A Two-Stage Supervised Learning Approach for Electricity Price Forecasting by Leveraging Different Data Sources

by

Shuman Luo

A Thesis Presented in Partial Fulfillment
of the Requirement for the Degree
Master of Science

Approved April 2019 by the
Graduate Supervisory Committee:

Yang Weng, Chair
Qin Lei
Jiangchao Qin

ARIZONA STATE UNIVERSITY

May 2019

ABSTRACT

Over the years, the growing penetration of renewable energy into the electricity market has resulted in a significant change in the electricity market price. This change makes the existing forecasting method prone to error, decreasing the economic benefits. Hence, more precise forecasting methods need to be developed. This paper starts with a survey and benchmark of existing machine learning approaches for forecasting the real-time market (RTM) price. While these methods provide sufficient modeling capability via supervised learning, their accuracy is still limited due to the single data source, e.g., historical price information only. In this paper, a novel two-stage supervised learning approach is proposed by diversifying the data sources such as highly correlated power data. This idea is inspired by the recent load forecasting methods that have shown extremely well performances. Specifically, the proposed two-stage method, namely the rerouted method, learns two types of mapping rules. The first one is the mapping between the historical wind power and the historical price. The second is the forecasting rule for wind generation. Based on the two rules, we forecast the price via the forecasted generation and the first learned mapping between power and price. Additionally, we observed that it is not the more training data the better, leading to our validation steps to quantify the best training intervals for different datasets. We conduct comparisons of numerical results between existing methods and the proposed methods based on datasets from the Electric Reliability Council of Texas

(ERCOT). For each machine learning step, we examine different learning methods, such as polynomial regression, support vector regression, neural network, and deep neural network. The results show that the proposed method is significantly better than existing approaches when renewables are involved.

ACKNOWLEDGEMENTS

My deepest gratitude goes first and foremost to my supervisor, Dr. Yang Weng, for his constant encouragement and guidance on this thesis research. I sincerely thank him for his consistent illuminating guidance and patience that walked me through all the stages of my master studies. He sets a good example of a researcher to our group by his enthusiasm and sharp insights into research.

Second, I would like to express my sincere thanks to my committee members, Dr. Qin Lei and Dr. Jiangchao Qin for giving me instructive advice and their direct and indirect help to me.

Special thank goes to my friend Jingyi Yuan who have put considerable time and efforts on listening to me and helping me work out the problems during the research.

Finally, I would like to express my heartfelt thank to my beloved family for their emotional support and great confidence in me throughout my time in graduate school.

TABLE OF CONTENTS

CHAPTER Page

LIST OF TABLES

Table Page

LIST OF FIGURES

Table Page

Chapter 1

INTRODUCTION

Since the industrial revolution, energy has become a key factor in everyday life [1]. Fossil fuels have become the most primary energy production in the world [1]. However, with the population growth and technological development, the current world is facing two vital problems, environmental pollution, and energy resource shortages [2]. One way to overcome problems is to improve efficiency and reduce emission [3]. The other way is to develop alternate energy resources [2]. People draw their eyes to renewable resources for their properties of environmental-friendly and sustainability. The most competitive renewables include water, wind, photovoltaic energy, and biofuel. Many of them have been proved to be advanced in addressing environmental and energy issues [4, 5]. Many renewables have been applied to the electricity market.

In the last few years, electricity market prices decreased a lot due to the close-to-zero marginal costs from renewable energies [6]. Therefore, the electricity market participants are seeking ways to be more competitive in the market. Many companies have adopted new electricity price plans [7], for example, time-of-use electricity price plans. These plans charge higher rates when demand is high, and lower rates when demand is low. This encourages customers to wisely decide their electricity usages and reduce on-peak energy usages [8]. This situation makes not only the producers but also the customers pursue more precise forecasts of the electricity market prices than ever. However, electricity price usually has complex features, such as highly volatile behavior and non-linearity, which makes it rather difficult to build a precise forecasting model [9, 10, 11].

In general, the electricity market price forecast has two classes of computing techniques. One is so-called 'hard computing techniques' [12], which can accurately predict the electricity prices if we know the exact model of the system. Time series models [13] and autoregressive integrated moving average (ARIMA) models [14] are two typical models. However, electricity prices are influenced by many factors, such as the volatile prices of generation resources, seasonal weather risks, the uncertain behavior of competitors in the market and so on [15]. These elements make it rather difficult to build the accurate model of the system. Besides, the solutions of hard computing techniques are solved according to physical regularity which needs high computation costs. Different from 'hard computing techniques', 'soft computing techniques' are proposed without needing to build the models of the systems [12]. This type of technique learns the mapping between the input and the output data, which needs less information and has higher computation efficiency [12].

Hence, I employ 'soft computing techniques', such as forecasting future real-time market (RTM) bus price using the historical bus price with different machine learning methods. However, this direct method from price to price has a relatively poor performance, no matter which learning method I try or if I perform the validation step for hyper-parameters like training or testing data size. This is because the model only considers a single feature with limited information. In order to improve this method, I add another important data type, namely the wind power generation, which directly impact price variation. Additionally, I also redesign the forecasting model by leveraging the fact that wind generation forecasting is with very high accuracy, e.g., mean absolute percentage error is less than 5% [16, 17, 18]. Specifically, the proposed method learns two types of mapping rules. The first one is the mapping between the historical wind power generation and the historical price. The second is the forecasting rule for wind power generation. Based on the two rules, I forecast the price via the forecasted generation and the first learned mapping rule between power generation and price. I name the proposed method the rerouted method (two-stage method).

As a highlight, I examine the advantages and disadvantages of each machine learning method for both direct method (price-to-price method) and the rerouted method (two-stage method), so that I can select the best method with the best hyper-parameters for the benchmark. Specifically, I choose machine learning methods that are widely used in real-world applications [19, 20], e.g., polynomial regression, support vector regression (SVR), neural network (NN), and deep neural network (DNN).

For numerical validation, I use RTM bus price data and system-wide wind power generation data from Electric Reliability Council of Texas (ERCOT). RTM bus price is the simple average of the time-weighted hub bus prices for each settlement interval in real-time, for each hub bus included in this hub. I preprocessed and removed some extreme data to make all the data in the normal range. The selected wind power generation contains the wind power generation all over the system. Simulation results show that the direct forecasting (price-to-price method) obtain its best testing accuracy when I employ polynomial regression. The rerouted method (two-stage method) obtain its best testing accuracy when I adopt deep learning. In general, the results show that the proposed method is significantly better than the direct forecasting (price-to-price method) when renewables are involved. All the references mentioned above can be better summered up in Table 1.1.

Current research work indicates that we may obtain higher forecasting accuracy if we consider additional highly correlated data sources such as solar energy and biofuels. The NN and DNN used in this work are basic networks, future research can explore more of the network structure.

The rest of the paper is organized as follows: Section 2 formulates the forecasting problem. Section 3 describes the machine learning methods I use. Section 4 describes the simulation setup and the numerical results. Section 5 concludes the paper.

Table 1.1: Reference Explanation

Reference number	Contents
[1, 2, 4, 3, 5]	The reason why renewables entered the electricity market.
[6, 7, 8]	The impact of renewables entering the electricity market on electricity prices.
[9, 10, 11]	The difficulties to build a precise electricity price forecasting model.
[13, 14, 15]	The difficulties to apply 'hard computing techniques' to build the electricity price forecasting model.
[12]	The advantages to use 'soft computing techniques' to build the electricity price forecasting model.
[16, 17, 18]	The high accuracy of the wind generation forecasting.
[19, 20]	The successful examples of employing machine learning methods into power systems.

Chapter 2

PROBLEM FORMULATION

In this section, I explain the direct method (price-to-price method) and the rerouted method (two-stage method) in detail using diagrams and mathematical formulas. To ensure an objective assessment of all the methods in this paper, I use the same dataset to test different approaches and models.

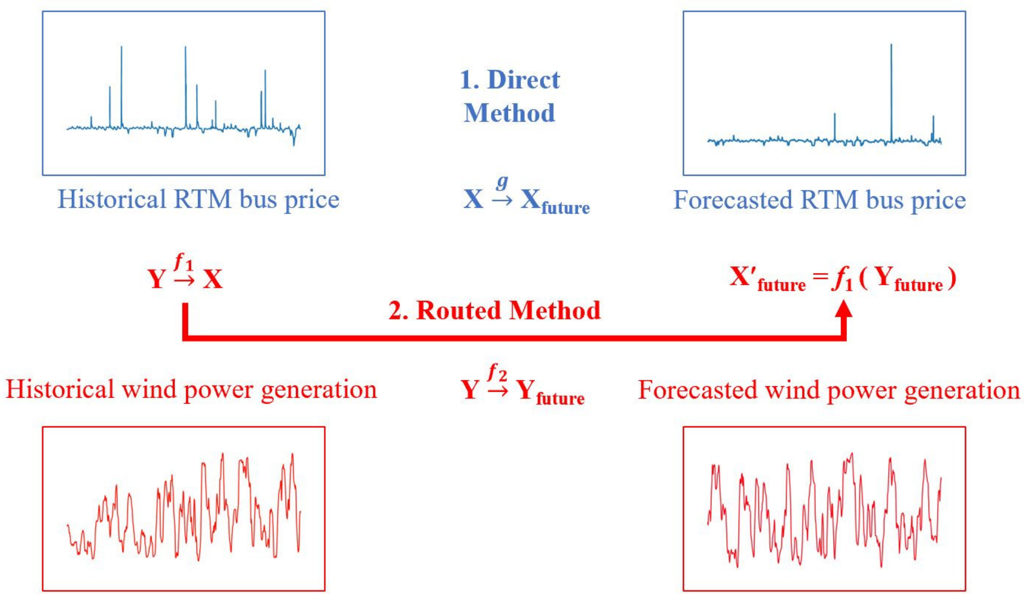


Figure 2.1: The Mechanism of the Methods

The ideas are shown in Figure 2.1. The blue pictures represents the direct method (price-to-price method). I use function g to learn the mapping between

X (Historical RTM Bus Price) and X_{future} (Forecasted RTM Bus Price). The red arrows and pictures form the rerouted method (two-stage method). The rerouted method (two-stage method) contains 3 steps. Step one is to use function f_1 to learn the mapping between X and Y (Historical Wind Power Generation). Step two is to use function f_2 to learn the mapping between Y and Y_{future} (Forecasted Wind Power Generation). Step three is to predict X_{future} using Y_{future} and the function f_1 learned before.

2.1 Direct Method (Price-to-Price Method)

The problem is defined as: forecast the real-time market (RTM) bus price for the following month using the historical RTM bus price. Specifically, I first preprocessed and removed some extreme data to make all the data in the normal range. Then, let M be the size of the input data, N be the size of the output data, I adjust the size of M and N to find the best pairs that obtain the highest testing accuracy. The parameters are formulated as follows.

- **Input:** The input matrix is the RTM bus price from January 2016: $X : M \times 1$.

- **Output:** The output is the predicted RTM bus price for February 2016 $X_{\text{future}} : N \times 1$, which is given by equation (1):

$$X_{\text{future}} = g(X), \tag{2.1}$$

  where X_{future} is the prediction of future RTM bus price. $g(\cdot)$ is the method chosen for forecasting.

In order to get $g(\cdot)$, I use historical data $(X,\ X_{\text{future}})$ to learn the mapping. By adjusting the sizes of the historical data, I can determine the best mapping $\hat{g}(\cdot)$ of the different methods presented in this paper.

2.2 Rerouted Method (Two-Stage Method)

The problem is defined as: forecast the RTM bus price for the following month using the historical RTM bus price and the system-wide wind power generation. Specifically, the rerouted method contains 3 steps. In step 1, I learn the mapping between the historical wind power generation and the historical price information. In step 2, I predict wind power generation using historical wind power generation. In step 3, I use the predicted wind power generation and the mapping function learned in step 1 to predict the future price. The parameters are formulated as follows. Figure 2.2 shows the flow chart of the rerouted method summarizing all the processes.

