UNIVERSITY OF CALCUTTA


# A Framework Towards Generalized Mid-term Energy Forecasting Model for Industrial Sector in Smart Grid

by

Sourasekhar Banerjee

Roll Number: 97/CSM/160014

Registration Number : A01-1112-0801-11

under guidance of

Prof. Nabendu Chaki

A thesis submitted in partial fulfillment for the degree of Master of Technology

in

Computer Science and Engineering

Department of Computer Science and Engineering

June 2018


## Department of Computer Science and Engineering

## University College of Science, Technology and Agriculture

## University of Calcutta

## Certificate

This is to certify that the project report entitled "A Framework Towards Generalized Mid-term Energy Forecasting Model for Industrial Sector in Smart Grid" submitted for partial fulfillment of the requirements for the degree of Master of Technology in Computer Science and Engineering from Department of Computer Science and Engineering, University of Calcutta, has been carried out by Sourasekhar Banerjee (Roll No.: 97/CSM/ 160014, Registration No.: A01-1112-0801-11) under the supervision of Prof. Nabendu Chaki, Professor, Department of Computer Science and Engineering, University of Calcutta, India.

| | |
|---|---|
| Prof. Nabendu Chaki<br>Professor,<br>Department of Computer Science and Engineering<br>University of Calcutta. | Dr.Banani Saha<br>Chairperson Board of Studies<br>Department of Computer Science and Engineering<br>University of Calcutta. |

External Examiner

# *Acknowledgements*

One of the joys of completion is to look over the journey past and remember all who have helped and suported me along this long but fulfilling road. I have received invaluable help and support from various individuals from institute as well as personal. It is the most opportune moment to convey my heartfelt gratitude to all of them.

First of all I would like to express my immense gratitude for my Thesis supervisor Prof.Nabendu Chaki, Professor, Department of Computer Science and Engineering, University of Calcutta for his guidance and encouragement all throughout in making this work possible. Without a single bit of exaggeration, I can say that it was him whose constant support, right from the beginning and till the end, provided me academic inspiration and psychological boost to complete the thesis successfully. Without his disciplined and rigorous support,this dissertation wouldn't have seen the light of the day.

I am indebted to Miss Manali Chakraborty , Senior Research Fellow, Department of Computer Science and Engineering, University of Calcutta. From the begin of this thesis she helped me a lot to understand the primary objective of this work. She was always by my side and helped me a lot to make it successful.

This acknowledgement remain incomplete without the mention of the support provided by the faculty and office staff of the Department of Computer Science and Engineering, University of Calcutta.

I am especially thankful to my friends specially Sounak Banerjee, Pranjal Sett, Satyam Raha, Gouri Kundu, Debanjan Nandan, Bishnu Prasad Bera, Chiranjit Sarkar. They have helped me a lot to overcome many obstacle.

Words are not enough to express my gratitude to my family. I sincerely thank to my parents Mr.Chandra Sekhar Banerjee and Mrs. Anuradha Banerjee for always being there with their faith on me. I want to give special thanks to my elder sister Dr.Somdutta Banerjee, Brother in law Dr.Montu Bose and maternal Uncle Prof. Ambar Ghosal.

Though all my teachers, friends, relations and mentors have extended their help and support at every stage, I am alone responsible for all the remaining errors.

---

Sourasekhar Banerjee

*"Prophesy is a good line of business, but it is full of risks. "*

Mark Twain in Following the Equator

UNIVERSITY OF CALCUTTA

# *Abstract*

Department of Computer Science and Engineering

Master of Technology

by Sourasekhar Banerjee
Roll Number: 97/CSM/160014
Registration Number : A01-1112-0801-11

One of the many improvements that Smart Grid offers over traditional power grid is a balanced supply-demand ratio. Now, as electricity is hard to store for future usage, it is important to be aware of the demand in order to generate enough electricity for uninterrupted power supply. Thus, forecasting plays a vital role in Smart Grid. However, with various range of rapidly fluctuating influential parameters towards electricity consumption patterns, it is next to impossible to design a single forecasting model for different types of users. Typically, electricity usage depends on the demographic, socio-economic and climatic environment of any region. Besides, the dependencies between influential parameters and consumption vary over different sectors, like, residential, commercial and industrial. In this paper, our main goal is to develop a generalized mid-term forecasting model for the industrial sector, that can accurately predict quarterly energy usage of a large geographic region with the diverse range of influential parameters. The proposed model is designed and tested on real-life datasets of industrial users of various states in the U.S.

# Contents

# List of Figures

# List of Tables

# Abbreviations

| | |
|---|---|
| **ARFF** | **A**ttribute-**R**elation **F**ile **F**ormat |
| **ANN** | **A**rtificial **N**eural **N**etwork |
| **ANOVA** | **An**alysis **O**f **Va**riance |
| **ARIMA** | **A**uto**r**egressive **I**ntegrated **M**oving **A**verage |
| **ARMA** | **A**uto**r**egressive **M**oving **A**verage |
| **BEA** | **B**ureau of **E**conomic **A**nalysis |
| **BLS** | **B**ureau of **L**abor **S**tatistics |
| **CC** | **C**orrelation **C**oefficient |
| **CSV** | **C**omma **S**eparated **V**alues |
| **EIA** | U.S. **E**nergy **I**nformation **A**dministration |
| **GDP** | **G**ross **D**omestic **P**roduct |
| **IDA** | **I**ndex **D**ecomposition **A**nalysis |
| **MAE** | **M**ean **A**bsolute **E**rror |
| **MAPE** | **M**ean **A**bsolute **P**ercentage **E**rror |
| **LMS** | **L**east **M**edian of **S**quare |
| **LTLF** | **L**ong-**T**erm **L**oad **F**orecasting |
| **MLP** | **M**ulti-**L**ayer **P**erceptron |
| **MTLF** | **M**id-**T**erm **L**oad **F**orecasting |
| **NCDC** | **N**ational **C**limatic **D**ata **C**enter |
| **NOAA** | **N**ational **O**ceanic and **A**tmospheric **A**dministration |
| **QP** | **Q**uadratic **P**rogramming |
| **RAE** | **R**elative **A**bsolute **E**rror |
| **RMSE** | **R**oot **M**ean **S**quare **E**rror |
| **RRSE** | **R**elative **R**oot **S**quare **E**rror |
| **SMO** | **S**equential **M**inimal **O**ptimization |

| | |
|---|---|
| **STLF** | **S**hort-**T**erm **L**oad **F**orecasting |
| **SVM** | **S**upport **V**ector **M**achine |
| **WEKA** | **W**aikato **E**nvironment for **K**nowledge **A**nalysis |

*For my Parents ...*

# Chapter 1

# Introduction

Technological progressions in recent years lead us to a more advanced, comfortable and machine dependent society, where the majority of these machines are fueled by electricity. As a result, the demand for electricity increases rapidly. In order to meet this demand, several distributed renewable energy sources are incorporated in the supply chain of electricity. Simultaneously, the electricity usage pattern also fluctuates throughout the day based on several driving factors, such as temperature, working hours of commercial, industrial and educational establishments, festivals etc [1]. Now, balancing the supply-demand ratio in the power grid, considering all these odds, is not only an easy job but requires additional technological support. Smart Grid is considered as an upgradation of the existing power grid, that incorporates a parallel data communication network along with the electricity network. This combined system can offer a two-way data exchange facility, where utilities can share supply related information with users, whereas, users can also share their demands, usage history, as well as additional data generation information (if, they produce electricity using some renewable power sources, such as solar, bio-fuel etc.) with utilities.

Forecasting is a data-driven procedure that helps utilities in planning, investment, and decision-making purposes. Many utilities already use forecasting to address their current challenges, but forecasting will increase in importance because of the growing complexity of challenges and the availability of more data inputs from a data-rich smart grid environment.

In this project, I am going to deliver a generalized model for forecasting industrial energy demand on Smart Grid. This work is an extension of the work of [2]. In [2], they have already proposed a generalized mid-term load forecasting model for residential and commercial sectors in Smart Grid. Their model can be applied to various geographical regions, having different socio-economic, demographic and weather parameters. The

model has a coefficient database, that stores the coefficients for different variations of predictor variables. The proposed model was tested on various states of the U.S and even on some other regions in India ( Himachal Pradesh). Most of the times their model delivers best results compared to existing works. However, when they tried to apply their model directly to the industrial sector, it was not performing as expected. Upon analyzing the results further, I have found that the energy demand in the industrial sector is guided by a different set of parameters than the other two sectors. Besides, same influential parameters have different effects on these sectors. As an example, weather variations and festivities do not affect the energy usage of industrial sector as much as residential sector.

U.S. Energy Information Administration (EIA) categorized the industrial sector in three distinct industry types [12]:

- *energy intensive industries*, generally consist of iron and steel manufacturing, petroleum refineries, nonferrous metals and nonmetallic minerals industries.
- *energy nonintensive industries* including various pharmaceuticals, electronic and electrical gadget manufacturing industries.
- *nonmanufacturing industries*, such as, agricultural, forestry, fishing, construction etc.

The energy demand in the industrial sector varies across regions and countries, depending on the level and mix of these above types of industries. Hence, I have selected industrial energy usage as the primary driving factor, that can be used to distinguish different regions for the proposed model.

The purpose of this model is to predict quarterly energy demand of any large geographic region e.g., a state of a particular country. At first, we need to identify proper predictor variables that govern the electricity usage in the industrial sector. We have used six different data mining techniques and identified the most suitable one according to the results. Then we have tried to find a multiplier constant which can be multiplied by the forecast value to introduce accuracy in prediction. Here We have categorized different geographical regions, based on their average electricity usage in the industrial sector and for each region we have tried to find a range of multiplier.

In chapter 2 we have discussed the motivation for this work and a brief literature review has given. The proposed model has defined in chapter 3 and in chapter 4 I have concluded this work.

# Chapter 2

# State of Art

Energy forecasting is about estimating future consumptions based on various data and information available and as per consumer behavior. Forecasting models can be categorized into 3 groups depending on the time period of prediction [3] namely:

- Short Term Load Forecasting (STLF): It refers to forecasting models that are used to predict the demand over hourly or day by day data.
- Mid-Term Load Forecasting (MTLF): It refers to the forecasting model that predicts demand over one week or several month or quarter.
- Long-Term Load Forecasting (LTLF): This type of model predicts the demand for several years.

In this project, we mainly focused on Mid-Term Load Forecasting. It is generally used to find peak load within a span of several weeks or months. It can also be used to analyze the impact of different factors on electrical demand profile. These factors are viz. weather profile, demographic profile, number of energy-intensive and nonintensive industrial establishments etc.[4]. Compared to other two forecasting models, MTLF provides better insights which help utilities to cope with any sudden changes in energy demand [2].

Over decades several forecasting techniques have been evolved to predict electrical load. Most of them are either from time series literature [5] or soft computing framework [6]. Different soft computing based technique viz. Artificial neural network based method (ANN), Fuzzy logic based, Genetic algorithm based method has taken good attention in the field of forecasting. These methods provide high accuracy in there results even though there is nonlinearity in data. But the problem is most of the time these techniques

act as a black box. It is quite difficult to find out the relationship between explanatory variable and response variable.

