Ensemble Learning-based Electricity Price Forecasting for Smart Grid Deployment

by

Bijay Neupane

A Thesis Presented to the
Masdar Institute of Science and Technology
in Partial Fulfillment of the Requirements for the Degree of
Master of Science
in
Computing and Information Science

©2013 Masdar Institute of Science and Technology
All rights reserved.
Abstract

In a deregulated electricity market, forecasting electricity prices is essential to help stakeholders with the decision making process. Electricity price forecasting is an inherently difficult problem due to its special characteristics of dynamicity and nonstationarity. Further deployment of the smart grid for power distribution enables a two-way communication between the supplier and the consumer. This enables the supplier to price the energy based on the consumption feedback from the consumer, and the consumer can schedule their consumption behavior to achieve optimal utilization. In our research, our goal is to tackle dynamicity involved in price forecasting. We employ three different machine learning-based prediction algorithms, and proposed two versions of ensemble based models (fixed weights and varying weights) which make predictions by taking into account the individual predictions from the three algorithms. The input features are selected from a pool of features derived from information such as past electricity price data, weather data, and calendar data. A wrapper method for feature selection is used in which the ANN model is continuously trained and updated in order to select the best feature set. The performance of the proposed method is evaluated and compared with the published results of the state-of-the-art Pattern Sequence-based Forecasting (PSF) method on the same data sets and our method is observed to provide superior results.
This research was supported by the Government of Abu Dhabi to help fulfill the vision of the late President Sheikh Zayed Bin Sultan Al Nayhan for sustainable development and empowerment of the UAE and humankind.
Acknowledgments

First of all I would like to extend my gratitude to my family member without their help and support, it is impossible.

I would like to take this opportunity to thanks those who actively guided and helped in this research. Foremost, I would like to express my deep appreciation to my advisor Dr. Zeyar Aung for his continuous support for my M.Sc. study and research. His guidance, patience, motivation, and support helped me to develop a deep understanding of the subject. Beside my advisor, I would like to thanks my thesis supervisor committee members: Dr. Wei Lee Woon and Dr. Jacob Crandall for the their valuable time, comments, and advice.

Bijay Neupane

Masdar City, May 29, 2013.
# Contents

1 Introduction 1

1.1 Price Forecasting in Deregulated Markets 1

1.2 Motivation 3

1.3 Thesis Statement - Objectives 4

1.4 Research Contribution 4

1.5 Thesis Organization 5

2 Price Prediction in Smart Grid 6

2.1 Basic Structure of Smart Grid 7

2.2 Price Prediction for Smart Grid Deployment 9

2.3 Price Prediction Challenges in Smart Grid 10

3 Literature Review 11

3.1 Overview of Price Forecasting Techniques 11

3.2 Existing Forecasting Techniques 13

3.3 Factors Influencing Electricity Price 17
### 3.4 Prerequisite for a STPF System

17

### 4 Proposed Prediction Model

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.1</td>
<td>Objective</td>
<td>19</td>
</tr>
<tr>
<td>4.2</td>
<td>Architecture</td>
<td>20</td>
</tr>
<tr>
<td>4.2.1</td>
<td>Fixed Weight Method</td>
<td>22</td>
</tr>
<tr>
<td>4.2.2</td>
<td>Varying Weight Method</td>
<td>23</td>
</tr>
</tbody>
</table>

### 5 Experimental Setup

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.1</td>
<td>Data Set</td>
<td>26</td>
</tr>
<tr>
<td>5.2</td>
<td>Evaluation Metrics</td>
<td>27</td>
</tr>
<tr>
<td>5.3</td>
<td>Explored Algorithm</td>
<td>28</td>
</tr>
<tr>
<td>5.3.1</td>
<td>Artificial Neural Networks</td>
<td>28</td>
</tr>
<tr>
<td>5.3.2</td>
<td>Support Vector Machine</td>
<td>31</td>
</tr>
<tr>
<td>5.3.3</td>
<td>Random Forests</td>
<td>36</td>
</tr>
<tr>
<td>5.4</td>
<td>Experimental Process Flow</td>
<td>38</td>
</tr>
<tr>
<td>5.4.1</td>
<td>Feature Engineering</td>
<td>40</td>
</tr>
<tr>
<td>5.4.2</td>
<td>Feature Selection</td>
<td>40</td>
</tr>
<tr>
<td>5.4.3</td>
<td>Incorporating Other Features</td>
<td>42</td>
</tr>
<tr>
<td>5.4.4</td>
<td>Normalization</td>
<td>43</td>
</tr>
<tr>
<td>5.5</td>
<td>Constructing Learning Model</td>
<td>43</td>
</tr>
</tbody>
</table>

### 6 Results and Discussions

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.1</td>
<td>Experiment I: With Fixed Weight Algorithm</td>
<td>46</td>
</tr>
<tr>
<td>6.1.1</td>
<td>Experiment IA: 2004–2006</td>
<td>46</td>
</tr>
<tr>
<td>6.1.2</td>
<td>Experiment IB: 2008–2012</td>
<td>55</td>
</tr>
<tr>
<td>6.2</td>
<td>Experiment II: With Varying Weight Algorithm</td>
<td>57</td>
</tr>
<tr>
<td>6.2.1</td>
<td>Experiment IIA: 2004–2006</td>
<td>58</td>
</tr>
<tr>
<td>6.2.2</td>
<td>Experiment IIB: 2008–2012</td>
<td>59</td>
</tr>
</tbody>
</table>
List of Tables

5.1 Features used for training and testing of New York (NYISO) dataset. 42

6.1 MAE, MER and MAPE in NYISO markets for year 2006 (Experiment IA). 47

6.2 MAE, MER and MAPE in ANEM markets for year 2006 (Experiment IA). 50

6.3 MAE, MER and MAPE in OMEL markets for year 2006 (Experiment IA). 53

6.4 MAPE, MAE, and MER performance indicators for NYISO and ANEM markets for years 2011–2012 (Experiment IB). 57

6.5 MAE, MER and MAPE in NYISO markets for year 2006 (Experiment IIA). 62

6.6 MAE, MER and MAPE in ANEM markets for year 2006 (Experiment IIA). 63

6.7 MAE, MER and MAPE in OMEL markets for year 2006 (Experiment IIA). 64
6.8 MAPE, MAE, and MER performance indicators for NYISO and ANEM markets for years 2011–2012 (Experiment IIB). . . . . . 65
List of Figures

2.1 Architecture and components of the smart grid (reproduced with permission from [28]). .................................................. 8

4.1 Architecture of prediction model. ........................................ 21

5.1 A simple architecture of neuron. ........................................ 29
5.2 Simple two-layer feed-forward Neural Network. ..................... 30
5.3 Overview of the proposed price forecasting process flow. ........... 39

6.1 The worst forecasted day in NYISO market in year 2006. .......... 48
6.2 The best forecasted day in NYISO market in year 2006. ............ 49
6.3 The worst forecasted days in NYISO market in year 2006. .......... 51
6.4 The best forecasted days in NYISO market in year 2006. .......... 52
6.5 The best forecasted days in NYISO market in year 2006. .......... 54
6.6 The worst forecasted days in NYISO market in year 2006. .......... 55
6.7 The worst forecasted day in NYISO market in year 2012. .......... 58
6.8 The best forecasted day in NYISO market in year 2012. .......... 59
6.9 The worst forecasted day in ANEM market in year 2012. .......... 60
6.10 The best forecasted day in ANEM market in year 2012.
1.1 Price Forecasting in Deregulated Markets

By liberalizing pricing in the electricity market of a smart grid, the suppliers will be able to reduce the waste of excess electricity when demand is low, and ensure sufficient supply when demand is high. This scenario exists since the usage of electricity will depend on the price per unit at a particular time of day and consumers have access to an electricity pool where they decide what time to buy electricity from the pool. Thus, a cost-conscious consumer will be concerned about the price in the coming hours, days, or even weeks, and will try to optimize their utilization and minimize their total bill through smart use of electricity. With dynamic pricing systems where the consumers would pay based on their time of consumption and the amount of load they consume, it is essential for the consumers to have some price prediction mechanism to assist in scheduling their energy consumption strategy in advance. However, with the majority of consumers making decisions based
on some prediction algorithm might bring some imbalance in energy consumption pattern in the society, which might increase the peak demand for certain hours, thus triggering the supplier to revise its pricing scheme again. Therefore, electricity price forecasting is an inherently difficult problem due to its unique characteristics of dynamicity and nonstationarity. Its complexity is further increased with the introduction of the smart grid which enable interactive and price-responsive environment.

Under a fixed pricing scheme in the electricity market, consumption of electricity follows a distinct peak demand curve. This peak demand forces producers to use resources to meet the peak demand. These resources are redundant for the rest of the time. To overcome this inefficiency, the concept of demand management is put forward as a part of the smart grid initiative. A smart grid utilizes the information about the behaviors of supplier and consumer of electricity and tries to optimize the production and distribution of electricity grid. It uses two-way communication between the consumer and the supplier to exchange information in order to optimizes utilization for the consumer and smooth the demand curve. The general idea of demand management is to design a pricing mechanism which decides the hourly prices that can persuade the consumers to change their usage patterns in order to lower the peak demand, with the expectation that the consumers will respond to it. Another objective of this mechanism is to eliminate fluctuations in the demand beyond a defined threshold. Thus every market player including consumer and retailer are also very concerned about the price in the coming hours, days or may be in the coming month.
1.2 Motivation

The problem of electricity price forecasting is related yet distinct from that of electricity load forecasting. Although the demand (load) and the price are correlated, their relation is non-linear. The load is influenced by various factors such as non-storability of electricity, consumers’ behavioral patterns, and seasonal changes in demand. The price, on the other hand, is affected by those aforesaid factors as well as additional aspects such as financial regulations, competitors’ pricing, dynamic market factors, and various other macro- and micro economic conditions. As a result, the price of electricity is a lot more volatile than the electricity load. Interestingly, when dynamic pricing strategies are introduced, prices become even more volatile, where the daily average price changes by up to 50% while other commodities may exhibit about 5% change [25].

In addition, very limited number of research works dedicated to the deregulated market in the smart grid can be found in the field of price forecasting. Because of the increased complexity of the deregulated market, traditional approaches of price forecasting usually exhibit poor performances in such markets. As extra flexibilities in pricing schemes are introduced and higher supplier-consumer interactions are available, it is very unlikely that traditional approaches of price forecasting can effectively model different factors affecting electricity price in smart grid.

Even though a lot of research has been performed on electricity price forecasting, none of the experiments provides a sufficiently accurate model with less than 5% mean absolute prediction error (MAPE) value in a consistent manner. Thus, a more accurate price forecasting system is necessary since many retailers and their businesses depend on the prices of electricity.
CHAPTER 1. INTRODUCTION

1.3 Thesis Statement - Objectives

Our objective is to build an accurate electricity price forecasting model for generating hour-ahead price forecast. This price forecasting is important for the transmission company to schedule short-term generator outages, design load response programs as well as bid into the market strategically and manage its assets optimally. When an accurate price forecasting system is available, large consumers can stem their electricity usage plan strategically to maximize their utility.

1.4 Research Contribution

Many papers have been published in the area of price forecasting using Artificial Neural Network (ANN) and other algorithms. Here, our main contribution lies on extracting the best features from a pool of features and training three different algorithms with these features in order to create a real-time forecasting model. As mentioned in [24], lagged prices are generally used in price forecasting due to its high auto correlation with electricity market prices. However, in a real-time setup, except system load and previous hour prices, other features are not available, thus needing us to choose features only from the available pool. The input features are selected from a pool of features derived from information such as past electricity price data, weather data, and calendar data. A wrapper method for feature selection is used in which the ANN model is continuously trained and updated in order to select the best feature set. (The preliminary results using ANN were published in [38].)

