SHORT TERM ELECTRICITY LOAD FORECASTING USING MACHINE LEARNING TECHNIQUES

by

Mustafa Yeşilyurt

B.S., Industrial Engineering, Boğaziçi University, 2014

Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Master of Science

Graduate Program in Industrial Engineering
Boğaziçi University
2018

ACKNOWLEDGEMENTS

I would like to thank first my thesis advisor Assistant Professor Mustafa Gökçe Baydoğan for his guidance and attention. Without his support and enthusiasm, this study could have never been completed. He provided me great educational experience which I could not be able to get from anyone else.

I would like to thank also to the member of thesis committee, Professor Gürkan Kumbaroğlu and Assistant Professor Ömer Faruk Beyca for their efforts.

I would like to express my gratefulness to TUBITAK for providing me scholarship via 2210-National Scholarship Programme for MSc Students.

I would like to express my gratitude to Center for Islamic Studies(ISAM) for allowing me to study in their peaceful library.

I am so grateful to my family especially. Without their support and courage they gave me unconditionally, I would not be able to accomplish anything.

ABSTRACT

SHORT TERM ELECTRICITY LOAD FORECASTING USING MACHINE LEARNING TECHNIQUES

Electricity is one of the most essential component of human life. Maintaining and operating electric power systems is a complex task. Short term electricity load forecasting plays an important role in operation of electric systems. Plenty of methodologies have been applied to perform short term load forecast. In this study, performances of several machine learning methodologies such as random forest, support vector machines and gradient boosting method are analyzed in short term load forecasting. Performances of these methods are compared with performances of three naive methods, several time series methods and linear regression. All methods are tested on a dataset which includes hourly electricity load information of a region of a country and hourly temperature values of cities in that region. For machine learning methods and linear regression, a feature set is constructed based on calendar, temperature and past load information. Minimum absolute percentage error(MAPE) is used as performance metric to compare methodologies. According to results of experiments, machine learning methods showed better performance than conventional methods in short term load forecasting.

ÖZET

MAKİNE ÖĞRENMESİ YÖNTEMLERİ KULLANILARAK KISA DÖNEM ELEKTRİK TALEP TAHMİNİ

Elektrik insan hayatının en önemli parçalarından biri haline gelmiştir. Elektrik sistemlerinin sürdürülmesi ve yönetilmesi ise kompleks bir iştir. Kısa dönemli elektrik talep tahmini elektrik sistemlerinin yönetilmesinde önemli bir rol oynar. Bugüne kadar birçok metod elektrik talep tahmininde kullanılmıştır. Bu çalışmada, rastsal orman, destek vektör makineleri ve meyil yükseltme metodu gibi bazı makine öğrenmesi metodlarının kısa dönemli elektrik talep tahminindeki performasnları analiz edilmiştir. Bu metodların performansları, üç yalın metodun, çeşitli zaman serisi metodlarının ve doğrusal regresyonun performansları ile karşılaştırılmıştır. Bütün metodlar, bir bölgenin saatlik elektrik taleplerini ve o bölgedeki şehirlerin saatlik sıcaklıklarını içeren bir veriseti üzerinde test edilmiştir. Makine öğrenmesi metodları ve doğrusal regresyon için takvim, sıcaklık ve geçmiş talep bilgilerine dayanan bir değişken seti oluşturulmuştur. Metodları karşılaştırmak için minimum mutlak yüzde hata performans ölçütü olarak kullanılmıştır. Deney sonuçlarına göre, makine öğrenmesi metodlarının geleneksel metodlara göre daha iyi sonuçlar gösterdiği görülmüştür.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS

b	Intercept term in SVM
C	Cost parameter of support vectors in SVM
d	Order of integration part in ARIMA
D	Order of seasonal integration part in ARIMA
k	Number of parameter levels in parameter tuning
l	Number of weeks first used in naive methods
m	Number of parameters in parameter tuning
n	Number of folds in parameter tuning
p	Order of auto-regressive part in ARIMA
P	Order of seasonal auto-regressive part in ARIMA
Q	Order of seasonal moving average part in ARIMA
q	Order of moving average part in ARIMA
t	Number of periods in a season
w	Parameter vector in SVM
x_i	i_{th} element of feature vector in SVM
y_i	i_{th} element of response vector in SVM
ε	Tolerance parameter in SVM
ξ_i	i_{th} support vector above prediction in SVM
ξ_i^*	i_{th} support vector below prediction in SVM

LIST OF ACRONYMS/ABBREVIATIONS

AAE	Absolute Average Error
AIC	Akaike Information Criterion
AICC	Akaike Information Criterion Corrected
ANN	Artificial Neural Networks
AR	Auto-Regressive
ARIMA	Auto-Regressive Integrated Moving Average
ARMA	Auto-Regressive Moving Average
BIC	Bayesian Information Criterion
CART	Classification and Regression Trees
GBM	Gradient Boosting Method
HW	Holt-Winters
LR	Linear Regression
LTLF	Long Term Load Forecasting
MA	Moving Average
MAPE	Mean Absolute Percentage Error
MTLF	Medium Term Load Forecasting
RF	Random Fores
RMSE	Root Mean Squared Error
STLF	Short Term Load Forecasting
SVM	Support Vector Machines
Tbats	Trigonometric Box-Cox transformation Auto-Regressive Moving Average Trend Seasonality
VSTLF	Very Short Term Load Forecasting

1. INTRODUCTION

Energy is one of the most crucial entity for a society to maintain its life-sustaining processes. Especially with the development of the countries and transformation of life structures, need for energy increases significantly. However, energy resources are limited. Therefore, energy related issues have become more important. Moreover, increase in worldwide concerns about the climate related issues has also contributed more interest on energy. This situation causes an increase in the interest on every stage of energy life cycle and related policies, too.

Electricity is a type of energy which is in secondary energy groups. It could only be produced by using different sources. Among the energy types, electricity seems as the most common and important energy type for human beings and its importance and usage will increase more in future especially with the digitalization era. Also in other fields of activities, electricity is increasing its share. Therefore, because importance of electricity is increasing, planning and organizing electricity is also increasing.

In order to provide safe and efficient electricity, countries or other organizations need to plan their energy policies, investments and operations sufficiently before. Because, it may take years to execute a decision. In order to perform these processes, a country, organization, companies, etc. need to forecast the future electricity demand and then plan everything which is related with it. These organizations use several methods to forecast the future electricity load in different time intervals according to their needs. These forecasts are usually categorized in 3 different part based on the time interval:

Short Term Load Forecasting (STLF): It usually refers to forecasts from few minutes to 24 hours.

Medium Term Load Forecasting (MTLF): It usually refers to forecasts from a few days to several months.

Long Term Load Forecasting (LTLF): It usually refers to forecast from one year to more years [7].

In all of these forecast types, electricity load curves show different characteristics and there are different factors affecting the behaviour of the load. For example, in long term electricity forecast, energy prices, economic parameters, population and etc. should be considered. In medium term, seasonal climate changes should be considered. In short term forecast, consumer daily behaviours, hourly weather conditions and etc. should be considered in the models [8].

Among these forecast types, STLF gains more importance in recent years. Accurate STLF is very important instrument to provide safety and efficiency to a power system. Electricity is not a storable material and there has to be always a proper balance of consumption and production in a power grid. It means, whatever amount of electricity is produced in a power system should be consumed instantly or vice versa. Otherwise, some failures such as high voltage, low voltage and etc. may occur. In order to avoid from those failures, the production and consumption of the system should be well planned. This way, safe and smooth energy to consumer could be provided, power system could be kept safe and unnecessary costs could be prevented. Since it is an instant system, the best option to achieve these objectives is forecasting. Forecasting accurately the power inputs and outputs on the system will increase the planning performance and decrease the necessity efforts to keep system in balance and reduce the probability of failures.

Furthermore, with the emergence of renewable energies by 21th century and smart grid phenomenon importance of electric load forecasting is increased [9]. Because the nature of renewable energy sources do not allow producers to fully control the production, it is important to at least forecast the production. Additionally, forecasting is one of the basic tool for smart grids. Together with instant information of the system, forecasting provides ability to manipulate and control the system. Another main factor that increases the importance of forecast is deregulation of markets. In deregulated market, there is not a fixed price other than the individual contracts. Price of electricity varies from time to time and determined by market regulator based on the announced demand and supply values of the players in the market. Hence, this situation pushes the players to sell electricity in high price hours and consume electricity in low price hours [10]. To achieve this objective they need to forecast both supply and demand of electricity. In other words, short term electricity load forecasting is not only an issue to keep system safe and efficient but also an issue to make more revenue for stakeholders.

Forecasting the load in short term is a complex task. There are some factors those need to be considered such as temperature, humidity, time of year, month, week and day, behaviour of consumers, random factors and etc. [8]. Including more explanatory factors in the model will increase the forecast performance. However, including more factors may also cause a computational burden. After preprocessing the data of a specific system, forecast may be done by using the most influential factor in order to achieve a good forecast performance without spending computational effort for the less important factors.

There are several methods used in short term load forecasting. In earlier times, mainly conventional methods were applied such as linear regression and time series methods, in STLF. Holt Winter's(HW) exponential smoothing and auto-regressive integrated moving average(ARIMA) methods are the most common time series methods used in STLF [11]. Conventional algorithms are performing well in terms of computational time and resources. However, there are some disadvantages of these methods. First of all, conventional methods could not properly represent non-linear relations between load and the influencing factors. [12]. Moreover, a disadvantage of exponential smoothing methods is that they involve an initialization and updating many terms [11]. For ARIMA models, the order selection process is considered as subjective and difficult to apply [11]. Because of these disadvantages, different models which are capable of capture the nature of the load curves are required.

As from 1990s, new approaches have been deployed in load forecasting. artificial intelligence techniques such as neural network(ANN), fuzzy logic and expert systems have been applied to deal with the non-linearity [13]. The well-known machine learning methods used for STLF are artificial neural networks, support vector machines, decision trees, random forests, gradient booting methods and etc.

In this study, regression techniques from machine learning domain such as random forest, gradient boosting method(GBM), support vector machines(SVM) are applied. These methods are compared with three different naive methods, with several time series methods such as exponential smoothing, auto-regressive integrated moving average, trigonometric Box-Cox transformation auto-regressive moving average trend seasonality(tbats) method and with linear regression. Advantage of machine learning methods is that these methods could be well fitted to non-linear data which is a crucial attribute of models used for load forecasting. However, these methods are more complex methods than conventional methods. Hence, they have some disadvantages such as having lack of generalization ability [11] and higher computational requirements. The performance of these machine learning methods are also dependent on feature set. A feature set is tried to be designed in order to maximize the performance of these methods. Moreover, several modified versions of the feature sets are also analyzed with same machine learning regression methods in order to see whether these methods' performances could be improved or not.

The remaining part of this thesis is structured as follows. In Section 2, literature review is made. In Section 3, methodologies used in this study are explained. In Section 4, the data used in this study and the results of the methods are presented. Finally, in Section 5, conclusions and future works are presented.

2. LITERATURE REVIEW

The approaches for STLF could be categorized into conventional time series methods, machine learning tools and hybrids methods.

2.1. Time Series Methods

The approaches in this category are white-box models. Output of the model has a parametric relation with inputs. Some of the methods used in this field are:

Auto Regressive Moving Average(ARMA) was applied on a dataset from Helenic Power Corporation(Greece) covering from January 1, 2004 to December 31, 2005 [14]. Before applying ARMA they applied deseasonality on the dataset and then parameter selection. In order to estimate the parameters of ARMA model they applied some methods such as, AIC, BIC, AICC and especially multi-model partitioning filter. They use RMSE as performance criteria and reach more than 90 percent.

Auto Regressive Integrated Moving Average(ARIMA) has also been applied. In one of them [15], researchers used a 168 hours dataset from Taipower. They applied standard ARIMA and ARIMA with transfer function in order to include temperature as an affecting factor. ARIMA with transfer function gave better result and their result was 4% AAE on average. In another study [16] , researchers used different practical load data types of 2007 from Taipower and they applied standard ARIMA and ARIMA with lifting scheme. Their model with lifting scheme gave better results which is lower than 1% MAPE.

Exponential smoothing methodologies have also been applied. In one study [17], researchers applied exponential smoothing with double seasonality on two half-hourly load dataset, Rio de Janerio data from 5 May 1996 to Sunday, 18 May 1996 and England and Wales data from 27 March 2000 to Sunday, 9 April 2000. They compared exponential smoothing with some other methods and found exponential smoothing gave best result which is lower than 3% MAPE. In another study[8], researchers used minutely load data from National Grid of Great Britain from 2 April 2006 to 28 October 2006. They performed VSTLF in this study for 10 minutes and 30 minutes ahead load forecast but they also checked their model beyond this values. They uses double seasonal Holt and Winters method as candidate method and they compared this method with some other methods. Up to 4 hours their method gave best result but after 4 hour, company's own weather based forecast method gave the best result. In another study [18], researcher analyzed single, double and seasonality with Holt and Winters, ARMA and some other methods on half-hourly 6 year-dataset from both Great Britain and France from 2001 to 2006. He claimed that triple seasonality with HW and ARMA gives the best result which is lower than 2% MAPE.

In another study [19], researchers used Kalman filter. They applied this on a dataset from Canada's biggest utility company for 1994-1995 years. They took temperature, wind, previous hours load and their derivatives as variables. They applied Kalman Filter in order to adjust the coefficients of these variables. It seems, for some particular days, they could predict the load with around 1% deviations.

