selection: A review on gene selection,” *IEEE/ACM Trans. Comput. Biol. Bioinforma.*, vol. 13, no. 5, pp. 971–989, Sept. 2016.

[26] H. Liu and L. Yu, “Toward integrating feature selection algorithms for classification and clustering,” *IEEE Trans. Knowl. Data Eng.*, vol. 17, no. 4, pp. 491–502, Apr. 2005.

[27] Q. Gu, Z. Li, and J. Han, “Generalized fisher score for feature selection,” *Proc. 27th Conf. Uncertain. Artif. Intell. UAI 2011*, pp. 266–273, 2011.

[28] P. A. Estévez, M. Tesmer, C. A. Perez, and J. M. Zurada, “Normalized mutual information feature selection,” *IEEE Trans. Neural Networks*, vol. 20, no. 2, pp. 189–201, Feb. 2009.

[29] J. Xu, B. Tang, H. He, and H. Man, “Semisupervised Feature Selection Based on Relevance and Redundancy Criteria,” *IEEE Trans. Neural Networks Learn. Syst.*, vol. 28, no. 9, pp. 1974–1984, Sept. 2017.

[30] K. Kira and L. A. Rendell, “A practical approach to feature selection,” in *Machine learning proceedings 1992*, Elsevier, 1992, pp. 249–256.

[31] N. Zhong, J. Dong, and S. Ohsuga, “Using rough sets with heuristics for feature selection,” *J. Intell. Inf. Syst.*, vol. 16, no. 3, pp. 199–214, May 2001.

[32] S. J. Reeves and Z. Zhe, “Sequential algorithms for observation selection,” *IEEE Trans. Signal Process.*, vol. 47, no. 1, pp. 123–132, Jan. 1999.

[33] W. Tang, Z. Shi, and Y. Wu, “Regularized simultaneous forward-backward greedy algorithm for sparse unmixing of hyperspectral data,” *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 9, pp. 5271–5288, Sept. 2014.

[34] S. Li and D. Wei, “Extremely high-dimensional feature selection via feature generating samplings,” *IEEE Trans. Cybern.*, vol. 44, no. 6, pp. 737–747, Jun. 2014.

[35] I. A. Hodashinsky and K. S. Sarin, “Feature Selection for Classification through Population Random Search with Memory,” *Autom. Remote Control*, vol. 80, no. 2, pp. 324–333, Apr. 2019.

[36] I. Guyon, J. Weston, S. Barnhill, and V. Vapnik, “Gene Selection for Cancer Classification using Support Vector Machines,” *Mach. Learn.*, vol. 46, no. 1, pp. 389–422, Jan. 2002.

[37] H. F. Zhou, J. W. Zhang, Y. Q. Zhou, X. J. Guo, and Y. M. Ma, “A feature selection algorithm of decision tree based on feature weight,” *Expert Syst. Appl.*, vol. 164, no. August 2020, p. 113842, Feb. 2021.

[38] N. Pudjihartono, T. Fadason, A. W. Kempa-Liehr, and J. M. O’Sullivan, “A review of feature selection methods for machine learning-based disease risk prediction,” *Front. Bioinforma.*, vol. 2, p. 927312, Jun. 2022.

[39] M. Kubus, “The problem of redundant variables in random forests,” *Acta Univ. Lodz. Folia Oeconomica*, vol. 6, no. 339, pp. 7–16, Feb. 2019.

[40] R. Tibshirani, “Regression shrinkage and selection via the lasso: A retrospective,” *J. R. Stat. Soc. Ser. B Stat. Methodol.*, vol. 73, no. 3, pp. 273–282, Apr. 2011.

[41] S. Paul and P. Drineas, “Feature Selection for Ridge Regression with Provable Guarantees,” *Neural Comput.*, vol. 28, Jun. 2015.

[42] C. S. Signorino and A. Kirchner, “Using LASSO to model interactions and nonlinearities in survey data,” *Surv. Pract.*, vol. 11, no. 1, pp. 1–10, 2018.

[43] Y. He, C. Cao, S. Wang, and H. Fu, “Nonparametric probabilistic load forecasting based on quantile combination in electrical power systems,” *Appl. Energy*, vol. 322, p. 119507, Sept. 2022.

[44] S. Zhang, Y. Wang, Y. Zhang, D. Wang, and N. Zhang, “Load probability density forecasting by transforming and combining quantile forecasts,” *Appl. Energy*, vol. 277, p. 115600, Nov. 2020.

[45] S. Ben Taieb, R. Huser, R. J. Hyndman, and M. G. Genton, “Forecasting uncertainty in electricity smart meter data by boosting additive quantile regression,” *IEEE Trans. Smart Grid*, vol. 7, no. 5, pp. 2448–2455, Sept. 2016.

[46] R. Koenker and G. Bassett Jr, “Regression quantiles,” *Econom. J. Econom. Soc.*, pp. 33–50, Jan. 1978.

[47] R. Koenker and K. F. Hallock, “Quantile regression,” *J. Econ. Perspect.*, vol. 15, no. 4, pp. 143–156, Fall 2001.

[48] B. Liu, J. Nowotarski, T. Hong, and R. Weron, “Probabilistic Load Forecasting via Quantile Regression Averaging on Sister Forecasts,” *IEEE Trans. Smart Grid*, vol. 8, no. 2, pp. 730–737, Mar. 2017.

[49] T. Hong, “Short Term Electric Load Forecasting”, Ph.D. dissertation, Dept. Elect. Eng., North Carolina State Univ., Raleigh, NC, 2010.

[50] A. Faustine and L. Pereira, “FPSeq2Q: Fully Parameterized Sequence to Quantile Regression for Net-Load Forecasting With Uncertainty Estimates,” *IEEE Trans. Smart Grid*, vol. 13, no. 3, pp. 2440–2451, May 2022.