In this research we employ three different machine learning-based prediction algorithms, and proposed two versions of ensemble based models (i.e., fixed weights and varying weights) which make predictions by taking into account the individual predictions from the three algorithms. The input features are selected from a pool of
features derived from information such as past electricity price data, weather data, and calendar data. A wrapper method for feature selection is used in which the ANN model is continuously trained and updated in order to select the best feature set.

1.5 Thesis Organization

This chapter mainly discusses the background, motivation for the research, objective and breakdown of the thesis work. Chapter 2 covers background of smart grid and the price prediction challenges in smart grid. Chapter 3 presents the literature review regarding the electricity price prediction. It includes an overview of price forecasting, different research carried in this field, different time horizon of price prediction, factor affecting the prediction and prerequisite of price prediction. Chapter 4 presents the core idea of the research. It discusses proposed model and algorithm used in the research. Chapter 5 presents experimental setup performed for the research. It explains data source, feature creation and selection technique, and complete prediction model using ensembles learning approach. Chapter 6 presents the experimental results of the proposed method in comparison with a state-of-the-art method, namely PSF. The last chapter (Chapter 7) of the thesis encapsulates conclusion and future outlook of the project.
Price Prediction in Smart Grid

This chapter discusses the basic structure of the smart grid, and challenges associated with the construction of smart grid. We analyze the effect of different components of smart grid on pricing schemes. At last, we focus on the requirements and challenges associated with price prediction methods for smart grid deployment.

The current electrical grid, which is considered the greatest engineering achievement of the 20th century, is gradually facing challenges due to its limitations. However, it is increasingly out of date and overburdened, leading to costly blackouts and brownouts. For example, the 2003 blackout in the northern and eastern U.S. and Canada caused a $6 billion loss in economic revenue [4].

Integration of different energy sources to the main power grid with almost no losses in energy, maintaining the integrity and security of the grid, is the biggest challenge of the present grid system. There exist a problem in transmission and distribution of electricity which further limit the grid reliability and efficiency. With the introduction of smart grid, real-time energy uses monitoring, dynamic pricing scheme,
two way communication, greater fault tolerance, better distribution of electricity
and better planning of electricity generation will be possible. Centralized electrical
supply networks are limited to fossil fuel resources and prone to complete system
failures. Decentralized solutions (Micro-Grids) for provision of electricity to every
part of the world are gaining popularity especially among the developing countries.
These systems can be powered through fossil fuel or renewable and can be linked
together to form an electrical network. Small achievement in efficiency, will give
higher reduction in carbon emission and increase in review generation.

Smart grid offer valuable technology that can be implemented in near future
and promise the long-term intelligence which increase the efficiency and reliability
in power distribution. It will enable dynamic power distribution and consumption
with two way communication between grid and consumer. It will allow the pro-
ducer to adopt different pricing scheme based on user consumption pattern, source
of generation, time of use etc. which help to maintain the production and demand
curve.

2.1 Basic Structure of Smart Grid

Smart grid refers to the modernization of the current electrical grid by utilizing
information, communication, and control technologies so as to have bi-directional
flow of information and electricity to intelligently integrate the actions of all users
connected to which generators, consumers and those that do both in order to effi-
ciently deliver sustainable, economic and secure electricity supplies [10].

The smart grid is characterized by several new trends and features: smart me-
ters, demand response mechanisms, online customer interactions though PCs/mobile
devices, dynamic electricity tariffs, online billing, incorporation of renewable en-
ergy generation (such as solar and wind energy) and electric vehicles, more reliable
power transmission and distribution, dynamic load balancing, better power quality, better power security, etc. [29, 30]. The architecture and components of the smart grid are illustrated in 2.1.

Figure 2.1: Architecture and components of the smart grid (reproduced with permission from [28]).

Communications technology and infrastructure is at the heart of improvements to the electrical grid, which will collate data provided by smart meters, sensors, computer systems, and many other devices into understandable and actionable information for consumers and utilities [10]. While Information Technology (IT) is one of the major driving forces behind a smart grid as well as various IT systems and techniques such as artificial intelligence, high performance computing, simulation and modeling, data network management, database management, data warehousing, and data mining are to be used to facilitate smooth running of the smart grid [20].
2.2 Price Prediction for Smart Grid Deployment

By liberalizing the pricing in the electricity market of the smart grid, the suppliers will be able to reduce the wastage of excess electricity when demand is low, and ensure sufficient supply when demand is high. Electricity provider always has the problem in designing pricing scheme based on the user uses pattern. In smart grid producer can design their pricing scheme based on the information collected from smart devices. Producer can analysis the user demand, uses pattern, cost associated to production of energy to full fill the demand and adopt the dynamic pricing scheme. User can schedule their smart devices like, smart washing machine, vacuum cleaner, Air condition to operate when the prices are low.

This scenario exists since the usage of electricity will depend on the price per unit at a particular time of day when consumers have access to an electricity pool where they decide what time to buy electricity from the pool. Thus, a cost-conscious consumer will be concerned about the price in the coming hours, days, or even weeks, and will try to optimize his/her utilization and minimize his/her total bill through smart use of electricity. With dynamic pricing system where the consumers would pay based on their consumption time and amount of load they consume, it is essential for the consumers to have some price prediction mechanism so that they can schedule their energy consumption strategy in advance.

However, with the majority of consumers making decisions based on some prediction algorithm might bring some imbalance in energy consumption pattern in the society, which might increase the peak demand for certain hours, thus triggering the supplier to revise its pricing scheme again. Therefore, electricity price forecasting is an inherently difficult problem due to its special characteristics of dynamicity, nonstationarity, and its complexity is even increased with introduction of the smart grid.
2.3 Price Prediction Challenges in Smart Grid

Dynamicity in pricing scheme, two-way communication and different components associated with smart grid creates great challenges in designing prediction algorithm. A Price forecasting algorithm should be able to overcome all the drawbacks of traditional grids and incorporate the scenario where users have greater flexibility and control over consumption of load. Some challenges associated with price prediction for smart grid deployment are:

- Improve the prediction accuracy.
- Model different scenario of consumer behavior, like uses pattern, response to the price, demand, etc.
- Consider type of resources used for electricity generation.
- Consider price differences between different community or area.
- Capture users motivation towards lowering the peak demand.
- Maintain prediction accuracy when most of the consumer in the society try to predict the price and consume load accordingly, which might cause drastic change in peak demand, which lead to unexpected fluctuation in prices.
CHAPTER 3

Literature Review

The research on electricity price forecasting has been in existence for decades, and a numbers of researches has been proposed and developed for deployment. This part of research focus on existing research on price forecasting, analyzing different research prospective, establishment of generalized research question, relevance and feasibility of proposed research project and explore areas of possible improvement. This chapter also contains the comparative study of various existing price forecasting technique, existing models, reviewed research papers and their finding and remarks.

3.1 Overview of Price Forecasting Techniques

Electricity price forecasting has become one of the most significant aspects in electricity market for trading and planning. The positive economic consequences have attracted many stake holders to invest time and money for development of new
algorithm for precise price prediction. This financial aspect has drawn alarming interest to many researchers, and has produced many significant research and contribution in electricity price forecasting. This research thrust is more deepen with the introduction of smart grid. The concept of dynamic pricing scheme, a component of smart grid has further extended the research interest in electricity price forecasting. Precise knowledge about the future electricity price will help consumer to plan consumption and supplier to plan the production based on user behavior. This chapter discusses some of the most popular algorithm, technique and research proposed till date in the field of electricity price forecasting.

Generally price forecasting models can be categorized in to two groups: time-series and game theory models [14, 26]. Time-series models use a non-dynamic approach, where prediction are made based on previously observed values only. The second category involves the dynamic model that recognizes the fact that the price is not only a function of the time of the day, but also depends upon the various factors. This approach tries to model the market participant’s strategy and find mathematical solution for predicting the price. In oligopolistic electricity markets, participants shift their bidding curve from their actual marginal cost to maximize their profits these models include mathematical solution for games and price evolution can be considered as the outcome of a power transaction game [5].

Further forecasting can be classified based on planning horizon duration.

**Short Term Price Forecasting (STPF)**

Short term forecasting is important for quick decision making process, markets can plan their bidding strategy using the forecasted price to maximize their profit in deregulated market [31]. In smart grid consumer can make decision of power consumption based on the current and predicted near future price. Short Term prediction includes next hour price prediction, day ahead price prediction.
Medium Term Price Forecasting (MTPF)

Medium term price forecasting include prediction for next week, next month or up to 1 year. In traditional pricing scheme MTPF is affected by seasonal effect, like rise in electricity price in summer, holidays due to higher demand and decline in winter. In smart grid deployment other factor like type of source, demand from a community, peak time of the day etc might influence electricity price. Medium-term foretasted price can be used by supplier to optimize their production cost by planning the resource allocation for generation of electricity.

Long Term Price Forecasting (LTPF)

Time horizon for Long term price forecasting varies from couple of year to decades. Long term price trends are used by policy maker to plan pricing scheme and management of resources. Investor uses it for analyzing recovery of investment in power plant construction, production and transmission. Based on the type of production, price forecast can be used to conserve resources like, nuclear plant can make planning regarding purchase of uranium, hydro plant can consider construction of reservoir and solar/wind farm can make further analysis on construction of new plan based on cost analysis.

3.2 Existing Forecasting Techniques

Due to the importance of accurate price forecasting in volatile electricity market, a number of approaches have been presented in the literature. These approaches range from traditional time series analysis to machine learning techniques for forecasting future prices. ARIMA [19, 25] and GARCH [22] models are some examples of traditional methods. On the other hand, artificial neural network (ANN) [7, 43], ARIMA model enhanced with wavelet transform [18], hidden markov
model [46], fuzzy inferred neural network [6], and support vector regression are some examples of machine learning techniques. In this section we will analyze few of these methods.

ARIMA

In [19], the authors used an ARIMA model based on wavelet transformation for electricity price forecasting. Historical data was first split using the wavelet transform before the application of ARIMA modeling. Then, the forecasted results were obtained by applying inverse wavelet transformation.

An autoregressive integrated moving average (ARIMA) model has been proposed by Jakasa et al. in [25]. They used this model to predict one-day ahead electricity prices from the German market. In the experiment, a method similar to Box and Jenkins [13] model has been used. Experiment runs on data from 2000 to 2011, in which 2557 observations were used to train the model and 1279 observations used to test the model.

Artificial Neural Network

As presented in [40], ANN is a well-known approach for short term forecasting, thus we use ANN for forecasting price in the next hour using the input data with specially designed features. The ANN promise to give very accurate results provided that the ANN trained with the correct set of input features and enough input data points.

In [41], the authors presented the ANN model for price prediction using history and other estimated factors in the future to fit and extrapolate the prices and quantities. They implemented three-layer back propagation (BP) network using historical market clearing price, system load and fuel price as input variable.

In [32], the authors present combination model including Probability Neural
Network and Orthogonal Experimental Design for electricity price prediction. He implemented PNN as classifier which has advantage of a fast learning process as it requires a single -pass network training stage for adjusting weight. OED was used to find the optimal smoothing parameter which help to increase prediction accuracy.

**Support Vector Machine (SVM)**

In [39], authors proposed a SVM model for price forecasting using historical price, demand data and Projected Assessment of System Adequacy (PASA) data as input variables. They performed their experiment using Australian National Electricity Market (NEM), New South Wales regional data over the year 2002.

In [23], the authors proposed model for time series prediction using Fish Swarm Algorithm (AFSA) for choosing the parameter of SVM. This model used Optimized support vector machine for electricity price forecasting.