On the other hand, regression-based techniques provide both accuracy and interpretability of dependent and independent variables. In [7, 8] the authors have used additive regression technique to build their model. Performance of regression models depends on the quality of data sets and computing capability of machines [8, 9]. In this work, our main objective is to provide a generalized model based on some explanatory variable. These independent variables must have some impact on the prediction. That is why we have chosen regression based Data mining technique.

In[2] the author has chosen an additive regression as their proposed model. Here their task was to provide a generalized model. They have produced a coefficient database to produce higher accuracy. Their model was able to cover residential and Commercial sector properly but was failed to provide the prediction for the industrial sector. In this manuscript, we have tried to propose a model which is able to provide average energy usage in the industrial sector.

Index decomposition analysis (IDA) [10, 11, 13] has also been used to track economy-wide and sectoral energy efficiency trends. The main purpose of this application is to identify the drivers of energy use for the energy consuming sectors. Suppose in industrial sector value added often taken as a driver. IDA analysis can be single level or multilevel [13] In [14] has proposed a fuzzy regression technique to efficiently forecast long-term energy consumption in the industrial sector of Iran. They have introduced four independent variables such as energy price, energy intensity, gross domestic production, and employment as inputs to the model. The prediction accuracy is better than traditional ANOVA and ARIMA model. A short-term load foresting model based on wavelet decomposition and the random forest has proposed in[15]. The traditional data mining algorithms have some drawbacks, viz. They can fall into local optimum or they have poor generalization capability. Wavelet decomposition algorithm is a valid method to extract the load of the different components as the training set and random Forest regression algorithm suffers less from the problem of overfitting and determining the difficulty of model parameters.

The industrial sector represents almost 54% [12] of global energy consumption, which makes it the largest energy consuming end-user sector in the economy. Besides, studies show that the industrial energy usage will increase by as much as 40% by next 15 years [12], especially in developing countries. Since the industrial sector is generally dominated by heavy machinery and mostly nonshiftable loads, it can have the major contribution in high peak demands, which in turn can challenge the generation process of utilities. In order to avoid these situations, it is necessary to acquire prior knowledge regarding

the future energy demand of this sector. There exist some good works which propose various methods to forecast future energy usage in industrial sectors, as well as, other sectors, such as, residential and commercial. However, most of these methods were developed over a small geographical region. The underlying algorithms and procedures were trained and tested on the dataset of that particular region. As a result, these methods can be effective as long as we apply them to regions which have same socio-economic, demographic characteristics as the region, based on which the model has been developed. Otherwise, every time we have to build a new model from scratch depending on the characteristics of our application region.

In order to solve this problem, we have proposed a generalized forecasting model for the industrial sector in Smart Grid. The proposed model can forecast accurately the industrial energy usage of different regions using only one underlying model and an associate multiplier database. We have tested our model on various states across the U.S.

# Chapter 3

# Proposed Methodology

The construction of the proposed model generally consists of two primary phases:

1. **Select underlying data mining algorithm:** using **training** and **testing** of various data mining algorithms and comparing the results.
2. **Construct multiplier database:** this phase helps us to generalize the proposed model so that we can apply it on a large geographical region.

Figure 3.1 depicts the basic block diagrams for the two phases separately.

In order to justify the correctness of our multiplier values, we have used Random Forest (as a base learner) to predict the multipliers and compared the obtained values with our calculated multipliers and found that they actually belong to the same range.

## 3.1 Select underlying data mining algorithm

The following series of steps will describe briefly the methodology of selecting proper data mining techniques for predicting industrial electricity usage.

- 1. Selection of the predictor variables for industrial sector.
- 2. Data collection
- 3. Data Preprocessing.
- 4. Experimental Setup and results
- 5. Verify the result using the data of 2015.
- 6. Predict the quarterly demand of energy for year of 2016 and 2017.

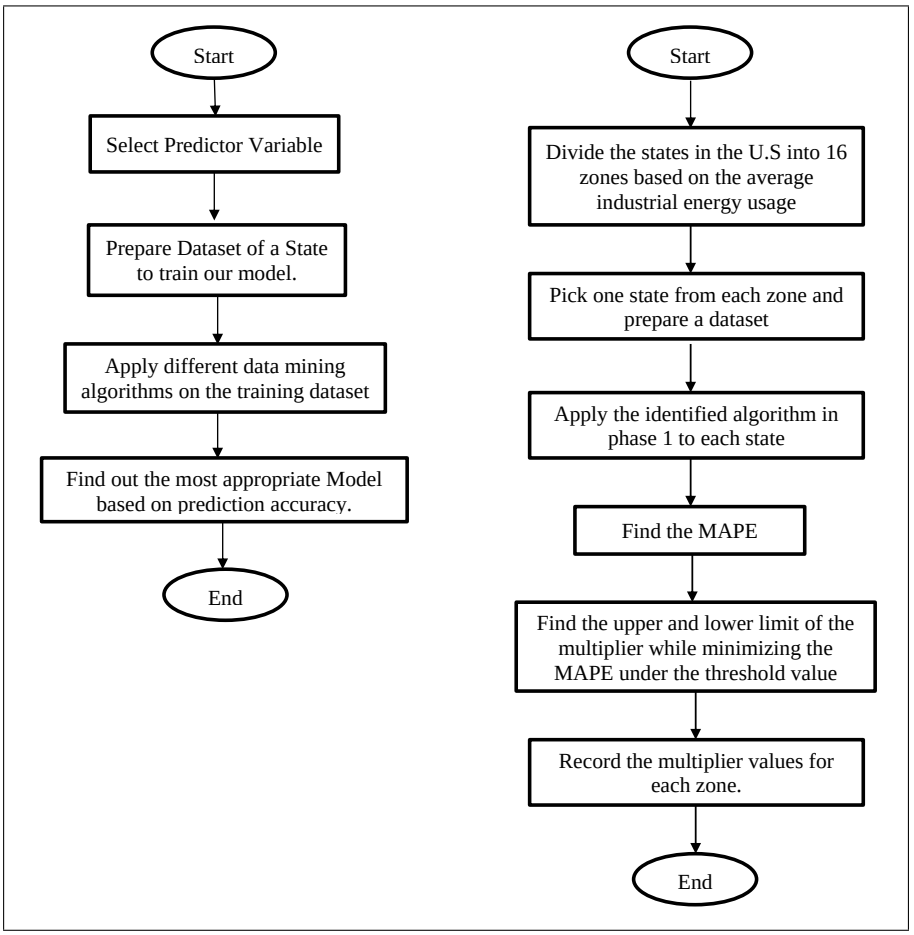


FIGURE 3.1: Block diagram for proposed forecasting model.

### 3.1.1 Selection of the predictor variable for industrial sector.

We have selected 10 industry related influential parameters as follows:

1. **Previous quarterly energy usage(Thousand megawatts) history**, the average of 10 years' usage data for each quarter.
2. **Number of Energy Intensive Industry**, Here we mainly focused on petroleum refinery, heavy metal, and manufacturing industries[18].
3. **Number of Energy Non-intensive Industry**, consists of mainly wood, plastic, rubber, computers and electronic equipment manufacturing etc [18].
4. **GDP growth rate** for each quarter of each state.
5. **Population density** of each state for each quarter.
6. **Average Maximum Temperature** of each state.
7. **Average Minimum Temperature** of each state.

8. **Average Mean Temperature** of each state.

9. **Industrial load share**, determines how much percentage of electrical energy is used for industrial sector amongst all sector.

10. **Retail sale of energy(cents per kilowatt-hour)** for each quarter to industries of each state.

### 3.1.2 Data Collection.

The next important task is to develop datasets on which various data mining techniques can be applied. All dataset has been prepared on the basis of different states of the U.S. We have collected 15 year long quarterly data. We have applied different data mining techniques on these datasets. Data for parameter 1, 9 and 10 has been collected from US Energy Information Administration (EIA)[19]. The establishment of energy intensive and energy nonintensive industry has been collected from Bureau of Labor Statistics (BLS)[20]. The GDP data has been collected from Bureau of Economic Analysis (BEA), U.S. Department of Commerce [21]. The population data has been collected from [22]. Parameter 6, 7 and 8 has been collected from national centers for environment Information(NOAA) [23]. We have prepared 9 decade long datasets, i.e.: 2000-2009, 2001-2010, 2002-2011, 2003-2012, 2004-2013, 2005-2014, 2006-2015, 2007-2016 for different states of USA. We have discussed more datasets in appendix H.

### 3.1.3 Data Preprocessing.

It is a common requirement for many machine learning estimators to standardize datasets. Here we scale features to lie between a range of 0 to 1. In this paper, the data representing each predictor variable $x_i$ for i = 1, 2, . . . , n are normalized using the following equation 3.1.

$$x_i\_norm = \frac{(x_i - x_i\_min)}{(x_i\_max - x_i\_min)} \tag{3.1}$$

where, $x_i\_max$ and $x_i\_min$ are the maximum and minimum values for the vector $x_i$, and $x_i\_norm$ is the resultened normalized value. n is the total number of predicted variables.

### 3.1.4 Experimental Setup and Results.

All experiments have been performed on a tool called WEKA: Waikato Environment for Knowledge Analysis [16]. It is a suite of machine learning software written in Java to perform data mining tasks. I have described details of WEKA in Appendix I.

#### 3.1.4.1 Selection of appropriate data mining algorithms :

I have selected 6 regression based data mining approach [1] to identify which approach gives the better result than other methods. These algorithms are :

1. Additive Regression
2. Least Median Square Regression
3. Random Forest
4. M5P tree
5. MLP Regressor
6. SMO

I have taken a decade-long training dataset (2006-2015) of North Carolina. Each year has been divided into quarters.6 different regression techniques have been applied to the training dataset to produce models. We have compared the performances of these models based on 7 metrics, [17] such as Correlation Coefficient(CC), Mean Absolute Error (MAE). Root Mean Square Error(RMSE), Relative Absolute Error(RAE), Relative Root Square Error(RRSE), Mean Absolute Percentage Error(MAPE) and for model testing we have used R2 value. The results of our experiments are given in table 3.1. Appendix G describes details about these metrics.

TABLE 3.1: Performance of Different Regression Methods

| **Regression Technique** | **Correlation Coefficient** | **MAE** | **RMSE** | **RAE (%)** | **RRSE (%)** | **MAPE** | **R2 Value** |
|---|---|---|---|---|---|---|---|
| Additive Regression | 0.9693 | 34.25 | 45.498 | 23.40 | 24.94 | 4.131 | 0.94 |
| LeastMedSq | 0.8944 | 50.661 | 84.239 | 34.62 | 46.17 | 3.741 | 0.87 |
| Random Forest | 0.9914 | 26.195 | 33.506 | 19.90 | 18.36 | 3.349 | 0.96 |
| M5P Tree | 0.9718 | 34.45 | 43.21 | 23.54 | 23.68 | 3.686 | 0.85 |
| MLP Regressor | 0.9719 | 34.91 | 43.05 | 23.86 | 23.60 | 3.739 | 0.95 |
| SMO | 0.9238 | 44.57 | 71.9 | 30.46 | 39.13 | 3.516 | 0.80 |

[1] Appendix A to F describes more on these algorithms

After analyzing all these data and taking consideration of every comparison metrics I have found that Random Forest gives better accuracy in forecasting amongst regression techniques.