2.2. Regression Methods

Linear regression is a method that builds linear relation between regressors and response variable. It considered as white-box model because it clearly shows the relation between regressor and response variable. It is easy to implement and used widely also in STLF. In one study, multiple linear regression was applied on a dataset from Sulewesi Island(Indonesia) [20]. They included temperature and time of day information as input variables. They reached 4% MAPE on average.

Regression methods from machine learning domain have also been used in STLF. They have capability of fitting to nonlinear datasets better than conventional methods. However, these models are usually called as gray-box models or black-box models because the model usually does not provide or partially provide the information of relation between inputs and outputs. Some of the methods have been used in this field are:

Decision tree algorithm is one of the basic machine learning algorithm. It has some variants. In one study [21], CART(Classification and Regression Trees) version of it was applied to a one year hourly electricity consumption dataset from a power region in Poland. They used weather related factors, structure of day related factors and calendar related factors as input features. They used R square as performance measure and reached more than 0.7 R square values.

Random forest method was applied as an ensemble of CART classifiers on electricity load data of Polish Power System from the period of 2002-2004 [11]. They used two different months as test set. They reached 1.16 MAPE on average, however, they did not provide the information of which inputs they used for prediction.

Gradient boosting method was applied in STLF in several studies. In one study [22], component wise version of gradient boosting was applied in a competition. In component wise version, only one parameter is selected for each iteration. They used a dataset of hourly electricity consumption of 20 different zones from USA from from 11 weather stations, from the first hour of 1 January 2004 to the sixth hour of 30 June 2008. They used weather information, past consumption information and calendar information as input features. They did not provide a performance value but they mentioned that they were fifth out of 105 groups in a challenging worldwide Kaggle competition. In another study [23], gradient boosting was applied to hourly load of the ISO New England Control Area (ISO-NE CA) for a 4-year period, from 2009 to 2012. They generated features from past load data and add information like hour,day and month of year then had 23 features. Finally they reached 1.32 MAPE. In that study they also applied random forest and get 1.97 MAPE for it.

Support vector machines method was also applied in STLF in several studies. In one of them [24], support vector machines was applied to hourly electricity load data from Eastern Province in Saudi Arabia is about 6 years from 1 January 1992. They used past load data as input. After removing seasonality they applied support vector machines. Their performance measure was RMSE and get 0.0028 RMSE. In another one [25], support vector machines with varying types of kernels was applied to hourly electricity load data from Northeast Region of Brazil from 6/1/2001 to 10/3/2001. They used different models for each hour and at the end combine the 24 model to make a full day forecast. They used past days load data values as input to model. They reached MAPE values from 1.01 to 3.69 for different kernel types.

Another machine learning tool is artificial neural networks. In literature this method and its variants have been applied in several studies to forecast short term load. In one study [26], ANN was applied to a hourly electricity load data from Korea Electric Power Company which includes 1 month for each season. They used only the some of the previous load data as inputs. By using these, they reached about 2 percent relative error. In another study, ANN was applied to a one year hourly load dataset from Greek Power System of 1993. Previous load values, temperature related information and day of week information were used as inputs. Consequently, they reached 2.35 average error rate for one day ahead forecast.

In this thesis, the objective is to analyze the performances of several methods proposed in literature rather than proposing a new method. In most of the studies, one-day ahead forecasting is performed. However, in this study, two-day ahead forecasting is done due to practical needs. A feature list is used which is designed by being inspired from studies in literature. Finally, all methods are compared based on several performance metrics which are widely accepted in literature.

3. METHODOLOGY

Short term load forecasting, as in the literature, could be done by using different types of methods. Because electricity load data is a time series data, it could be analyzed by using time series methods. Moreover, electricity load is affected from different factors. Regression methods construct relation between a response variable and independent variables. Therefore, regression methods could also be used in STLF. Then the methods those construct relation between a response and independent variables could also be used.

3.1. Naive Methods

Naive methods are lean approaches mainly based on similarity of the time periods. The objective is to create a model which does not require complex modelling and computational effort. Basic idea is to forecast a particular future time period by using the previous data which are similar to the time period to be forecasted. Naive methods' accuracy could vary from application to application depending on the dataset size, recurrency of the data and complexity of the problem. Even the performance is not good enough in a study, their performance could be used as a lower limit in benchmarking of the other methods.

3.2. Time Series Methods

Short term electricity load data are time series type of data. They have a continuous nature and a pattern. Because of this structure, load data could be analyzed by conventional time series analysis methods. There are some generic and well known methods such as ARIMA, exponential smoothing, tbats etc.

3.2.1. Auto-Regressive Integrated Moving Average(ARIMA) With Seasonality

ARIMA is one of the well-known and most used time series methodology in statistics. It is a generalization of auto-regressive(AR) and moving average(MA) methods. Both of these methods are used to analyze time series for better understanding of underlying mechanism or to forecast the future values.

Auto-regressive part of the method aims to construct a regression model based on the past values of time series. Moving average part of the model aims to construct a regression model based on the previous error terms. These two part are works in a stationary data set. Integrated part(I) is applied to obtain a stationary data by applying differencing between particular instances and their previous instances when the data set does not show a stationary behaviour.

ARIMA models usually denoted by ARIMA(p,d,q). p denotes the number of previous values used in auto-regressive part, q denotes the number of previous error terms used in moving average part and d denotes how many number of times data is subtracted with past values. Seasonal ARIMA models are usually denoted as ARIMA(p,d,q)$(P,D,Q)_t$ where t denotes the number of period in a season, P,Q,D denotes the autoregressive, moving average and differencing terms of seasonal part.

3.2.2. Exponential Smoothing

Exponential smoothing is another well-known time series method in statistics. The main idea is to construct a model by using past values of the data. However, in exponential smoothing rather than giving equal weights to previous values like in simple moving average, exponentially decreasing weights over time are assigned to previous values. It is an easily implemented procedure based on user defined assumptions, such as seasonality.

3.2.3. Tbats

Tbats is another methodology [27] used in statistics. It is much newer than other conventional time series analysis methods, so, not widely used as much. This method is proposed especially for time series which includes complex seasonalities. It includes trigonometric regressors(T), Box-Cox transformation(B), ARMA errors(A), trend(T) and seasonality(S).

3.3. Regression Methods

Load forecast could be done by using conventional methods at some extend. However, due to the nature of the load curves, conventional methods' performances could not better than some degree. Thus, load prediction remains as a difficult task. There some reasons cause this situation. Firstly, load time series is strongly nonlinear and exhibits several levels of seasonality and periodicity [28]. Secondly there are many exogenous variables those must be considered and the relation among those variables and the load is usually difficult to model [29]. Because of these reasons, there is a need for models those could handle these situations. Machine learning methods come to the forefront at these points. Machine learning methods have the capability to explain non-linear and complex time-series. Moreover, machine learning methods could explain and utilize the complex relations between exogenous variables and response variables. Hence, machine learning methods are strong candidates to forecast short term load data. There are several generic regression methods such as linear regression, random forest, support vector machines, gradient boosting method, artificial neural networks and etc.

Among these methods, although they are used in the literature artificial neural networks methods have some drawbacks. Firstly, neural networks are completely black-box models. Hence, they do not provide any intuition about the relation between factors and response. Secondly, the neural networks require very large amount of data to ensure that the results are statistically accurate [30]. Thirdly, neural networks require great computational resources to train the model. Finally, neural networks are prone to overfit [31]. Hence, this may cause neural networks perform poorly on test data although it perform well in training data. Due to these drawbacks, artificial neural networks methods are not included in this study.

3.3.1. Linear Regression

Linear regression is an essential method in statistics and it could be a baseline for any performance comparison in any study. It has a simple structure, is easy to implement and provides information about the model. Another advantage of this method is that it does not require any parameter to adjust. Basic idea is to fit a linear model to the data and then make forecast by using that model. The model is fitted to the data by minimizing sum of square of errors between fitted values and real values. It also provides importance information about the regressors. Linear regression is computationally cheaper than other methods. However, linear regression tries to explain the relation between regressors and response variable linearly. Thus, in the case of non-linear relations, linear regression may not show good performance.

3.3.2. Tree Based Regression

3.3.2.1. Decision Trees. Decision tree algorithm is one of the basic machine learning method proposed in 1984 [32]. The goal of decision tree is to determine the influencing factors and how they influence the response variable. This relation between these factors and response variable is described in the form of decision rules. These rules are expressed in tree structure [21]. At every iteration, one factor which gives the best improvement is selected and split from the point at which the factor gives the best improvement. This improvement could be calculated by using different techniques such as information gain, variance reduction and etc. During these iterations a decision tree is constructed. Every particular instance to be forecasted is entered to the tree from the top, goes over the rule set until it reaches to a terminal node and a value is assigned to it. This method could be used for both classification and regression(Figure3.1). This method is the foundation of the tree based methods such as random forest, gradient boosted trees which will be used in this study.

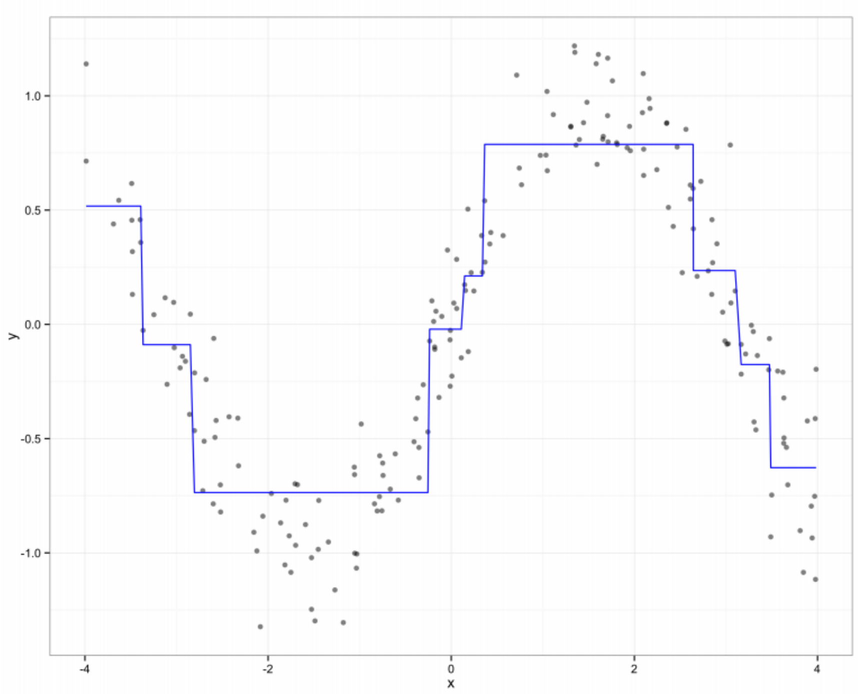

Figure 3.1. Approximation of the function using Decision Tree. The blue line displays the prediction from the tree [1].

<u>3.3.2.2. Random Forest.</u> Random forest is one of the most used and generic machine learning methods, which is in class of ensemble learners. Basically, random forest is a collection of decision trees. Each tree is trained on a randomly selected observation set and each split is made based on a randomly selected predictor set. This is the reason why this method is called as random. Different learners are based on different observations and different predictors, hence all models are different from each other. To forecast a new instance, all models produce a prediction and the final predictions is made as average of the results (Figure 3.2, Figure 3.3). This method is useful because it prevents the system from over-fitting. Moreover, it could handle the categorical input data. However, there are important parameters to define from user such as, number of trees, number of observation at each node, number of features for each tree and etc..

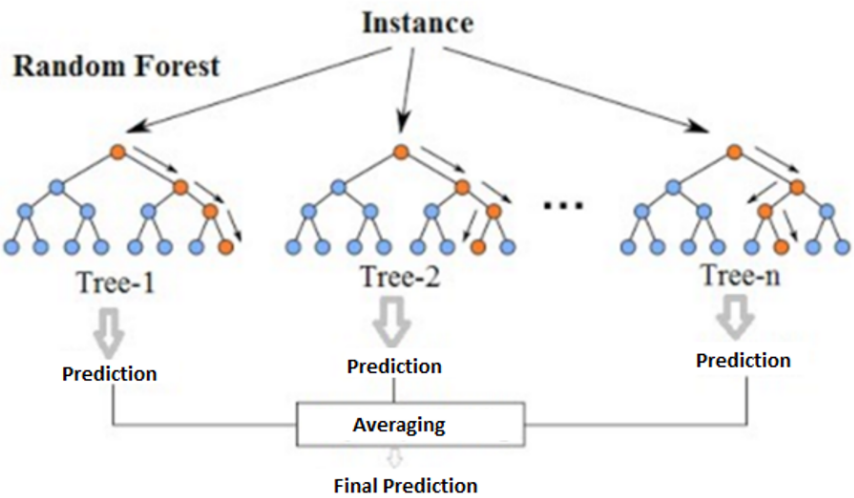

Figure 3.2. Simple random forest model [2]

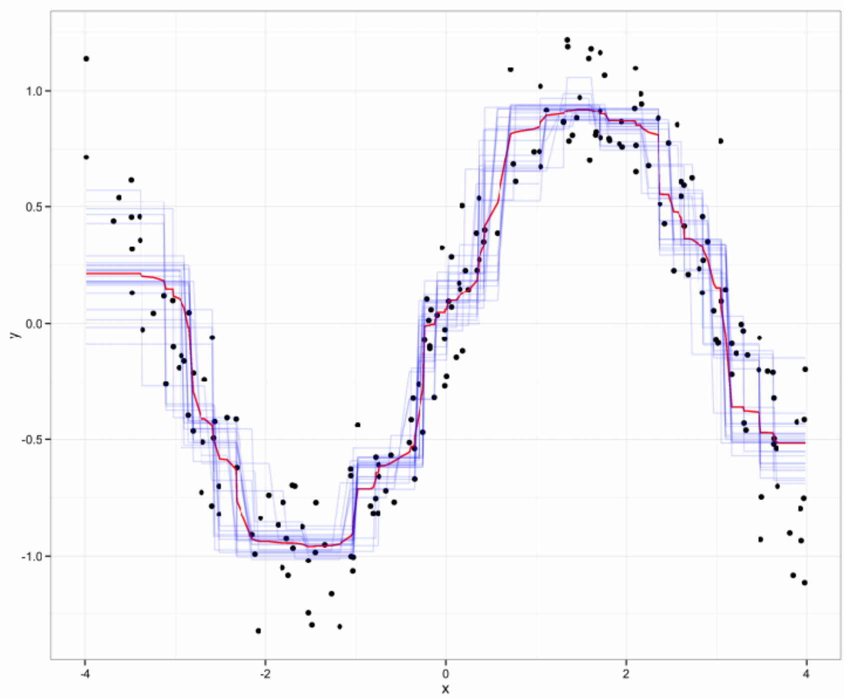

Figure 3.3. 25 randomly selected trees (shown in blue) in a Random Forest (prediction shown in red) [1].