**Fuzzy Inference**

Neural network and Fuzzy inference log based hybrid model was proposed in [6]. In their model, a feed forward network is proposed with three layers, input, hidden and output layer. In the hidden layer, it performs the fuzzification.

In [37], the authors proposed fuzzy inference net (FIN) method for price forecasting. proposed method uses FIN to extract fuzzy rules with FSOM to evaluate the probability of unknown data for the predetermined cluster. This technique is used in groups the users determine freely to evaluate the attribute. FIN has an advantage of not necessarily containing similar group so the grouping can be freely as desired.
SOM

In [21], presents a method for short term price forecasting using self-organized map (SOM). They used two-stage method, SOM network is used in the first stage and a support vector machine which used to fit the output from SOM to each subset in the second stage in a supervised manner. Having removed the anomalies from the training set they obtained mean absolute percentage error (MAPE) as 10.24% in hybrid network approach for the ISO New England market. A neural network based on similar days method has been explored in [35] by Mandal et al. for daily and weekly electricity price forecasting. In the experiment they select similar price days corresponding to forecasted day using Euclidean Norm. In order to forecast the price in a single day they use 45 days prior to forecasting day as well as 45 days prior and 45 days after the same day in last year data as input values to the ANN model. Using their model they obtained MAPE 6.93% for daily forecasting in PJM market data in year 2006.

Pattern Sequence Similarity

In [36], Martínez-Álvarez et. al presented a method which was based on pattern sequence similarity. In this approach, a clustering technique was first used on the data before application of the Pattern Sequence-based Forecasting (PSF) algorithm to produce one step ahead forecasts of the electricity prices. In the experiment, \(k\)-means clustering was used to cluster the training data. The training is performed using data collected from 3 different electricity markets, namely New York (NYISO) [2], Australia (ANEM) [1] and Spain (OMEL) [3], for the years 2004–2005, while testing is carried out for using data from 2006. The authors perform a detailed analysis using these three different datasets, and compared their results to those obtained using other methods. As this work is quite recent and the three datasets used are publicly available, we use the results described in this paper as
benchmarks in order to evaluate our proposed method.

### 3.3 Factors Influencing Electricity Price

An electricity price refers to the clearing price in the deregulated market or the price paid by the consumers per unit of electricity consumed. Electricity price are highly unstable in open market or for consumers, and its instability further increases by deployment of smart grid as it is influenced by many visible and invisible factor. It is unfeasible to engineer all the influencing factor as feature in predicting model but can be analyzed through correlation matrix between price and influencing factor. Factor influencing price may differ based on the prediction time horizon explained in previous section. Short term price depend on current demand, type of energy used for generation, historical price trend, hour of days etc. Medium term and Long term price is influence by factors like energy reserve (oil and gas), expected demand, population growth and various economic factor. Most of the research on price prediction uses this factor as input features for prediction models.

Deployment of smart grid increases the complexity in predicting price with increase in influencing factor. Factor involving human behavior, response for the price from community, dynamicity in price may affect the price in short and long term. Thus predicting model for smart grid should be able to incorporate all this factor along with traditional factors affecting electricity price.

### 3.4 Prerequisite for a STPF System

Most of the existing prediction techniques used by the energy wholesaler or the consumers are for short-term price forecasting (STPF). Each entity intends to have a STPF which has higher reliability and accuracy, so as to maintain smooth functioning of electricity market. Dynamic pricing system make it more important for
consumer to utilize STPF to plan their load consumption. Though there are many prerequisites for a good STPF system, the most important ones are listed below.

- Higher prediction accuracy.
- Low processing time.
- Lower manual tuning or fully automation.
- Effective gathering and handling of data.
- User friendly interface and manual.
- Robustness and reliability.
- Automated noise detection and filtering.
Proposed Prediction Model

4.1 Objective

Predicting electricity price with higher accuracy is one of the important part of electricity market in smart grid. Traditional models forecast the future electricity prices based on the assumption that all members of the society share a generic consumption pattern in a static manner. However, because of this simplistic assumption, they fail to perform effectively especially in the smart grid era where the consumer’s usage patterns can be quite dynamic based on the real-time information available through increased interactions between the supplier and the consumers. A robust model which can maintain their performance irrespective of the type of grid is required for smart grid deployment. This research proposes the prediction model which attains higher accuracy for traditional electricity market and expects to maintain its performance when deployed to smart grid. Resource constrains might be one of the main issue regarding smart grid deployment, however the pro-
posed model provide flexibility in choosing type of algorithm for price prediction. This flexibility enables the user to choose algorithm based on available resource, time constrain and computational complexity.

4.2 Architecture

As mentioned above a new approach has been proposed in this research. We believe that incorporating the modified ensemble learning scenario into the well-known prediction methods will help to improve the performance of our prediction model. With the current research on price prediction, expert machine learning algorithms like Neural Network, Random Forest, and Least square Support Vector Machine (LSSVM) showed promising results. We propose a strategy which enables these algorithms to learn from the environment and update their parameters based on the information they have collected.

In short-term price forecasting, prediction is made by the all the participating expert algorithms but the decision will be made only by the algorithm whose performance was best in previous day. On the first day of the model deployment, the master algorithm will be chosen randomly. From the second day onward, for each hour of the day performance of each algorithm will be evaluated and the best algorithm for that hour will be chosen based on their prediction accuracy. The best predictor will be the master predictor, whose predicted value for next day same hour will be the decisive value. At the end of each day every algorithm will analyze their performance for that day. If the performance is within the range of the threshold they maintain their models else they update their models by adjusting the parameter or update their threshold with certain learning rate. The proposed ensemble models, using fixed weights and varying weights respectively, are described in Algorithms 1 and 2.
Figure 4.1: Architecture of prediction model.

Model proposed for electricity price prediction for smart grid deployment is depicted in Figure 4.1. Presented architecture performs ensemble learning using three different algorithms which showed promising result when analyzed separately. A number of different algorithms can be used under this model depending upon the processing power and time constrain. Proposed model performs feature engineering on the price data collected from the de-regulated electricity market, which is followed by feature selection, learning, predicting and model updating steps. Chap-
CHAPTER 4. PROPOSED PREDICTION MODEL

4.2.1 Fixed Weight Method

Algorithm 1 describes the steps for the Fixed weight Method (FWM). In FWM, at the beginning of the deployment, a weight of 0 is assigned for all the participating algorithms except the randomly chosen master algorithm for the first day. Weight for randomly chosen master algorithm is assigned to 1. Same process is carried out for assigning weight and selecting master algorithm for remaining hours of the first day. In the next step, we build models for each participating algorithm using training dataset. These models are used to predict target value for unseen test data vector. Once we obtain the prediction from each model, expert selector selects prediction from the model with highest weight as the final predicted value. When we have access to actual price, model evaluator will evaluate the performance of each model and update the weight based on prediction accuracy achieved by respective models. Weight updater updates the weight for the model with highest prediction accuracy to 1 and assigns 0 weights to rest of the models. The model with weight equals to 1 for current hour will be master model for next day same hour i.e. prediction from model with weight equals to 1 for current hour will be final prediction value for next day same hour. This process is repeated for all 24 hours of the day and at the end of the day model evaluator will evaluate the performance of each model for that day. If the performance of the model is below threshold value a retrain signal is send to the system, which will initiate the retraining of the model with respective algorithm. Newly build model will replace the old model but will maintain the weight of the previous model. In this method, weight of each model alternate between 0 or 1 based on the prediction accuracy achieved by it. Final prediction for each hour of the day is obtained from the largest weight model (master model) for respective hours. Weight and performance of each models are indepen-
dent to each other, weight of any model depends only on the performance of that model for particular hour of the day.

**Algorithm 1** Fixed Weights

choose set of expert algorithm $S$
choose $M$ such that $M \in S$
set $\text{weight}_M \leftarrow 1$
set $\text{master} \leftarrow \text{null}$
for each day $d$
do for each hour $h$
if $d = 1$ then
predicted $= \text{getPredictedValue}()$
for each algorithm $e$ in $S$
error($e$) $\leftarrow$ predicted($e$) $- \text{actualPrice}$
end for
sort error
$\text{master}(h) \leftarrow \text{algorithm with min(error)}$
choose prediction from $M$ as final predicted value
else
predicted $= \text{getPredictedValue}()$
$\text{finalPrediction} \leftarrow$ predicted($\text{master}(h)$)
for each algorithm $e$ in $S$
error($e$) $\leftarrow$ predicted($e$) $- \text{actualPrice}$
end for
sort error
set $\text{master}(h) \leftarrow \text{algorithm with min(error)}$
end if
end for
end for

### 4.2.2 Varying Weight Method

Algorithm 2 describes the steps for Varying Weight Method (VWM). In VWM on the first day of the deployment, weights for all participating algorithms are set to 1, and randomly one algorithm is chosen to be master algorithm for the first day. In the next step, we build models for each participating algorithm using training dataset. These models are used to predict target value for the unseen test data vector. Once we obtained the prediction value from each model, expert selector se-
lects, prediction from the model with highest weight as final predicted value. When we have access to actual price, model evaluator will evaluate the performance of each model and update the weight based on the function of their prediction accuracy, and learning rate $\lambda$. Algorithm with highest accuracy will have its weight increased and other algorithm with lower accuracy will have their weight decreased based on prediction accuracy achieved by them. Here, $\lambda$ is learning rate or weight adjusting rate, which decide the rate of change of weight. Higher value of $\lambda$ will decrease the weight by larger value and lower value of $\lambda$ decrease weight by lower value. The model with highest weight for current hour will be master model for next day same hour i.e. prediction from model with highest weight for current hour will be final prediction value for next day same hour. This process is repeated for all 24 hours of the day and at the end of the day model evaluator will evaluate the performance of each model for that day. If the performance of the model is below threshold value a retrain signal is send to the system, which will initiate the retraining of the model with respective algorithm. Newly build model will replace the old model but will maintain the weight of the previous model. In this method weights for each model are updated based on prediction accuracy achieved by them. Final prediction for each hour of the day is obtained from the model (master model) for those hours. Weight and performance of each models are independent to each other, it only depend on the performance of that model for particular hour of the day.
Algorithm 2 Varying Weights

choose set of expert algorithm $S$
choose $M$ such that $M \in S$
assign $weight(\forall x \in S) \leftarrow 1$
set value for $\lambda$

for each day $d$ do
  for each hour $h$ do
    if $d = 1$ then
      $predicted = getPredictedValue()$
      for each algorithm $e$ in $S$ do
        $error(e) \leftarrow predicted(e) − actualPrice$
        $weight(e) \leftarrow$ updated weight of $e$ by factor of $\lambda$ and $error(e)$
      end for
      choose prediction from $M$ as final predicted value
    else
      $predicted = getPredictedValue()$
      $finalPrediction \leftarrow prediction(x | x \in S, weight(x) > weight(y) \forall y \in S \setminus \{x\})$
      for each algorithm $e$ in $S$ do
        $error(e) \leftarrow predicted(e) − actualprice$
        if $error(e) < error(y) \forall y \in S \setminus \{e\}$ then
          $weight(e) \leftarrow$ Increase weight of $e$ by factor of $\lambda$ and $error(e)$
        else
          $weight(e) \leftarrow$ Decrease weight of $e$ by factor of $\lambda$ and $error(e)$
        end if
      end for
    end if
  end for
end for
5.1 Data Set

We use our method to forecast electricity prices in three different electricity markets of New York [2], Australia [1], and Spain [3]. We select these market as mentioned in [36], due to the vast research carried out in the area based on these market’s data. From NYISO we selected “Capita” as the reference area to benchmark with previous experiments. The NYISO electricity market is deregulated and the system provides hour- and day-ahead predicted electricity prices. The Australian Electricity Industry deregulated from 1995 and currently Australian National Electricity Market (ANEM) conducts market clearing in every half hour. We select Queensland area as the reference since we benchmark the results with above mentioned research. For the Spanish market we use the same data used by [36] in their experiment.
5.2 Evaluation Metrics

Below mentioned are performance measures used to validate our method. To facilitate direct comparison with results obtained in other similar studies a number of different measures were used.