### 3.1.5 Verification of the proposed model.

here I have forecast the quarterly usage of energy in the industrial sector for the year 2015 using these regression models and compared prediction with the actual energy usage. The results are given in figure 3.2.

Analyzing these results, I can conclude that though every model has the similar pattern with the actual value, the performance of Random Forest is quite better than the others.

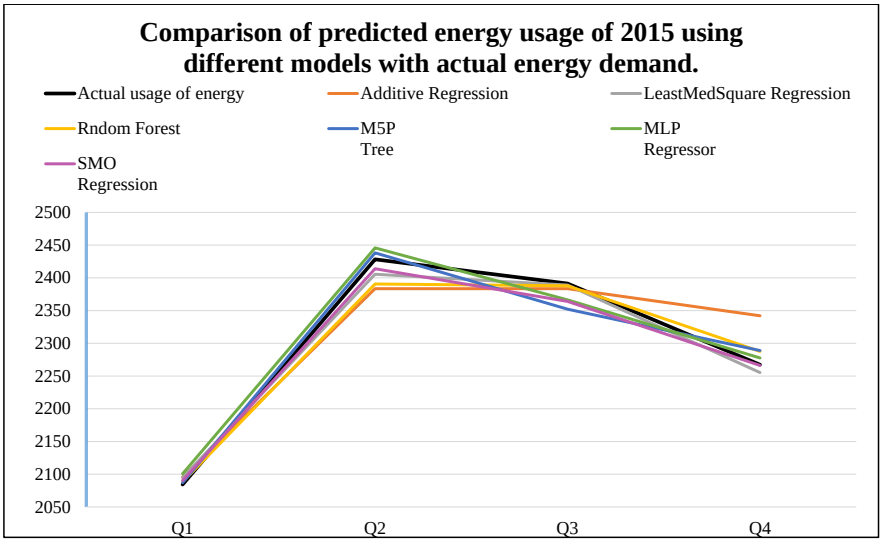


FIGURE 3.2: Energy demand prediction for North Carolina - 2015.

### 3.1.6 Predict quarterly usages of energy for the year 2016 and 2017.

In this section, I have tried to predict the quarterly energy usage in the industrial sector of North Carolina for the year 2016 and 2017 using the model based on Random Forest. I have compared our results with the actual energy usages. Figure 3.3 shows the comparison of actual and predicted usages. The pattern of prediction curve is more or less similar to the actual curve.

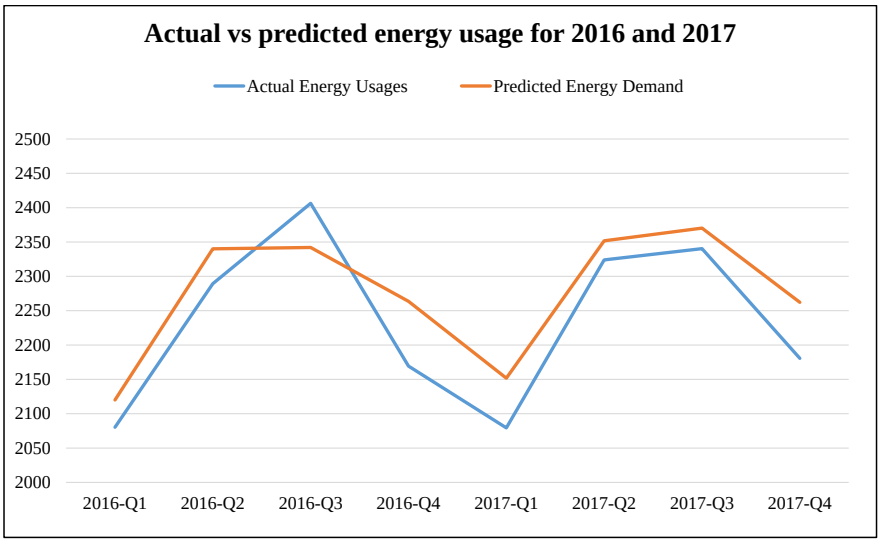


FIGURE 3.3: Comparison results using Random Forest for North Carolina.

## 3.2 Construct multiplier database

In section 3.1 I have created a model based on Random Forest to forecast the energy usages of North Carolina. Here I'm going to measure the performance of our proposed model using datasets of the different geographic region of the same country, which have same socio-economic parameters. We have studied 15 years long quarterly data for average energy usages in the industrial sector of all states of the U.S. After that we have divided these states into 16 regions, based on average industrial energy usage. This division is given in Table 3.2.

I have selected datasets of at least one state from each zone and have applied our proposed model on them. We have selected New Hampshire, New Jersey, New York, Wisconsin, Lousiana, Michigan, Illinois, Indiana, Ohio and Texas from zone A-I and p respectively. Figure 3.4(a)-(d) shows the comparison between actual and predicted quarterly energy demand for the year 2014 and 2015. It has been observed from the figure 3.4(a)-(d) that for all states, the prediction is in the range of 2000-3000. However, interestingly there is some similarity in patterns. Most of the cases the actual energy usages curve and prediction curve has produced similar patterns. So I have to find a multiplier which should be multiplied by the predicted value to improve prediction accuracy. In the next section, I will produce a multiplier database which I have calculated after doing experiments on states of the different region. I have also predicted multiplier for each quarter of the year 2015 for a state and verified it with its multiplier range. For each region, there is a range of multiplier. By using multiplier from this range we can improve the accuracy of the predicted result.

TABLE 3.2: Division of region based on Average energy usage in previous 15 years

| **Region** | **Average industrial energy usage (Thousand Megawatt)** | **States** |
|---|---|---|
| A | 0-500 | Maine, New Hampshire, Rhode Island, Vermont, North Dakota, South Dakota, Delware, Montana, Alaska, Hawaii |
| B | 500-1000 | New Jersey, Kansas, Nebraska, Idaho, New Mexico, Utah,Wyoming |
| C | 1000-1500 | Massachusetts, New York, Missouri, Florida, Virginia, West Virginia, Mississippi, Arkansas, Oklahoma,Arizona,Colorado,Nevada, Oregon |
| D | 1500-2000 | Wisconsin, MInnesota |
| E | 2000-2500 | North Carolina, South Carolina, Tennessee, Louisiana |
| F | 2500-3000 | Michigan,Georgia,Alabama |
| G | 3000-3500 | Illinois,Kentucky |
| H | 3500-4000 | Pennsylvania,Indiana,California |
| I | 4000-4500 | Ohio |
| J, K, L, M, N, O | 4500-5000-. . . -7500 | NIL |
| P | >7500 | Texas |

### 3.2.1 Multiplier database.

In this section, my objective is to find a multiplier database, which can be used to improve the accuracy and scalability of the proposed forecasting model. We have applied our model to the datasets of each state representing each zone and calculated MAPE from the predicted and the actual energy usage. If the value of MAPE is less than 10 then we can say that prediction has higher accuracy. If the MAPE is not below 10 then we have to minimize the value of MAPE by changing the predicted value. Here the main objective is to find a range of multiplier says $x$ for each zone. Algorithm 1 describes the procedure for finding $x$, for which the minimum value of MAPE is less than 10. Using algorithm 1 on various states, I have found the upper and lower limit of the multiplier. The results are given in table 3.3 and figure 3.5

Table 3.3 clearly shows that Louisiana-North Carolina, and Indiana-California both are from region E and H respectively and both of them has similar multiplier range. So by calculating the average of individual multiplier limits of all states belong to the same region we can produce the approximate range of multiplier for that region. It has also


FIGURE 3.4: Actual usage vs. predicted energy demand for (a) Florida (b) Indiana (c) Louisiana (d) Ohio

**Algorithm 1:** *Find_Multiplier*

**Input:** Quarterly data of Predicted and Actual usages of energy.

**Output:** $mul_L$ and $mul_U$:Lower and Upper limit of multiplier respectively.

Start;
let x = 0 ,min_mape = 0, flag = 2;
`/* flag is a variable which is used to determine the upper and lower limit value of the multiplier. */`
**while** $(flag \neq 0)$ **do**
    $min\_mape+ = \frac{\sum_1^n abs((actual_i - predicted_i)/actual_i)}{n} \times 100$ ;
    **if** $(min\_mape < 10$ && $flag == 2)$ **then**
        $mul_L = x$;
        flag = 1;
    **else**
        **if** $(min\_mape \geqslant 10$ && $flag == 1)$ **then**
            $mul_U = (x - 0.001)$;
            flag = 0;
        **end**
    **end**
    $x+ = 0.001$;
**end**
End;

TABLE 3.3: Upper and Lower Limit of Multiplier for different states

| Region | State | Upper limit of Multiplier | Lower limit of Multiplier |
|---|---|---|---|
| A | New Hampshire | 0.06 | 0.08 |
| B | New Jersey | 0.28 | 0.31 |
| C | Florida | 0.57 | 0.7 |
| C | New York | 0.52 | 0.7 |
| D | Wisconsin | 0.77 | 1 |
| E | Louisiana | 0.9 | 1.1 |
| E | North Carolina | 0.9 | 1.1 |
| F | Michigan | 1.01 | 1.2 |
| H | Indiana | 1.4 | 1.7 |
| H | California | 1.5 | 1.8 |
| I | Ohio | 1.7 | 2.1 |
| P | Texas | 3 | 3.7 |

been observed from figure 3.5 that actual energy usage curve belongs in between the prediction range.

I have done another experiment on multiplier range. I have taken the 7-decade long dataset of Ohio from the region I and found out that both upper and lower limits are almost same for each dataset. Figure 3.6 depicts the results .

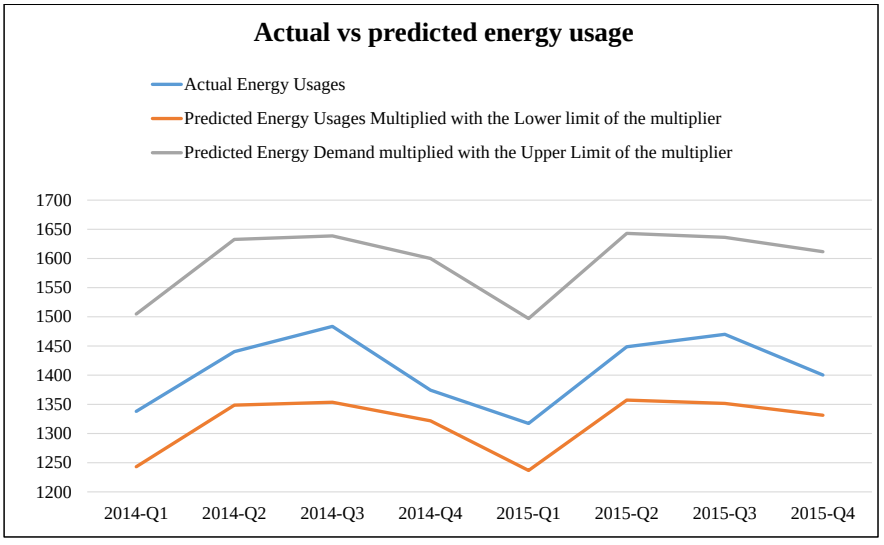


FIGURE 3.5: Actual vs predicted industrial energy demand of Florida - 2014 and 2015.