- **Input 1:** The input 1 matrix is RTM bus price from January 2016: X.
- **Input 2:** The input 2 matrix is system-wide wind power generation from January 2016: Y.

1. Step 1: I use historical data $(Y,\ X)$ to learn the mapping function $f_1(\cdot)$ between the historical system-wide wind power generation and the RTM bus price.

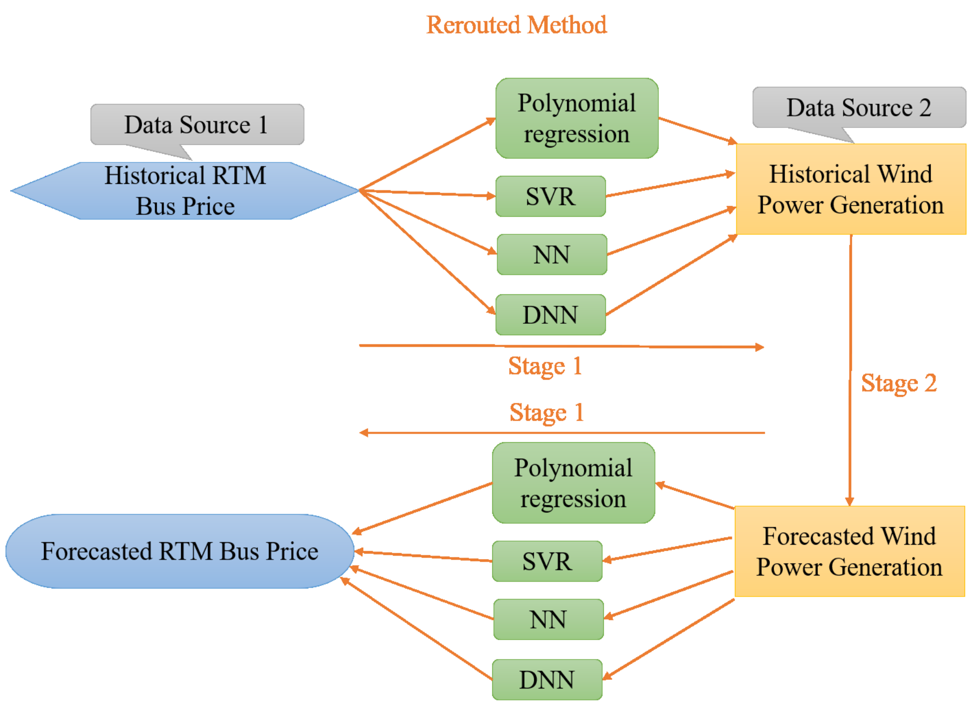


Figure 2.2: The Flow Chart of the Rerouted Method

2. Step 2: Let Y_{future} be the prediction of future system-wide wind power generation. I use historical data $(Y,\ Y_{\text{future}})$ to learn the mapping function $f_2(\cdot)$ between the historical system-wide wind power generation and the future system-wide wind power generation.

- **Output X_{future}:** (Step 3) I use the predicted wind power generation Y_{future} and the mapping function $f_1(\cdot)$ learned in step 1 to predict the future price X_{future}. The output is given by equation (2):

$$X_{\text{future}} = f_1(Y_{\text{future}}), \tag{2.2}$$

where X_{future} is the prediction of RTM bus price for February 2016, $f_1(\cdot)$ is the method chosen for forecasting.

Chapter 3

MACHINE LEARNING METHODS

In this chapter, I explain existing and popular machine learning methods for the proposed learning process in the last section.

3.1 Methods Overview

3.1.1 Polynomial Regression

In general, the polynomial regression model is given by equation (3):

$$y_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \cdots + \beta_m x_i^m + \epsilon_i,\ i = 1,\ 2,\ \cdots,\ n. \tag{3.1}$$

It can also be written as equation (4):

$$\vec{y} = X\vec{\beta} + \vec{\epsilon}, \tag{3.2}$$

where X is a design matrix, $\vec{y}$ is a target vector, $\vec{\beta}$ is a coefficient vector, and $\vec{\epsilon}$ is a vector of random errors.

The vector of the estimated polynomial regression coefficient can be calculated using equation (5):

$$\vec{\beta} = (X^T X)^{-1} X^T \vec{y},\ m < n. \tag{3.3}$$

3.1.2 Support Vector Regression (SVR)

SVR is a regression analysis of the data when we do the fitting [21]. It uses the idea of support vectors and the Lagrange multiplier. SVR constructs a hyperplane or set of hyperplanes in a high- or infinite-dimensional space by minimizing the margin on all the training data [22]. The support vector regression is obtained in equation (6).

$$\begin{aligned} \min \quad & F(w) = \frac{1}{2}\|w\|^2 \\ \text{s.t.} \quad & |y_i - (w^T x_i + b)| \leq \epsilon,\ i = 1,\ 2,\ \cdots,\ n, \end{aligned} \tag{3.4}$$

where x_i is a training sample with target value y_i. $w^T x_i + b$ is the prediction for that sample and ϵ is a free parameter that serves as a threshold. For example, all predictions have to be within an ϵ range of the true predictions. The method can be better illustrated by Figure 3.1.

The mapping of SVR to higher dimensions results in a serial of problems. It's hard to obtain the form of the mapping and to compute the coordinates of the data in that space. Hence, kernel methods are introduced to solve the problem. A kernel function can compute the dot product between the two mapping transforms in the feature space without knowing the mapping transform function itself. Assume $X_i,\ X_j \in R_n$, nonlinear function Φ implements the mapping from input space X to feature space F, where $F \subseteq R_m,\ n \ll m$. Refer to the kernel method, we have:

$$K(X_i,\ X_j) = \Phi(X_i) \cdot \Phi(X_j), \tag{3.5}$$

where $K(X_i,\ X_j)$ is the kernel function. From equation (7), we can see that the

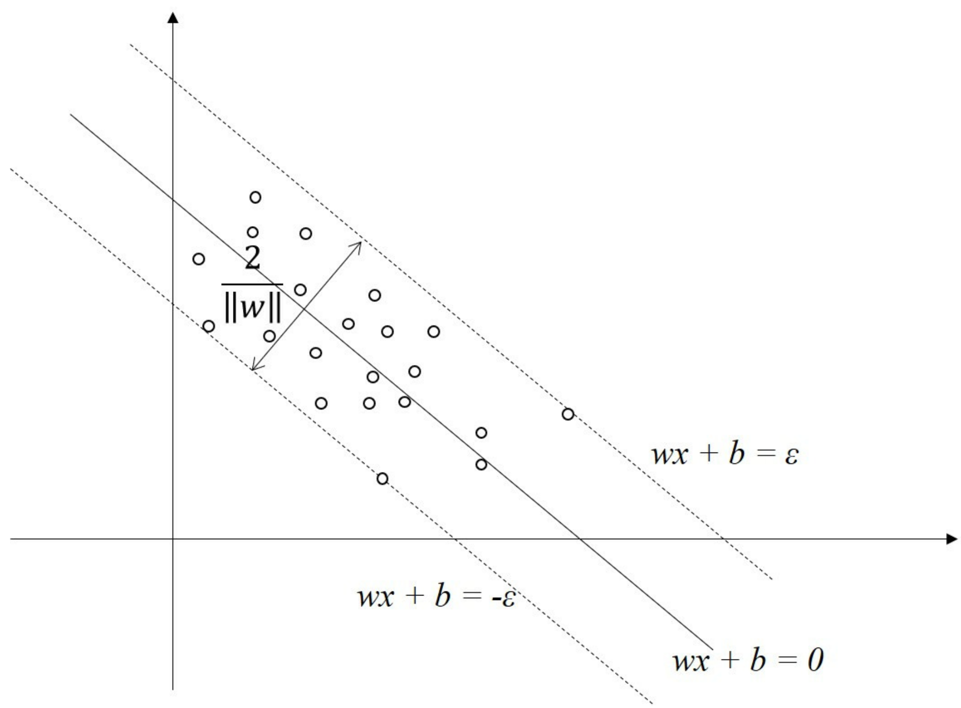


Figure 3.1: SVR Diagram

kernel function transforms the inner product operation in the high-dimensional space into the kernel function in the low-dimensional space. Commonly used kernel functions include linear kernel, polynomial kernel, and Gaussian kernel, also known as radial basis function (RBF) kernel.

3.1.3 Neural Network (NN)

NNs are highly-interconnected-computing systems inspired by modern biology [23]. NNs are built up from a number of processing units, also called neurons. Each neuron is a weighted sum of the inputs formed by a linear function with a biased term [24]. The sum is then passed through a transfer function, also called an activation function, which is often a unit step, sigmoid and Gaussian [24]. Neurons can be grouped into layers. Typically, the first layer and the last layer of a basic NN is called the input layer and the output layer. The layers between the input and output layers are called the hidden layers.

NNs can be represented in equation (8):

$$
\begin{aligned}
&x_{i,j} = g(h_{i,j}),\ h_{i,j} = w_{i,j}^{(0)} + \sum_k w_{i,j}^{(k)} x_{k,j-1}, \\
&i = 1,\ 2,\ \cdots,\ n,\ j = 1,\ 2,\ \cdots,\ m,\ k = 1,\ 2,\ \cdots,\ t,
\end{aligned}
\tag{3.6}
$$

where $x_{i,j}$ is the input for the current layer, $x_{k,j-1}$ is the input for the last layer, $w_{i,j}^{(k)}$ is the weights of the k^{th} neuron, $w_{i,j}^{(0)}$ is the biased term, g is the transfer function. The transfer function is introduced to increase the non-linearity. We conduct many experiments on different activation functions and find sigmoid function can achieve the highest accuracy. We also provide a diagram illustrating the structure of a basic NN shown in Figure 3.2.

Backpropagation (BP) is a method to calculate the gradient of the loss function (produces the cost associated with a given state) with respect to the weights in an artificial neural network (ANN) [25]. Backpropagation neural networks (BPNNs) have the ability to implement any complex nonlinear mapping from input to output, to learn by themselves and to adapt to changes [26]. Furthermore, BPNNs have generalization ability and error tolerance. In the meanwhile, BPNNs have many shortcomings such as the local minimization problem. With different initializations of the weights, a BPNN will converge to different local minimums. So

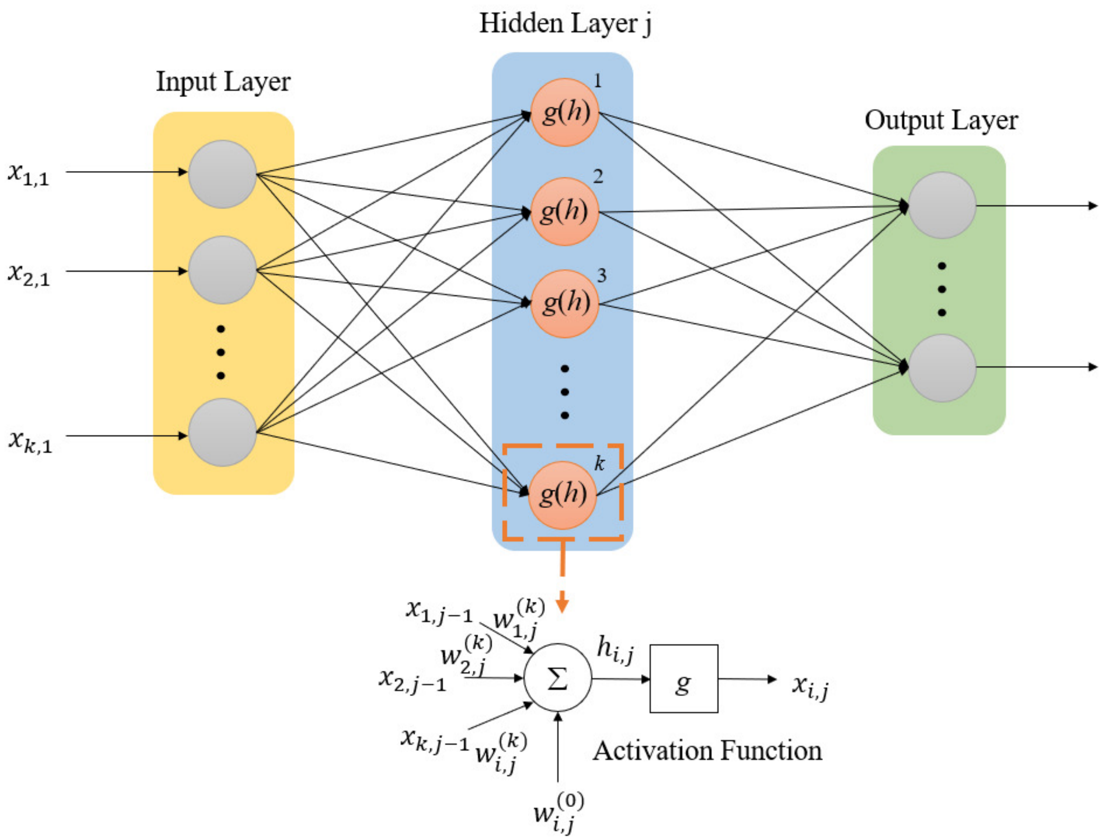


Figure 3.2: Neural Network Diagram

every time we train, we get a different result.