[51] H. He, J. Pan, N. Lu, B. Chen, and R. Jiao, “Short-term load probabilistic forecasting based on quantile regression convolutional neural network and Epanechnikov kernel density estimation,” *Energy Reports*, vol. 6, pp. 1550–1556, Dec. 2020.

[52] S. Haben and G. Giasemidis, “A hybrid model of kernel density estimation and quantile regression for GEFCom2014 probabilistic load forecasting,” *Int. J. Forecast.*, vol. 32, no. 3, pp. 1017–1022, Jul.-Sep. 2016.

[53] F. Amara, K. Agbossou, Y. Dubé, S. Kelouwani, A. Cardenas, and J. Bouchard, “Household electricity demand forecasting using adaptive conditional density estimation,” *Energy Build.*, vol. 156, pp. 271–280, Dec. 2017.

[54] J. Xie, T. Hong, T. Laing, and C. Kang, “On Normality Assumption in Residual Simulation for Probabilistic Load Forecasting,” *IEEE Trans. Smart Grid*, vol. 8, no. 3, pp. 1046–1053, May 2017.

[55] B. Wang, M. Mazhari and C. Y. Chung, "A Novel Hybrid Method for Short-Term Probabilistic Load Forecasting in Distribution Networks," *IEEE Trans. Smart Grid*, vol. 13, no. 5, pp. 3650-3661, Sept. 2022

[56] J. Xie and T. Hong, “GEFCom2014 probabilistic electric load forecasting: An integrated solution with forecast combination and residual simulation,” *Int. J. Forecast.*, vol. 32, no. 3, pp. 1012–1016, 2016.

[57] Y. Yang, S. Li, W. Li, and M. Qu, “Power load probability density forecasting using Gaussian process quantile regression,” *Appl. Energy*, vol. 213, pp. 499–509, Mar. 2018.

[58] P. Kou and F. Gao, “A sparse heteroscedastic model for the probabilistic load forecasting in energy-intensive enterprises,” *Int. J. Electr. Power Energy Syst.*, vol. 55, pp. 144–154, 2014.

[59] P. Wang, B. Liu, and T. Hong, “Electric load forecasting with recency effect: A big data approach,” *Int. J. Forecast.*, vol. 32, no. 3, pp. 585–597, Jul.2016.

[60] H. Alkharusi, “Categorical Variables in Regression Analysis: A Comparison of Dummy and Effect Coding,” *Int. J. Educ.*, vol. 4, no. 2, p. 202, Apr. 2012.

[61] G. W. Brier, “Verification of forecasts expressed in terms of probability,” *Mon. Weather Rev.*, vol. 78, no. 1, pp. 1–3, Jan. 1950.

[62] R. L. Winkler *et al.*, “Scoring rules and the evaluation of probabilities,” *Test*, vol. 5, no. 1, pp. 1–60, Jun. 1996.

[63] A. H. Murphy, “The ranked probability score and the probability score: A comparison,” *Mon. Weather Rev.*, vol. 98, no. 12, pp. 917–924, Dec. 1970.

[64] J. E. Matheson and R. L. Winkler, “Scoring rules for continuous probability distributions,” *Manage. Sci.*, vol. 22, no. 10, pp. 1087–1096, Jun. 1976.

[65] T. Hong, P. Pinson, S. Fan, H. Zareipour, A. Troccoli, and R. J. Hyndman, “Probabilistic energy forecasting: Global Energy Forecasting Competition 2014 and beyond,” *Int. J. Forecast.*, vol. 32, no. 3, pp. 896–913, 2016.

[66] T. Gneiting, F. Balabdaoui, and A. E. Raftery, “Probabilistic forecasts, calibration and sharpness,” *J. R. Stat. Soc. Ser. B Stat. Methodol.*, Apr. 2007.

[67] B. Wang, “Short Term Probabilistic Load Forecasting at Local Level in Distribution Networks”, M.Sc. dissertation, Dept. Elect. Eng., Univ. Sask., Saskatoon, SK, Canada, 2019.

[68] S. Kotz, T. J. Kozubowski, and K. Podgórski, *The Laplace Distribution and Generalizations*, Boston, MA: Birkhäuser, 2001, pp. 239-272.

[69] J. Lucas, C. Carvalho, Q. Wang, and A. Bild, “Sparse statistical modelling in gene expression genomics,” *Bayesian Inference Gene Expr. Proteomics*, pp. 1–25, 2006.

[70] I. Yildirim, “Bayesian inference: Gibbs sampling,” *Tech. Note, Univ. Rochester*, 2012.

[71] W. Hörmann, J. Leydold, “Generating generalized inverse Gaussian random variates,” *Stat. Comput.*, vol. 24, pp. 547–557, 2014

[72] M. Robnik-Šikonja and I. Kononenko, “Theoretical and empirical analysis of ReliefF and RReliefF,” *Mach. Learn.*, vol. 53, no. 1, pp. 23–69, 2003.

[73] W. Yang, K. Wang, and W. Zuo, “Neighborhood component feature selection for high-dimensional data.,” *J. Comput.*, vol. 7, no. 1, pp. 161–168, 2012.

[74] A. J. Cannon, “Quantile regression neural networks: Implementation in R and application to precipitation downscaling,” *Comput. Geosci.*, vol. 37, issue 9, pp. 1277-1284, Sept. 2011

[75] P. J. Huber, “Robust regression: asymptotics, conjectures and Monte Carlo,” *Ann. Stat.*, pp. 799–821, Sept. 1973.

[76] N. Meinshausen and G. Ridgeway, “Quantile regression forests.,” *J. Mach. Learn. Res.*, vol. 7, no. 6, Jun. 2006.

[77] L. Breiman, “Random forests,” *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.