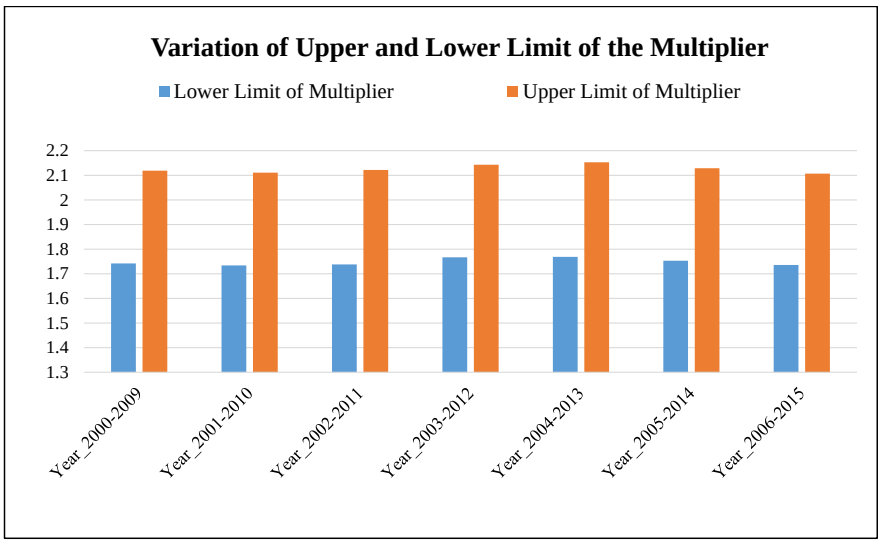


FIGURE 3.6: Variations of Upper and Lower limits of multiplier for Ohio

### 3.2.2 Forecasting Multiplier.

In the previous section, I have discussed the process of finding multiplier range for the different region. In this section, I have tried to predict the multiplier for a state and compared it with our calculated values. I have collected 15 years long quarterly data which contains actual energy usage for each quarter and predicted energy demand for each quarter which has been calculated by applying the proposed model, discussed in [26]. We have selected the dataset of Florida for experimental purpose. We have calculated the multiplier for each quarter where MAPE is minimum. The result has been shown in table 3.4.

TABLE 3.4: Minimum MAPE and corresponding Multiplier for Florida(2000-2014).

| Year with Quarter | Actual Energy Usages | Predicted Energy Usages | Multiplier | Minimum MAPE |
|---|---|---|---|---|
| 2000-Q1 | 1527.33 | 1526.16 | 0.578 | 0.077 |
| 2000-Q2 | 1578.33 | 1579.34 | 0.584 | 0.064 |
| 2000-Q3 | 1575.33 | 1576.51 | 0.577 | 0.075 |
| 2000-Q4 | 1502.33 | 1503.61 | 0.577 | 0.085 |
| ... | ... | ... | ... | ... |
| 2014-Q1 | 1338 | 1338.56 | 0.618 | 0.042 |
| 2014-Q2 | 1440.33 | 1439.21 | 0.622 | 0.078 |
| 2014-Q3 | 1483.67 | 1482.65 | 0.637 | 0.069 |
| 2014-Q4 | 1374.33 | 1374.44 | 0.613 | 0.007 |

TABLE 3.5: Prediction of multiplier for each quarter of 2015 and corresponding actual and predicted industrial energy usages of Florida

| **Year with Quarter** | **Predicted Multiplier** | **Actual Energy** | **Predicted Energy** |
|---|---|---|---|
| 2015-Q1 | 0.615 | 1317.33 | 1334.4 |
| 2015-Q2 | 0.621 | 1448.67 | 1478.697 |
| 2015-Q3 | 0.624 | 1470 | 1479.654 |
| 2015-Q4 | 0.619 | 1400.33 | 1445.74 |

Now I have done time series forecasting[2] in WEKA[17] using these obtained multiplier values in table 3.4. In time series forecasting we have applied Random Forest as a base learner and Maximum Lag creation is 12. I have forecast energy usage for each quarter of 2015 for Florida. The output of predicted multiplier has given in table 3.5. It has been observed that predicted multipliers are in the range of region C as mentioned in table 3.3 in section 3.2.1. These results, in turn, justify the effectiveness and correctness of our multiplier database, as well as, our proposed forecasting model.

[2]Appendix I section I.3 describes more on Time series forecasting.

# Chapter 4

# Conclusion

In order to plan the generation, transmission, and distribution in a Smart Grid, the energy consumption pattern of individual sectors must be known in advance. Especially in the industrial sector, quarterly demand forecasting can motivate the utilities to design the supply side of the process accordingly, so that peak demands can be reduced and an uninterrupted flow of electricity can be maintained with optimized economical gain for both users and utilities.

In this project, my main objective is to find a generalized model which can be able to predict industrial energy demand for a large geographic region. I have achieved this goal by using Regression Forest technique as an underlying data mining algorithm for our model. It has also been shown that prediction, using this model always belongs in the range of 2000-3000 KW/hr. Thus, we have constructed a multiplier database, storing different multipliers for each zone. The zones are classified using the average industrial energy usage. While forecasting energy demand for a state, firstly we have to find out in which zone it belongs. Now we will select multiplier for that zone and multiply it with the predicted value. As a future extension of this work, we are planning to incorporate more granularity in the parametric modeling for forecasting. Besides, in this paper, we have only considered the variation of industrial energy usage and its effect on the dependencies between influential parameters and consumption pattern. In future, we would like to work on the variations of several economic factors, like GDP, the price of electricity and its effect on industrial energy usage.

# Appendix A

# Additive Regression

Additive models are referred to generating predictions by summing up contributions obtained from other models. Most learning algorithms for additive models do not build the base models independently but ensure that they complement one another and try to form an ensemble of base models that optimize predictive performance according to some specified criteria. Boosting implements forward stagewise additive modeling. This class of algorithms starts with an empty ensemble and incorporates new members sequentially. At each stage, the model that maximizes the predictive performance of the ensemble as a whole is added, without altering those already in the ensemble. Optimizing the ensemble's performance implies that the next model should focus on those training instances on which the ensemble performs poorly. This is exactly what boosting does by giving those instances larger weights.

## A.1 Numeric Prediction

There is a well known forward stage-wise additive modeling method for numeric prediction.

- First build a standard regression model for example Regression tree. This model produce predicted values. Calculate the residuals from actual and predicted values
- Correct those errors by learning a second model say another regression tree. This second model tries to predict the observed residuals. To do this, simply replace the original class values by their residuals before learning the second model. Adding the predictions made by the second model to those of the first one automatically yields lower error on the training data.

- Usually some residuals still remain because the second model is not a perfect one, so we continue with a third model that learns to predict the residuals of the residuals, and so on.

If the individual models minimize the squared error of the predictions, as linear regression models do, this algorithm minimizes the squared error of the ensemble as a whole. Forward stagewise additive regression is prone to overfitting because each model added fits the training data closer and closer. To decide when to stop, use cross-validation.

## A.2 Illustration

Here an Illustration of how Additive Regression works has given below. We have chosen a dataset of North Carolina (2007-2017) from appendix H figure H.1. The result produces 10 models. It takes regression tree as a base learner and initial prediction of average demand is 2241.3030302727275

- Model number 0
  Decision Stump
  Classifications
  mean_temp $\leq$ 0.340416 : -122.35303022272748
  mean_temp $\geq$ 0.340416 : 101.96085851893919
  mean_temp is missing : -2.0670334131202914E-13

- Model number 1
  Decision Stump
  Classifications
  population$\leq$ 0.2476695 : 119.30198427738095
  population $\geq$ 0.2476695 : -22.570645674099087
  population is missing : 1.0012193094801412E-14

- Model number 2
  Decision Stump
  Classifications
  Number_of_energy_intensive_industry $\leq$ 0.434903 : -57.1617009183076
  Number_of_energy_intensive_industry $\geq$ 0.434903 : 26.67546042854354
  Number_of_energy_intensive_industry is missing : -6.136505445200865E-15

- Model number 3 Decision Stump
  Classifications

Number_of_energy_non_intensive_industry $\leq$ 0.4960785 : -42.28347956688682
Number_of_energy_non_intensive_industry $\geq$ 0.4960785 : 24.161988323935315
Number_of_energy_non_intensive_industry is missing : -5.006096547400706E-15

- Model number 4
  Decision Stump
  Classifications
  max_temp $\leq$ 0.155708 : -52.770680091666804
  max_temp $\geq$ 0.155708 : 9.98364217950453
  max_temp is missing : 2.4223047810003414E-16

- Model number 5
  Decision Stump
  Classifications
  Number_of_energy_intensive_industry $\leq$ 0.9833795 : 4.91338100235957
  Number_of_energy_intensive_industry $\geq$ 0.9833795 : -103.18100104955096
  Number_of_energy_intensive_industry is missing : 8.074349270001138E-16

- Model number 6
  Decision Stump
  Classifications
  mean_temp $\leq$ 0.956148 : -4.61558507368985
  mean_temp $\geq$ 0.956148 : 63.079662673761284
  mean_temp is missing : -6.055761952500853E-16

- Model number 7
  Decision Stump
  Classifications
  min_temp $\leq$ 0.6694175 : 18.527143674826945
  min_temp $\geq$ 0.6694175 : -24.377820624772287
  min_temp is missing : 3.4719701861004896E-15

- Model number 8
  Decision Stump
  Classifications
  Number_of_energy_non_intensive_industry $\leq$ 0.25294099999999997 : 40.556682109418674
  Number_of_energy_non_intensive_industry $\geq$ 0.25294099999999997 : -7.672885804484619
  Number_of_energy_non_intensive_industry is missing : -4.360148605800615E-15