3.1.4 Deep Learning

Deep learning is a class of machine learning algorithms that use multiple layers of nonlinear processing units for feature extraction and transformation [27]. Each successive layer uses the output from the previous layer as input. Most deep learning models nowadays are based on ANNs [28]. It is very time-consuming to train a model, and the validation is very complex and troublesome. However, a well trained deep learning model can be easily applied to other problems by doing some simple refinements.

3.1.5 Method Comparison

In the polynomial regression, all the features are determined by us, which may contain useless features. Hence, NNs and DNNs are brought up for not needing to decide how to construct the features. We directly input the raw data, if we achieve high accuracy, the model is useful. However, NNs and DNNs involve the random initialization of weights. So training on the same data may give different results. Besides, considerable parameters are set concerning the architecture of the ANNs as well as the learning algorithms. The optimizations of these parameters can only be carried out through a trial-and-error process which consumes much time and resources [29]. The training of an SVR is a convex quadratic optimization which has one unique solution and it does not involve the random initialization of weights like NNs and DNNs [30]. Any SVR with the same parameter settings trained on identical data will give the same results. This greatly reduces the number of training required to find the optimum.

3.2 Performance Evaluation Metric

The performances of all the methods are determined by the mean square errors (MSEs). Let K be the size of the output, the computational formula is defined as follows:

$$\mathrm{MSE} = \frac{1}{\mathrm{N}} \sum_{\mathrm{t}=1}^{\mathrm{N}} (\widehat{\mathrm{Y}}_{\mathrm{t},1} - \mathrm{y}_{\mathrm{t},1})^2,$$

$$\widehat{Y}_{t,1} = \begin{cases} \frac{1}{t}(\hat{y}_{1,t} + \hat{y}_{2,t-1} + \cdots + \hat{y}_{t,1}), & \text{if } t \leq K-1 \\ \\ \frac{1}{K}(\hat{y}_{t-K+1,K} + \hat{y}_{t-K+2,K-1} + \cdots + \hat{y}_{t,1}), & \text{if } t > K-1 \end{cases} \tag{3.7}$$

where $\widehat{\mathrm{Y}}_{\mathrm{t},1}$ is the forecasted price at hour t, $\mathrm{y}_{\mathrm{t},1}$ is the real price at hour t, and N is the number of the total hours.

Chapter 4

NUMERICAL RESULTS

4.1 Data Preparation

Electric Reliability Council of Texas (ERCOT) is an independent system operator managing about 90 percent of the states electric load. ERCOT made significantly large investments in the renewable energy sector, particularly in wind energy and continues to lead as the top wind production in the nation [31]. ERCOT has an adequate market and grid information which can be easily accessed and downloaded from its website[32]. If you need some specific range of data that is not available on the website, you can contact ERCOT by submitting an information request form[33]. ERCOT is eager to help and responds quickly. The raw data I get from ERCOT is excel files containing the information of all districts. I extract all the data I need and build vectors of RTM price and system-wide wind power generation that are hourly measured. To ensure the RTM price data in the normal range, let μ be the mean of the data and T be the threshold, the normal range is defined as: $\mu \pm T$ in this specific problem.

4.2 Benchmark

For rerouted method (two-stage method), I do following simulations using electricity price data and the data of system-wide wind power generation. I use a 744×1 vector of system-wide wind power generation as the input data and a 744×1 vector of real-time market (RTM) bus price as the target data for training. Both data came from the same time slot that is January 2016 which contains 31 (days) $\times$ 24 (hours) $=$ 744 data points. The training data can be visualized in Figure 4.1, x-axis is the system-wide wind power generation from January 2016 and y-axis is the RTM bus price from January 2016.

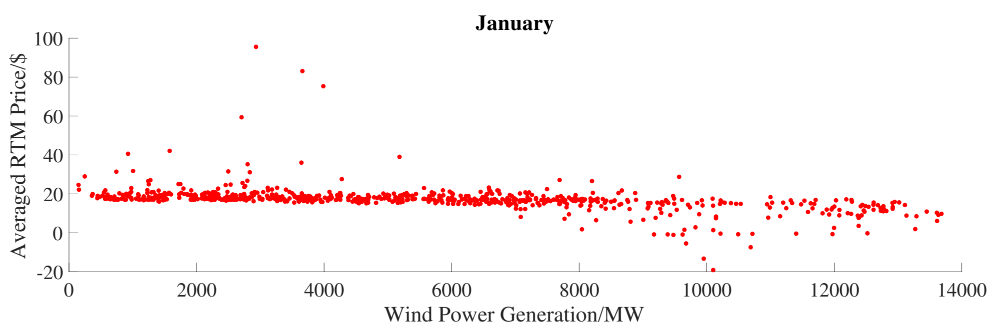


Figure 4.1: Initial Data Distribution

For direct method (price-to-price method), let M be the input data size, N be the output data size. By adjusting these two hyper-parameters, I am able to find the best M and N that make the mapping between the prices reach the highest accuracy. In this method, M is chosen from $2, 3, 6, 12, 24, \cdots, 384$; and N is chosen from 1, 2, 3, 6. I start from 2 for M, because if I only use one historical data to predict one or more data, the uncertainty is so huge that a high accuracy is hard to achieve. For other numbers, I let the latter number be twice the former to study the tendency of the testing accuracy.

General results show that the rerouted method can guarantee better accuracies when compared to the direct method, for all the machine learning methods I used. This statement can be confirmed by Table 4.1, where I compare the results of both methods. As I can observe in Table 4.1, the rerouted method gain its highest accuracy when 14-layer DNN is used. To ensure consistency, the direct method also employs the same machine learning methods. And the result shows that it obtains its highest frequency when using polynomial regression. The detailed results and comparisons will be listed in the following subsections.

4.2.1 Polynomial Regression

For rerouted method (two-stage method), I use the system-wide wind power generation from January 2016 as the training data and that from February 2016 as the testing data. I vary the degrees of the polynomial model from 1 to 4. Figure 4.2 shows the training regression curves and the error histograms of all the degrees and Figure 4.3 shows the testing regression curves and the error histograms of all the degrees. For direct method (price-to-price method), I fix the predicting data size N for each time and adjust the historical data size M to obtain the best testing results. The process of determining the size of the data can be illustrated in Figure 4.4. As shown in Figure 4.4, the training mean squared error (MSE) is fluctuating around the minimum value when I increase the historical data size M, while the

Table 4.1: MSE Comparison

(a) Rerouted Method

Method	1 degree polynomial regression	2 degree polynomial regression	3 degree polynomial regression	4 degree polynomial regression
Training	36.9341	36.5456	36.0694	35.9630
Testing	22.3945	23.0906	22.6684	22.2578
Method	NN with 30 hidden neurons	NN with 60 hidden neurons	linear kernel SVR	polynomial kernel SVR
Training	36.2109	33.4186	38.7031	37.9564
Testing	24.6695	26.23	25.6771	24.0645
Method	Gaussian kernel SVR	7 layers DNN	11 layers DNN	14layers DNN
Training	38.1197	34.9439	36.8437	36.1379
Testing	23.7623	22.7658	23.2444	**20.5065**

(b) Direct Method

Method	polynomial regression	NN with 30 hidden neurons	NN with 60 hidden neurons	linear kernel SVR
M	12	6	96	192
N	3	3	3	1
Training	40.6163	28.3315	23.4281	25.5157
Testing	**27.1534**	27.3736	30.0632	29.8577
Method	polynomial kernel SVR	Gaussian kernel SVR	11 layers DNN	14layers DNN
M	192	96	3	3
N	1	1	2	2
Training	25.5097	35.6567	23.3813	43.3681
Testing	29.865	37.1421	32.3289	32.1427

testing MSE becomes extremely large. This reveals the overfitting problem when I perform the training of the data set. The highest testing accuracy of each figure is determined by the gap between the training MSE and the testing MSE. I select the ones with the smallest gaps and merge them into Table 4.2(b) in order to be compared with the rerouted method. The MSEs for both methods are shown in Table 4.2.

Table 4.2: The MSEs of Polynomial Regression

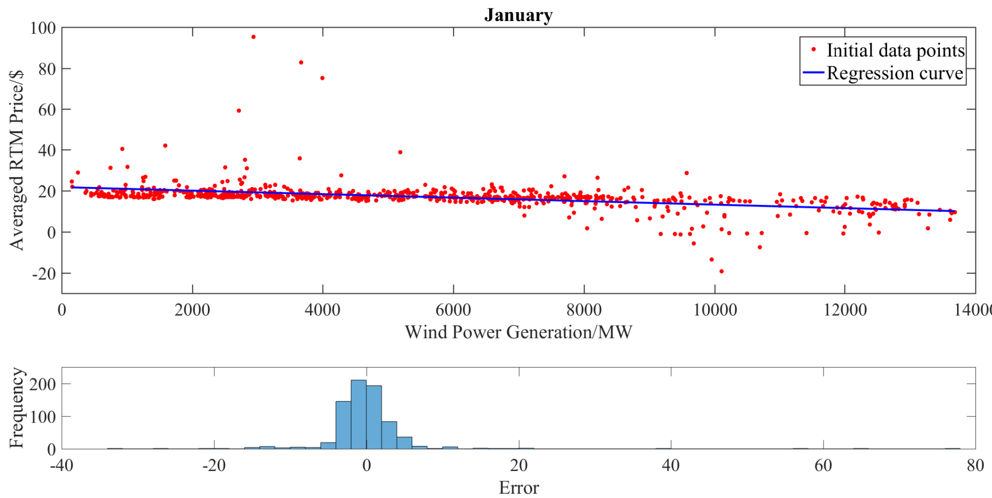
(a) Rerouted Method

MSE \ Degree	1	2	3	4
Training	36.9341	36.5456	36.0694	35.9630
Testing	22.3945	23.0906	22.6684	**22.2578**

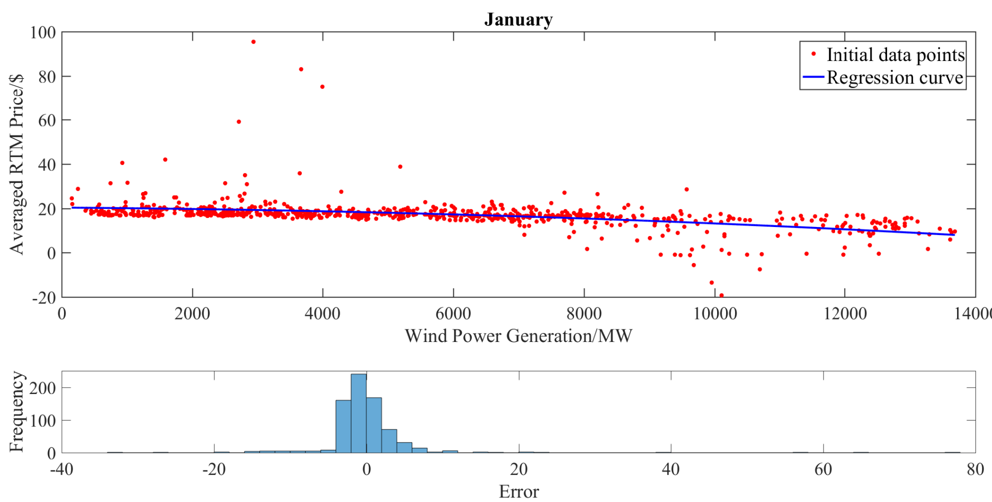
(b) Direct Method

N	1	2	3	6
M	48	96	12	24
Training	24.9384	31.8074	40.6163	54.1
Testing	35.2281	28.5926	**27.1534**	30.3579