Mean Error Relative to \( x \) (MER)

\[
MER = 100 \times \frac{1}{N} \sum_{i=1}^{N} \frac{|x_i - \hat{x}_i|}{\bar{x}}
\]  
(5.1)

where \( x_i \) defines the actual price and \( \hat{x}_i \) defines the predicted price. \( \bar{x} \) is the mean price for the period of interest and \( N \) is the number of predicted hours. This indicator is irrespective of the absolute values.

Mean Absolute Error (MAE)

\[
MAE = \frac{1}{N} \sum_{i=1}^{N} |x_i - \hat{x}_i|
\]  
(5.2)

Mean Absolute Percentage Error (MAPE)

\[
MAPE = 100 \times \frac{1}{N} \sum_{i=1}^{N} \frac{|x_i - \hat{x}_i|}{x_i}
\]  
(5.3)

We use this indicator since its irrespective of the absolute values whereas MAE depends on the absolute range of the electricity prices. If the range is vast then it gives a high MAE value but accuracy may be high.

Standard Deviation for MER, MAR and MAPE is defined as:

\[
\sigma = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (e_i - \bar{e})^2} \text{ where } e_i = \frac{\hat{x}_i - x_i}{\bar{x}}
\]  
(5.4)
CHAPTER 5. EXPERIMENTAL SETUP

5.3 Explored Algorithm

Many promising algorithms for price prediction have been proposed in literature as described in chapter 3. Number of algorithms that can be used in proposed model can vary according to the requirement and time constrains. For our research we implemented three algorithms which show good results in our previous research. Algorithms used in our proposed model are described below:

5.3.1 Artificial Neural Networks

Many models of ANN have been proposed for prediction and classification problems in machine learning. History of using ANN for predictive model can be tracked long time back. Many research works applying ANN for STPF have been proposed earlier. ANN was proposed as an algorithm to combine regression and time series analysis. Artificial Neural Network is the part of information technology that is constructed with the architecture of human brain consisting of neurons and information flow. This architecture learns from the experience and generalizes from historical information and provides new output by extracting essential information and characteristics from inputs in the pattern of variable interconnection weights among the processing elements. ANN has shown great advantage over the traditional statistical predictive model, ANNs are data-driven self-adaptive methods, which can capture subtle functional relationship among the data even if the underlying relationships are unknown or hard to describe. Basic structure of ANN consist of 3 layers. The first layer is processing layer or input layer, the second layer is hidden layer, and the third is output layer. There can be many numbers of nodes in input layer which might contain the shared data but are not connected to each other. Input layer forward information via synapses to hidden layer which contain number hidden nodes. ANN can have one or more than one hidden unit in hidden layer based on type of application. The size and number of nodes in each
layer depend on model complexity and data.

Figure 5.1: A simple architecture of neuron.

Figure 5.1 describe typical structure of artificial neurons. Input layer receive data in a form of vector $A = (a_1, a_2, a_3, \cdots, a_n)$. Weight $w_1, w_2, \cdots, w_n$ are bounded to each input connection and a bias $b$ is added to the neuron. The neuron function $f(x)$ can be defined as the composition of these input, weight and bias term.

$$f(x) = K(\sum_{i=1}^{n} (w_i a_j + b_j)),$$

where $K$ is activation function.

Out of all the models, multi-layer perceptron is the best known and most widely used. The network where information flows in only one direction without any loop is feed-forward neural network and network where information flow in both direction and has loop is known as Recurrent or non-feed-forward networks. Figure 5.2 shows the architecture of a generic two-layered feed-forward neural network model used for this research.

To perform prediction using a neural network, two basic steps are required: training, and learning. Assuming availability of historical data, given a task $t$ and class of functions $f$. Learning refer to using set of observation to find $f^* \in F$ which solve the $t$ in some optimal sense. Let us define a cost function $C : F \rightarrow \mathbb{R}$ such that for optimal solution $f^*, C(f^*) \leq C(f), \forall f \in F$, in other words no solution
gives better solution than optimal solution. Cost function is important parameter for learning as it give measure about how far is the particular solution from the optimal solution. In our research we have used mean square error as the cost function. So the learning algorithm tries to minimize the square error in the dataset during learning. Selection of adequate training data and features has high influence over the performance of the neural network. In the learning step neural network construct input-output mapping and update the weight of inputs and biases at the end of each iteration. Backpropagation is the most common learning algorithm, in which at the end of each iteration output error is propagated back to input adjusting the weight and biases. Backpropagation learning algorithm is a gradient descend algorithm which minimize sum of square errors. To overcome the drawback of slower convergence and numerically inefficient of backpropagation algorithm, two parameter learning rate and momentum can be adjusted. In this research we have implemented Levenberg-Marquardt algorithm for training because of its faster learning rate.
5.3.2 Support Vector Machine

Support vector machine (SVM) [45] is a kind of maximum margin classifier which was originally proposed to solve the problem of binary classification. Among a large number of training data vectors, only a few are selected as “support vectors” that define the maximum margin. Only the support vectors are utilized in predicting the classes of the testing data vectors, thus leading to a good generalization.

Later, it was realized that SVM can be adapted to solve the problem of regression [42]. Suykens and Vandewalle proposed a least-squares version of support vector regression (SVR) [44] which is particularly suitable to solve regression problems in time series data. The least-squares SVR tries to find the solution by solving a set of linear equations instead of a convex quadratic programming for classical SVMs.

A brief description of the least-squares SVR is given in the below subsections. This description is adapted and modified from the original ones given in [33, 44].

General SVM Formulation

Suppose we have a training set of n samples \( \{x_i, y_i\} (i = 1, \cdots, n) \) with input data vector \( x_i \in \mathbb{R}^m \) (where m is the dimensionality of \( x_i \)) and corresponding binary class labels \( y_i \in \{-1, +1\} \). In Vapnik’s original formulation [45], the SVM classifier is defined by the conditions:

\[
\begin{align*}
    w.\phi(x) + b &\geq 1, \quad \text{if} \quad y = +1 \\
    w.\phi(x) + b &\leq -1, \quad \text{if} \quad y = -1
\end{align*}
\]  

These formulations can be rewritten as a single condition:

\[
y_i (w.\phi(x_i) + b) \geq 1, \quad i = 1, \cdots, n
\]
where $\phi(x)$ is a nonlinear mapping function of a vector from original space to the high (possibly infinite) dimensional space, $w$ is a weight vector which defines the separation hyperplane, and $b$ is an offset of the separation hyperplane from the origin $(0,0)$. If the given data set is inseparable (i.e., separating hyperplane does not exist), a slack variable $\xi_i$ is introduced in such a way that:

$$
y_i(w, \phi(x_i) + b) \geq 1 - \xi_i, \quad i = 1, \ldots, n
$$

subject to the constraints:

$$
\begin{align*}
\xi_i &\geq 0, & i = 1, \ldots, n \\
\end{align*}
$$

By applying the structural risk minimization principle, the risk bound (i.e., learning error) of the classifier can be minimized by solving the following minimizing problem:

$$
\min J_1(w, \xi) = \frac{1}{2} \|w\|^2 + C \sum_{i=1}^{n} \xi_i
$$

subject to the constraints:

$$
\begin{align*}
y_i[w, \phi(x_i) + b] &\geq 1 - \xi_i, & i = 1, \ldots, n \\
\xi_i &\geq 0, & i = 1, \ldots, n \\
\end{align*}
$$

where $C$ is the slack penalty parameter to control the net effect of the slack variables.

In order to remove the complex constraints of the above minimization problem in 5.3.2, we introduce Lagrangian multipliers $\alpha_i \geq 0 (i = 1, \ldots, n)$. Thus, the minimization problem becomes:

$$
\min L_1(w, \xi, \alpha) = \frac{1}{2} \|w\|^2 + C \sum_{i=1}^{n} \alpha_i (y_i(w, \phi(x) + b) - 1 - \xi_i)
$$

subject to the constraints:

$$
\xi_i \geq 0, \quad i = 1, \ldots, n
$$
The optimal point will in the saddle point of the Lagrangian function. Thus, we have:

\begin{align}
\frac{\partial L}{\partial w} &= 0 \quad \Rightarrow \quad w = \sum_{i=1}^{n} \alpha_i y_i \phi(x_i) \\
\frac{\partial L}{\partial b} &= 0 \quad \Rightarrow \quad \sum_{i=1}^{n} \alpha_i y_i = 0 \\
\frac{\partial L}{\partial \xi_i} &= 0 \quad \Rightarrow \quad 0 \leq \alpha_i \leq C, \quad i = 1, \cdots, n
\end{align}

(5.10)

By substituting \(w\) by its expression, we get the following quadratic programming problem:

\[
\max \mathcal{Q}_1(\alpha) = \sum_{i=1}^{n} \alpha_i \alpha_j y_i y_j K(x_i, x_j)
\]

subject to the constraints:

\[
0 \leq \alpha_i \leq C, \quad i = 1, \cdots, n
\]

Here, \(K(x_i, x_j) = \phi(x_i) \cdot \phi(x_j)\) is called the kernel function (which will be elaborated below). By solving this quadratic programming problem subject to the constraints, we will get the separating hyperplane in the high dimensional space, that is, the classifier in the original space.

**Least Squares SVM Formulation**

Suykens and Vandewalle derived the least squares version of the SVM classifier by reformulating the minimization problem as below [44]:

\[
\min J_2(w, b, e) = \mu \left( \frac{1}{2} \| w \|^2 \right) + \zeta \left( \frac{1}{2} \sum_{i=1}^{n} e_i^2 \right)
\]

subject to the equality constraints:

\[
y_i \left( w \cdot \phi(x_i) + b \right) = 1 - e_i, \quad i = 1, \cdots, n
\]
CHAPTER 5. EXPERIMENTAL SETUP

The least-squares SVM classifier formulation above implicitly corresponds to a regression interpretation with binary targets $y_i = \pm 1$. Both $\mu$ and $\zeta$ are parameters to tune the amount of regularization versus the sum squared error. The solution does only depend on the ratio $\gamma = \mu / \zeta$, therefore the original formulation uses only $\gamma$ as tuning parameter. Therefore, we have:

$$
\min J_2(w, b, e) = \frac{1}{2} \| w \|^2 + \gamma \frac{1}{2} \sum_{i=1}^{n} e_i^2 
$$

(5.13)

The solution of the least-squares regressor is obtained after the Lagrangian function is constructed as follows:

$$
L_2(w, b, e, \alpha) = J_2(w, b, e) - \sum_{i=1}^{n} (\alpha_i (y_i (w \cdot \phi(x_i) + b) - 1 + e_i))
$$

$$
= \frac{1}{2} \| w \|^2 + \gamma \frac{1}{2} \sum_{i=1}^{n} e_i^2 - \sum_{i=1}^{n} \alpha_i (y_i (w \cdot \phi(x_i) + b) - 1 + e_i) 
$$