- Model number 9
  Decision Stump

Classifications

load_share $\leq$ 0.18547 : -34.59116908305668

load_share $\geq$ 0.18547 : 5.461763539430002

load_share is missing : 1.6148698540002277E-16

# Appendix B

# Least Median of Square Regression (LMS)

LeastMidSq [32] is a robust linear regression method that minimizes the median rather than means of the square of divergence from the regression line. It repeatedly applies standard linear regression to subsamples of the data and outputs the solution that has the smallest median squared error. Classical least square regression consists of minimizing the sum of the squared residuals.Here the sum is replaced by the median of the squared residuals. The least median of squares (LMS) method has been proposed by Rousseeuw in [34] to provide a very robust estimate of parameters in linear regression problems. The LMS estimates can be obtained as a solution of the following optimization problem. Let $x_i^T = (x_{i1}, .., x_{ip}), i = 1, ..., n$ and $y = (y_1, ...y_n)^T$ be given real vectors.Let assume that $n/2 \geq p$ and the $(n \times p)$ matrix $X = [x_{ij}]$ is of full rank to avoide degenerate cases. Let $\theta = (\theta_1, ...\theta_p)^T$ be a vector of regression parameters. The optimization problem that arises out of the LMS method is to find $\theta^*$ providing

$$\theta = \min med(y_i - x_i{}_T\theta)^2 \tag{B.1}$$

## B.1 Illustration

Here an Illustration of how LeastMedSq works has given below. We have chosen a dataset of North Carolina (2007-2017) from appendix H figure H.1.It implements a least median square linear regression utilizing the existing weka LinearRegression class to form predictions. The equation for calculating average demand has given in B.2.

$$\begin{aligned} avg\_demand = & -257.5918 \times Number\_of\_energy\_intensive\_industry \\ & + 199.4011 \times Number\_of\_energy\_non\_intensive\_industry \\ & - 31.8673 \times gdp\_growth \\ & + 117.9599 \times population \\ & + 120.2105 \times max\_temp \\ & + 222.1062 \times min\_temp \\ & + 117.5651 \times load\_share \\ & + 15.6879 \times average\_retail\_sales \\ & + 2016.8296 \end{aligned} \tag{B.2}$$

# Appendix C

# MLP Regressor

MLP regressor trains a multilayer perceptron with one hidden layer using WEKA's Optimization class by minimizing the given loss function plus a quadratic penalty with the BFGS method. Note that all attributes are standardized, including the target. There are several parameters. The ridge parameter is used to determine the penalty on the size of the weights. The number of hidden units can also be specified. Note that large numbers produce long training times. Finally, it is possible to use conjugate gradient descent rather than BFGS updates, which may be faster for cases with many parameters. To improve speed, an approximate version of the logistic function is used as the default activation function for the hidden layer, but other activation functions can be specified. In the output layer, the sigmoid function is used for classification. If the approximate sigmoid is specified for the hidden layers, it is also used for the output layer. For regression, the identity function is used activation function in the output layer. Also, if delta values in the backpropagation step are within the user-specified tolerance, the gradient is not updated for that particular instance, which saves some additional time. Parallel calculation of loss function and gradient is possible when multiple CPU cores are present. Data is split into batches and processed in separate threads in this case. Note that this only improves runtime for larger datasets. Nominal attributes are processed using the unsupervised NominalToBinary filter and missing values are replaced globally using ReplaceMissingValues.

## C.1 Illustration

Here we have illustrated the application of MLP regressors on the dataset from appendix H figure H.1.

- batch size : 100
- activation function : approximate sigmoid $f(x) = 1/(1 + e^{-x})$
- loss function : squared error

MLP Regressor with ridge value 0.01 and 2 hidden units (useCGD=false)
Output unit 0 weight for hidden unit 0: 2.168832695486436
Hidden unit 0 weights:

-0.13841353803408102 Number_of_energy_intensive_industry
1.1096227315546543 Number_of_energy_non_intensive_industry
0.5236086593331005 gdp_growth
-0.8556224913981205 population
1.067748551904085 max_temp
0.8293347231781815 mean_temp
0.6147505068272534 min_temp
-2.2319670003473444 load_share
2.202944154954385 average_retail_sales

Hidden unit 0 bias: -2.824622253042397
Output unit 0 weight for hidden unit 1: 3.996829164969948
Hidden unit 1 weights:

0.2669473295055625 Number_of_energy_intensive_industry
0.08563880141548813 Number_of_energy_non_intensive_industry
0.13586773006710592 gdp_growth
-0.2146812974767929 population
0.7065346063369397 max_temp
0.17747273515372478 mean_temp
-0.5797215920188247 min_temp
0.941041104630402 load_share
-0.38141189593985525 average_retail_sales

Hidden unit 1 bias: -0.6944159203258778
Output unit 0 bias: -2.0656394306817076

# Appendix D

# Sequential Minimal Optimization (SMO)

Sequential Minimal Optimization (SMO) is used for training Support Vector Machine (SVM) for regression. Training SVM requires the solution of a very large quadratic programming (QP) optimization problem. SMO breaks this QP problem into a series of smallest possible QP problems. These small QP problems are solved analytically, which avoids using a time-consuming numerical QP optimization as an inner loop. The amount of memory required for SMO is linear in the training set size, which allows SMO to handle very large training sets.SMO's computation time is dominated by SVM evaluation, hence SMO is fastest for linear SVMs and sparse data sets.

## D.1 Algorithm

SMO is an iterative algorithm for solving the optimization problem. SMO breaks this problem into a series of smallest possible sub problems, which are then solved analytically. Because of the linear equality constraint involving the Lagrange multipliers $\alpha_i$, the smallest possible problem involves two such multipliers. Then, for any two multipliers $\alpha_1$ and $\alpha_2$, the constraints are reduced to:
$0 \leq \alpha_1, \alpha_2 \leq C$,
$y_1\alpha_1 + y_2\alpha_2 = k$, and this reduced problem can be solved analytically: one needs to find a minimum of a one-dimensional quadratic function. $k$ is the negative of the sum over the rest of terms in the equality constraint, which is fixed in each iteration. The algorithm proceeds as follows:

1. Find a Lagrange multiplier $\alpha_1$, that violates the $Karush–Kuhn–Tucker(KKT)$ conditions for the optimization problem.
2. Pick a second multiplier $\alpha_2$ and optimize the pair ($\alpha_1$,$\alpha_2$).
3. Repeat steps 1 and 2 until convergence.

When all the Lagrange multipliers satisfy the KKT conditions (within a user-defined tolerance), the problem has been solved. Although this algorithm is guaranteed to converge, heuristics are used to choose the pair of multipliers so as to accelerate the rate of convergence. This is critical for large data sets since there are $n(n-1)/2$ possible choices for $\alpha_i$ and $\alpha_j$.

## D.2 Illustration

Here an Illustration of how SMO works has given below. We have chosen a dataset of North Carolina (2007-2017) from appendix H figure H.1.

- Batch Size : 100
- Kernel : Poly Kernel

weights (not support vectors)

$$
\begin{aligned}
&: \\
&- 0.0103 \times (normalized) Number\_of\_energy\_intensive_industry \\
&+ 0.2191 \times (normalized) Number\_of\_energy\_non\_intensive\_industry \\
&+ 0.0934 \times (normalized) gdp\_growth \\
&- 0.1346 \times (normalized) population \\
&+ 0.2779 \times (normalized) max\_temp \\
&+ 0.1747 \times (normalized) mean\_temp \\
&+ 0.0807 \times (normalized) min\_temp \\
&+ 0.1515 \times (normalized) load\_share \\
&- 0.1024 \times (normalized) average\_retail\_sales \\
&+ 0.1829
\end{aligned}
\tag{D.1}
$$

Number of kernel evaluations: 990 (96.141% cached)

# Appendix E

# Random Forest

Decision Trees and their extension Random Forests are robust and easy to interpret machine learning algorithms for Classification and Regression tasks [33]. Random Forests are an ensemble of k untrained Decision Trees (trees with only a root node) with M bootstrap samples (k and M do not have to be the same) trained using a variant of the random subspace method or feature bagging method. Note the method of training random forests is not quite as straightforward as applying bagging to a bunch of individual decision trees and then simply aggregating the output. The procedure for training a random forest is as follows:

1. At the current node, randomly select p features from available features D. The number of features p is usually much smaller than the total number of features D.
2. Compute the best split point for tree k using the specified splitting metric (Gini Impurity, Information Gain, etc.) and split the current node into daughter nodes and reduce the number of features D from this node on.
3. Repeat steps 1 to 2 until either a maximum tree depth l has been reached or the splitting metric reaches some extrema. Repeat steps 1 to 3 for each tree k in the forest.
4. Vote or aggregate on the output of each tree in the forest.
5. Compared with single decision trees, random forests split by selecting multiple feature variables instead of single features variables at each split point. Intuitively, the variable selection properties of decision trees can be drastically improved using this feature bagging procedure. Typically, the number of trees k is large, on the order of hundreds to thousands for large datasets with many features.

In simple words, Random Forest builds multiple decision trees and merges them to get more accurate and stable prediction. figure E.1 depicts the working principle of random forest. One big advantage of random forest is that it can be used for both classification

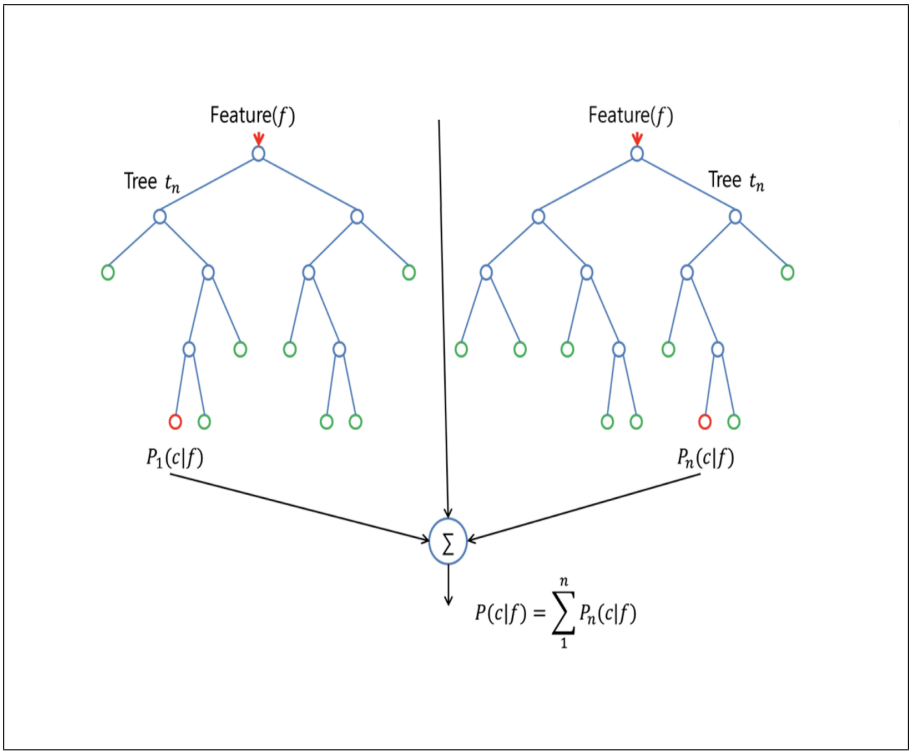


FIGURE E.1: Working principle of Random Forest

and regression problems, which form the majority of current machine learning systems. The random-forest algorithm brings extra randomness into the model when it is growing the trees. Instead of searching for the best feature while splitting a node, it searches for the best feature among a random subset of features. This process creates a wide diversity, which generally results in a better model.