As I can see from Table 4.2(a), when I increase the order of the polynomial regression the testing accuracy does not change much, which means that the data has great linearity as a whole. Combined with Figure 4.4 and Table 4.2(b), the rerouted method has smaller testing MSE than any of the direct method no matter how you



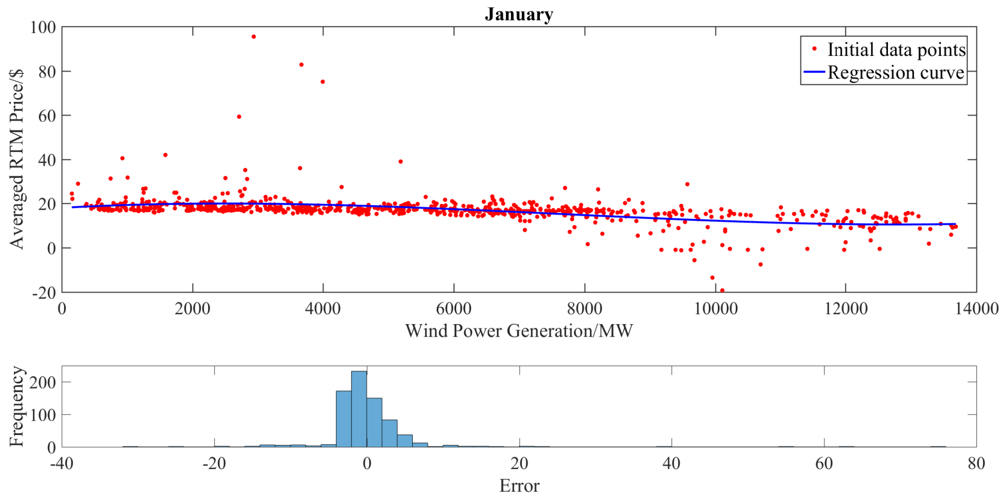
(a) Degree 1 Polynomial Regression



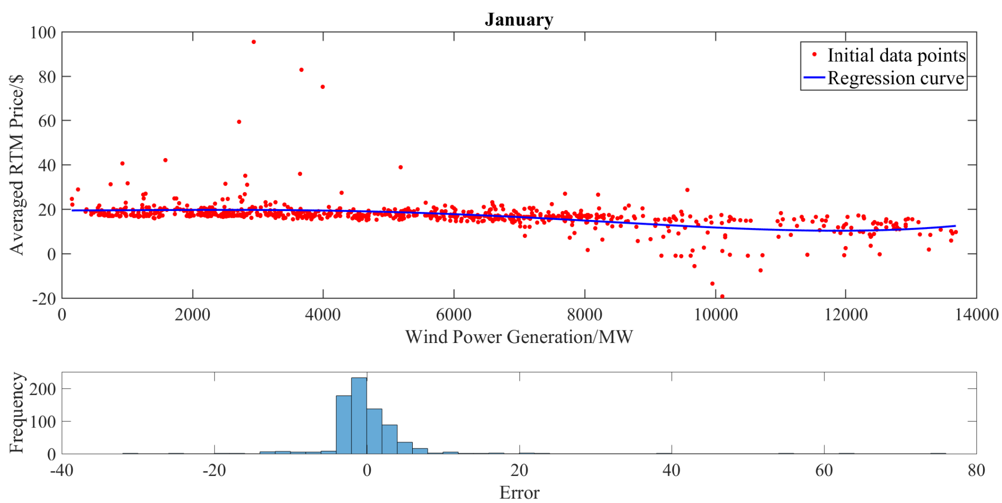
(b) Degree 2 Polynomial Regression

Figure 4.2: Training Results of Other Orders Polynomial Regression

resize M and N. In this case, when I use the rerouted method and polynomial regression with degree 1 or 4, I can obtain good testing performance. As you can observe from Figure 4.2 and Figure 4.3, the polynomial regression with a reason-

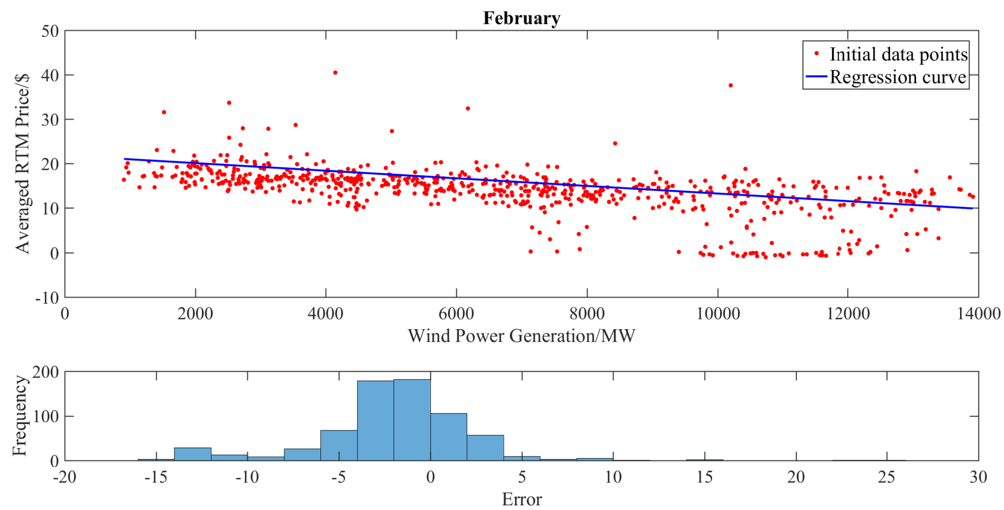


(c) Degree 3 Polynomial Regression

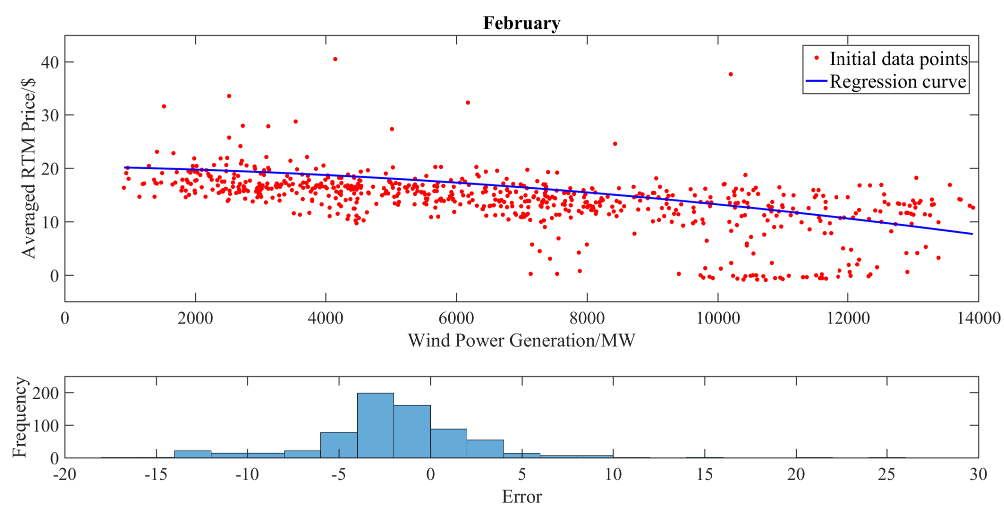


(d) Degree 4 Polynomial Regression

Figure 4.2: Training Results of Other Orders Polynomial Regression



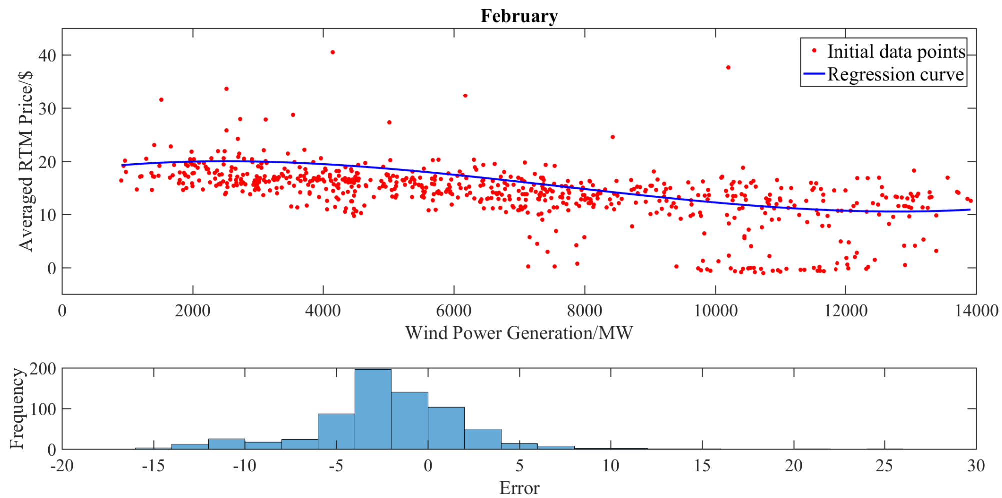
(a) Degree 1 Polynomial Regression



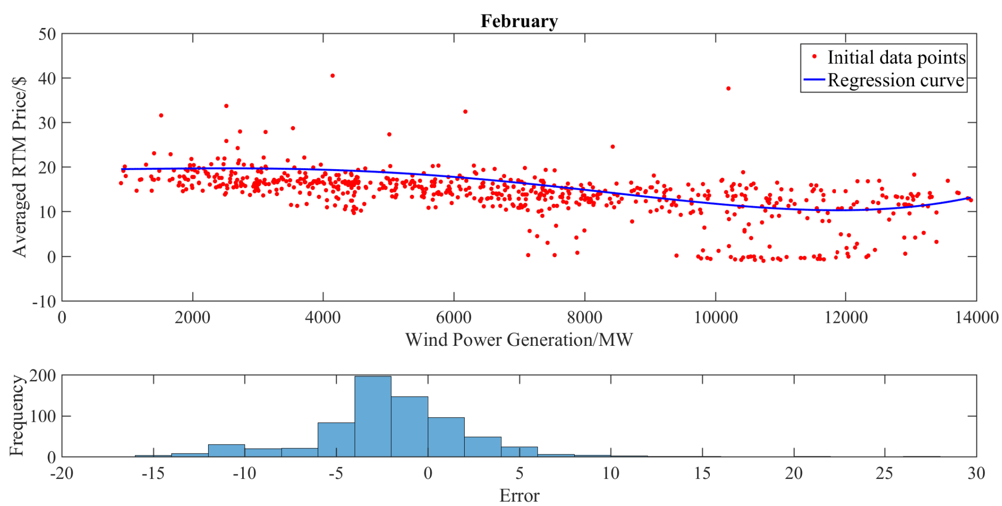
(b) Degree 2 Polynomial Regression

Figure 4.3: Testing Results of All Orders Polynomial Regression

able degree has a poor behavior detecting outliers. Hence, I employ support vector regression (SVR) for our next step to see if it can reach a higher accuracy when I map the data into higher dimensions.