(5.14)

where $\alpha_i \in \mathbb{R} (i = 1, \ldots, n)$ are the Lagrange multipliers. Again, the conditions for optimality are:

$$
\begin{align*}
\frac{\partial L_2}{\partial w} = 0 & \Rightarrow w = \sum_{i=1}^{n} \alpha_i y_i \phi(x_i) \\
\frac{\partial L_2}{\partial b} = 0 & \Rightarrow \sum_{i=1}^{n} \alpha_i y_i = 0 \\
\frac{\partial L_2}{\partial e_i} = 0 & \Rightarrow \alpha_i = \gamma e_i, \quad i = 1, \ldots, n \\
\frac{\partial L_2}{\partial \alpha_i} = 0 & \Rightarrow y_i (w \cdot \phi(x_i) + b) - 1 + e_i = 0, \quad i = 1, \ldots, n
\end{align*}
$$

(5.15)

By the elimination of $w$ and $e$, we will have a linear programming problem instead
of a quadratic programming one:

\[
\begin{bmatrix}
0 & y^T \\
y & \Omega + y^{-1}I_n
\end{bmatrix}
\begin{bmatrix}
b \\
\alpha
\end{bmatrix}
= \begin{bmatrix}
0 \\
1_n
\end{bmatrix}
\]  

(5.16)

where \(y = [y_1, \ldots, y_n], 1_n = [1, \ldots, 1], \alpha = [\alpha_1, \ldots, \alpha_n], \) and \(I_n\) is an \(n \times n\) identity matrix. Here \(\Omega \in \mathbb{R}^{n \times n}\) is the kernel matrix whose individual element \(\Omega_{i,j}(i, j = 1, \ldots, n)\) is defined as follows [16, 33].

\[
\Omega_{i,j} = \phi(x_i) \cdot \phi(x_j) = K(x_i, x_j)
\]  

(5.17)

Thus, the regression function is:

\[
y = w \cdot \phi(x) + b = \sum_{i=1}^{n} \alpha_i K(x_i, x_j) + b
\]  

(5.18)

**Kernel Function**

For the kernel function \(K(., .)\) one typically has the following choices [33]: Linear kernel:

\[
K(x_i, x_j) = x_i \cdot x_j
\]  

(5.19)

Polynomial kernel of degree \(p\):

\[
K(x_i, x_j) = (\frac{x_i \cdot x_j}{c})^p
\]  

(5.20)

Radial basis function (RBF) kernel:

\[
K(x_i, x_j) = \exp(-\frac{\|x_i - x_j\|^2}{\sigma^2})
\]  

(5.21)

Where \(p, c, \) and \(\sigma\) are constants. Here, Mercer’s condition holds for all \(c, \sigma \in \mathbb{R}^+\) and \(p \in \mathbb{N}\) values in the polynomial and RBF cases. The scale parameters
and $\sigma$ determine the scaling of the inputs in the polynomial and RBF kernel functions. This scaling affects the kernel’s bandwidth, which is an important factor in generalization of a kernel method [33].

### 5.3.3 Random Forests

Random forests were proposed by Leo Breiman for designing the ensemble based predictor with a set of decision tree that grows in randomly selected subspace of features. In this approach each tree in ensemble is created by randomly selecting the features to split at each node and and these features on training set are used to calculate best split. The tree is grown using CART methodology to maximize size, without pruning [12]. Let us assume that, training data contain $N$ instances and $M$ set of variable or feature. We are given $m$, where $m$ is the set of variable $m \subset M$. Learning set is created by choosing $n$ instance from $N$ with replacement, and rest of the instances are used to calculate error of the model. At each node, randomly select $m$ variable and make split decision based on these variable. At the end fully grown unpruned tree is constructed.

#### Error Generalization in Random Forests

In this section we provide brief summary about the generalization of error in random forest as derived by L. Breiman [15]. A random forest is the ensemble of classifier $h(x, \theta_k)$ where $k \in 1, \cdots, n$ and $h \in h_1, \cdots, h_m$. For each input $x$ random forest $h_m(x, \theta_k)$ will generate an output of $y_k$. Here $h_m$ represent the individual random forest created from independent samples $\theta_k$ drawn from the distribution. This implies that random forest randomly chooses $n$ features for each tree from same training data and features are chosen independent to each other [11]. Each tree cast vote for the most probable class for any input $X$. For a general classifier $h_k(x)$ in the ensemble of classifier $h_1, \cdots, h_k$, Bresman defined as the margin function [15].
CHAPTER 5. EXPERIMENTAL SETUP

\[ mg(X, Y) = av_k I(h_k(X) == Y) - \max_{j \neq x} av_k I(h_k(X) = j) \]  \quad (5.22)

where we represent the function and \( av_k \) is an average. This margin measure the extend of the difference between average number of vote at \( X,Y \) for correct class and average number of vote for any other class. Confidence in classifying any class is directly proportional to margin value, i.e. the larger the margin the higher confidence in classification and vice-versa [15]. The generalization error is given by:

\[ PE^* = P_{X,Y} (mg(X, Y) < 0) \]  \quad (5.23)

which is the probability over the space of input vector \( X \) and Label \( Y \).

For calculation of strength and correlation between the random forest Breiman defined margin function for random forest as,

\[ mr(X, Y) = P_{\Theta} (h(X, \Theta) == Y) - \max_{j \neq x} P_{\Theta} (h(X, \Theta) = j) \]  \quad (5.24)

where \( \Theta \) is the random sampling, this equation closely resemble the equation for generalization error. Thus strength for set of classifier is defined as

\[ s = E_{X,Y} mr(X, Y) \]  \quad (5.25)

where \( E \) is the mathematical expectation. The mean correlation \( \rho \) between the classifiers is derived as a function of \( E \), the standard deviation and the random vector \( \theta_k \). Breiman [15] obtains the upper bound on the generalization error as:

\[ PE^* \leq \rho \left( 1 - \frac{s^2}{s^2} \right) \]  \quad (5.26)
The equation above gives us a clear indication that the generalization error is influenced by two factors, the correlation between the random trees and their individual strengths. Further, the mean correlation is related to the independence of $\theta_k$ and thus, the generalization error is directly proportional to the randomization and the independence of $\theta_k$. Breiman [15] further derives that as the number of random trees becomes large (tends to infinity), the generalization error converges to a limit.

**Characteristics of Random Forest**

As described by Breisman [15], few characteristic and advantages of using random forests are:

- Better visualization for high dimension data.
- Better resistance to outliers and errors.
- Better technique to deal with missing values.
- Automatic detection of valuable predictors.
- Faster than other traditional method like Adaboost, bagging.
- As trees are independent to each other they can be easily parallelized.
- It gives useful internal estimates of error, strength, correlation and variable importance.

### 5.4 Experimental Process Flow

Feature creation and selection is the first step in classification or regression (i.e., forecasting in our context). It is a widely used process in machine learning and involves either the creation of new features or the selection of an optimal subset from a pool of existing features. The selected subset will contain key features
which contribute to the accuracy of the forecasts and also help reduce over-fitting of the model. 5.3.

Figure 5.3: Overview of the proposed price forecasting process flow.
CHAPTER 5. EXPERIMENTAL SETUP

5.4.1 Feature Engineering

The electricity market data follows a time series form (time, value) pair and does not provide any specific features to use with predicting models. Thus we have to create features from the available past data to be used as inputs to the predicting models. We analyzed the input data using a similar approach to the method in [8] and create about 20 features from the available electricity market data. Hourly data is extracted for 24-hour windows, yielding 24 features with which we seek to forecast the electricity price for the following hour. The best features that give short term trend in the price market are past 24 hour data which has been verified in [9]. But this data does not capture the seasonal behaviors and long term trends.

In price forecasting, it is important to take into account both short and long term trends and also seasonal patterns. Sudden changes in the price might be caused by seasonal behaviors and other factors. In order to capture this behavior we create putatively relevant features based on historical data which lasts for longer period. We create features such as last year same day same hour data, last year same day same hour price fluctuation, last week same day same hour price, last week same hour price fluctuation etc.

Though we create many features from historical data, we cannot use all generated features since training accuracy depends on the number of features. To achieve higher forecasting accuracy, over-fitting and over training of models should be avoided. In order to select the best features we use feature selection techniques as described in the next section.

5.4.2 Feature Selection

Feature selection is a very important step towards building robust forecasting model. In this paper we have implemented wrapper method [27] using WEKA for subset selection from a large pool of features. Wrappers implements search algorithm
for finding the subset of features in feature space and evaluate the subset using
the model or learning algorithm. Wrapper evaluates the estimated accuracy ob-
tained from learning algorithm by adding or removing features from the features
subset. Estimation of accuracy is done using cross validation, in this research we
have implemented 10 fold cross validation on the training set and ANN was used as
learning algorithm. Wrappers methods are most widely used in supervised learning
problem where labels are available. It can also be used for unsupervised learning
problem where labels are not available by setting some target or objective function
to optimize, which result in best cluster.

After creating around 20 features that capture long term trends and past 24 hour
data as 24 features we perform feature selection to find the best set of features. Due
to large pool of features we divide the feature set into two as 24 hourly features in
one pool and rest of the features in another pool. As described in [9], past 24 hourly
price features capture the current short term trend and selecting only a subset of
these features will diminish the accuracy of the model. Thus, we used this verified
approach and select all the 24 features to be used in our prediction model.

To select the best features from rest of the features we have implemented wrap-
per method which used for subset selection from a large pool of features. Wrap-
ners implements search algorithm for finding the subset of features in feature space
and evaluate the subset using the model or learning algorithm. Wrapper evaluates
the estimated accuracy obtained from learning algorithm by adding or removing
features from the features subset. Estimation of accuracy is done using cross vali-
dation, in this paper we have implemented 10 fold cross validation on the training
set.

In selecting the best feature set, the wrapper method was applied to all features
except those associated with the past 24 hours. Technically, this training accuracy
may be less than the best possible accuracy since we did not incorporate the past
CHAPTER 5. EXPERIMENTAL SETUP

<table>
<thead>
<tr>
<th>Attribute ID</th>
<th>Feature Description</th>
<th>Formula</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 to 24</td>
<td>Previous 24 Hrs electricity price</td>
<td>( p_{t-1} ) to ( p_{t-24} )</td>
</tr>
<tr>
<td>25</td>
<td>Previous year same hours price</td>
<td>( p_{y-1} )</td>
</tr>
<tr>
<td>26</td>
<td>Previous year same day average price</td>
<td>( p_{\text{avg}(y-1)} )</td>
</tr>
<tr>
<td>27</td>
<td>Previous week same hour price</td>
<td>( p_{d-7} )</td>
</tr>
<tr>
<td>28</td>
<td>Previous hour price increase/decrease</td>
<td>(</td>
</tr>
<tr>
<td>29</td>
<td>Previous year same day, same hour, price increase/decrease</td>
<td>(</td>
</tr>
<tr>
<td>30</td>
<td>Temperature</td>
<td>( T_d )</td>
</tr>
<tr>
<td>31</td>
<td>Day of the week</td>
<td>( W_d )</td>
</tr>
<tr>
<td>32</td>
<td>Holiday</td>
<td>( H_d )</td>
</tr>
</tbody>
</table>

Table 5.1: Features used for training and testing of New York (NYISO) dataset.

24 hour data. Once we apply all the data, including the past 24 hour data, to the wrapper, it take a much longer time since the wrapper method is computationally very expensive. Due to the verified importance of the past 24 hour data, we choose to drop it from our feature selection using wrapper and select the best features from the remaining ones with the wrapper using 10 fold cross validation. As an example, the final feature set obtained for New York (NYISO) dataset after feature selection process is shown is Figure 5.1.

5.4.3 Incorporating Other Features

Features that generated using historical prices gave good forecasting results, but it is possible to improve the accuracy by considering more features that are not associated directly with price data. Thus, we consider other parameters that affect the load or price in the market and found that according to [5], the temperature, day of the week and the occurrence of holidays can all affect the electricity load and pricing. Therefore, we incorporate these features into generated feature set.