# Appendix F

# M5 based Model Tree (M5P Tree)

The M5 Model tree is a decision tree learner for regression tasks [30]. This technique is used to predict the values of the numerical dependent variable. The approach of the M5 tree is same as CART tree [31]. M5 model tree chooses to mean square error as impurity function, as CART tree does. But unlike CART tree, M5P tree assigns a multivariate linear regression model instead of assigning constant directly to the leaf node. The M5 tree is much smaller than the regression tree and provides higher accuracy in prediction. This type of tree can handle data with high dimensionality and can learn efficiently instead of CART tree method.

## F.1 Construct M5 Model Tree

Construction of M5 model tree is a recursive node splitting approach, same as the construction of decision tree. M5 tree partition the data into a collection of set T and the set T is either associated with a leaf or some test is chosen that splits T into subsets corresponding to the test outcomes. The same process is applied recursively to each subset. This approach may cause overfitting, that must be solved by using pruning. Information gain in M5 tree is measured by the reduction in standard deviation before and after the test. Initial step is to calculate the standard deviation of the response values of cases in $T$. $T$ is split on the outcome of a test unless $T$ contains very few cases. Let $T_i$ denote subset of cases corresponding to $i$th outcome of a specific test.The expected reduction in error can be written as follows:

$$\Delta error = sd(T) - \sum_{i=1}^{n}(\frac{|T_i|}{|T|} \times sd(T_i)) \tag{F.1}$$

where $sd(T_i)$ is standard deviation of $i$th subset cases.

## F.2 Pruning M5 Tree

Pruning is proceeded from the leaf to root node. At every internal node, the M5 tree compares estimated error of that node with the error of the subtree below. If there is no improvement of the performance of the tree then the subtree is pruned.

### F.2.1 Error Based estimation

The M5 model tree uses error-based method for pruning. This method estimates the error of the model on unseen cases. M5 tree computes this number by first averaging the absolute difference between response values of observations and predicted values. This will generally underestimate the error on unseen cases, so M5 multiplies it by $(n+v)/(n-v)$, where n is the number of training cases and v is the number of parameters in the model. The estimated error of a subtree is calculated as the sum of estimated error of the left and right tree below that node multiplying with the proportion of samples that goes down to the left and right tree.

### F.2.2 Linear Models

A multivariate linear model is fit into each node of the tree using standard regression technique. However M5 tree does not use all features in the dataset, instead, it is restricted to features that are referenced by tests or linear models in the subtrees below this node. As M5 will compare the accuracy of a linear model with the accuracy of a subtree, this ensures a level playing field in which the two types of models use the same information. After the linear model is built, M5 tree simplifies it by greedily eliminate coefficients one by one. This way might generally result in the increase in the averaged residual, however, it also reduces the multiplicative factors above, so the estimated error can decrease.

## F.3 Smoothing

Smoothing is a process of improving the accuracy of prediction for tree based model. When the value of a case is predicted by a model tree, the value given by the model at the appropriate leaf is adjusted to reflect the predicted values at nodes along the path from the root to that leaf. The process of smoothing is given below.

- The predicted value comes from the leaf. The value is computed by the model at that leaf.

- If the case follows branch $S_i$ of subtree S, let $n_i$ be the number of training cases at $S_i$, $PV(S_i)$ the predicted value at $S_i$, and $M(S)$ is the value given by the model at $S$. The predicted value backed up to $S$ is

$$PV(S) = \frac{n_i.PV(S_i) + k.M(S)}{n_i + k} \tag{F.2}$$

## F.4 Illustration

Here an Illustration of how M5P tree works have given below. We have chosen a dataset of North Carolina (2007-2017) from appendix H figure H.1. The resultant tree has shown in figure F.1. The tree has 3 Linear model at the leaf node. The number of rule is 3 and they are given below.

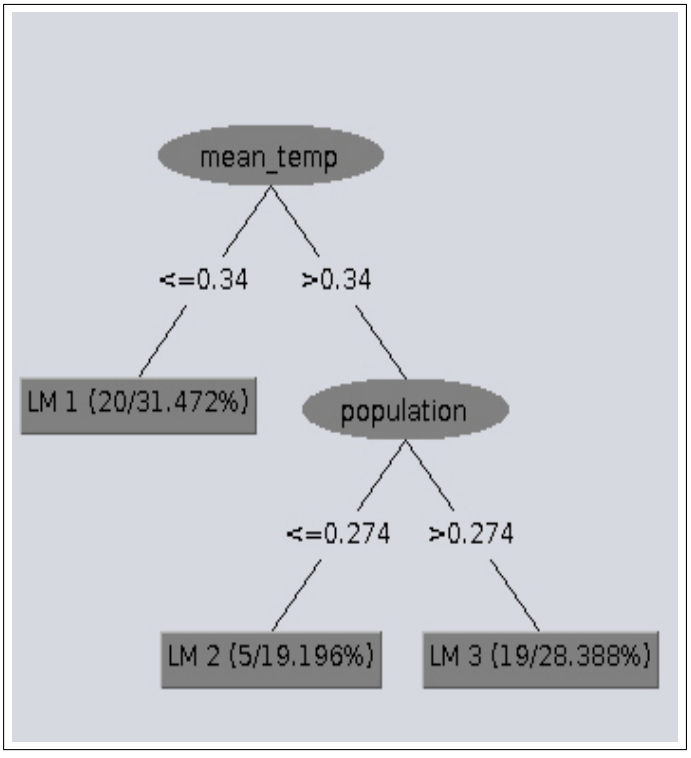


FIGURE F.1: M5 pruned model tree: (using smoothed linear models)

- LM num:1

$$\begin{aligned} avg\ demand = {} & 42.5253 \times Number\ of\ energy\ intensive\ industry \\ & + 56.759 \times Number\ of\ energy\ non\ intensive\ industry \\ & + 99.2377 \times gdp\ growth \\ & - 82.7536 \times population \\ & - 71.1759 \times max\ temp \\ & - 60.7089 \times mean\ temp \\ & + 361.3181 \times load\ share \\ & + 1912.506 \end{aligned} \tag{F.3}$$

- LM num: 2

$$\begin{aligned} avg\ demand = {} & 71.2455 \times Number\ of\ energy\ intensive\ industry \\ & + 50.9375 \times Number\ of\ energy\ non\ intensive\ industry \\ & + 147.8885 \times gdp\ growth \\ & - 203.407 \times population \\ & + 392.3102 \times max\ temp \\ & - 54.4823 \times mean\ temp \\ & + 150.7955 \times load\ share \\ & + 1915.4595 \end{aligned} \tag{F.4}$$

- LM num: 3

$$\begin{aligned} avg\ demand = {} & 177.0115 \times Number\ of\ energy\ intensive\ industry \\ & + 50.9375 \times Number\ of\ energy\ non\ intensive\ industry \\ & + 106.6241 \times gdp\ growth \\ & - 150.2313 \times population \\ & + 483.5275 \times max\ temp \\ & - 54.4823 \times mean\ temp \\ & + 156.6258 \times load\ share \\ & + 1761.3017 \end{aligned} \tag{F.5}$$

# Appendix G

# Evaluating Numerical Prediction

The performance of numeric prediction can be evaluated by using several metrics. In this Chapter we have described 7 evaluation metrics which are summarized in table G.1. The predicted values on the test instances are $p_1, p_2, \ldots, p_n$; the actual values are $a_1, a_2, \ldots, a_n$. Where $p_i$ and $a_i$ refers to the numerical value of the prediction and actual respectively for the $i$th test instance.

TABLE G.1: Metrics for Performance evalution

| Performance Metric | Formula |
|---|---|
| $CorrelationCoefficient$ | $\frac{\sum_{i=1}^{n}(p_i-\overline{p})(a_i-\overline{a})}{\sqrt{\sum_{i=1}^{n}(p_i-\overline{p})^2(a_i-\overline{a})^2}}$ |
| $MAE$ | $\frac{\sum_{i=1}^{n}\lvert p_i-a_i\rvert}{n}$ |
| $RMSE$ | $\sqrt{\frac{\sum_{i=1}^{n}(p_i-a_i)^2}{n}}$ |
| $RAE(\%)$ | $\frac{\sum_{i=1}^{n}\lvert p_i-a_i\rvert}{\sum_{i=1}^{n}\lvert a_i-\overline{a}\rvert} \times 100$ |
| $RRSE(\%)$ | $\sqrt{\frac{\sum_{i=1}^{n}(p_i-a_i)^2}{\sum_{i=1}^{n}(a_i-\overline{a})^2}} \times 100$ |
| $MAPE(\%)$ | $\frac{\sum_{1}^{n}\lvert\frac{(a_i-p_i)}{a_i}\rvert}{n} \times 100$ |
| $R^2value$ | $1-\frac{SS_{res}}{SS_{tot}}$ |

## G.1 Correlation Coefficient(CC)

Correlation Coefficient measures the statistical correlation between the actual and predicted values. The range of Correlation Coefficient is $[-1, 1]$. If the value of CC is 1 that means predicted and actual results are perfectly correlated. If the value of CC is -1 then they are perfectly correlated negatively. It the value is 0 i.e., there is no correlation between predicted and actual values. This type of measurement is scale independent i.e., if we take a particular set of predictions, the error is unchanged if all the predictions are multiplied by a constant factor and the actual values are left unchanged. The formula of calculating correlation coefficient has given in table G.1.

## G.2 Mean Absolute Error (MAE)

Mean Absolute Error is an alternative to Mean-squared error. Some mathematical technique viz. Linear regression use mean-square error but in performance measurement it is not advantageous. Mean squared error tends to exaggerate the effect of outliers i.e., instances when the prediction error is larger than the others. MAE can solve this problem by just doing average the magnitude of the individual errors without taking account the sign. The formula for calculating MAE has given in table G.1.

## G.3 Root Mean Square Error (RMSE)

Root Mean Square Error has been referred in table G.1. RMSE is standard deviation of the residuals or prediction errors. Residuals are a measure of how far the data points are present from the regression line. This type of measurement is helpful for forecasting.

## G.4 Relative Absolute Error (RAE)

The relative absolute error is very similar to the relative squared error in the sense that it is also relative to a simple predictor, which is just the average of the actual values. In this case, though, the error is just the total absolute error instead of the total squared error. Thus, the relative absolute error takes the total absolute error and normalizes it by dividing by the total absolute error of the simple predictor. Mathematically, the relative absolute error $e_i$ of an individual program $i$ is evaluated by the equation given in table G.1.

## G.5 Relative Root Square Error(RRSE)

The root relative squared error is relative to what it would have been if a simple predictor had been used. More specifically, this simple predictor is just the average of the actual values. Thus, the relative squared error takes the total squared error and normalizes it by dividing by the total squared error of the simple predictor. By taking the square root of the relative squared error one reduces the error to the same dimensions as the quantity being predicted. Mathematically, the root relative squared error $e_i$ of an individual program $i$ is evaluated by the equation given in table G.1.

## G.6 Mean Absolute Percentage Error(MAPE)

Mean Absolute Percentage Error is a measure of prediction accuracy of a forecasting method in statistics. It expresses accuracy as a percentage . This metric is defined by the formula ,given in table G.1. The calculation of MAPE is very easy but it has some drawbacks in practical applications.