(c) Degree 3 Polynomial Regression



(d) Degree 4 Polynomial Regression

Figure 4.3: Testing Results of All Orders Polynomial Regression

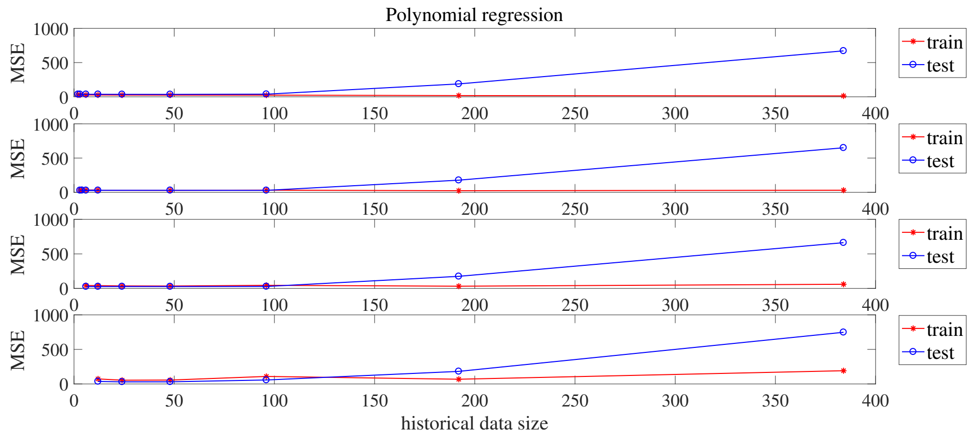


Figure 4.4: Direct Method: Data Size Determination

4.2.2 *Support Vector Regression (SVR)*

For both methods, I use command 'fitrsvm' in MATLAB and choose three different kernel functions: linear kernel, polynomial kernel and Gaussian kernel to do SVR. For the direct method (price-to-price method), the predicting data size N is fixed to be 1 and the historical data size M is adjustable. The procedure is displayed in Figure 4.5. The determination of the best testing accuracy obeys the same rule stated in the last section. In order to be better compared with the rerouted method, the highest testing accuracy of each figure is merged into Table 4.3(b). For the rerouted method (two-stage method), Figure 4.6 shows the training regression curves along with the error histograms of the SVR with linear kernel, polynomial kernel and Gaussian kernel. Figure 4.7 shows the testing regression curves along with the error histograms of the SVR with linear kernel, polynomial kernel and Gaussian kernel. The MSEs for both methods of SVR with 3 different kernels are shown in Table 4.3.

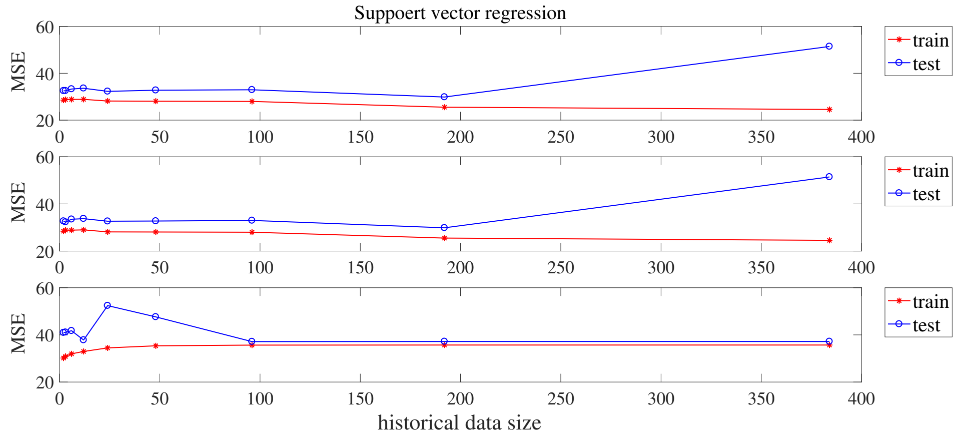


Figure 4.5: Direct Method: Data Size Determination

Table 4.3: The MSEs of SVR

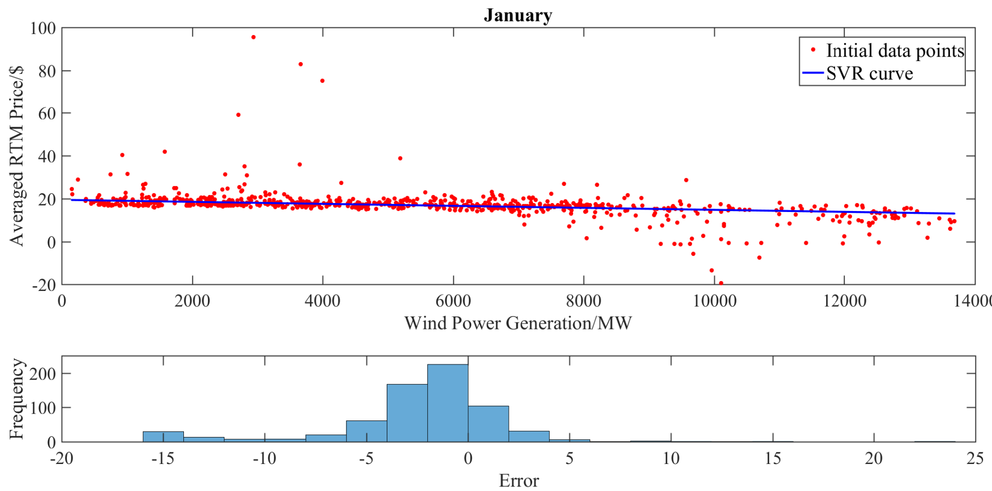
(a) Rerouted Method (Two-Stage Method)

Kernel / MSE	Linear	Polynomial	Gaussian
Training	38.7031	37.9564	38.1197
Testing	25.6771	24.0645	**23.7623**

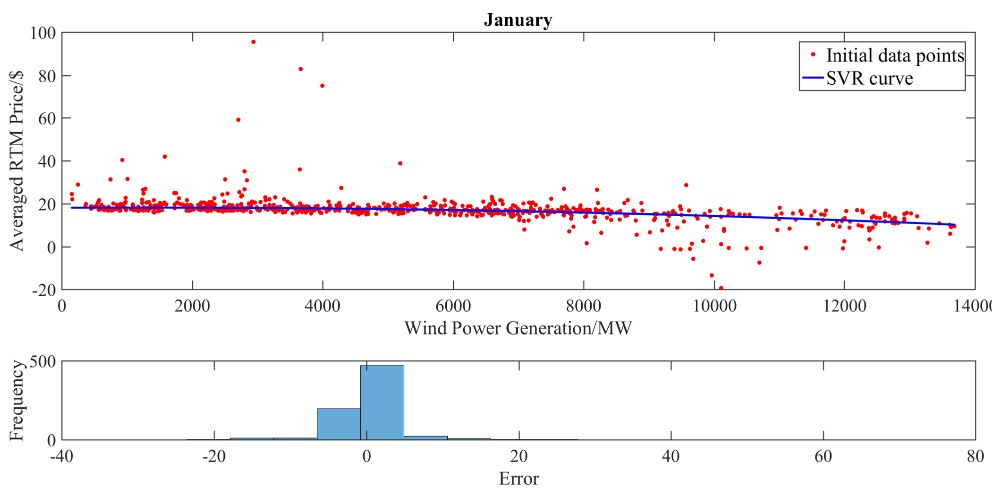
(b) Direct Method (Price-to-Price Method)

Kernel	Linear	Polynomial	Gaussian
M	192	192	96
Training MSE	25.5157	25.5097	35.6567
Testing MSE	**29.8577**	29.8650	37.1421

It's clearly shown in Table 4.3 that when I adopt the rerouted method, all the



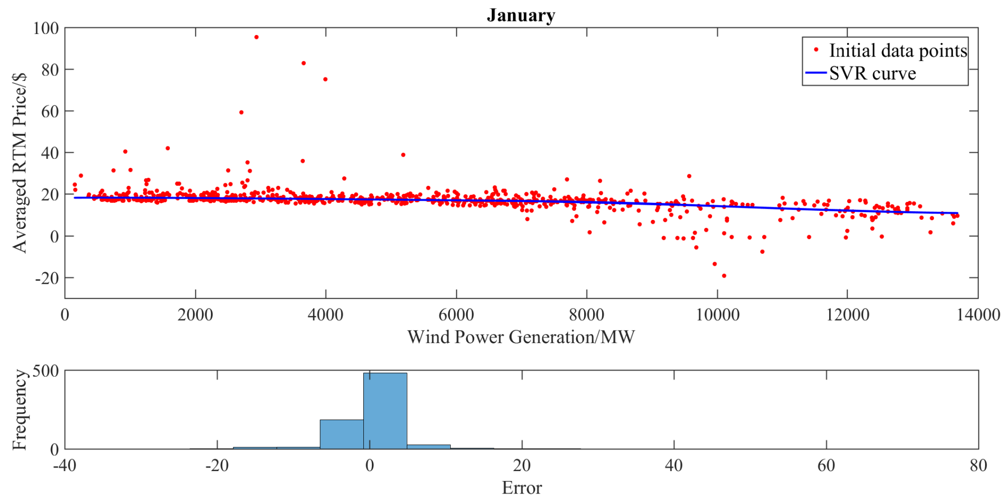
(a) Linear Kernel SVR



(b) Polynomial Kernel SVR

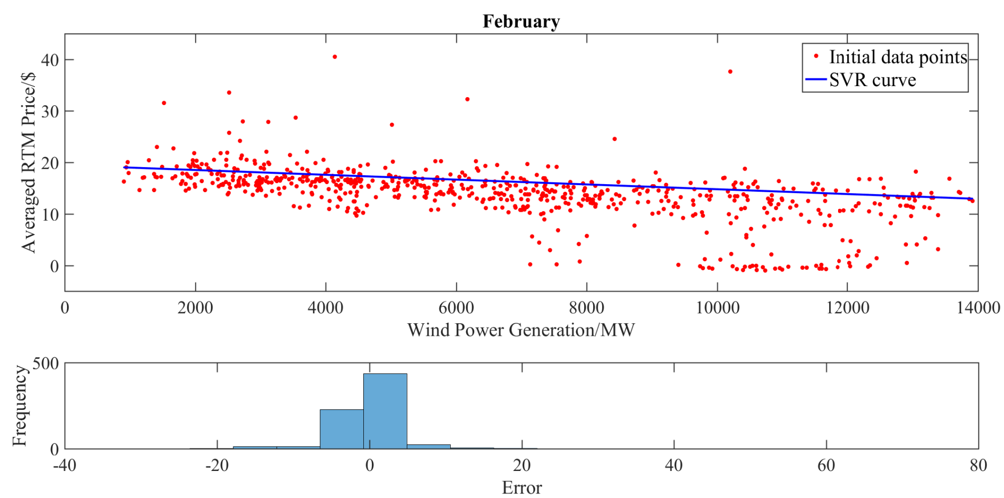
Figure 4.6: Training Results

SVR with different kernels have similar training and testing MSEs and the testing MSE of the SVR with Gaussian kernel is slightly better than all the others. The rerouted method has smaller testing MSEs than any of the direct method. In this case, when I use the rerouted method and SVR with Gaussian kernel, I can obtain