In order to forecast the next hour price we need the temperature in that particular hour which is not available in real time. Thus we use predicted temperature that is provided by a weather forecasting service. For the training and testing purpose
historical data was used but when the model use in real time we can use a service such as wounderground.com to get the forecasted temperature value. For the holiday data we use predefined holidays in the tested region and directly bind them with existing data.

We train and test all models without temperature and holiday features together with other selected features, and then we incorporate temperature and holiday features into feature set and train and test the models. It was seen that the use of temperature and calendar-based features improved the accuracy of the system; as such, these features were used in the final model. Some may argue that oil and gas prices and many other factors also affects the pricing. However, it is not easy to have a correctly predicted oil and gas prices to be incorporated into our model. Thus, we decide not use such not-easy-to-predict features in our model because using them may aggravate the error in the final price forecasting.

5.4.4 Normalization

Normalization is one of the best approach to deal, where the input data is different from each other and ranging in different measurements and scales. In our approach we defined various input data that has different scales, thus we need to normalize the data to achieve a consistent form. We use mapminmax function available in Matlab to normalize our input data. Mapminmax returns a normalized matrix by normalizing each row to the range of provided min and max values. In our method we normalize the data into the range (-1.0, 1.0).

5.5 Constructing Learning Model

Our experiment was performed using three different algorithm. All the process for experiment were arranged according to the figure 5.3. For learning process, 3
layer ANN was constructed using 15 nodes in hidden layer. Random Forest algorithm train the model using 100 trees by selecting 15 variable at each node. Value for sigma and gamma for constructing SVM model was obtained by performance iterative analysis using training data. Our model as described above utilize each algorithm based on their previous performance, where parameters for each algorithm can be changed in each run for improving the performance. This model was used to perform research on training and test set for three different electricity market. Each algorithm runs independently and prediction result obtained from them will be used to evaluate the performance of the algorithm and adjust weight. At the end of the day, prediction model is updated by changing parameters for the participating algorithms. Result obtained and performance evaluation of proposed model is explained in Chapter 7.
CHAPTER 6

Results and Discussions

We use our method to forecast electricity prices in three different electricity markets as New York (NYISO) [2], Australia (ANEM) [1], and Spain (OMEL) [3] markets. We select these market as mentioned in [36], due to the vast research carried out in the area based on these data markets. From NYISO we selected “Capita” as the reference area to benchmark with previous experiments. The NYISO electricity market is deregulated and the system provides hour- and day-ahead forecasted electricity prices. The Australian Electricity Industry was deregulated in 1995 and currently known as Australian National Electricity Market (ANEM) conducts market clearing in every half hour. We select Queensland area as the reference since we benchmark the results with above mentioned research. For the Spanish market we use the same data that used by [36] in their experiment.

Two different set of experiment were performed using two different algorithms for updating the weight of participating algorithm. First experiment was performed with Fixed Weight algorithm and its result is discussed in first section and result
obtained from *Varying Weight* algorithm is discussed in next section.

### 6.1 Experiment I: With Fixed Weight Algorithm

This section discuss the result from the first experiment using *Fixed Weight* algorithm.

#### 6.1.1 Experiment IA: 2004–2006

For all the NYISO, ANEM, and OMEL datasets, For 2004–2006 dataset we used data of March 2004–March 2006 as training set and April 2006–December 2006 as testing set. This experiment was performed by providing fixed weight to entire algorithms. At each iteration weight of algorithm is set to 1 if it has the least prediction error and else 0.

**NYISO Dataset**

Table 6.1 shows mean error rate (MER), mean absolute error (MAE), and mean absolute percentage error (MAPE) obtained for all datasets used for validation of the model for year 2006. In spite of less training data in Table 6.1, we can see that the result obtained for the New York data has MER of 3.92% with the S.D. (standard deviation) of 1.03, MAE of 2.25 with S.D. of 0.53 and MAPE of 3.97% with S.D. of 0.75. The worst month was December where MER is 5.61%, MAE is $3.02/MWHr$ and MAPE is 5.46%.

Figure 6.1 depicts the worst and Figure 6.2 depicts the best forecasted days in 2006 testing dataset. It is visible that in NYISO market, our model forecasts the prices with higher accuracy. In this experiment we found results from neural network were dominating other algorithm, more than 60% of the prediction from NN was selected as final prediction.
Table 6.1: MAE, MER and MAPE in NYISO markets for year 2006 (Experiment IA).

<table>
<thead>
<tr>
<th></th>
<th>NYISO (2006)</th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ANEM Dataset</td>
<td>MER (%)</td>
<td>Our Method</td>
<td>MAE (%)</td>
<td>PSF (%)</td>
</tr>
<tr>
<td>jan</td>
<td>4.45</td>
<td>2.25</td>
<td>3.51</td>
<td>3.82</td>
<td></td>
</tr>
<tr>
<td>feb</td>
<td>5.53</td>
<td>3.02</td>
<td>4.63</td>
<td>3.07</td>
<td></td>
</tr>
<tr>
<td>mar</td>
<td>6.30</td>
<td>3.97</td>
<td>4.30</td>
<td>3.75</td>
<td></td>
</tr>
<tr>
<td>apr</td>
<td>3.79</td>
<td>2.22</td>
<td>3.51</td>
<td>3.82</td>
<td></td>
</tr>
<tr>
<td>may</td>
<td>2.79</td>
<td>1.64</td>
<td>4.63</td>
<td>3.07</td>
<td></td>
</tr>
<tr>
<td>jun</td>
<td>3.47</td>
<td>2.03</td>
<td>2.31</td>
<td>3.64</td>
<td></td>
</tr>
<tr>
<td>jul</td>
<td>3.89</td>
<td>2.28</td>
<td>2.28</td>
<td>3.75</td>
<td></td>
</tr>
<tr>
<td>aug</td>
<td>4.58</td>
<td>2.68</td>
<td>3.49</td>
<td>3.76</td>
<td></td>
</tr>
<tr>
<td>sep</td>
<td>2.72</td>
<td>1.59</td>
<td>4.49</td>
<td>3.42</td>
<td></td>
</tr>
<tr>
<td>oct</td>
<td>3.19</td>
<td>1.87</td>
<td>4.23</td>
<td>3.89</td>
<td></td>
</tr>
<tr>
<td>nov</td>
<td>5.25</td>
<td>2.90</td>
<td>3.53</td>
<td>4.91</td>
<td></td>
</tr>
<tr>
<td>dec</td>
<td>5.61</td>
<td>3.02</td>
<td>3.08</td>
<td>5.46</td>
<td></td>
</tr>
<tr>
<td>Mean</td>
<td>3.92</td>
<td>2.25</td>
<td>3.40</td>
<td>3.97</td>
<td></td>
</tr>
<tr>
<td>S.D. (%)</td>
<td>1.03</td>
<td>0.53</td>
<td>0.84</td>
<td>0.75</td>
<td></td>
</tr>
</tbody>
</table>

ANEM Dataset

The Australian market data is highly volatile with a large number of unexpected abnormalities and outliers. Australian market has high fluctuation in the electricity price with highest price of $9739/MWhr in the month of January 2006 and lowest value of $7.81/MWhr in the 1st day of February 2004. Due to this high fluctuating data and outliers, forecasted price for this market has high range of error. From Table 6.2, we can see that performance of our approach for Australian market has
9.41% of MER with S.D. of 3.92 and MAE of $3.14/MWh with S.D. of 1.31 and MAPE of 8.17% with S.D. 2.90. Though the result of our approach has higher error percentage, still its performance is better than the other researches performed on this dataset. MER of our approach for this market is slightly better than the previous research, whereas there is greater accuracy achievement in the MAE.

The worst performance was in the months of January, July and August with worst performance of 17.97% MER in July and the best performance of 4.59% MER was obtained in the month of March. The best MAPE of 5.40% was obtained for the month of October and the worst MAPE of 13.84% in the month of July.
Figure 6.2: The best forecasted day in NYISO market in year 2006.

Figure 6.3 depicts the worst and Figure 6.4 depicts the best forecasted days in 2006 testing dataset. It is visible that in ANEM market our model forecasts the prices with higher accuracy. Though this accuracy is less than that of NYISO market data, results revealed that our forecasting accuracy is higher than that of its competitor method, namely PSF [36]. In this experiment final prediction follow the same trend as in NYISO data where most of the prediction were based on NN.
Table 6.2: MAE, MER and MAPE in ANEM markets for year 2006 (Experiment IA).

<table>
<thead>
<tr>
<th></th>
<th>ANEM (2006)</th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>MER</td>
<td>PSF (%)</td>
<td>Our Method (%)</td>
<td>MAE</td>
<td>PSF (%)</td>
</tr>
<tr>
<td>jan</td>
<td>13.96</td>
<td>5.8</td>
<td>4.66</td>
<td>1.51</td>
<td>13.34</td>
</tr>
<tr>
<td>feb</td>
<td>7.26</td>
<td>8.59</td>
<td>2.42</td>
<td>5.15</td>
<td>7.19</td>
</tr>
<tr>
<td>mar</td>
<td>4.59</td>
<td>7.84</td>
<td>1.53</td>
<td>1.73</td>
<td>4.79</td>
</tr>
<tr>
<td>apr</td>
<td>5.60</td>
<td>9.92</td>
<td>1.87</td>
<td>1.98</td>
<td>5.82</td>
</tr>
<tr>
<td>may</td>
<td>10.07</td>
<td>12.85</td>
<td>3.36</td>
<td>3.21</td>
<td>9.59</td>
</tr>
<tr>
<td>jun</td>
<td>11.00</td>
<td>22.04</td>
<td>3.67</td>
<td>6.81</td>
<td>7.85</td>
</tr>
<tr>
<td>jul</td>
<td>17.97</td>
<td>17.11</td>
<td>5.99</td>
<td>8.16</td>
<td>13.84</td>
</tr>
<tr>
<td>aug</td>
<td>11.28</td>
<td>11.71</td>
<td>3.76</td>
<td>3.32</td>
<td>7.55</td>
</tr>
<tr>
<td>sep</td>
<td>8.77</td>
<td>8.23</td>
<td>2.93</td>
<td>2.34</td>
<td>7.03</td>
</tr>
<tr>
<td>oct</td>
<td>5.70</td>
<td>7.66</td>
<td>1.90</td>
<td>1.92</td>
<td>5.40</td>
</tr>
<tr>
<td>nov</td>
<td>10.59</td>
<td>6.76</td>
<td>3.53</td>
<td>2.09</td>
<td>9.25</td>
</tr>
<tr>
<td>dec</td>
<td>6.12</td>
<td>6.42</td>
<td>2.04</td>
<td>1.41</td>
<td>6.35</td>
</tr>
<tr>
<td>Mean</td>
<td>9.41</td>
<td>10.41</td>
<td>3.14</td>
<td>3.30</td>
<td>8.17</td>
</tr>
<tr>
<td>S.D (σ)</td>
<td>3.92</td>
<td>4.87</td>
<td>1.31</td>
<td>2.23</td>
<td>2.90</td>
</tr>
</tbody>
</table>

OMEL Dataset

For Spanish electricity market we collected data from 2004-2006. Data from 2004 and 2005 was used for training and data for 2006 was used to test the resulting model. The results obtained from the Spanish market are shown in Table 6.3, we can see that the average MER for year 2006 is 5.34% with S.D. of 0.64, which shows that the monthly error are around the average error. MAE for the Spanish
MAE for Spanish data is very low compared to other market because of its relatively very low average energy prices compared to others. The best and the worst forecasted days for year 2006 is depicted in Figures 6.5 and 6.6 respectively.