- 1. It cannot be used if there are zero values (which sometimes happens for example in demand data) because there would be a division by zero.
- 2. For forecasts which are too low the percentage error cannot exceed 100%, but for forecasts which are too high there is no upper limit to the percentage error.
- 3. When MAPE is used to compare the accuracy of prediction methods it is biased in that it will systematically select a method whose forecasts are too low. This little-known but serious issue can be overcome by using an accuracy measure based on the ratio of the predicted to actual value (called the Accuracy Ratio), this approach leads to superior statistical properties and leads to predictions which can be interpreted in terms of the geometric mean

## G.7 Model Testing using Coefficient of Determination or $R^2$ Score

The Coefficient of Determination is the proportion of the variance in the dependent variable that is predictable from the independent variable(s). The formulae of calculating $R^2$ score has given in table G.1. The value of $R^2$ is between 0 to 1 or 0 to 100%. A data set has n values marked $a_1$,...,$a_n$.,each associated with a predicted (or modeled) value $p_1$,...,$p_n$. Define the residuals as $e_i = a_i$ - $p_i$ (forming a vector e).

if $\overline{a}$ is the mean of the observed data then :

$$\overline{a} = \frac{1}{n}\sum_{i=1}^{n} a_i \tag{G.1}$$

Then the variability of data set can be measured using three sums of squares formulas:

- Total Sum of Squares ($SS_{tot}$):

$$SS_{tot} = \sum_{i=1}^{n}(a_i - \overline{a}^2) \tag{G.2}$$

- The regression sum of squares ($SS_{reg}$):

$$SS_{reg} = \sum_{i=1}^{n}(p_i - \overline{a})^2 \tag{G.3}$$

- residual sum of squares ($SS_{res}$) :

$$\begin{aligned} SS_{res} &= \sum_{i=1}^{n}(a_i - p_i)^2 \\ &= \sum_{i=1}^{n} e_i^2 \end{aligned} \tag{G.4}$$

- Coefficient of Determination : $R^2$ score can easily be derived by using equation G.2 and G.4.

$$R^2 \equiv 1 - \frac{SS_{res}}{SS_{tot}} \tag{G.5}$$

Here the high $R^2$ value (.85 to 1) means the data has fitted very well in our model. Low $R^2$ value indicates that data has not fit well with the proposed model.

# Appendix H

# Data

Here the overview of the dataset has given. This type of datasets has been used as training or test set. We have collected 15 years long(2000-2017) data of North Carolina, Florida, New Jersey, New Hampshire, Indiana, California, Michigan, Illinois, Louisiana, New York, Ohio, Texas, Wisconsin. We have divided the 15-year dataset into 9 decade long data. i.e., Data for year $2000-2009, 2001-2010, 2002-2011, 2003-2012, 2004-2013, 2005-2014, 2006-2015, 2007-2016$ for each of the above mentioned states of U.S.A. In our dataset there are nine independent variable and one dependent variable as mentioned in chapter 3 section 3.1.1. The illustration of the data of North Carolina(2007-2016) has given in figure H.1. Each dataset contains quarterly data of 10 consecutive years for a particular state. In the dataset, there is 40 row. Each row is representing a value of all parameter for each quarter. Each column is representing the value of each independent variable.

Relation: Data_N_C_2007_2017

| No. | 1: average_demand | 2: No_of_energy_intensive_industry | 3: no_of_energy_non_Intensive_Industry | 4: gdp_growth | 5: popultion | 6: max temp | 7: mean temp | 8: min temp | 9: load_share | 10: **average_retail_sales** |
|---|---|---|---|---|---|---|---|---|---|---|
| | Numeric | Numeric | Numeric | Numeric | Numeric | Numeric | Numeric | Numeric | Numeric | Numeric |
| 1 | 2248.0 | 2819.0 | 5084.0 | -0.044017 | 9043.96... | 57.566667 | 45.7 | 33.866667 | 21.002149 | 5.056667 |
| 2 | 2471.666667 | 2835.0 | 5082.0 | 0.578077 | 9092.84... | 78.166667 | 65.8 | 53.466667 | 24.093449 | 5.146667 |
| 3 | 2536.0 | 2832.0 | 5070.0 | 0.532996 | 9142.47... | 87.8 | 76.266667 | 64.733333 | 19.914666 | 5.953333 |
| 4 | 2423.0 | 2871.0 | 5083.0 | 1.463129 | 9192.85... | 65.9 | 54.133333 | 42.333333 | 23.829662 | 5.583333 |
| 5 | 2242.666667 | 2885.0 | 5091.0 | 1.668308 | 9243.99... | 57.666667 | 45.766667 | 33.8 | 20.951669 | 5.226667 |
| 6 | 2459.666667 | 2896.0 | 5086.0 | 1.817618 | 9290.63... | 78.733333 | 66.866667 | 55.0 | 23.908113 | 5.37 |
| 7 | 2408.333333 | 2918.0 | 5092.0 | 0.77538 | 9332.76... | 84.6 | 74.3 | 63.933333 | 19.759333 | 6.0 |
| 8 | 2087.666667 | 2910.0 | 5088.0 | -1.784517 | 9370.39... | 61.733333 | 50.366667 | 39.0 | 21.040785 | 5.786667 |
| 9 | 1863.666667 | 2899.0 | 5025.0 | -1.686899 | 9403.52... | 54.733333 | 43.6 | 32.466667 | 17.3273 | 5.793333 |
| 10 | 2078.666667 | 2904.0 | 4987.0 | 0.255431 | 9435.19... | 78.266667 | 67.1 | 55.933333 | 21.017863 | 5.896667 |
| 11 | 2183.666667 | 2938.0 | 5028.0 | 0.414232 | 9465.39... | 83.866667 | 74.066667 | 64.266667 | 18.184594 | 6.32 |
| 12 | 2070.333333 | 2943.0 | 5048.0 | 0.239048 | 9494.14... | 60.866667 | 50.766667 | 40.633333 | 21.164724 | 5.97 |
| 13 | 1972.333333 | 2954.0 | 4930.0 | 0.409831 | 9521.43... | 51.4 | 41.0 | 30.633333 | 16.911996 | 5.786667 |
| 14 | 2291.0 | 2947.0 | 4904.0 | 0.423716 | 9547.34... | 81.833333 | 69.9 | 57.866667 | 22.062081 | 6.03 |
| 15 | 2331.333333 | 2998.0 | 4971.0 | 0.707572 | 9571.86... | 87.9 | 77.033333 | 66.1 | 17.822287 | 6.59 |
| 16 | 2184.666667 | 3009.0 | 4962.0 | 1.807339 | 9595.01... | 59.766667 | 48.166667 | 36.533333 | 21.101774 | 5.886667 |
| 17 | 2091.666667 | 3016.0 | 4926.0 | -0.469136 | 9616.77... | 55.566667 | 44.166667 | 32.766667 | 18.501047 | 5.83 |
| 18 | 2272.0 | 3027.0 | 4932.0 | 2.190446 | 9639.15... | 80.566667 | 68.666667 | 56.766667 | 21.559386 | 5.85 |
| 19 | 2332.333333 | 3038.0 | 4938.0 | -0.155063 | 9662.15... | 86.6 | 76.233333 | 65.833333 | 18.638785 | 6.476667 |
| 20 | 2158.0 | 3053.0 | 4966.0 | -0.163674 | 9685.77... | 64.466667 | 52.5 | 40.533333 | 22.299532 | 6.03 |
| 21 | 2067.666667 | 3061.0 | 5026.0 | 1.91747 | 9710.01... | 61.2 | 49.466667 | 37.666667 | 20.318386 | 6.186667 |
| 22 | 2325.666667 | 3079.0 | 5043.0 | -0.335196 | 9734.17... | 78.6 | 67.033333 | 55.466667 | 22.712328 | 6.326667 |
| 23 | 2275.0 | 3049.0 | 5035.0 | 1.101821 | 9758.25... | 85.3 | 75.266667 | 65.2 | 18.59369 | 6.85 |
| 24 | 2251.333333 | 3042.0 | 5031.0 | -0.550349 | 9782.25... | 62.1 | 51.133333 | 40.133333 | 22.401327 | 6.133333 |
| 25 | 2057.666667 | 3051.0 | 5020.0 | 1.819566 | 9806.16... | 53.866667 | 43.3 | 32.666667 | 18.985668 | 6.076667 |
| 26 | 2297.333333 | 3077.0 | 5027.0 | 0.82052 | 9829.90... | 76.6 | 66.033333 | 55.366667 | 22.647213 | 6.156667 |
| 27 | 2388.333333 | 3051.0 | 5036.0 | 1.713867 | 9853.45... | 82.866667 | 73.6 | 64.366667 | 19.931013 | 6.753333 |
| 28 | 2189.0 | 3081.0 | 5072.0 | 1.566192 | 9876.83... | 62.266667 | 51.3 | 40.3 | 21.338056 | 6.306667 |
| 29 | 2077.666667 | 3127.0 | 5118.0 | 0.5057 | 9900.03... | 52.8 | 41.266667 | 29.666667 | 17.917098 | 6.46 |
| 30 | 2339.333333 | 3128.0 | 5113.0 | 0.39555 | 9923.06... | 79.466667 | 67.633333 | 55.8 | 22.547067 | 6.386667 |
| 31 | 2378.0 | 3112.0 | 5107.0 | 1.703392 | 9945.91... | 82.8 | 73.733333 | 64.666667 | 19.84975 | 6.736667 |
| 32 | 2222.666667 | 3101.0 | 5110.0 | 1.381541 | 9968.58... | 61.9 | 50.766667 | 39.6 | 21.537468 | 6.123333 |
| 33 | 2084.333333 | 3149.0 | 5159.0 | 1.299684 | 9991.08... | 52.633333 | 41.666667 | 30.666667 | 18.037327 | 6.293333 |
| 34 | 2428.333333 | 3169.0 | 5159.0 | 1.366448 | 10017.7... | 79.866667 | 68.633333 | 57.433333 | 23.078629 | 6.273333 |
| 35 | 2391.333333 | 3152.0 | 5108.0 | 1.365135 | 10048.5... | 85.233333 | 75.4 | 65.566667 | 19.161325 | 6.826667 |
| 36 | 2267.333333 | 3179.0 | 5112.0 | 0.91332 | 10083.4... | 66.466667 | 56.366667 | 46.3 | 23.315281 | 6.19 |
| 37 | 2080.333333 | 3169.0 | 5082.0 | 0.902496 | 10122.5... | 56.6 | 45.966667 | 35.266667 | 18.997321 | 5.846667 |
| 38 | 2289.333333 | 3180.0 | 5059.0 | 0.963841 | 10151.8... | 77.933333 | 66.6 | 55.333333 | 22.241653 | 6.15 |
| 39 | 2406.333333 | 3138.0 | 5004.0 | 1.073221 | 10171.3... | 87.666667 | 77.866667 | 68.033333 | 18.303288 | 6.82 |
| 40 | 2169.333333 | 3158.0 | 5046.0 | 1.751429 | 10181.1... | 64.6 | 52.966667 | 41.333333 | 21.452352 | 6.026667 |

FIGURE H.1: Data of North Carolina(2007-2016)

# Appendix I

# Weka 3: Data Mining Software in Java

**Waikato Environment for Knowledge Analysis** (Weka) is a suite of machine learning software written in Java, developed at the University of Waikato, New Zealand. It is free software licensed under the GNU General Public License [16, 17]. The algorithms can either be applied directly to a dataset or called from your own Java code. Weka contains tools for data preprocessing, classification, regression, clustering, association rules, and visualization. It is also well-suited for developing new machine learning schemes.