(c) Gaussian Kernel SVR

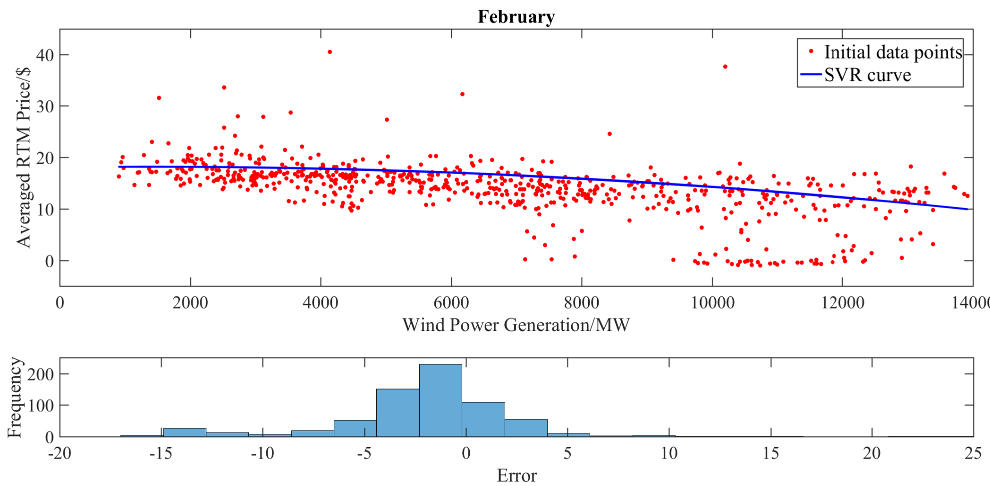
Figure 4.6: Training Results



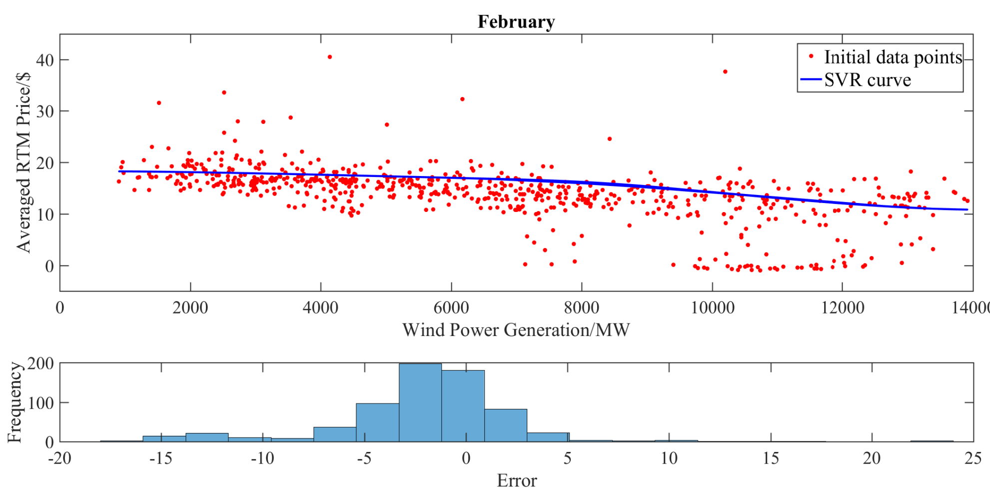
(a) Linear Kernel SVR

Figure 4.7: Testing Results

the best testing performance. Compared to Table 4.2, the testing MSEs become worse indicating that SVR ignores outliers. Hence, I employ neural networks to



(b) Polynomial Kernel SVR



(c) Gaussian Kernel SVR

Figure 4.7: Testing Results

make sure they capture the outliers.

4.2.3 Neural Network (NN)

Past researches have shown great success forecasting electricity price using NNs [9, 29, 24]. Drawing on their ideas, I conduct comparisons between the direct method and the rerouted method based on the same NN. For both methods, I use command 'nftool' in MATLAB to build a simple NN with 3 layers (1 input layer, 1 hidden layer, and 1 output layer). The hidden layer size is 30 and the NN is trained by the Levenberg-Marquardt algorithm. The structure of the NN is shown in Figure 4.8.

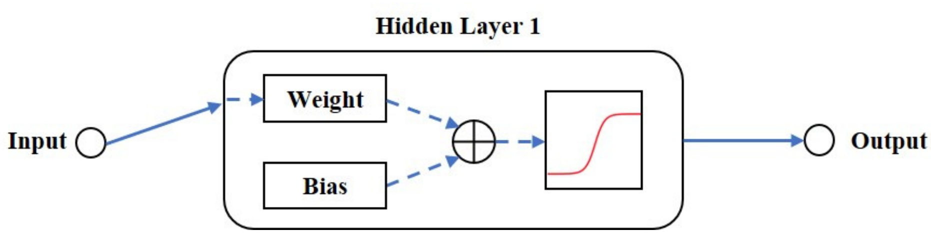
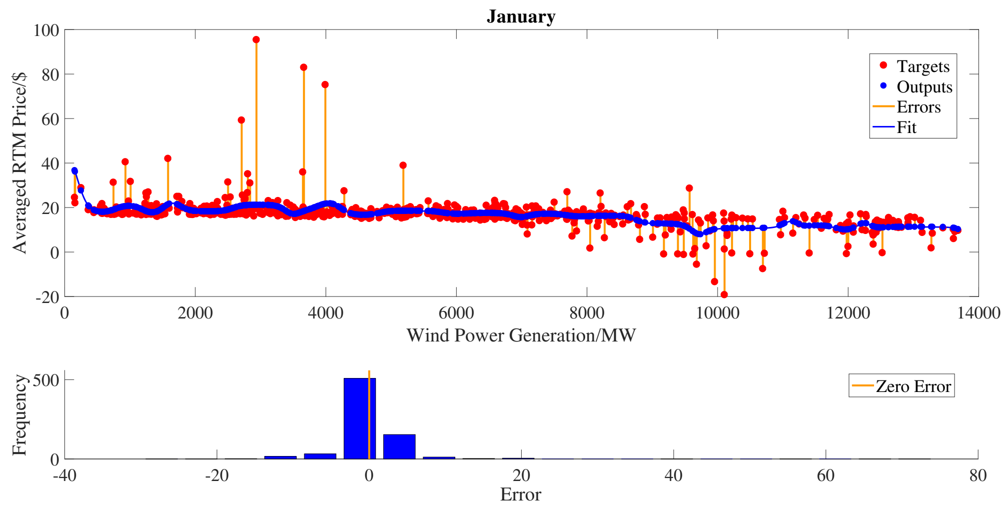


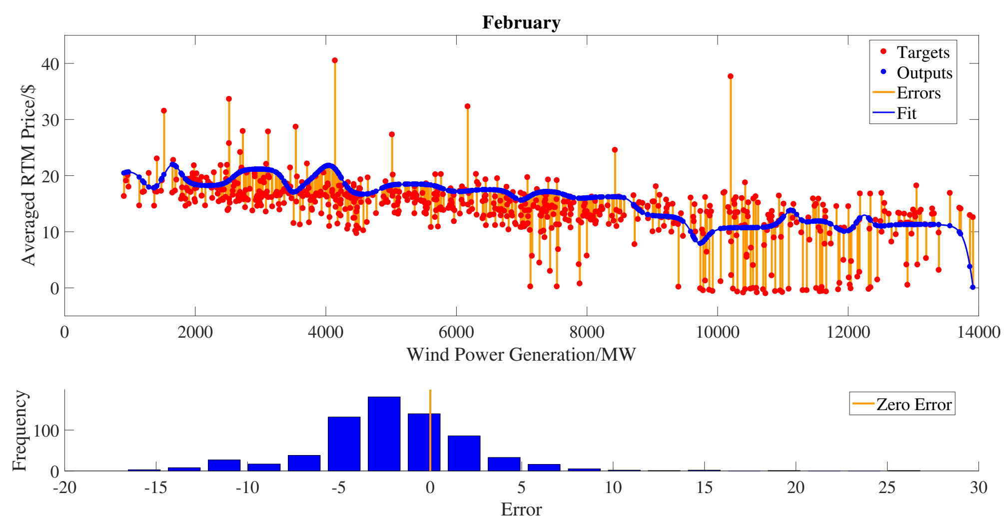
Figure 4.8: The Structure of NN

For rerouted method (two-stage method), the training and testing regression curves along with the error histogram are shown in Figure 4.9. The training MSE is 34.8300 and the testing MSE is 24.0220. For direct method (price-to-price method), I fix the predicting data size N for each time and adjust the historical data size M to obtain the best testing results. The procedure is shown in Figure 4.10 and the best results for each case are shown in Table 4.4.

As we can observe from Table 4.4, the rerouted method can reach a smaller



(a) Training Result



(b) Testing Result

Figure 4.9: Rerouted Method: NN with 30 Hidden Neurons

testing MSE than any of the direct method no matter how you resize M and N. In this case, when I use the rerouted method and NN with 30 hidden neurons, I can obtain the best testing performance. Compared to the results of the polyno-

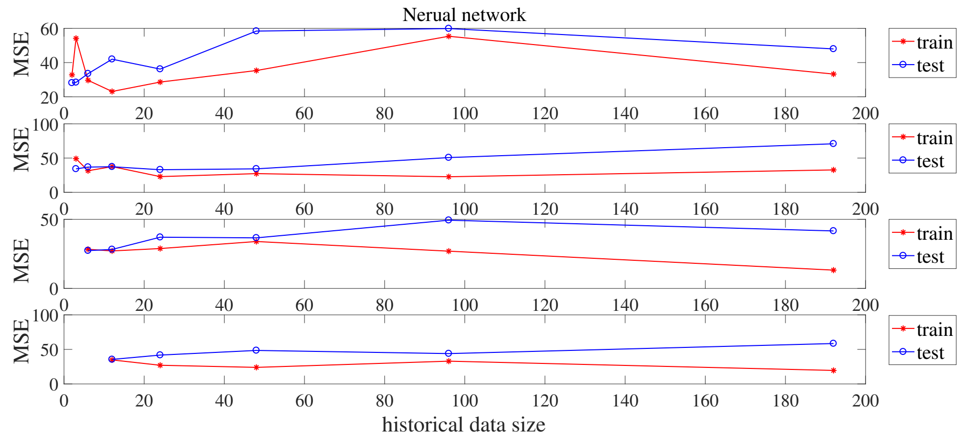
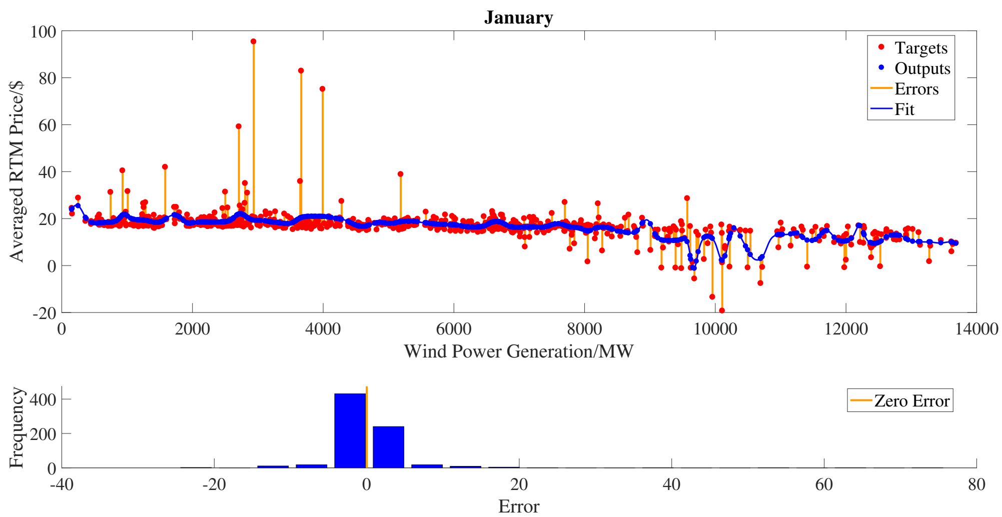


Figure 4.10: Direct Method: Data Size Determination

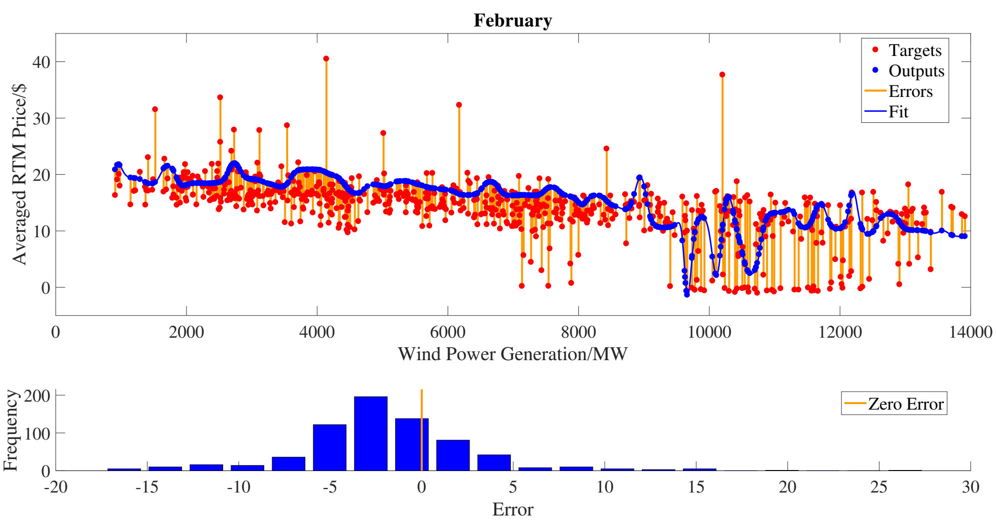
Table 4.4: Direct Method: the Best Testing Results According to N

N	1	2	3	6
M	2	24	6	12
Training MSE	32.8374	22.9490	28.3315	34.7806
Testing MSE	28.1420	32.9945	**27.3736**	35.4926

mial regression, I can see that this neural network has a better training MSE, but a worse testing MSE. Therefore, I try to find out if I can gain a better testing result by increasing the hidden layer size from 30 to 60. The simulation result is shown in Figure 4.11. The MSE for training is 33.4186, the MSE for testing is 26.3436. The result shows that when I increase the hidden layer size, I get a better training MSE, but a worse testing MSE. This means that increasing the sizes of the hidden neurons does not help us to obtain a better testing MSE but results in an over-fitting in this specific problem.