Number of trees parameter determines that how many tree will be constructed. Each tree is a different learner which is constructed by using different part of the training set and different set of feature list. For a specific instance, all of the trees are making a prediction and final prediction will be done by combination of those predictions. Number of observation parameter determines the minimum size of the terminal nodes. A node could not be split more, if the resulting splitted node will be lower than this value. Larger node size parameter values will result smaller trees and less computational time. Setting this value smaller will result larger trees and higher computational times. Additionally, setting node size small may cause over-fitting. Number of features for each tree parameter determines that how many feature will be selected for each different trees. Those features are selected randomly and prediction of a particular tree will be done based on only selected features.

3.3.2.3. Gradient Boosting Method. Gradient boosting method is another most common model type in machine learning literature and It is one of the most used technique. In international competitions gradient boosting is usually included in the models. It is also an ensemble model, which constituted from different weak learners (trees). In random forest, trees are linked parallel and the final decisions are made based on the average results of trees. These learners are independent learners. However, in boosting, trees are linked sequentially (Figure 3.4). All trees are learning based on the error of the previous trees (Figure 3.5, Figure 3.6, Figure 3.7). Therefore, models are not independent and the final prediction is made by summing the results of all of the trees. Since there can be over-fitting, user should define carefully the stopping conditions. If decision trees are used as learners, it could also handle the categorical data. There are important parameters those need to be adjusted such as, number of trees, learning rate, interaction depth etc.

Number of trees parameter determines that how many tree will be included in the model. Each tree is constructed consecutively in order to model the residuals from the previous cumulative learners. It could also be considered as iteration number in function of additive expansion. Learning rate parameter determines the weight of a newly added learner to the model. Learning too fast could cause over-fitting then it is better to slow down the learning process. Every new tree model the residuals, however, they are added to the model after being multiplied by this weight. Larger values of learning rate cause over-fitting but takes less time. Smaller values of learning rate

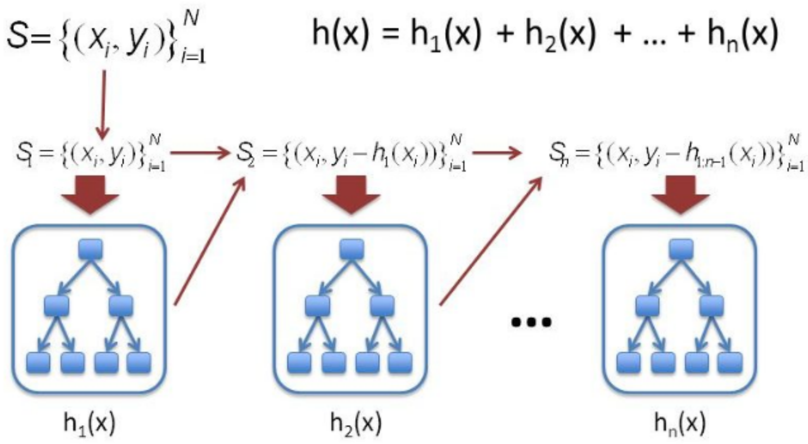

Figure 3.4. Simple Gradient Boosting Model [3]

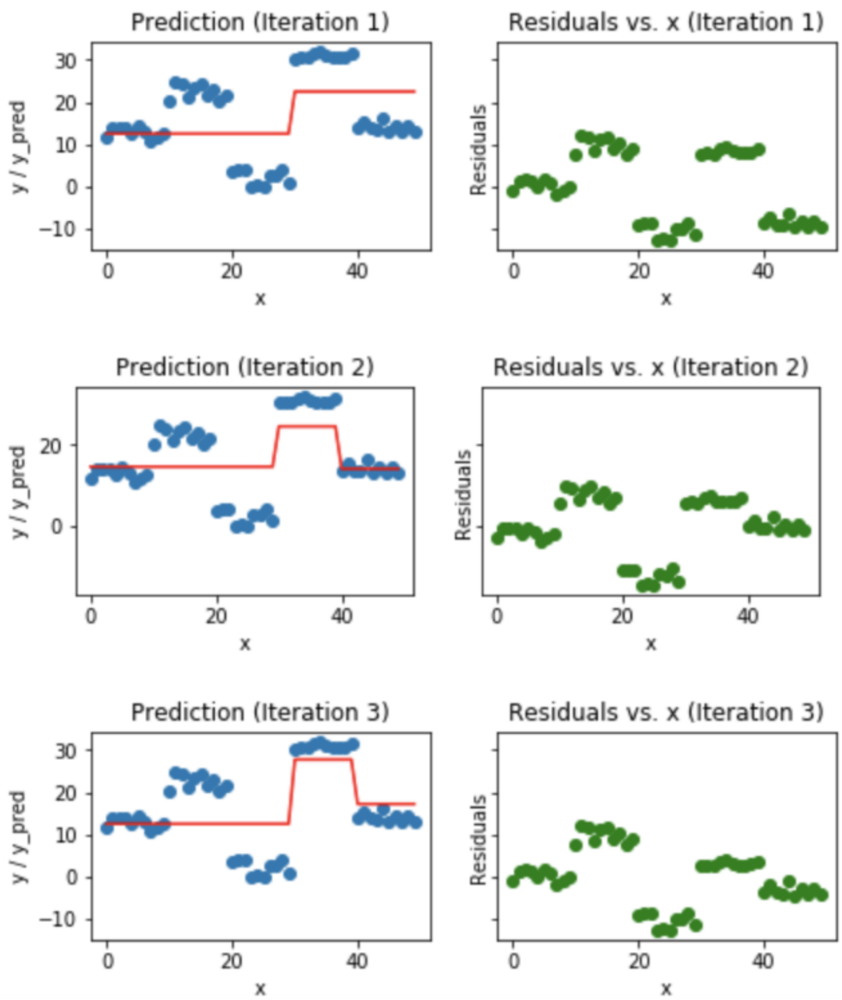

Figure 3.5. A Gradient Boosting Model Iterations 1-3 [4]

could prevent from over-fitting and construct the model in a smooth way, however, takes more time. Interaction depth is the number of splits that has to be performed

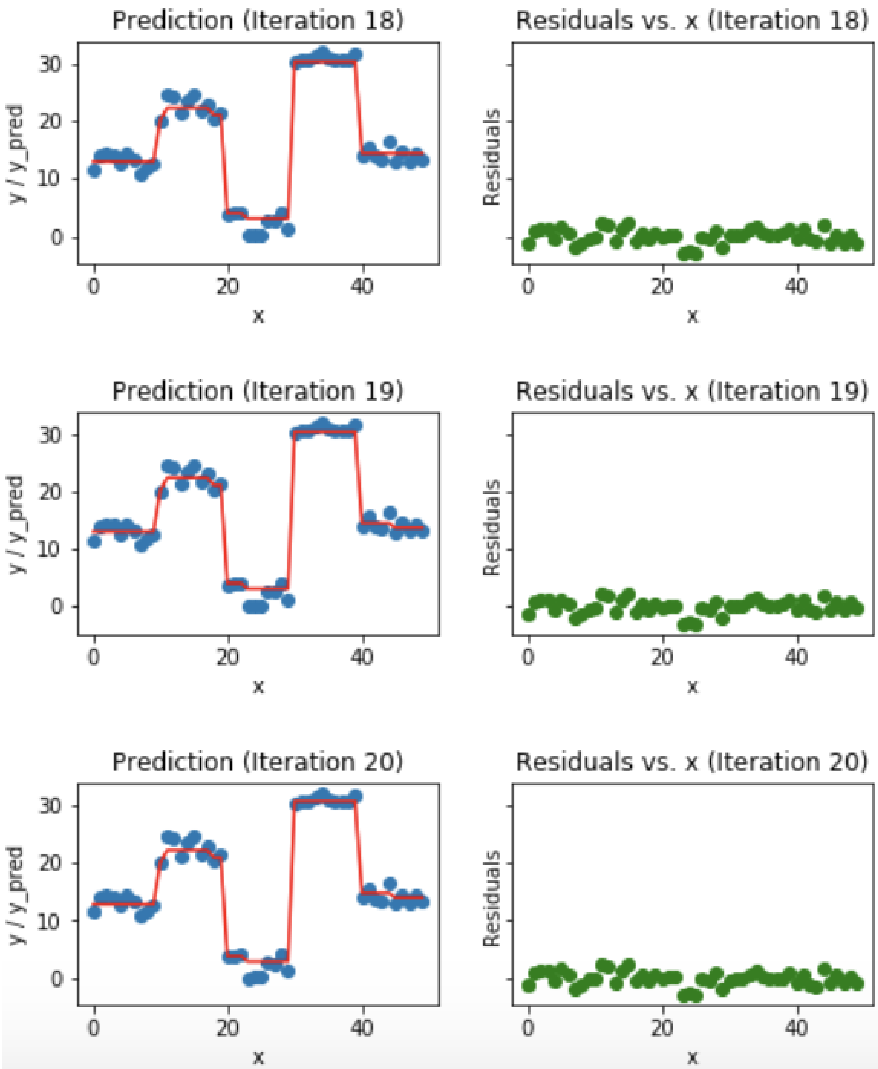

Figure 3.6. A Gradient Boosting Model Iterations 18-20 [4]

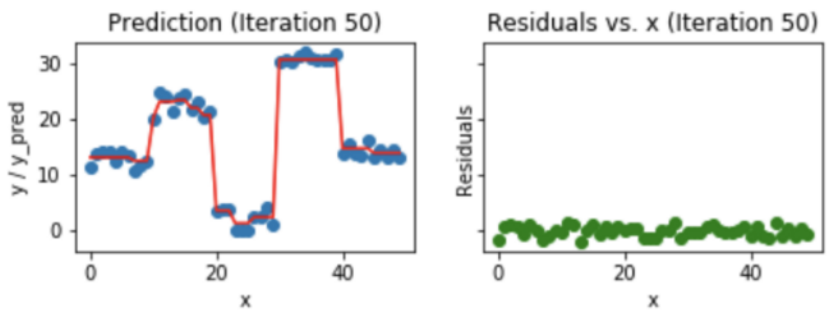

Figure 3.7. A Gradient Boosting Model Iterations 50 [4]

on each tree in gradient boosting method. It also determines the maximum number of variables will be interacted during these splits. When the depth is one, on each tree only one split is performed and only one variable is included. When the depth is greater than one, then on each tree more than one split is performed. These splits does not have to be on the same variable. The interaction between variables could also be considered. It means, in these splits different variables could be combined. For example, if interaction depth is three then interactions of variables up to three are considered. In practical applications small trees are provide considerably accurate results [33].

3.3.3. Support Vector Machines

The non-tree based machine learning method used in this study is support vector machine which is also quite common. In this method, the objective is to find a function which tries to deviate from all data points at most ϵ and be flat as much as possible(Figure 3.9). Support vector machines perform this task by solving an optimization problem. This problem's objective function consists of cost of deviation between predicted values and real value of a data point and cost of being non-flat that comes from adding more support vectors(Equation 3.1). Support vectors are the data points which at the borders of acceptable region. Adding more of them is costly because they harm linearity. However, adding less support vectors will increase errors. Algorithm optimizes these costs and tries to find the best model(Figure 3.8). Support vector machines usually used with kernel functions which provide great power to support vector machines. Support vector machines requires numerical data to produce a model. Hence, it is required to convert categorical data to numerical values somehow. There are important parameters need to be adjusted such as, kernel type, tolerance, epsilon etc.