## I.1 Description

Weka contains a collection of visualization tools and algorithms for data analysis and predictive modeling, together with graphical user interfaces for easy access to these functions. Weka 3 is a JAVA based version, mostly used for educational and research purposes. Advantages of Weka are given below:

1. Availability of this software is free under the GNU General Public License.
2. Advantageous on Portability, since it is fully implemented in the Java programming language, thus runs on almost any modern computing platform.
3. An extensive collection of data preprocessing and modeling techniques.
4. It is easy to use because of its GUI

Weka supports several standard data mining tasks, more specifically, data preprocessing, clustering, classification, regression, visualization, and feature selection. In this work, we mainly focused on regression. All of Weka's techniques are predicated on the assumption that the data is available as one flat file or relation, where each data point is described by a fixed number of attributes (normally, numeric or nominal attributes, but some other attribute types are also supported). Weka provides access to SQL databases using JDBC and can process the result returned by a database query. Weka provides access to deep learning too.

## I.2 Graphical User Interface (GUI)

Weka's main user interface is the Explorer, but essentially the same functionality can be accessed through the component-based Knowledge Flow interface and from the command line. There is also the Experimenter, which allows the systematic comparison of the predictive performance of Weka's machine learning algorithms on a collection of datasets.

The Explorer interface features several panels providing access to the main components of the workbench:

- **Preprocess panel :** It has facilities for importing data from a database, a comma-separated values (CSV) file, an Attribute-Relation File Format (ARFF) etc., and for preprocessing this data using a so-called filtering algorithm. These filters can be used to transform the data (e.g., turning numeric attributes into discrete ones) and make it possible to delete instances and attributes according to specific criteria.
- **Classify panel :** It enables applying classification and regression algorithms (indiscriminately called classifiers in Weka) to the resulting dataset, to estimate the accuracy of the resulting predictive model, and to visualize erroneous predictions, receiver operating characteristic (ROC) curves, etc., or the model itself (if the model is amenable to visualization like, e.g., a decision tree).
- **Associate panel :** It provides access to association rule learners that attempt to identify all important interrelationships between attributes in the data.
- **Cluster panel :** It gives access to the clustering techniques in Weka, e.g., the simple k-means algorithm. There is also an implementation of the expectation maximization algorithm for learning a mixture of normal distributions.
- **Select attributes panel :** It provides algorithms for identifying the most predictive attributes in a dataset.

- **Visualize panel :** It shows a scatter plot matrix, where individual scatter plots can be selected and enlarged and analyzed further using various selection operators.

## I.3 Time Series Forecasting

Time series analysis is the process of using statistical techniques to model and explains a time-dependent series of data points. Time series forecasting is the process of using a model to generate predictions (forecasts) for future events based on known past events. Time series data have a natural temporal ordering - this differs from typical data mining/machine learning applications where each data point is an independent example of the concept to be learned, and the ordering of data points within a data set does not matter.

$Weka \geq 3.7.3$ has dedicated time series analysis environment that allows forecasting models to be developed, evaluated and visualized. This environment takes the form of a plug-in tab in Weka's graphical "Explorer" user interface and can be installed via the package manager. Weka's time series framework takes a machine learning/data mining approach to modeling time series by transforming the data into a form that standard propositional learning algorithms can process. It does this by removing the temporal ordering of individual input examples by encoding the time dependency via additional input fields. These fields are sometimes referred to as "*lagged*" variables. Various other fields are also computed automatically to allow the algorithms to model trends and seasonality. After the data has been transformed, any of Weka's regression algorithms can be applied to learn a model. An obvious choice is to apply multiple linear regression, but any method capable of predicting a continuous target can be applied - including powerful non-linear methods such as support vector machines for regression and model trees (decision trees with linear regression functions at the leaves). This approach to time series analysis and forecasting is often more powerful and more flexible than classical statistical techniques such as ARMA and ARIMA. In this work, we have used the Random forest as a base learner and maximum lag creation is 12. We have seen that Random Forest as a base learner gives better results compared to other techniques.

# Bibliography


[1] Bakker,V.,"Triana a control strategy for Smart Grids Forecasting, planning and real-time control", Ph.D Thesis,ISBN:978-90-365-3314-0, ISSN: 1381-3617 ( CTIT Ph.D.Thesis Series No. 11-215 ) DOI 10.3990/1.9789036533140

[2] Chakraborty,M.,Chaki,N.,Das,S.K.,"Securing and Maintaining Automatic Functionalities for Advanced Metering Infrastructure in Smart Power Grid", A technical report on pre-submission seminar,University of Calcutta, June, 2018.

[3] Ghalehkhondabi,I.,Ardjmand,E.,Weckman,G.R., et.al., "An overview of energy demand forecasting methods published in 2005–2015", Energy System, vol. 8, Issue 2, pp. 411–447, May 2017.

[4] Khodayar,M.E.,Wu,H., "Demand Forecasting in the Smart Grid Paradigm: Features and Challenges",The Electricity Journal, vol. 28, no. 6, pp. 51-62, July 2015.

[5] Zafer,D.,Hunt,LC., "Industrial electricity demand for Turkey: A structural time series analysis", Energy Economics, vol. 33,Issue 3,pp. 426-436, May 2011.

[6] Socares,L.J.,Mederios,M.C., "Modeling and forecasting short-term electricity load: A comparison of methods with an application to Brazilian data",International Journal of Forecasting, vol. 24, Issue 4, pp. 630-644, October–December 2008.

[7] Ali,S.M.,Mehmood,C.A.,Khan,B.,Jawad,M.,et al., "Stochastic and Statistical Analysis of Utility Revenues and Weather Data Analysis for Consumer Demand Estimation in Smart Grids."PLoSONE , vol. 11, no. 6 :e0156849., June 2016. DOI 10.1371/journal.pone.0156849

[8] Hong,T.,Gui,M.,Baran,M.E.,Willis,H.L, " Modeling and Forecasting Hourly Electric Load by Multiple Linear Regression with Interactions",Power and Energy Society General Meeting,USA, pp. 1-8, 2010.

[9] Song,K.,Baek,Y.,Hong,D.H.,Jang,G., "Short-term load forecasting for the holidays using fuzzy linear regression method," IEEE Transactions on Power Systems, vol. 20, no.1, pp. 96-101, Feb. 2005.

[10] Ang,B.W.,Xu, X.Y., "Tracking industrial energy efficiency trends using index decomposition analysis",Energy Economics, vol. 40 , pp. 1014-1021, November 2013.

[11] Ang,B.W.,Wang,H.,"Index decomposition analysis with multidimensional and multilevel energy data",Energy Economics, Vol. 51, pp. 67-76, September 2015

[12] U.S. Energy Information Administration, "Industrial sector Energy Consumption", International Energy Outlook , pp. 113-126, 2016.

[13] Xu,X.Y.,Ang,B.W., "Multilevel index decomposition analysis : Approaches and application",Energy Economics, vol. 44 , pp. 375-382,July 2014.

[14] Mehr,M.N.,Samavati,F.F.,Jeihoonian,M., "Annual energy demand estimation of Iran industrial sector by Fuzzy regression and ARIMA", Eight International Conference on Fuzzy Systems and Knowledge Discovery (FSKD),IEEE, vol. 1,pp. 593-597,2011.

[15] Huang,Q.,Li,Y.,Liu,S.,Liu,P.," Short term load forecasting based on wavelet decomposition and random forest",Proceeding of the workshop on Smart Internet of Things, Article No. 2,SmartIoT 2017.

[16] Weka 3.8.1 : Data Mining software in Java, [Online]Available :https://www.cs.waikato.ac.nz/ml/weka/

[17] Witten,I.H.,Frank,E., and Hal,M.A.,"Data Mining: Practical Machine Learning Tools and Techniques",The Morgan Kaufmann Series in Data Management Systems, 3rd Edition,April 2004.

[18] U.S. EIA,Today in Energy, [Online]Available :https://www.eia.gov/todayinenergy/detail.php?id=8110

[19] U.S.Energy Information Administration,Electricity, [Online]Available : https://www.eia.gov/electricity/monthly/backissues.html

[20] U.S. Bureau of Labor,[Online]Available : Statistics(BLS),https://www.bls.gov/

[21] Bureau of Economic Analysis(BEA)U.S. Department of Commerce ,[Online]Available :https://www.bea.gov

[22] U.S. Census Bureau ,[Online]Available : https://www.census.gov/

[23] National Centers for Environmental Information,NOAA,[Online]Available :https://www.ncdc.noaa.gov

[24] Weka 3:Data Mining Software in Java,Class AdditiveRegression [Online]Available:http://weka.sourceforge.net/doc.dev/weka/classifiers/meta/AdditiveRegression.html

[25] Weka 3:Data Mining Software in Java,Class LeastmedSquare [Online]Available:http://weka.sourceforge.net/doc.packages/leastMedSquared/weka/classifiers/functions/LeastMedSq.html

[26] Weka 3:Data Mining Software in Java,Class Random Forest [Online]Available: http://weka.sourceforge.net/doc.dev/weka/classifiers/trees/RandomForest.html

[27] Weka 3:Data Mining Software in Java, Class M5P [Online]Available:http://weka.sourceforge.net/doc.dev/weka/classifiers/trees/M5P.html

[28] Weka 3: Data Mining Software in Java, Class MLP [Online]Available:http://weka.sourceforge.net/doc.packages/multiLayerPerceptrons/weka/classifiers/functions/MLPRegressor.html

[29] Weka 3: Data Mining Software in Java, Class SMO [Online]Available:http://weka.sourceforge.net/doc.dev/weka/classifiers/functions/SMO.html

[30] Quinlan,R.J.,"Learning with Continuous Classes",5th Australian Joint Conference on Artificial Intelligence,Singapore, pp. 343-348, 1992.

[31] Breiman,L., Friedman,J.,Olshen,R., and Stone.C., "Classification and Regression Trees", Wadsworth Statistics/Probability, 1st edition, January , 1984. ISBN-13: 978-0412048418

[32] Simpson,D.G.,"Introduction to Rousseeuw(1984) Least Median of Squares Regression",Breakthroughs in Statistics, vol. 3, pp. 433-461, 1997.

[33] Breiman,L.,"Random Forests",Machine Learning,vol. 45, Issue 1, pp. 5–32 , October, 2001.

[34] Rousseeuw,P.J., "Least Median of squares regression", Journal of American Statistical Association vol. 79, pp. 871-880, 1984.