(a) Training Result



(b) Testing Result

Figure 4.11: Rerouted Method: NN with 60 Hidden Neurons

4.2.4 Deep Neural Network (DNN)

Owing to the poor performance of the simple neural networks, I employ DNNs to see if they can do better. DNNs are known for their powerful ability to learn the essential features of data sets from a small sample set. I find that there are not many papers make use of DNN to forecast electricity price. Therefore, I hold a detailed discussion here to present the advance of using DNN.

For rerouted method (two-stage method), I use 'nntool' in MATLAB to build DNNs with 3 different number of layers. The first DNN has 14 layers: 1 input layer, 13 hidden layers, and 1 output layer. The second DNN has 11 layers: 1 input layer, 10 hidden layers, and 1 output layer. The last DNN has 7 layers: 1 input layer, 6 hidden layers, and 1 output layer. All of them has a layer size of 30 for each hidden layer. The transfer function for the last hidden layer is 'purelin', the transfer function for all the other hidden layers is 'tansig'. All of the DNNs listed above are trained by the Levenberg-Marquardt algorithm. The structure of the DNN is shown in Figure 4.12.

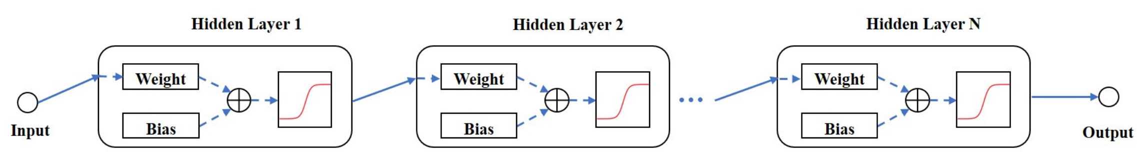
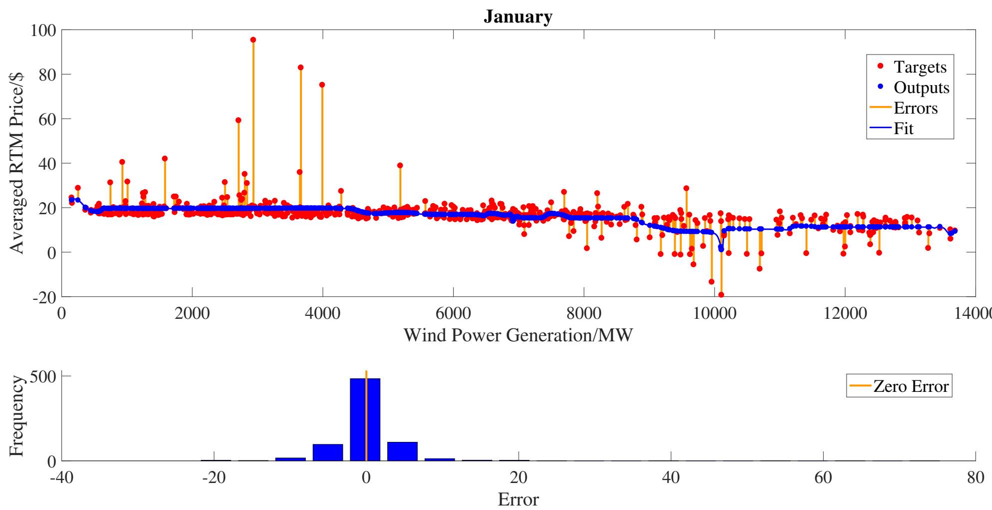


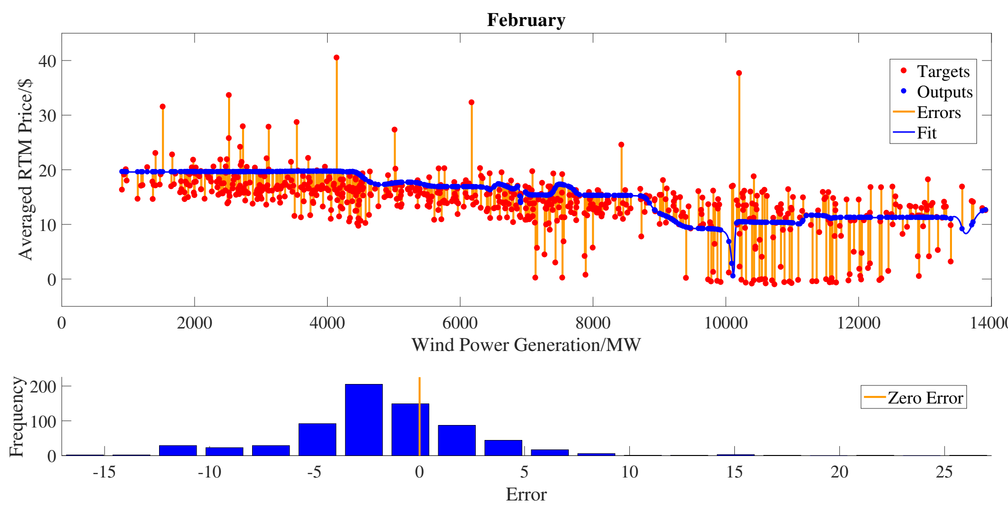
Figure 4.12: The structure of DNN

The training and testing regression curves along with the error histogram for the deep neural network with 7 layers are shown in Figure 4.13. The training and testing regression curves along with the error histogram for the deep neural network with 11 layers are shown in Figure 4.14. The training and testing regression curves along with the error histogram for the deep neural network with 14 layers are shown in Figure 4.15. The MSEs of the deep neural networks are shown in

Table 4.5.



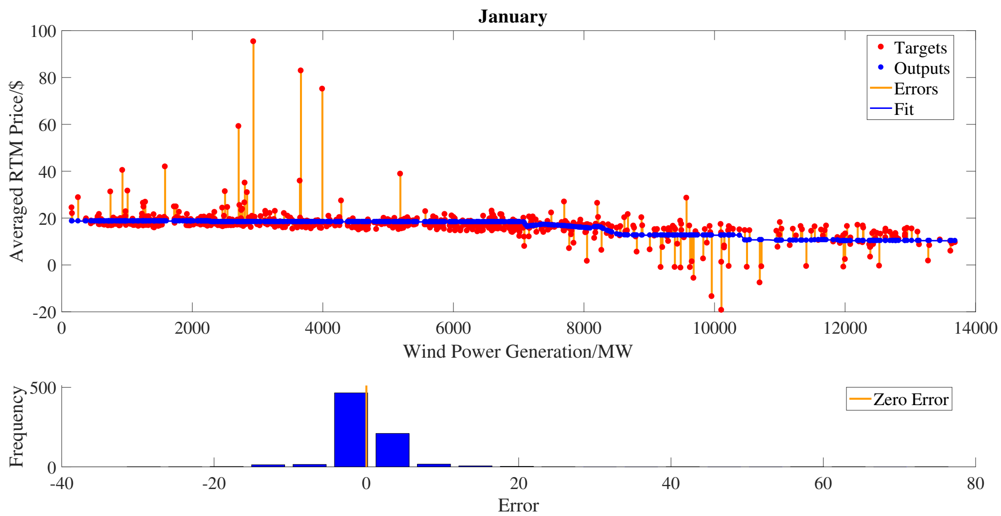
(a) Training Result



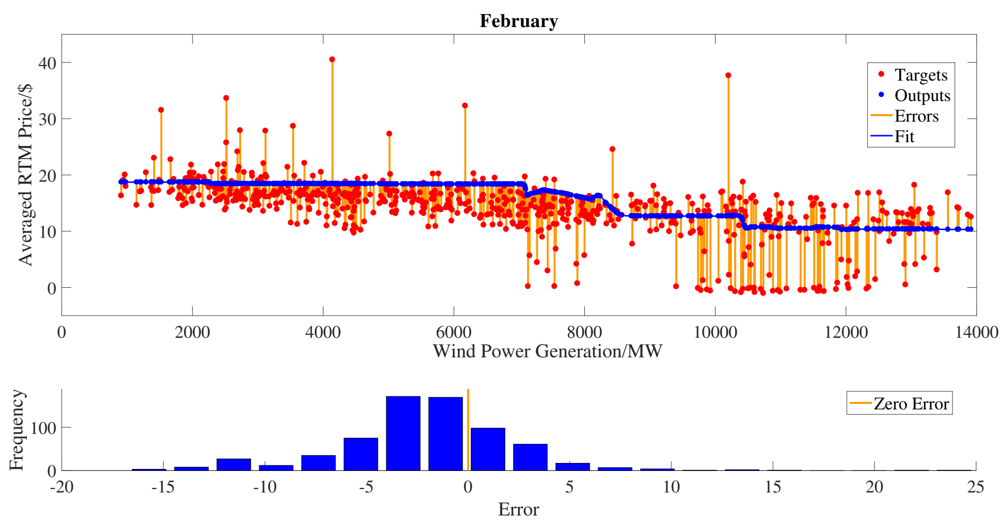
(b) Testing Result

Figure 4.13: Rerouted Method: DNN with 7 Layers

As we can see from Table 4.5, when I increase the number of hidden layers, the training MSE doesn't increase much but the testing MSE becomes smaller. This