Kernel is a set of mathematical functions. It is widely used in machine learning especially in support vector machines. Kernel function takes data as input and transform it into required and useful form. Practically it transforms nonlinear spaces into linear spaces. In other words, one could say kernel function is ısed to produce new feature dimensions bu using existing dimensions. Hence, data could become separable by using linear methods. There are types of kernel functions those apply different methods to transform data such as polynomial kernel, gaussian kernel, radial basis kernel etc. Depending on the dataset, kernel types parameter could influence the performance of the model. Tolerance parameter is another parameter in support vector machine which determines the the tolerance for stopping conditions. Taking this parameter as high values may result faster but worse performance. Taking this parameter as lower values may result better performance but may take long times to terminate the process. Lastly, epsilon parameter is another important parameter. This parameter determines that the errors to be treated as zero. If an error term between observed value and predicted value for a particular instance is lower than this epsilon value, it is assumed as zero. In other words, it determines the size of the loss-insensitive zone. Setting this value too high results poor performing models. On the other hand, setting this value too small force model to add more support vectors to not produce errors then it will cause over-fitting. Hence, epsilon must be selected to reflect data in some way.

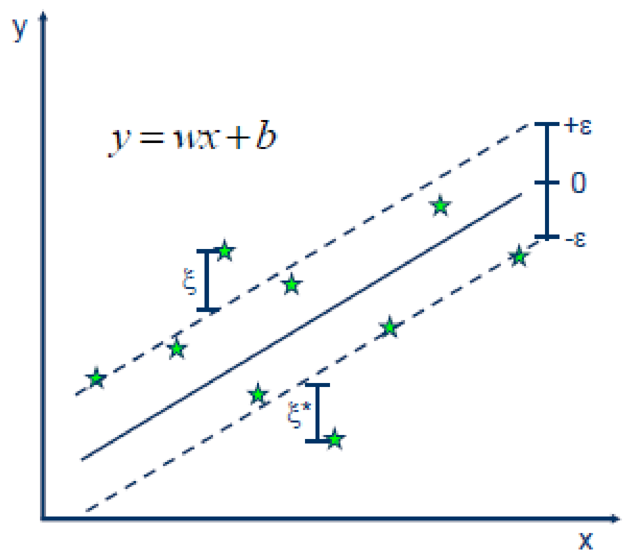

Figure 3.8. Support Vector Regression Model [5]

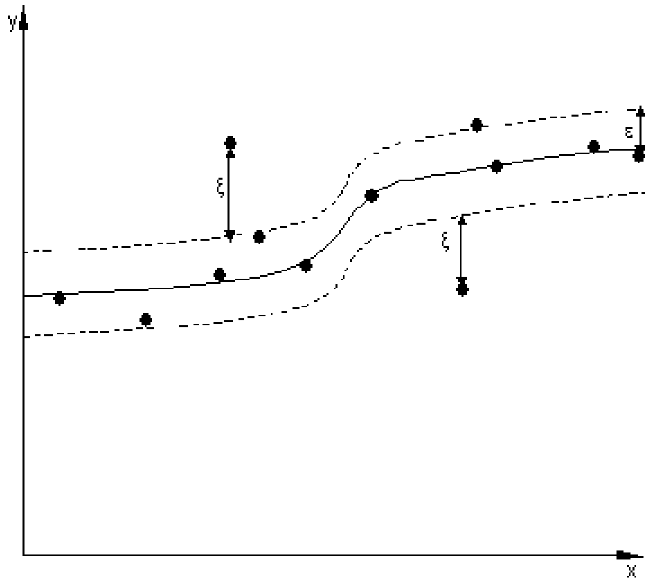

Figure 3.9. Nonlinear support vector reggression [6]

$$\begin{aligned} &Minimize: \\ &\frac{1}{2}\|w\|^2 + C\sum_{i=1}^{N}(\xi_i + \xi_i^*) \\ &Constraints: \\ &y_i - wx_i - b \le \varepsilon + \xi_i \\ &wx_i + b - y_i \le \varepsilon + \xi_i^* \\ &\xi_i, \xi_i^* \ge 0 \end{aligned} \tag{3.1}$$

4. EXPERIMENTS

4.1. Data

In order to make analysis, hourly electricity consumption data of a region of a country is used. The data starts from January 1, 2015 and end at 13 July, 2017. The data includes hourly temperature values of cities in that region and total hourly consumption of those cities.

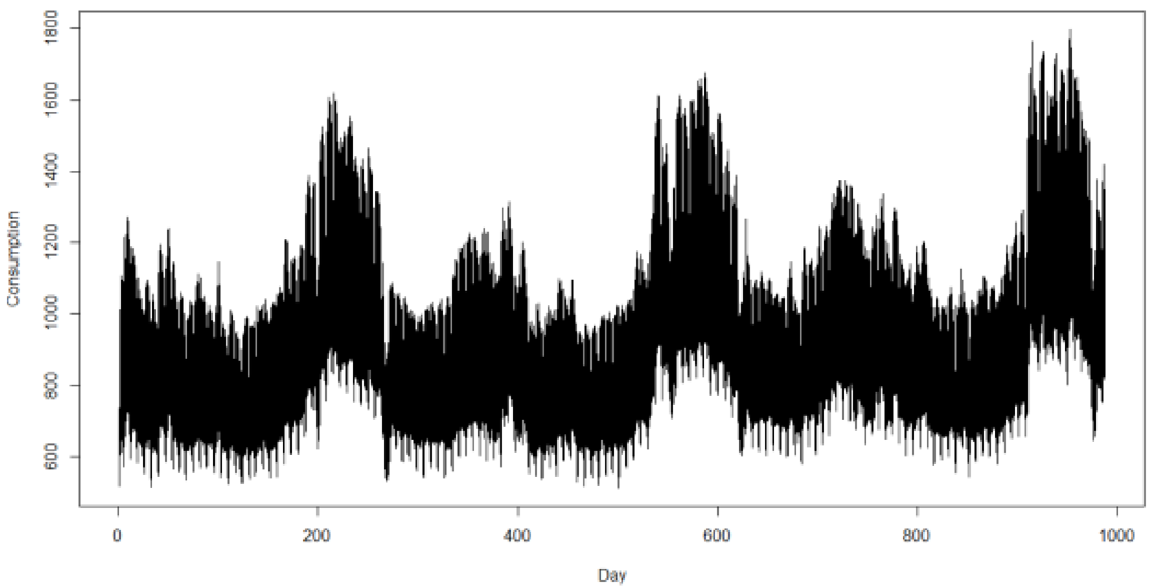

Figure 4.1. Electricity load data of complete horizon

4.1.1. Data Preprocess

There are some missing or wrong values in the data. Some of them are due to the yearly time adjustments. When the time was adjusted one hour forward, a zero value is assigned to skipped hour. When the time was adjusted one hour backward, then because that hour repeated in the system, two hours consumption amount is assigned to that hour. In these types of cases, the wrong values were replaced by an approximate value which is decided according to the consumption values of the previous and next hours and the consumption pattern of that day of week.

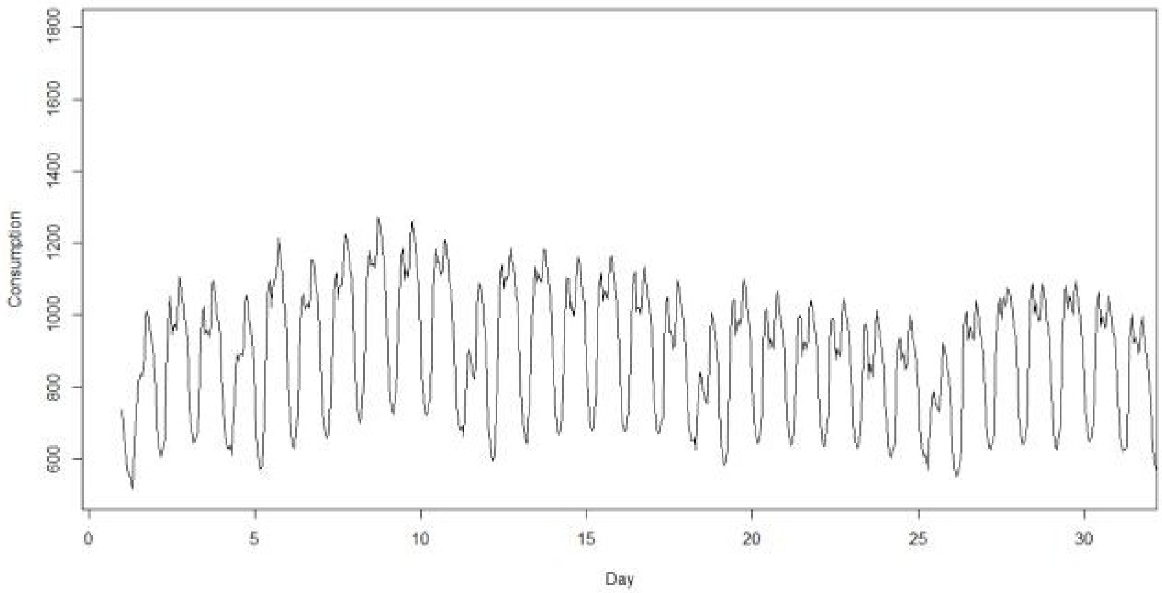

Figure 4.2. Monthly electricity load data(January,2015)

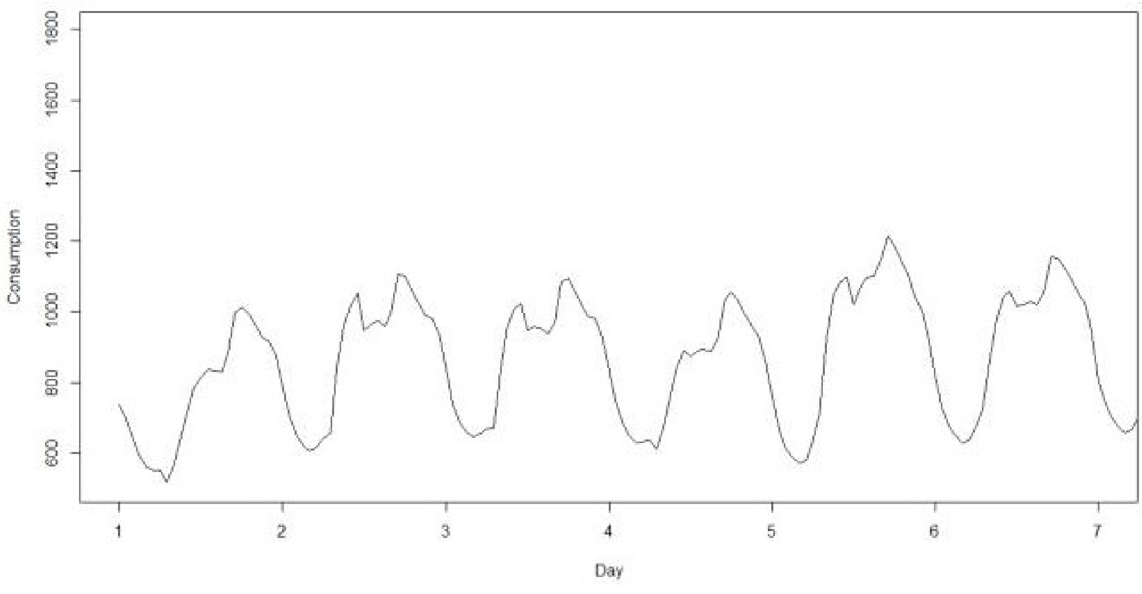

Figure 4.3. Weekly electricity load data(January 1-7 ,2015)

There are also some wrong values due to the electricity shortages. In this type of cases, wrong values are replaced by new values according to pattern of the same day of week which is adjusted to that day's expected consumption level.

There are 3 temperature values for 3 different cities in the region. Populations of the cities are close enough to take average of them. Then their temperature values are averaged an done temperature value is used.

Lastly, there are some features which are calculated by using data itself such as lagged consumption, lagged temperature, weekly differences etc. In order to calculate those values, previous data points are used. Hence, those features do not exist for the first instances. For example, for first 6 instances, any value is not assigned to 6 hours lagged temperature feature. In regression part, because some of the methods could not handle missing values, first two weeks of the data is not included in training set in order to make all instances have values for all features.

4.2. Methodology

In this study, main objective is to whether short term load forecasting could be done by using machine learning methods or not and if it could, what are their performances. Hence, the machine learning methods such as random forest, support vector machines and gradient boosting methods are applied to electricity load data of the region. Moreover, conventional time series methods such as ARIMA, exponential smoothing and tbats, linear regression and some naive methods are also applied. Main motivation to apply these methods is to compare their performances with machine learning method in order to see whether machine learning methods provide better performance values compared to conventional methods or naive methods or not.

For almost all methods, there are parameters to be set to particular appropriate values. Some of the methods are requiring more number of parameters and some of those require less number of parameters. For example, machine learning methods usually require more parameters, however linear regression do not require any parameter. Because of the cost of parameter tuning phase, for every method, 3 or 4 important parameters are selected according to their impact on results based on literature. In this study, there is not any method require more than 4 parameters which affect the results significantly. Parameter selection and tuning are made for those selected3 or 4 parameters at most for every method.

In order to decide the parameter values for every method, firstly the default parameter values are accepted as initial parameter values or the parameter values those come from automatic adjusting functions of particular methods are accepted as initial parameter values. Then, parameter tuning procedure is applied for every method.

There are some ways to tune parameters of a method in literature such as brute-force tuning, heuristic tuning, random hyper parameter optimization [34], Bayesian optimization [35] and etc. In this study, the proposed approaches do not require tuning of large number of parameters, therefore, brute-force parameter optimization method is used. This method evaluates each parameter combination on subsamples of dataset. It means that if there are m parameters and k parameter levels for each of them than there are k^m groups for testing. This search could also be done by n-fold cross validation, hence at that time there are nk^m runs are needed to tune the parameters. In this study, 10-fold cross validation is applied. Parameter tuning processes could be found in Appendix.