Comparison with PSF

To validate our method, we compared our results with those obtained in other studies found in the literature. We select Pattern Sequence-based Forecasting (PSF) [36] method as our benchmark for analysis since in their experiment, it proves that...
results are above aforementioned other concurrent works. Thus without comparing with other techniques, we perform a comparison with PSF to validate our results. In the PSF model they compare their results with ARIMA model, naïve Bayes, ANN, WNN (Weighted Nearest Neighbor) [34], STR [17] and other mixed models. As PSF paper performs testing on year 2006 data from all three markets we perform the testing on same data and results depicted in Tables 6.1, 6.3, and 6.2. As showed in the tables, we can argue that our method with FWM, forecasts with 1.61% MER improved accuracy in NYISO market for year 2006. It has a slight improvement of 1% MER over the ANEM market and 0.81% MER improvement.
Table 6.3: MAE, MER and MAPE in OMEL markets for year 2006 (Experiment IA).

<table>
<thead>
<tr>
<th></th>
<th>Our Method (%)</th>
<th>PSF(%)</th>
<th>Our Method</th>
<th>PSF</th>
<th>Our Method (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>jan</td>
<td>6.00</td>
<td>7.26</td>
<td>0.40</td>
<td>0.53</td>
<td>5.070545</td>
</tr>
<tr>
<td>feb</td>
<td>5.25</td>
<td>4.93</td>
<td>0.35</td>
<td>0.36</td>
<td>3.661152</td>
</tr>
<tr>
<td>mar</td>
<td>5.28</td>
<td>5.88</td>
<td>0.35</td>
<td>0.43</td>
<td>6.188545</td>
</tr>
<tr>
<td>apr</td>
<td>5.90</td>
<td>3.62</td>
<td>0.39</td>
<td>0.28</td>
<td>5.73097</td>
</tr>
<tr>
<td>may</td>
<td>6.00</td>
<td>8.11</td>
<td>0.40</td>
<td>0.64</td>
<td>6.108788</td>
</tr>
<tr>
<td>jun</td>
<td>5.14</td>
<td>3.76</td>
<td>0.34</td>
<td>0.29</td>
<td>5.270606</td>
</tr>
<tr>
<td>jul</td>
<td>4.59</td>
<td>4.3</td>
<td>0.30</td>
<td>0.33</td>
<td>4.549394</td>
</tr>
<tr>
<td>aug</td>
<td>5.17</td>
<td>5.37</td>
<td>0.34</td>
<td>0.42</td>
<td>5.56503</td>
</tr>
<tr>
<td>sep</td>
<td>5.66</td>
<td>6.41</td>
<td>0.37</td>
<td>0.50</td>
<td>5.418727</td>
</tr>
<tr>
<td>oct</td>
<td>6.08</td>
<td>7.89</td>
<td>0.40</td>
<td>0.58</td>
<td>6.64497</td>
</tr>
<tr>
<td>nov</td>
<td>3.94</td>
<td>8.3</td>
<td>0.26</td>
<td>0.64</td>
<td>7.245879</td>
</tr>
<tr>
<td>dec</td>
<td>5.04</td>
<td>8.02</td>
<td>0.33</td>
<td>0.59</td>
<td>7.609515</td>
</tr>
</tbody>
</table>

| Mean  | 5.34           | 6.15   | 0.35       | 0.47| 5.76           |
| S.D (σ)| 0.64           | 1.76   | 0.04       | 0.13| 1.11           |

in OMEL market for year 2006 data.

For numerical analysis, MAE measure also proved high accuracy with $1.15, $0.16, $0.12 per MWHr improvements for NYISO, ANEM and OMEL markets respectively. With our model, it provides better forecasting as indicated by MER factor, and S.D. in MER for NYISO is also less with compared to PSF model as it indicates 1.03% with our model and 1.29% with PSF model. For MAE value the S.D. is a lesser than PSF model and our model also provides better forecasting
Figure 6.5: The best forecasted days in NYISO market in year 2006.

having a less MAE value compared to PSF model. For ANEM market which has significant abnormalities in the dataset, our model forecasts with MAE S.D. 1.31 whereas in PSF model MAE S.D. is 2.23. This indicates a higher forecasting accuracy since our MAE value is less than that for PSF model. For MER in ANEM market, our model has a less S.D. 3.92 with compared to PSF which is 4.87. Forecasting for OMEL market also provides better accuracy as indicated by the relevant S.D. values indicated in Tables 6.1, 6.3, and 6.2.

For NYISO market we test our model from April 2006 to December 2006 due to unavailability of data in first 3 months in year 2004 to train with our model. But
we can argue that our model provides a less MER with very less S.D. affirming that for this months also we can get better results.

6.1.2 Experiment IB: 2008–2012

To further verify our methods’ good performance, we run a second experiment using the data which is more recent than those used in Experiment I. For this second experiment, from NYISO and ANEM datasets, we took June 2008–May 2011 data as training set and June 2011–May 2012 data as testing set. (Note: OMEL dataset is not available for the period 2008–2012.) It is observed that the results of this
Experiment II are even slightly better than those in Experiment I.

**NYISO Dataset**

From Table 6.4, we can see that the performance of our model is as good as with the data from 2006 though the 2011–2012 data contains many spikes and outliers. For 2011–2012 data, we obtain the MER of 3.99% with S.D. of 1.45, MAE of $1.48/MWhr with S.D. 0.54, and MAPE 3.86% with S.D. of 0.57. We can see that there is a slight improvement in the forecasting result for the latest data with a reduction in MAE by $0.9/MWhr. The worst forecasting obtained was 5.82% of MER in January, $2.68/MWhr of MAE in July and 4.87% of MAPE in January. The main cause of higher forecasting error in this month was due to higher no of spikes and outliers in New York market data. Table 6.4 also indicates the MAPE values for the 2011–2012 test dataset which gives the performance irrespective of the market numerical values.

Figure 6.7 depicts the worst and Figure 6.8 depicts the best forecasted day in 2006 testing dataset. It is visible that in NYISO market our model forecasts the prices with higher accuracy.

**ANEM Dataset**

In case of 2011–2012 dataset, the worst performance was in the months of January-March with 12.72% MER in the month of February and best performance of 3.83% MER in the month of May. The best and the worst MAPEs were 3.72% in May and 14.36% in February respectively. There is a slight increase in S.D. which indicate that the errors in the years 2011–2012 are much more deviated from mean error. This is because of improvement in the forecasting accuracy for few months in 2011–2012 dataset. For this latest data our approach shows better performance with the increased accuracy of more than 3% MER, $0.89/MWhr MAE and more
Table 6.4: MAPE, MAE, and MER performance indicators for NYISO and ANEM markets for years 2011–2012 (Experiment IB).

<table>
<thead>
<tr>
<th></th>
<th>Prediction for 2011/12</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>NYISO</td>
</tr>
<tr>
<td></td>
<td>MER (%)</td>
</tr>
<tr>
<td>jun</td>
<td>5.32</td>
</tr>
<tr>
<td>jul</td>
<td>7.25</td>
</tr>
<tr>
<td>aug</td>
<td>3.99</td>
</tr>
<tr>
<td>sep</td>
<td>3.73</td>
</tr>
<tr>
<td>oct</td>
<td>3.56</td>
</tr>
<tr>
<td>nov</td>
<td>3.47</td>
</tr>
<tr>
<td>dec</td>
<td>3.68</td>
</tr>
<tr>
<td>jan</td>
<td>5.82</td>
</tr>
<tr>
<td>feb</td>
<td>2.83</td>
</tr>
<tr>
<td>mar</td>
<td>3.48</td>
</tr>
<tr>
<td>apr</td>
<td>2.56</td>
</tr>
<tr>
<td>may</td>
<td>2.25</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Mean</th>
<th>S.D (σ)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>3.99</td>
<td>1.45</td>
</tr>
<tr>
<td></td>
<td>1.48</td>
<td>0.54</td>
</tr>
<tr>
<td></td>
<td>3.86</td>
<td>0.57</td>
</tr>
<tr>
<td></td>
<td>6.22</td>
<td>3.01</td>
</tr>
<tr>
<td></td>
<td>2.09</td>
<td>1.37</td>
</tr>
<tr>
<td></td>
<td>7.16</td>
<td>3.98</td>
</tr>
</tbody>
</table>

than 1% of MAPE, as shown in Table 6.4. The best and the worst forecasted days for the years 2011–2012 are depicted in Figures 6.10 and 6.9 respectively.

6.2 Experiment II: With Varying Weight Algorithm

This section discuss the result obtained from second experiment using Varying Weight algorithm. For all the NYISO, ANEM, and OMEL datasets, For 2004–2006 dataset we used data of March 2004–March 2006 as training set and April 2006–
December 2006 as testing set and for 2008–2012 we used data of June 2008–May 2011 data as training set and June 2011–May 2012 data as testing set. This experiment was performed by providing varying weight algorithm. In first iteration, all algorithms are assigned weight 1, and in each next iteration weight is adjusted based on the accuracy obtained by individual algorithm.

6.2.1 Experiment IIA: 2004–2006

The result obtained in this experiment with varying weight algorithm does not differ greatly with the result obtained with fixed weight algorithm (experiment I),
Figure 6.8: The best forecasted day in NYISO market in year 2012.

though we can see slight improvement in evaluation metrics. We can see from Figures 6.5, 6.6, and 6.7, that is an improvement of 0.06% of MER, 0.05 MAE and 0.05% of MAPE for NYISO dataset. Similarly there was an improvement 0.17%, 0.03, 0.24% of MER, MAE, MAPE respectively for ANEM and 0.08%, 0.0, 0.14% of of MER, MAE, MAPE respectively for ANEM for OMEL dataset.

6.2.2 Experiment IIB: 2008–20012

Similar to previous experiment, result obtained in this experiment does not vary greatly with the result obtained from ExperimentIB.
This experiment also produced slight improvement in evaluation metrics with improvement of 0.05%, 0.0, 0.01% of MER, MAE, MAPE respectively for NYISO and improvement of 0.06%, 0.04, 0.18% MER,MAE,MAPE respectively for ANEM dataset. Figure 6.8 illustrate the result obtained from the experiment.
Figure 6.10: The best forecasted day in ANEM market in year 2012.
Table 6.5: MAE, MER and MAPE in NYISO markets for year 2006 (Experiment IIA).

<table>
<thead>
<tr>
<th></th>
<th>NYISO (2006)</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>MER</td>
<td>MAE</td>
<td>MAPE</td>
</tr>
<tr>
<td></td>
<td>Our Method (%)</td>
<td>PSF (%)</td>
<td>Our Method</td>
</tr>
<tr>
<td>jan</td>
<td>4.45</td>
<td>2.25</td>
<td></td>
</tr>
<tr>
<td>feb</td>
<td>5.53</td>
<td>3.02</td>
<td></td>
</tr>
<tr>
<td>mar</td>
<td>6.30</td>
<td>3.97</td>
<td></td>
</tr>
<tr>
<td>apr</td>
<td>3.79</td>
<td>4.94</td>
<td>2.22</td>
</tr>
<tr>
<td>may</td>
<td>2.79</td>
<td>7.59</td>
<td>1.64</td>
</tr>
<tr>
<td>jun</td>
<td>3.47</td>
<td>3.34</td>
<td>2.03</td>
</tr>
<tr>
<td>jul</td>
<td>3.89</td>
<td>3.93</td>
<td>2.28</td>
</tr>
<tr>
<td>aug</td>
<td>4.58</td>
<td>5.37</td>
<td>2.68</td>
</tr>
<tr>
<td>sep</td>
<td>2.72</td>
<td>6.24</td>
<td>1.59</td>
</tr>
<tr>
<td>oct</td>
<td>3.19</td>
<td>7.43</td>
<td>1.87</td>
</tr>
<tr>
<td>nov</td>
<td>5.25</td>
<td>5.19</td>
<td>2.90</td>
</tr>
<tr>
<td>dec</td>
<td>5.61</td>
<td>6.04</td>
<td>3.02</td>
</tr>
<tr>
<td>Mean</td>
<td>3.92</td>
<td>5.53</td>
<td>2.25</td>
</tr>
<tr>
<td>S.D (σ)</td>
<td>1.03</td>
<td>1.29</td>
<td>0.53</td>
</tr>
</tbody>
</table>
Table 6.6: MAE, MER and MAPE in ANEM markets for year 2006 (Experiment IIA).