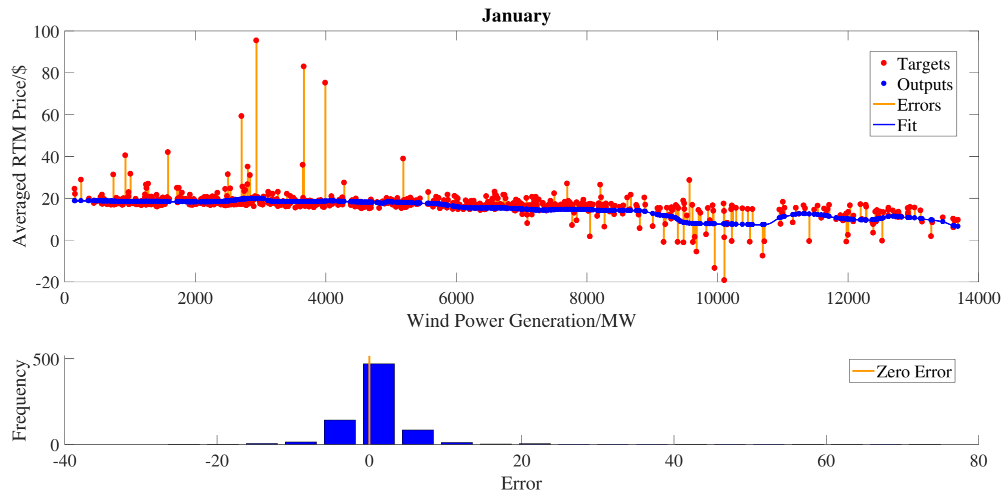
(a) Training Result



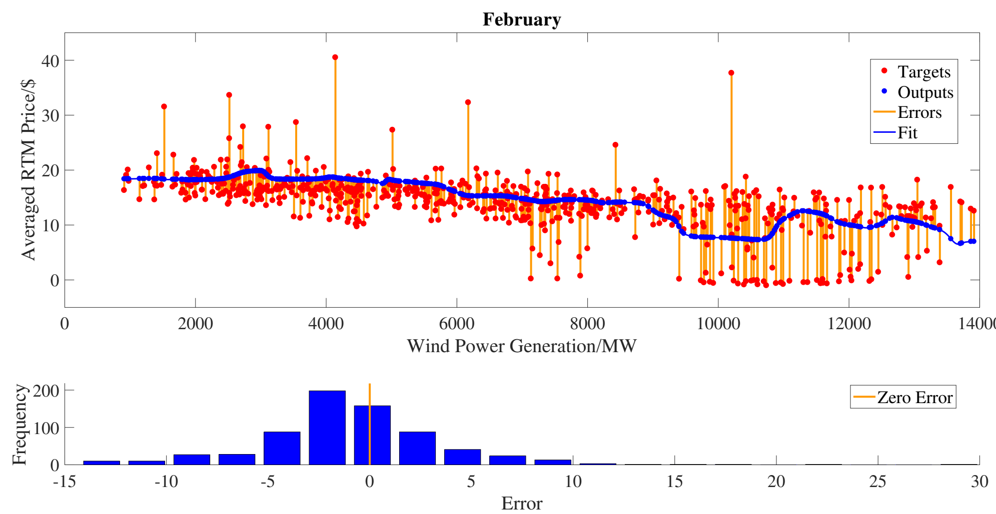
(b) Testing Result

Figure 4.14: Rerouted Method: DNN with 11 Layers

indicates that I may achieve better testing MSEs if I add more hidden layers. However, the training time will surely be longer if I increase the number of hidden layers. Hence, there's a trade-off between the number of layers and the time for



(a) Training Result



(b) Testing Result

Figure 4.15: Rerouted Method: DNN with 14 Layers

training. Therefore, we should carefully choose the number of layers to get the relatively good results.

Table 4.5: The MSEs of Deep Neural Networks

MSE \ Layers	7	11	14
Training	34.9439	36.8437	36.1379
Testing	22.7658	23.2444	**20.5065**

4.2.5 Additional Discussion

In order to testify the effectiveness and the generalization ability, real-time market(RTM) price and wind power generation from other months are chosen to perform the verification. All the simulation results are shown in Table 4.6 and it is obvious that the proposed two-stage method is better than the direct method.

Table 4.6: The MSEs of Other Months

Method \ Month	Index	April	May	June	July
Direct	MSE	81.5258	49.5156	120.1935	159.2903
Rerouted	MSE	65.7889	41.4192	56.4694	53.2651

In addition, to testify the effectiveness and the generalization ability, I also make use of Monte Carlo tools to test the model stability against noise. As wind power generation will be influenced by the noise, I add different noise levels to test the model endurance towards the noise. The experimental results are shown

Table 4.7: Model Endurance Towards Noise

Noise level	5%	10%	12%	15%
MSE (Feb)	22.7429	25.2443	26.0906	29.4196

in Table 4.7. Recalling previous results, the direct method can achieve the best testing accuracy of 27.1534. Compared to Table 4.7, I can draw the conclusion that the proposed method can endure at most 12% noise from the system and environment.

Chapter 5

CONCLUSION

This paper develops a novel two-stage method to forecast the real-time market (RTM) price. This new method, namely the rerouted method (two-stage method), predict the future price using historical RTM bus price along with system-wide wind power generation. The main contributions of this work are the diversified input data resources such as highly correlated power data and the validation step to quantify the the best training interval for different data set.

By conducting a comparison to the conventional method, namely direct method (price-to-price method), we confirm our conjecture that we can obtain higher accuracy if we diversify the data source. Furthermore, when we examine the relationship between the input and the output, we find that they are actually in a causal relationship. This causal relationship combined with some physical models can guarantee us with better results.

To verify the effectiveness and the generalization ability of the model, we conduct simulations over the other four months. The result shows that the proposed method is much more accurate than the direct method. To further explore the model stability against noise, we set up different noise levels of the wind power generation. The results show that the proposed model also has good stability towards the noise.

Other related subjects of interest for further research could be the improvement of the prediction accuracy by taking into consideration other renewable energies, solar energy most likely. These features should have causal relationships towards electricity price. In this paper, most of the methods are simple models that do not have a lot of parameters. We can develop more complex models to achieve better results.

[1] Lin, Lin and Cunshan, Zhou and Vittayapadung, Saritporn and Xiangqian, Shen and Mingdong, Dong, "Opportunities and challenges for biodiesel fuel," *Applied energy*, vol. 88, no. 4, pp. 1020–1031, 2011.

[2] Dhinesh, B and Raj, Y Maria Ambrose and Kalaiselvan, C and KrishnaMoorthy, R, "A numerical and experimental assessment of a coated diesel engine powered by high-performance nano biofuel," *Energy Conversion and Management*, vol. 171, pp. 815–824, 2018.

[3] Vigneswaran, R and Annamalai, K and Dhinesh, B and Krishnamoorthy, R, "Experimental investigation of unmodified diesel engine performance, combustion and emission with multipurpose additive along with water-in-diesel emulsion fuel," *Energy Conversion and Management*, vol. 172, pp. 370–380, 2018.

[4] Dhinesh, B and Annamalai, M, "A study on performance, combustion and emission behaviour of diesel engine powered by novel nano nerium oleander biofuel," *Journal of Cleaner Production*, 2018.

[5] Nomura, Noboru and Inaba, Atsushi and Tonooka, Yutaka and Akai, Makoto, "Life-cycle emission of oxidic gases from power-generation systems," *Applied Energy*, vol. 68, no. 2, pp. 215–227, 2001.

[6] A. A. Sánchez de la Nieta, V. González, and J. Contreras, "Portfolio decision of short-term electricity forecasted prices through stochastic programming," *Energies*, vol. 9, no. 12, p. 1069, 2016.

[7] A. Najafi, H. Falaghi, J. Contreras, and M. Ramezani, "Medium-term energy hub management subject to electricity price and wind uncertainty," *Applied Energy*, vol. 168, pp. 418–433, 2016.

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

[9] D. Wang, H. Luo, O. Grunder, Y. Lin, and H. Guo, "Multi-step ahead electricity price forecasting using a hybrid model based on two-layer decomposition technique and bp neural network optimized by firefly algorithm," *Applied Energy*, vol. 190, pp. 390–407, 2017.

[10] C. García-Martos, J. Rodríguez, and M. J. Sánchez, "Modelling and forecasting fossil fuels, co2 and electricity prices and their volatilities," *Applied Energy*, vol. 101, pp. 363–375, 2013.

[11] Z. Yang, L. Ce, and L. Lian, "Electricity price forecasting by a hybrid model, combining wavelet transform, arma and kernel-based extreme learning machine methods," *Applied Energy*, vol. 190, pp. 291–305, 2017.

[12] J. P. d. S. Catalão, S. J. P. S. Mariano, V. Mendes, and L. Ferreira, "Short-term electricity prices forecasting in a competitive market: A neural network approach," *Electric Power Systems Research*, vol. 77, no. 10, pp. 1297–1304, 2007.

[13] F. J. Nogales, J. Contreras, A. J. Conejo, and R. Espínola, "Forecasting next-day electricity prices by time series models," *IEEE Transactions on power systems*, vol. 17, no. 2, pp. 342–348, 2002.

[14] J. Contreras, R. Espinola, F. J. Nogales, and A. J. Conejo, "Arima models to predict next-day electricity prices," *IEEE transactions on power systems*, vol. 18, no. 3, pp. 1014–1020, 2003.

[15] A. Bello, D. W. Bunn, J. Reneses, and A. Muñoz, "Medium-term probabilistic forecasting of electricity prices: A hybrid approach," *IEEE Transactions on Power Systems*, vol. 32, no. 1, pp. 334–343, 2017.

[16] J. Ma, M. Yang, X. Han, and Z. Li, "Ultra-short-term wind generation forecast based on multivariate empirical dynamic modeling," *IEEE Transactions on Industry Applications*, vol. 54, no. 2, pp. 1029–1038, 2018.

[17] C. Wan, Z. Xu, P. Pinson, Z. Y. Dong, and K. P. Wong, "Probabilistic forecasting of wind power generation using extreme learning machine," *IEEE Transactions on Power Systems*, vol. 29, no. 3, pp. 1033–1044, 2014.

[18] J. Dowell and P. Pinson, "Very-short-term probabilistic wind power forecasts by sparse vector autoregression," *IEEE Transactions on Smart Grid*, vol. 7, no. 2, pp. 763–770, 2016.

[19] J. Che and J. Wang, "Short-term electricity prices forecasting based on support vector regression and auto-regressive integrated moving average modeling," *Energy Conversion and Management*, vol. 51, no. 10, pp. 1911–1917, 2010.

[20] B. Szkuta, L. A. Sanabria, and T. S. Dillon, "Electricity price short-term forecasting using artificial neural networks," *IEEE transactions on power systems*, vol. 14, no. 3, pp. 851–857, 1999.

[21] H. Drucker, C. J. Burges, L. Kaufman, A. J. Smola, and V. Vapnik, "Support vector regression machines," in *Advances in neural information processing systems*, 1997, pp. 155–161.

[22] Y. Chen, P. Xu, Y. Chu, W. Li, Y. Wu, L. Ni, Y. Bao, and K. Wang, "Short-term electrical load forecasting using the support vector regression (svr) model to

calculate the demand response baseline for office buildings," *Applied Energy*, vol. 195, pp. 659–670, 2017.

[23] M. van Gerven and S. Bohte, *Frontiers Research Topic: Artificial Neural Networks as Models of Neural Information Processing*. Frontiers in Computational Neuroscience, Retrieved 2018-02-20, https://www.frontiersin.org/research-topics/4817/artificial-neural-networks-as-models-of-neural-information-processing.

[24] P. Bento, J. Pombo, M. Calado, and S. Mariano, "A bat optimized neural network and wavelet transform approach for short-term price forecasting," *Applied Energy*, vol. 210, pp. 88–97, 2018.

[25] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, http://www.deeplearningbook.org.

[26] I. P. Panapakidis and A. S. Dagoumas, "Day-ahead electricity price forecasting via the application of artificial neural network based models," *Applied Energy*, vol. 172, pp. 132–151, 2016.

[27] Deng, Li and Yu, Dong, "Deep learning: methods and applications," *Foundations and Trends® in Signal Processing*, vol. 7, no. 3–4, pp. 197–387, 2014.

[28] Bengio, Yoshua, "Learning deep architectures for ai," *Foundations and trends® in Machine Learning*, vol. 2, no. 1, pp. 1–127, 2009.

[29] D. Keles, J. Scelle, F. Paraschiv, and W. Fichtner, "Extended forecast methods for day-ahead electricity spot prices applying artificial neural networks," *Applied Energy*, vol. 162, pp. 218–230, 2016.

[30] Sansom, Damien C, Tom Downs, and Tapan K. Saha, "Evaluation of support vector machine based forecasting tool in electricity price forecasting for australian national electricity market participants," *Journal of Electrical & Electronics Engineering, Australia*, vol. 22, no. 3, p. 227, 2003.

[31] H. Y. S. Tao, A. K. Srivastava, R. L. Pineda, and P. Mandal, "Wind power generation impact on electricity price in ercot," in *Power and Energy Society General Meeting, 2012 IEEE*. IEEE, 2012, pp. 1–7.

[32] http://www.ercot.com/, accessed 20-November-2018.

[33] http://www.ercot.com/about/contact/inforequest, accessed 20-November-2018.