In practice, companies are trying to make forecast for tomorrow without having the information of today yet. Hence, forecast of tomorrow is done by using information up to yesterday, in other words, two-days ahead forecast is done. Because of this reason, in this study, proposed models are tested and compared based on two-day ahead forecast. All methods are trained by using appropriate sample of dataset for each of them. Performances are calculated based on the test results of the last year. Models are compared based on their average, hourly, daily and monthly performances. Finally, based on the results of the tests, additional approaches are also tested in order to see whether performances of the proposed methods could be improved or not.

The performance metric used in this study is mean absolute percentage error (MAPE). MAPE is the absolute percentage deviation from observed value(Equation 4.1). Usage of MAPE is become a convention in STLF. In addition tho MAPE, bias of the methods are analyzed, too. Bias is the percentage deviation from observed value(Equation 4.2).

$$MAPE = \frac{100}{n} \sum_{i=1}^{n} \frac{|\ Observed\ Value_i - Predicted\ Value_i\ |}{Observed\ Value_i} \tag{4.1}$$

$$Bias = \frac{100}{n} \sum_{i=1}^{n} \frac{Predicted\ Value_i - Observed\ Value_i}{Observed\ Value_i} \tag{4.2}$$

4.2.1. Naive Methods

In this study alternative naive methods are proposed. In these methods, basic idea is to assign a forecast value for a particular instance by using the data of the most similar observation in past. Three alternative naive methods have been applied.

In the first one, the observations which have same day feature are selected. Then, the observations which have same hour feature are selected from those. At the last stage, the energy consumption of the observation which has smallest temperature difference from the instance to be forecasted is assigned as forecasted energy consumption value to that instance. In naive methods in this study, the objective is to find the most similar past data and assign it as forecast. In the first naive method, this similarity is tried to be caught by using same day and same hour information first. Because, every day of a week has more or less different characteristic. Moreover, every hours in a day also has different characteristic. Because of these two strong characteristics, in the first naive method, similarity is tried to be caught first by using these two information. Then, at later step, temperature is used as final decision criteria because electricity shows a strong relation with temperature.

In the second one, the past data of only the last last weeks are used. Because closer dates' curves show similar average consumption level, this approach is expected

Algorithm 1

1: Choose the data points which have same day feature

2: Choose the data points which have same hour feature

3: Choose the data point which have smallest temperature difference and assign its value as forecast

Figure 4.4. Algorithm of Naive Method 1

to help to capture the seasonal effect on load curve. Hence, in the second naive method, temporal proximity is emphasized. The observations which have same day feature are selected. Then, the observation which have same hour feature are selected from those. Then the observations from the last l weeks are selected(depends on the run, could be last 4, 6, 8 or 12 weeks). At the last stage, the energy consumption of the observation which has smallest temperature difference from the instance to be forecasted is assigned as forecasted energy consumption value to that instance.

Algorithm 2

1: Choose the data points which have same day feature

2: Choose the data points which have same hour feature

3: Choose the data points of last l weeks

4: Choose the data point which have smallest temperature difference and assign its value as forecast

Figure 4.5. Algorithm of Naive Method 2

In the third one, temperature is emphasized rather than temporal proximity. Firstly, the observations which have same day an hour feature are selected. Then the observations which have temperature difference of less than 0.4 degrees are grouped then the consumption value of the most closest one is assigned as forecast. If there is not such an observation then the consumption value of the observation in last 12 weeks which has the smallest temperature difference from the instance to be forecasted is assigned as forecast.

The average MAPE values are 10.51 for the first one, around 6.52 for the second one and 10.29 for the last one. First two year is used as training set and the last year instances are used as test set for this analysis.

Algorithm 3

1: Choose the data points which have same day feature
2: Choose the data points which have same hour feature
3: **if** Any observation which have temperature difference lower than 0.4 degree **then**
4: Assign the consumption value of the observation which has smallest temperature difference
5: **else**
6: Choose the data point which have smallest temperature difference in last 12 weeks and assign its value as forecast
7: **end if**

Figure 4.6. Algorithm of Naive Method 3

According to results, Naive 1 and Naive 2 methods are performing similarly but Naive 2 method is performing better than them in terms of days (Figure 4.7). All methods are performing better on Wednesdays and Thursdays, performing worse on weekends and Mondays. In terms of bias, Naive 2 method has a bias close to 0%. Naive 1 and Naive 3 methods have negative bias(Figure 4.8). It means that, these methods are predicting lower than the actual consumption on the average. The reason is probably the long run increasing trend of consumption. Newer dates have higher consumption values than the older ones. However, in Naive methods, forecast is made based on past data points. Then, this situation cause assigning lower values as forecast. and so negative bias. One important point to remark here is that, Naive 2 method performs fairly good in terms of bias. It means that, increasing the importance of temporal proximity approach to capture trend is worked.

In analyzing naive methods' performances based on hours, it is seen that Naive 2 method gives the best performance again(Figure 4.9). One point to remark here is that, all models are performing worse at peak hours. In terms of bias, Naive 2 method performed better than the other naive methods again. All methods have negative bias(Figure 4.10). The bias is more negative during morning hours and more positive

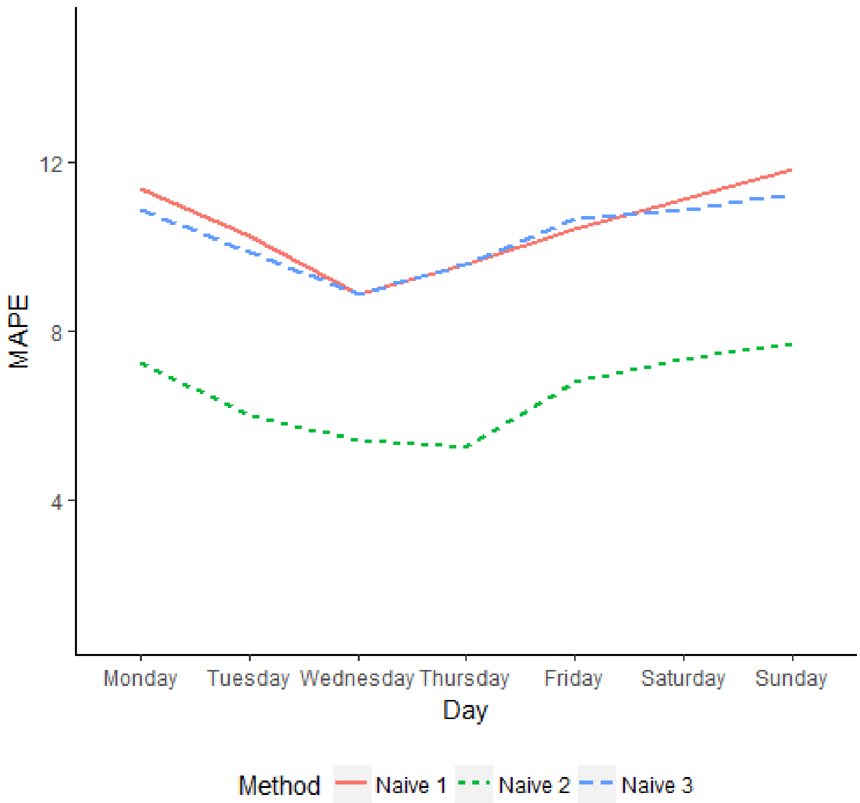

Figure 4.7. Average MAPE Results For Naive Methods Based on Days

during evening hours.

Finally, when the performances of the naive models are analyzed, it is seen that all models are performing better in April and August, performing worse in June and September(Figure4.11). This result is probably because the religious holidays. In the test year, religious holidays are in June and September. Relatively higher MAPE values of those holidays worsening the performances of the models in those months. Moreover, the last year is used as test set. Because, the last year data is up to middle of September, the months after September could not be analyzed.

In terms of bias, Naive 1 and Naive 3 methods shows similar patterns. Naive 2 shows perform differently than others. Naive 1 and Naive 2 methods show negative bias in first months of the year but Naive 2 methods shows positive bias. Naive 1 and Naive 3 methods show positive bias in May and June but Naive 2 shows negative bias. However, average bias values based on days are around zero and does not show

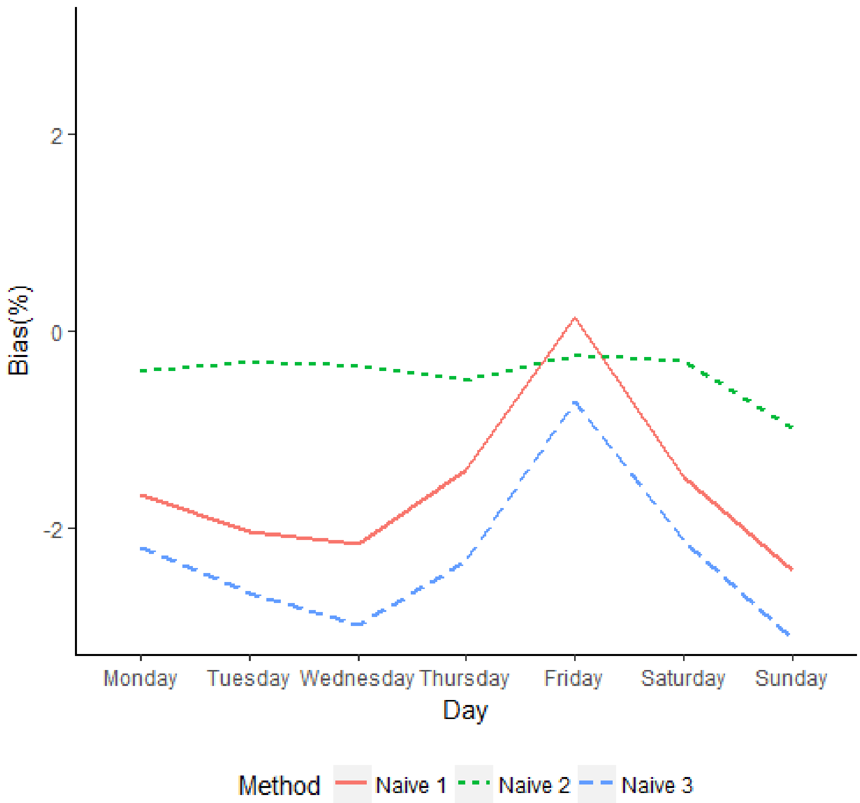

Figure 4.8. Average Bias Results For Naive Methods Based on Days

a meaningful pattern to extract(Figure 4.12).

4.2.2. Time Series Methods

In these approaches two weeks of data before the forecasted day are used as training set. While the day to be forecasted is shifted ahead, the training set is also shifted one day ahead(Figure 4.13). The reason to use two weeks of data is that, two week of data carry most of the information those are needed by a time series method, such as daily pattern, weekly pattern, trend between two weeks, average consumption level of that time of year and etc..

Time series methods give more weights to recent observations, too previous data become less important. Rather than training model by using all data, training the model by using only two weeks provide computational efficiency.

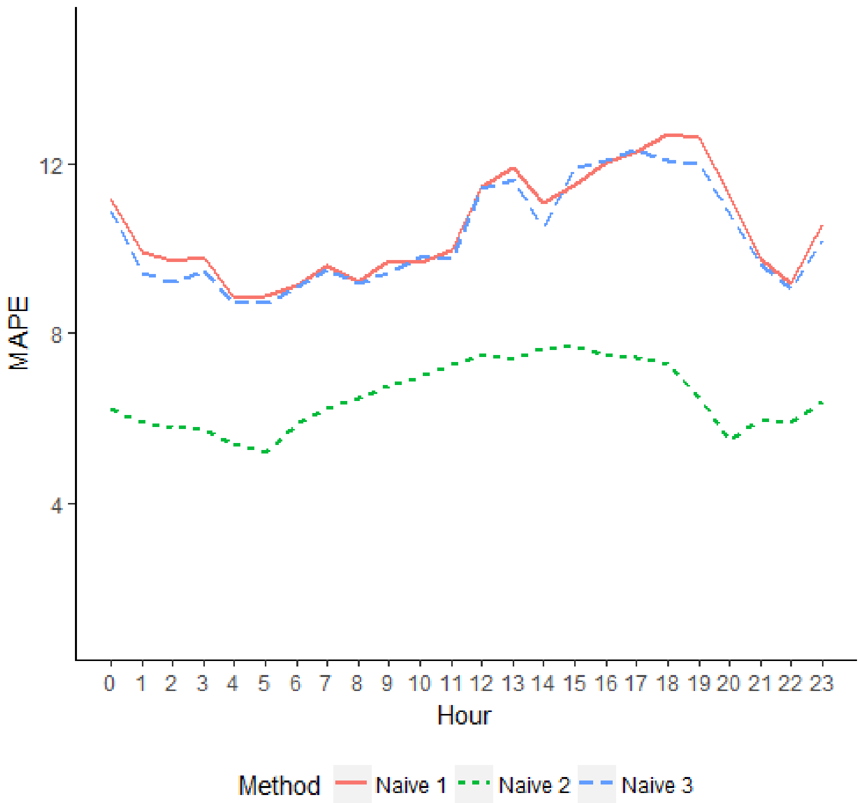

Figure 4.9. Average MAPE Results For Naive Methods Based on Hours

<u>4.2.2.1. ARIMA.</u> In this study ARIMA with seasonality is applied in three different way.