<table>
<thead>
<tr>
<th></th>
<th>ANEM (2006)</th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>MER</td>
<td></td>
<td>MAE</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Our Method (%)</td>
<td>PSF (%)</td>
<td>Our Method</td>
<td>PSF (%)</td>
<td>Our Method (%)</td>
</tr>
<tr>
<td></td>
<td>jan</td>
<td>14.03</td>
<td>5.8</td>
<td>4.61</td>
<td>1.51</td>
</tr>
<tr>
<td></td>
<td>feb</td>
<td>7.04</td>
<td>8.59</td>
<td>2.37</td>
<td>5.15</td>
</tr>
<tr>
<td></td>
<td>mar</td>
<td>4.44</td>
<td>7.84</td>
<td>1.47</td>
<td>1.73</td>
</tr>
<tr>
<td></td>
<td>apr</td>
<td>5.53</td>
<td>9.92</td>
<td>1.78</td>
<td>1.98</td>
</tr>
<tr>
<td></td>
<td>may</td>
<td>10.09</td>
<td>12.85</td>
<td>3.22</td>
<td>3.21</td>
</tr>
<tr>
<td></td>
<td>jun</td>
<td>11.20</td>
<td>22.04</td>
<td>3.72</td>
<td>6.81</td>
</tr>
<tr>
<td></td>
<td>jul</td>
<td>17.14</td>
<td>17.11</td>
<td>6.02</td>
<td>8.16</td>
</tr>
<tr>
<td></td>
<td>aug</td>
<td>11.12</td>
<td>11.71</td>
<td>3.81</td>
<td>3.32</td>
</tr>
<tr>
<td></td>
<td>sep</td>
<td>8.36</td>
<td>8.23</td>
<td>2.83</td>
<td>2.34</td>
</tr>
<tr>
<td></td>
<td>oct</td>
<td>5.59</td>
<td>7.66</td>
<td>1.89</td>
<td>1.92</td>
</tr>
<tr>
<td></td>
<td>nov</td>
<td>10.41</td>
<td>6.76</td>
<td>3.54</td>
<td>2.09</td>
</tr>
<tr>
<td></td>
<td>dec</td>
<td>5.94</td>
<td>6.42</td>
<td>2.00</td>
<td>1.41</td>
</tr>
<tr>
<td></td>
<td>Mean</td>
<td>9.24</td>
<td>10.41</td>
<td>3.11</td>
<td>3.30</td>
</tr>
<tr>
<td></td>
<td>S.D (σ)</td>
<td>3.83</td>
<td>4.87</td>
<td>1.34</td>
<td>2.23</td>
</tr>
</tbody>
</table>
Table 6.7: MAE, MER and MAPE in OMEL markets for year 2006 (Experiment IIA).

<table>
<thead>
<tr>
<th></th>
<th>OMEL(2006)</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>MER</td>
<td>MAE</td>
<td>MAPE</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Our Method (%)</td>
<td>PSF(%)</td>
<td>Our Method</td>
<td>PSF</td>
</tr>
<tr>
<td>jan</td>
<td>5.91</td>
<td>7.26</td>
<td>0.40</td>
<td>0.53</td>
</tr>
<tr>
<td>feb</td>
<td>5.20</td>
<td>4.93</td>
<td>0.34</td>
<td>0.36</td>
</tr>
<tr>
<td>mar</td>
<td>5.05</td>
<td>5.88</td>
<td>0.34</td>
<td>0.43</td>
</tr>
<tr>
<td>apr</td>
<td>5.68</td>
<td>3.62</td>
<td>0.39</td>
<td>0.28</td>
</tr>
<tr>
<td>may</td>
<td>5.88</td>
<td>8.11</td>
<td>0.40</td>
<td>0.64</td>
</tr>
<tr>
<td>jun</td>
<td>5.12</td>
<td>3.76</td>
<td>0.33</td>
<td>0.29</td>
</tr>
<tr>
<td>jul</td>
<td>4.68</td>
<td>4.3</td>
<td>0.29</td>
<td>0.33</td>
</tr>
<tr>
<td>aug</td>
<td>5.10</td>
<td>5.37</td>
<td>0.33</td>
<td>0.42</td>
</tr>
<tr>
<td>sep</td>
<td>5.61</td>
<td>6.41</td>
<td>0.36</td>
<td>0.50</td>
</tr>
<tr>
<td>oct</td>
<td>6.04</td>
<td>7.89</td>
<td>0.41</td>
<td>0.58</td>
</tr>
<tr>
<td>nov</td>
<td>3.83</td>
<td>8.3</td>
<td>0.25</td>
<td>0.64</td>
</tr>
<tr>
<td>dec</td>
<td>5.06</td>
<td>8.02</td>
<td>0.33</td>
<td>0.59</td>
</tr>
<tr>
<td>Mean</td>
<td>5.26</td>
<td>6.15</td>
<td>0.35</td>
<td>0.47</td>
</tr>
<tr>
<td>S.D (σ)</td>
<td>0.62</td>
<td>1.76</td>
<td>0.05</td>
<td>0.13</td>
</tr>
</tbody>
</table>
Table 6.8: MAPE, MAE, and MER performance indicators for NYISO and ANEM markets for years 2011–2012 (Experiment IIB).

<table>
<thead>
<tr>
<th></th>
<th>NYISO</th>
<th></th>
<th>ANEM</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>MER (%)</td>
<td>MAE</td>
<td>MAPE (%)</td>
<td>MER (%)</td>
</tr>
<tr>
<td>jun</td>
<td>5.19</td>
<td>1.92</td>
<td>4.25</td>
<td>4.42</td>
</tr>
<tr>
<td>jul</td>
<td>7.12</td>
<td>2.71</td>
<td>4.40</td>
<td>6.59</td>
</tr>
<tr>
<td>aug</td>
<td>3.87</td>
<td>1.50</td>
<td>3.39</td>
<td>4.07</td>
</tr>
<tr>
<td>sep</td>
<td>3.72</td>
<td>1.38</td>
<td>3.56</td>
<td>4.02</td>
</tr>
<tr>
<td>oct</td>
<td>3.60</td>
<td>1.33</td>
<td>3.49</td>
<td>4.08</td>
</tr>
<tr>
<td>nov</td>
<td>3.41</td>
<td>1.31</td>
<td>3.57</td>
<td>5.52</td>
</tr>
<tr>
<td>dec</td>
<td>3.68</td>
<td>1.35</td>
<td>3.73</td>
<td>4.38</td>
</tr>
<tr>
<td>jan</td>
<td>5.71</td>
<td>2.19</td>
<td>4.82</td>
<td>8.58</td>
</tr>
<tr>
<td>feb</td>
<td>2.84</td>
<td>1.06</td>
<td>3.22</td>
<td>12.61</td>
</tr>
<tr>
<td>mar</td>
<td>3.48</td>
<td>1.27</td>
<td>4.74</td>
<td>10.96</td>
</tr>
<tr>
<td>apr</td>
<td>2.44</td>
<td>0.96</td>
<td>3.84</td>
<td>4.93</td>
</tr>
<tr>
<td>may</td>
<td>2.18</td>
<td>0.84</td>
<td>3.22</td>
<td>3.76</td>
</tr>
<tr>
<td>Mean</td>
<td>3.94</td>
<td>1.48</td>
<td>3.85</td>
<td>6.16</td>
</tr>
<tr>
<td>S.D (σ)</td>
<td>1.42</td>
<td>0.54</td>
<td>0.56</td>
<td>2.98</td>
</tr>
</tbody>
</table>
6.3 Discussions

Two different experiments were performed with different approaches for updating the weight of the participating algorithms. From the result we can see that, performance of varying weight algorithm was slightly better than the fixed weight algorithm. This was because, in the varying weight algorithm, weights were adjusted based on the prediction accuracy i.e the change in weight is higher if the difference in real and predicted value is higher. Whereas in the fixed weight algorithm, changes in weight do not depend upon the level of accuracy of the algorithm, it just looks at which algorithm performs best. In, varying weight, algorithm is evaluated based on all the previous errors made by the algorithm which is directly correlated with final prediction accuracy of the model. Both approach shows high improvement in accuracy compared with our bench-mark method. Comparing the result obtained, varying weight shows improvement of 0.06–0.18% of MER, 0.03–0.04 of MAE, and 0.05%–0.24% of MAPE compared to result obtained from fixed weight model. Analyzing the results, we found the performance of Neural Network to be comparatively better than LSSVM and Random Forest when the price has higher fluctuation, whereas performances of LSSVM and Random Forests were better when the price has lower fluctuation. ANN shows consistent performance with both models whereas LSSVM fails to perform well in varying weight model as the prediction error for LSSVM was very high during the peak price, which decreased the weight of LSSVM by large margin. Random forest also shows consistence performance but its performance also degrade in some cases of peak price. Due to the higher and sudden variation in the electricity price which is influenced by many unforeseen factor causes degradation in performance of model. We can see that the performance for few months is below the average performance of the model due to many sharp peaks on that month. We can see better performance for the month when the prices are stable.
Electricity price forecasting in deregulated electricity market is essential to ease the decision making process. Though extensive research has been carried out in this field, significant improvements are yet to be achieved. Results will be more degraded in smart-grid deployment, due to increase in complexity with increase in influencing factor. We proposed ensembles based model, using three different algorithm participating for electricity price forecasting. We proposed two different approaches to update the weight of participating algorithm, which decide the master algorithm, whose prediction will be final prediction of the model. In the experiment we performed comparative analysis with recent, highly regarded study found in the literature and it is visible that our model has better forecasting accuracy as compared to models in literature. We performed analysis both on 2006 and 2012 data and it proves that our model can forecast robustly and accurately even in the various times. Both fixed weight approach and varying weight approach showed better forecasting accuracy as compared to models in literature, and among them
varying weight approach was better.

For the future work we want to apply our model on other electricity markets. Also we want to consider applying features such as gas prices, method of electricity generation, electricity demand and supply, etc. Furthermore, we want to incorporate different feature which can model the dynamicity associated with smart grid, so as to maintain accuracy for smart grid deployment. In addition, we plan to conduct further research on better weight adjusting algorithm which might increase the accuracy.
APPENDIX A

Abbreviations

STPF  Short Term Price Forecasting
MTPF  Medium Term Price Forecasting
STPF  Long Term Price Forecasting
ARIMA  Auto-regressive Integrated Moving Average
ANN  Artificial Neural Network
LS-SVM  Least Square- Support Vector Machine
SOM  Self-Organized Map
FWM  Fixed Weight Method
VWM  Varying weight Method
NYISO  New York Independent System Operator
APPENDIX A. ABBREVIATIONS

ANEM  Australian National Electricity Market

MER   Mean Relative Error

MAPE  Mean Absolute Percentage error

MAE   Mean Absolute Error

RF    Random Forest

SD    Standard Deviation

PSF   Pattern Sequence-based Forecasting

RF    Random Forest
Bibliography


BIBLIOGRAPHY


[30] D. Li, Z. Aung, S. Sampalli, J. Williams, and A. Sanchez. Privacy preser-