Firstly, seasonal ARIMA model is trained by a two week data and requested to forecast second day instances after training period. ARIMA model with parameters (p,d,q,P,D,Q) = (0,0,0,2,0,2) is constructed for this dataset. By using these parameters and two weeks of data as training set, the average MAPE for last year forecast is 6.82.

Secondly, seasonal ARIMA model is trained by using complete past data as training set to forecast a particular day. It means if we try to forecast the load of 500. day, then we are using past 498 days for forecast. In this model, first two year is used as initial training set and forecasts starts from second day of third year. ARIMA model with parameters (p,d,q,P,D,Q) = (0,0,2,2,0,2) is constructed for this dataset. By using these parameters and whole past data of each particular as training set, the average MAPE for last year forecast is 7.20.

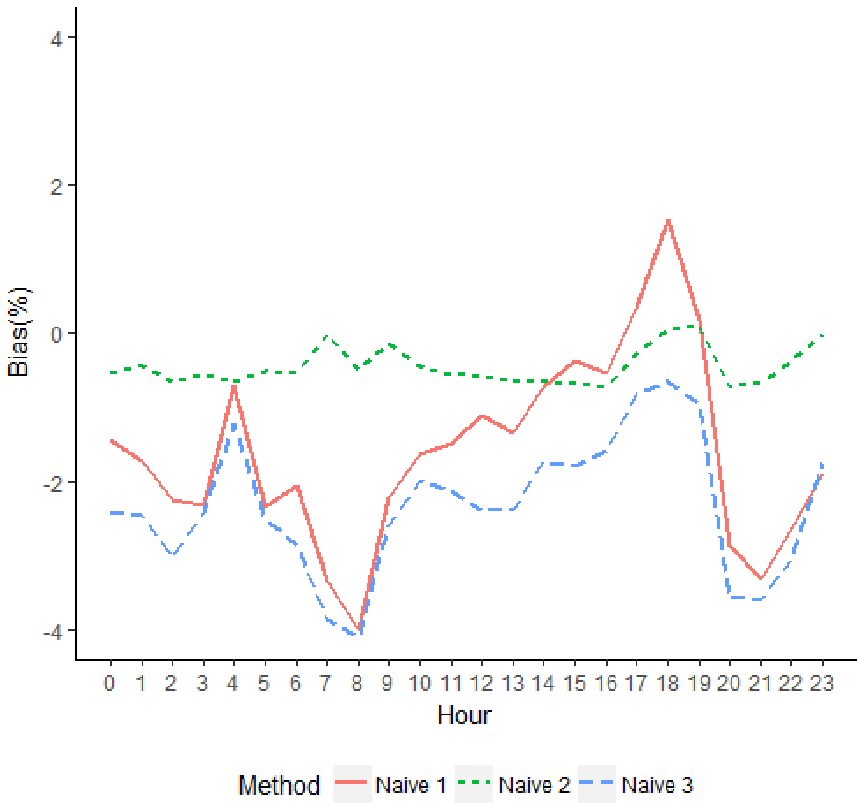

Figure 4.10. Average Bias Results For Naive Methods Based on Hours

Lastly, seasonal ARIMA model is trained by including temperature as regressor. Three separate temperature data of three different city of the region is added to the model. Training period is selected as two weeks and the same parameter set is used which is decided for the second version of ARIMA model. By using these parameters and two weeks of data as training set, the average MAPE for last year forecast is 6.81.

According to seasonal ARIMA results, there is no significant performance difference between seasonal ARIMA model trained with two weeks of data and seasonal ARIMA model trained with whole past data. These results prove that using only two weeks of data for training the model for the sake of computational time is a valid approach.

4.2.2.2. Exponential Smoothing. For exponential smoothing model, two weeks of data are used for training to forecast each particular instances. Exponential smoothing

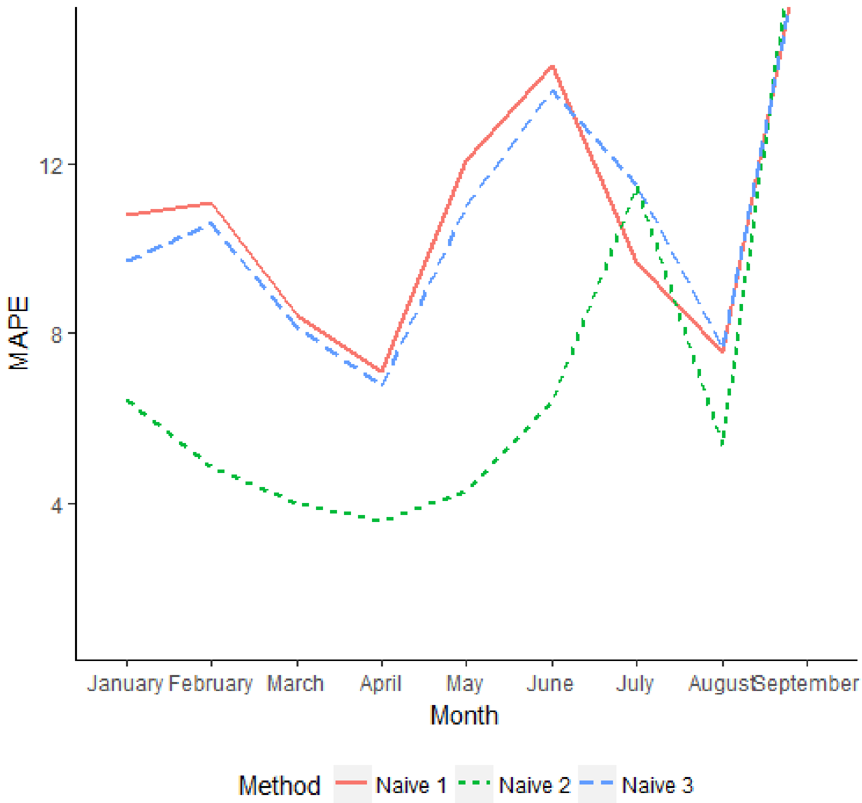

Figure 4.11. Average MAPE Results For Naive Methods Based on Months

method with parameters (Alpha, Beta) = (0.8, 0.01) is constructed for this dataset. By using these parameters and two weeks of data as training set, the average MAPE for last year forecast is 6.61.

<u>4.2.2.3. Tbats.</u> In Tbats method any parameter selection process is not applied because tbats function tries all possible options for parameters. Hence, the parameter selection process is left to function itself. For tbats model, two weeks of data are used for training. By this way whole data is forecasted except the first 15 days. By using these parameters and two weeks of data as training set, the average MAPE for last year forecast is 6.73.

According to results based on days, it is seen that models are performing more or less similarly(Figure 4.15). All models are performing worse on Tuesdays and Sundays. For Sundays, the reason is probably the daily load difference of Sundays. Sundays show

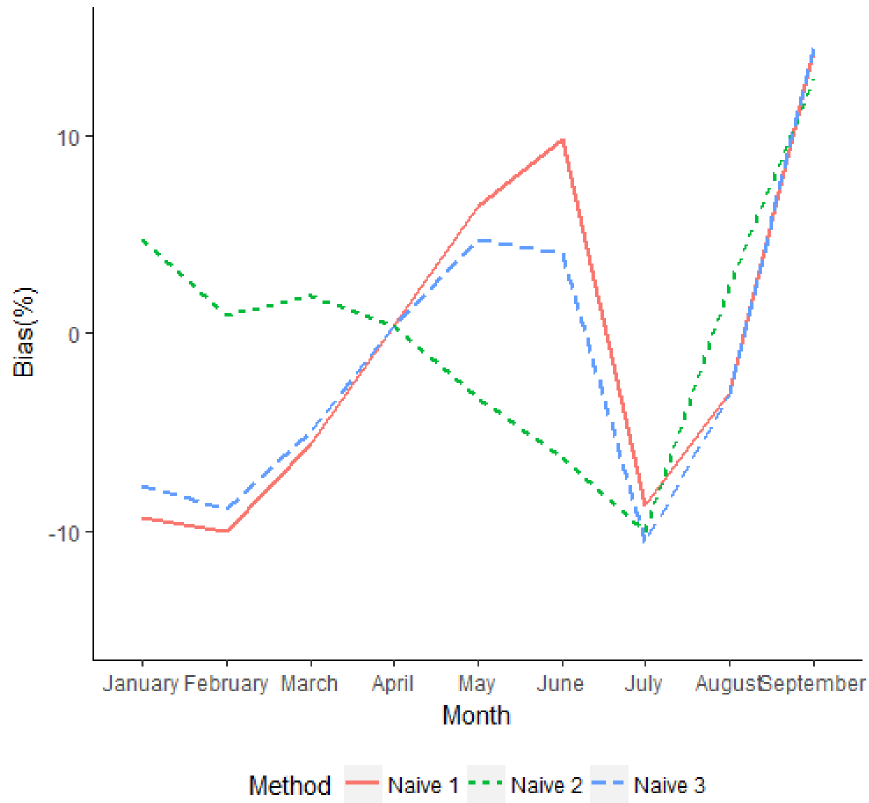

Figure 4.12. Average Bias Results For Naive Methods Based on Months

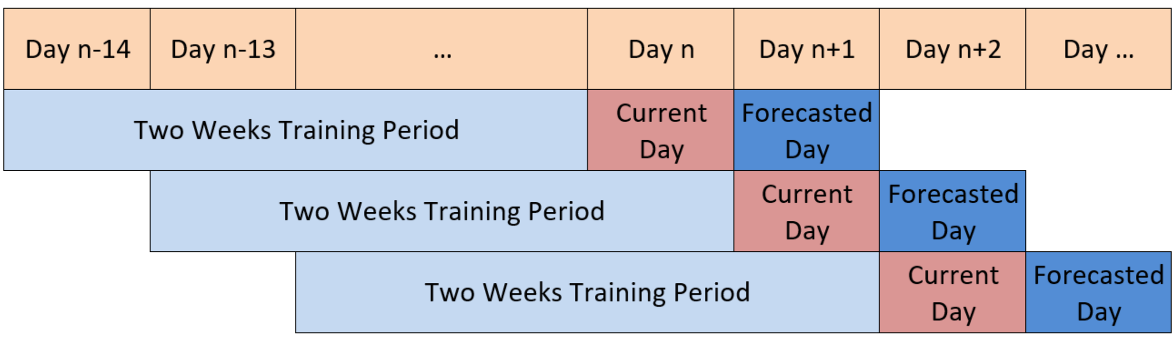

Figure 4.13. Two Weeks Training and Test Process

the most different load curve comparing to other days(Figure 4.14). Hence, predicting Sundays by using previous days data may cause problem. For Tuesdays, the reason is still the curve of Sundays. Because, Tuesdays forecast are made on Sundays. Hence, when models are forecasting, the most recent and important data are Sundays' data. Then the models show worse performance also on Tuesdays. In terms of bias, it is seen that all models are performing similarly(Figure 4.16). It is worth to mention that, on Tuesdays all models show negative bias and on Sundays all models shows positive bias. It is probably because of the load curve difference of Sundays'. Sundays load level is lower than the other days, however, while forecasting Sunday the most recent data is Thursday's data. Hence, forecasting a low demand day from a high demand day causes a positive bias. For Tuesdays, the situations is vice versa. Forecasting from a high demand day from a low demand day(Sunday) causes a negative bias.

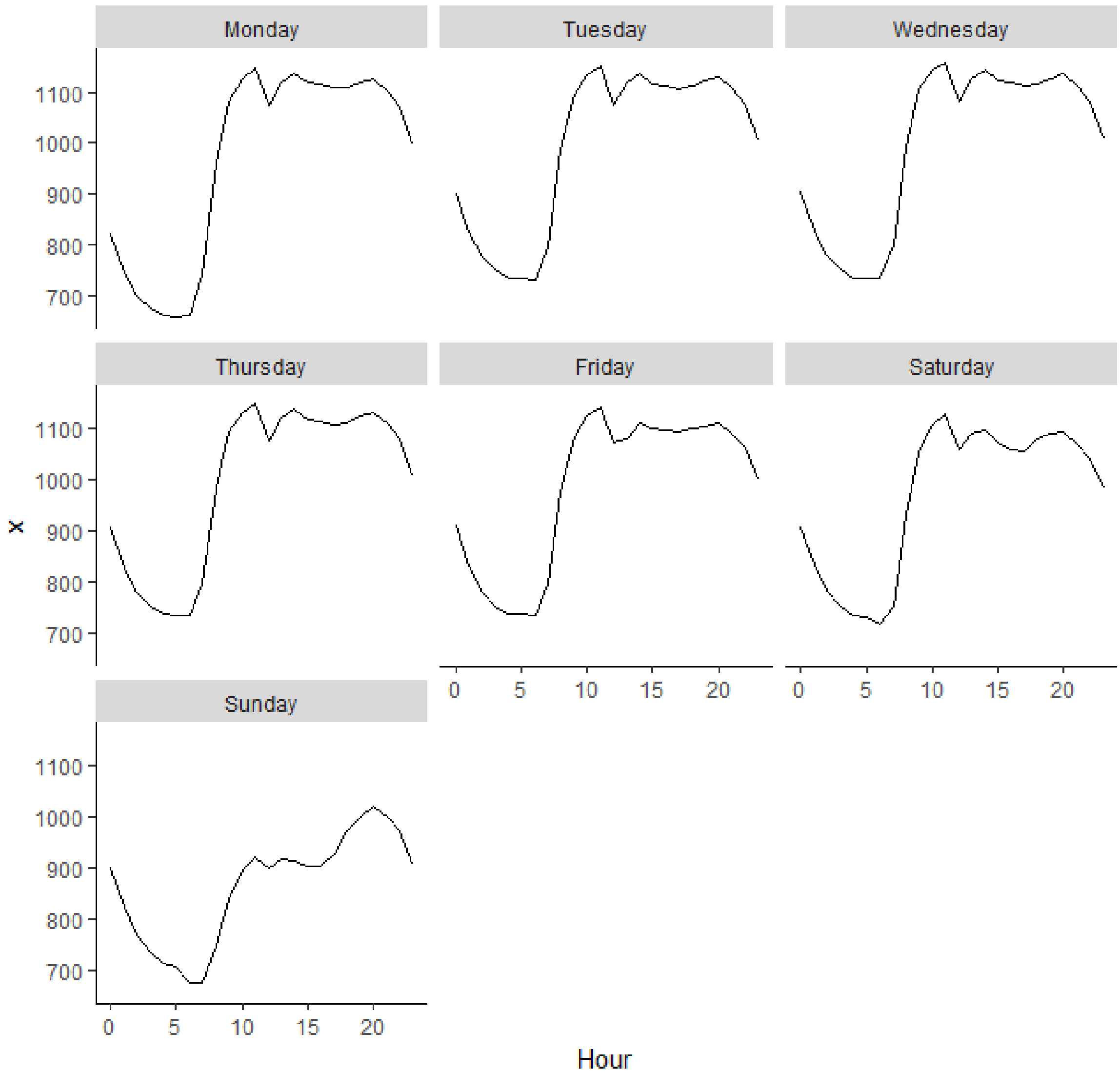

Figure 4.14. Average Daily Loads

In analyzing the model performances based on hours, it is seen that all models show similar performance(Figure 4.17). It is remarkable that time series models are performing worse at the hours when the consumption values are high. In terms of bias, it is seen that all models have positive bias based on hours(Figure 4.18). However, especially in the first half of the day ARIMA models show higher positive bias but

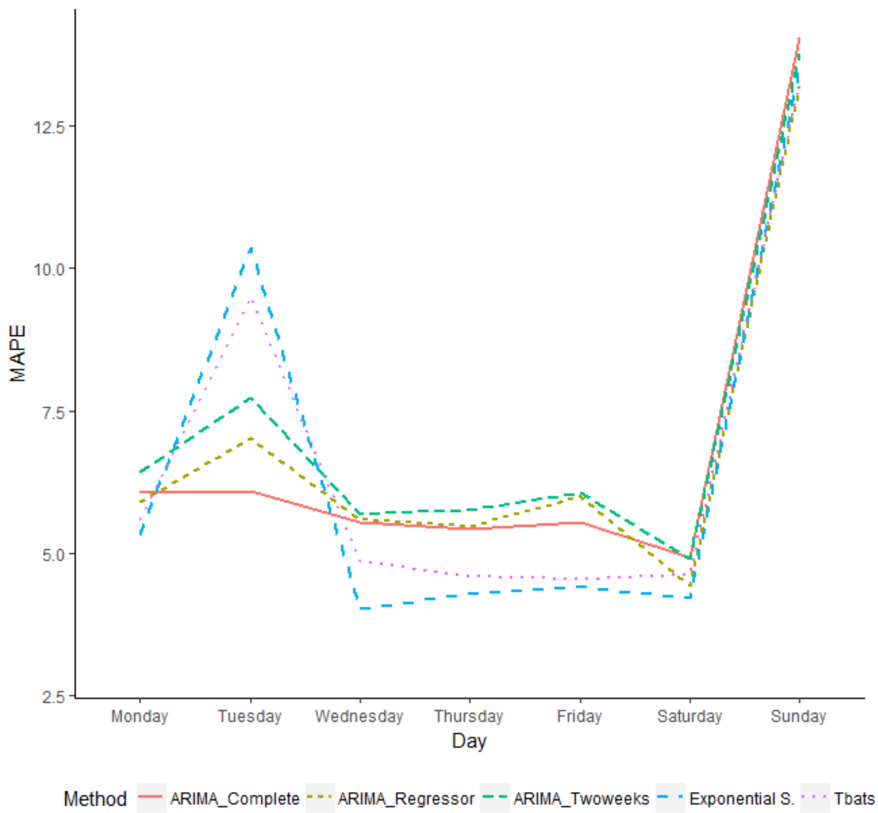

Figure 4.15. Average MAPE Results of Time Series Methods Based on Days

exponential smoothing and tbats models show a good performance.

Finally, when the performances of the models are analyzed based on months, it is again seen that the models are performing similarly(Figure 4.19). Like naive methods, time series methods are performing worse in September due to the religious holidays. In also July, these methods show slightly worse performance. However, comparing to naive methods their performance in July is better. In terms of bias, all models show positive bias during first months of year and show negative bias in June(Figure 4.20). In September, ARIMA models show positive bias, however, exponential smoothing and tbats models show negative bias. This situation may be related with sudden increase or decrease in temperature. For example in June, if the weather is increased very rapidly then the consumption also increases rapidly. Hence, the models could not catch this unexpected increase and could have bias on negative side.

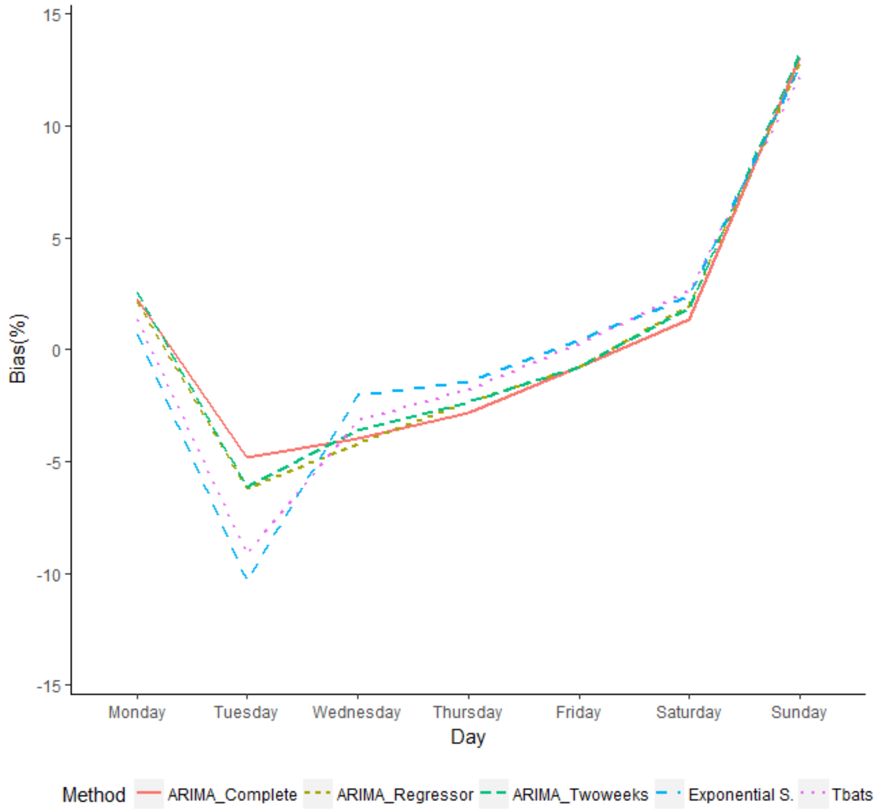

Figure 4.16. Average Bias Results of Time Series Methods Based on Days

4.2.3. Regression Methods

In this study, regression methods are applied by using first two years of data as training set and a feature list that will be described in this chapter. Models are tested over the last year. When a particular day is forecasted, then the former day of forecasted day is also added to the training set. The next forecast is done by using new training set. In other words, training set's size is increased by one day at every forecast(Figure 4.21).

<u>4.2.3.1. Feature List.</u> Regression models are the models those relate the response variable with the regressors. Then, first challenge is to find appropriate regressors and the second challenge is to find appropriate relation between regressors and response. In short term load forecasting, there are some factors and their variants could be included in a model based on the nature of the load curve. These are mainly the calendar effects,

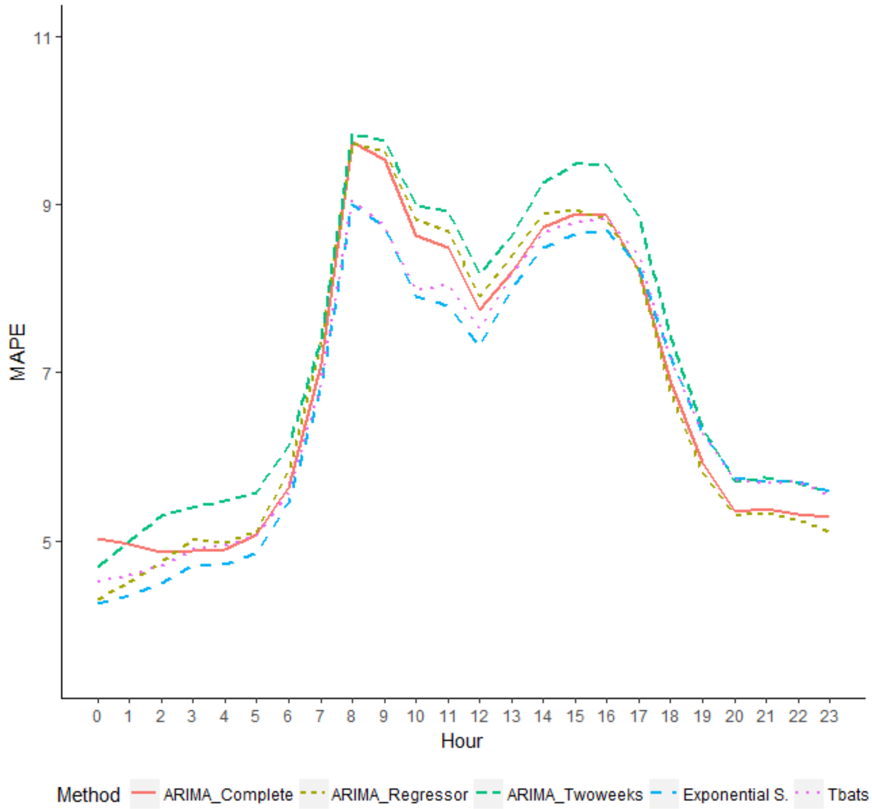

Figure 4.17. Average MAPE Results of Time Series Methods Based on Hours

temperature effects, lagged demand effects [22], trends etc.

Calendar is an important factor that determines the demand. The hour information is important because load curves show different behavior by changing hours. Electricity demand could be low after midnight and could be high during afternoon or evening times because of the economic activities and consumer behaviours. Day of week is also an important factor. During weekdays the demand is usually high especially peak hours however during weekends demand is usually lower than the weekdays. Even between the weekdays there are some differences could be observed. Season is another important factor. Depending on the types of heating and cooling devices, electricity consumption shows different behaviours based on the time of the year. In this study, it is tried to catch this effect by including week information. Another important calendar factor is holiday. During holidays the demand could be effected importantly. Hence, the holiday information should be included in the model. In this study, holidays are included in the model as national holidays and religious holidays because their effects

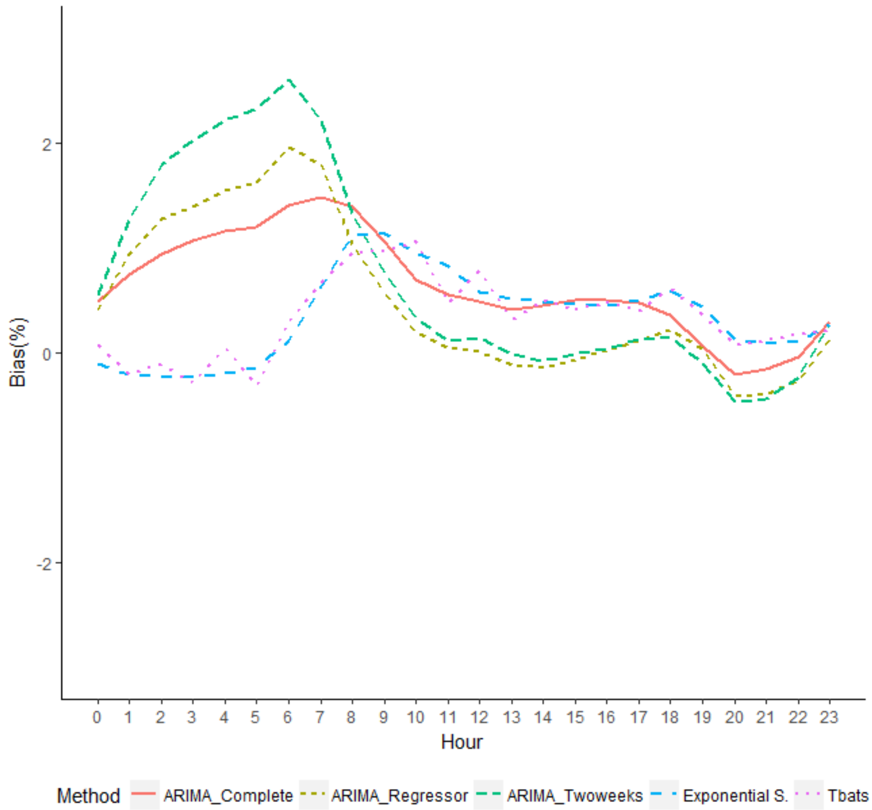

Figure 4.18. Average Bias Results of Time Series Methods Based on Hours

could be different. First of all religious holidays do not have fixed dates, their dates are changing and their effect on consumption is more distinguishable than the national holidays.

Temperature is another important factor on electricity consumption(Figure 4.22). Especially in the countries in which heating and cooling devices are electrical devices because people try to keep the room temperature in certain limits. The lagged values of temperature is also important because of the thermal inertia of the buildings. Hence, not only the outside temperature but also the lagged temperature values are important.

Lagged consumption is one of the other important factor to forecast the future consumption. Because the consumption of an hour could be very similar to the previous days' same hour, it is logical to extract information from those data points. Additionally, the consumption of a specific hour could be related with the previous hours. For example, if there is an occasion that affects demand of a particular hour it is possible

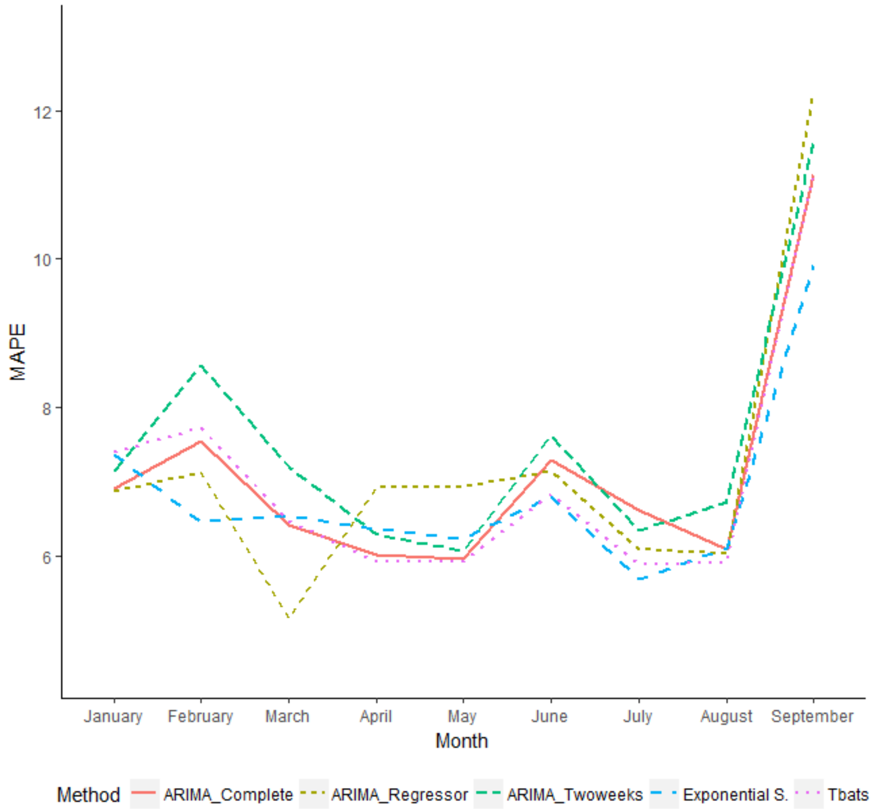

Figure 4.19. Average MAPE Results of Time Series Methods Based on Days

that it will affect the proceeding hours. Hence, also the immediate preceding hours consumption could be helpful information.

Finally, there are be some features those could be included or structured to model the different behaviors of the load curves. These features could be some generic features or could be data specific. In this study, there are some features included to capture the Saturday's and Sunday's effects more accurately. Moreover, to capture seasonal fluctuations and trends accurately, there are some features are calculated by using past weeks data and added to the models.

A feature list is designed based on the previous information(Table 4.1).

<u>4.2.3.2. Linear Regression.</u> In this study, for linear regression, first two year of data is used as initial training set and the last year is used as test set. After forecasting a

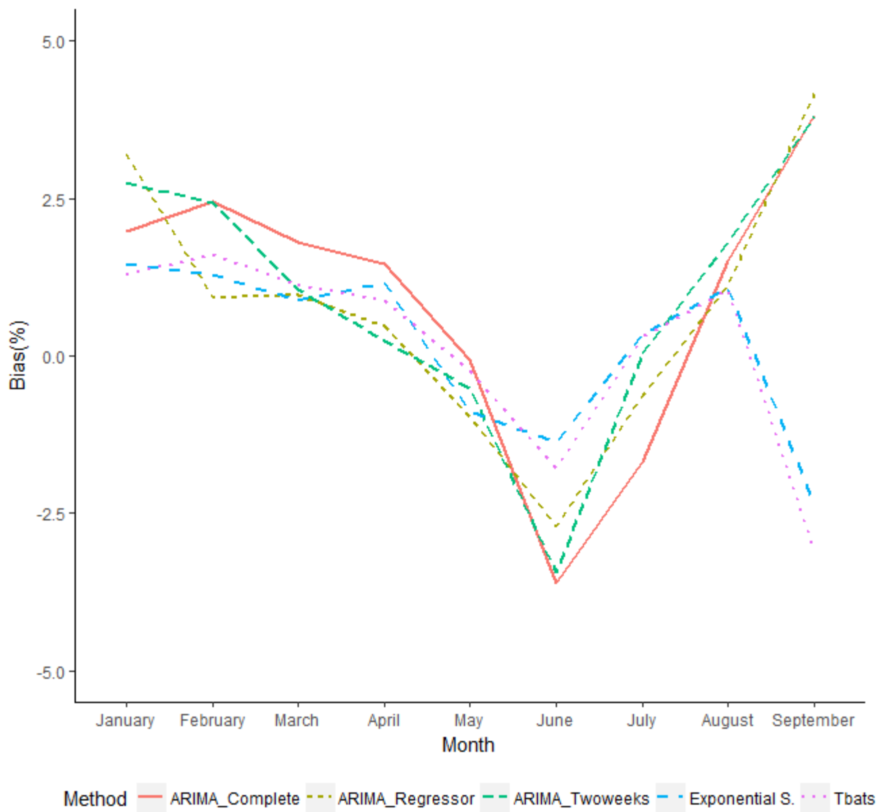

Figure 4.20. Average Bias Results of Time Series Methods Based on Months

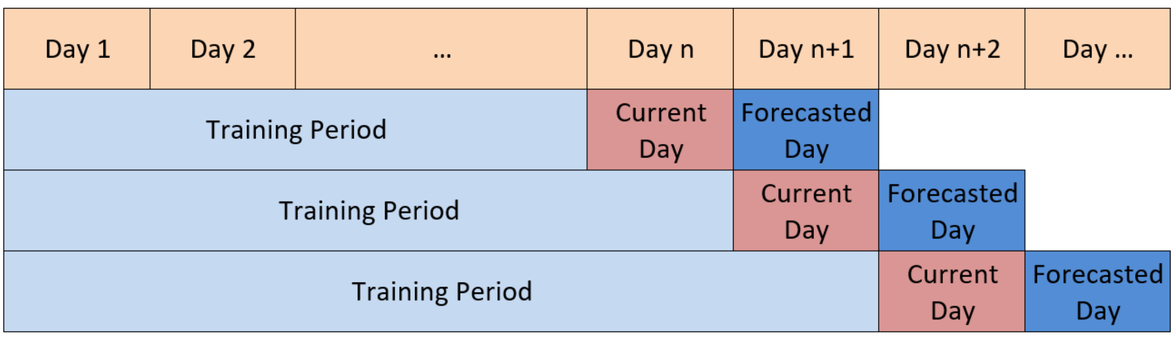

Figure 4.21. Complete Data Training and Test Process

day, that day is also added to the training set and the model is trained again with the additional day and make forecast for the next day. Because linear regression does not require any parameter, parameter selection stage is not applied in this part. In this study, the model is tested for 256 day data and the acquired average MAPE value is 4.87.

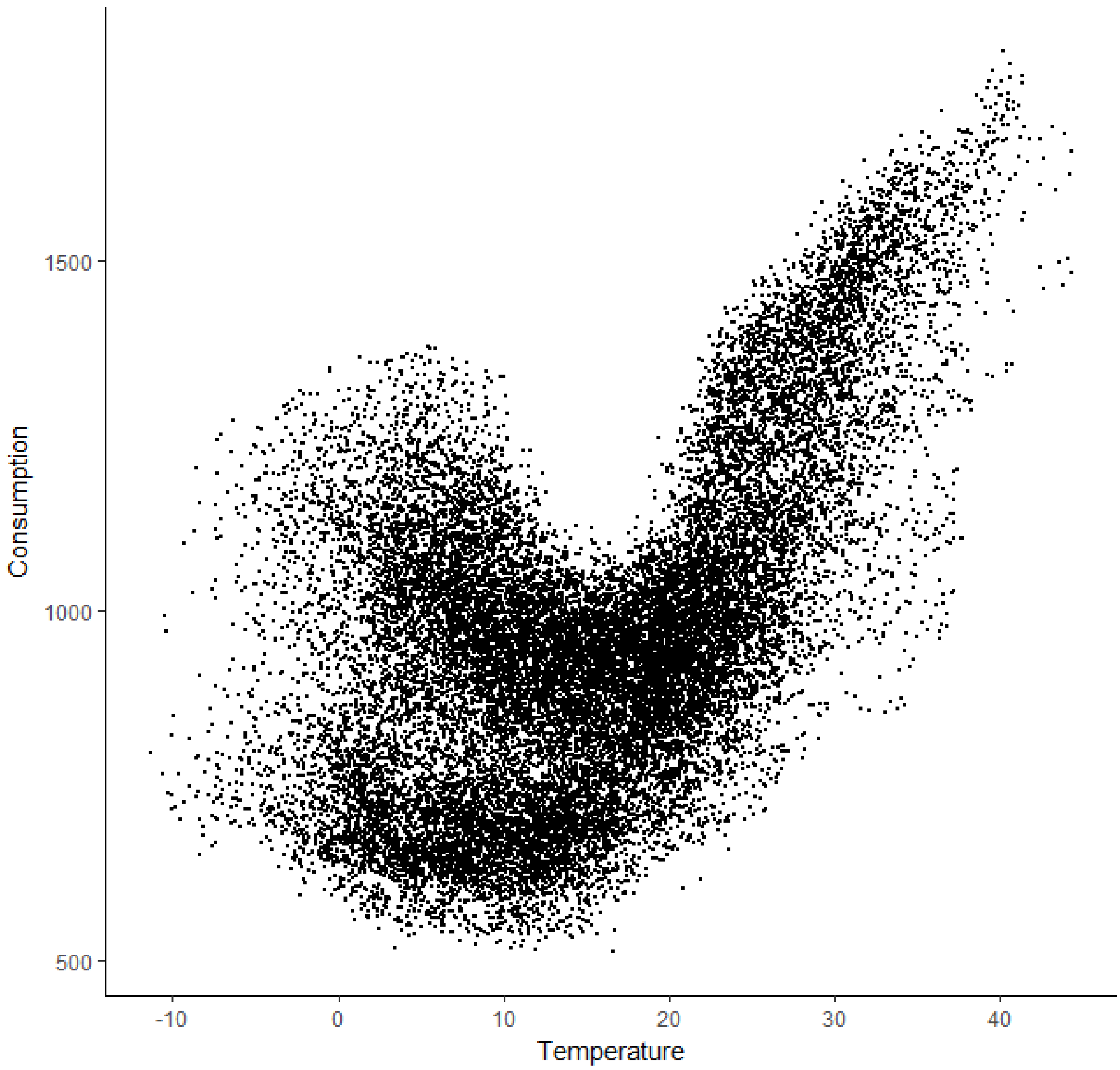

Figure 4.22. Electricity Consumption Versus Temperature

4.2.3.3. Random Forest. In this study, the random forest model is constructed by using parameters (Number of tree, Number of predictors for each tree, Node size) = (450, 12, 2) and the feature list that is described above. Then, up to end of data horizon hourly forecasts are done day-by-day. After forecasting a day, that day is also added to training set and the model is retrained with new training dataset. At the end of analysis the model is tested for 256 day data and the acquired average MAPE value is 3.46.

Table 4.1. Feature List

Day
Hour
Week
Temperature
Week Continious
48 hours before load value
49 hours before load value
50 hours before load value
51 hours before load value
52 hours before load value
53 hours before load value
54 hours before load value
60 hours before load value
72 hours before load value
1 hour before temperature value
2 hours before temperature value
3 hours before temperature value
4 hours before temperature value
5 hours before temperature value
6 hours before temperature value
24 hours before temperature value
isSunday
isSaturday
Religious holiday
National holiday
Weekly average increase
Weekly average increase of weekly average inc.

4.2.3.4. Gradient Boosting Method. In this study, the gradient boosting method model is constructed by using parameters as (Number of tree, Learning rate, interaction depth) = (22000, 0.012, 2) and the feature list that is described above. Then, up to end of data horizon hourly forecasts are done day-by-day. After forecasting a day, that day is also added to training set and the model is retrained with new training dataset. At the end of analysis the model is tested for 256 day data and the acquired average MAPE value is 2.81.

4.2.3.5. Support Vector Machines. In this study, the support vector machine model is constructed by using parameters (Kernel, Tolerance, Epsilon) = (Radial, 0.001, 0.03) and the feature list that is described above. Then, up to end of data horizon hourly forecasts are done day-by-day. After forecasting a day, that day is also added to training set and the model is retrained with new training dataset. At the end of analysis the model is tested for 256 day data and the acquired average MAPE value is 3.24.

According to results it seems that models have difficulty while making forecast on national and religious holidays. Hence, it is very obvious that there is a need to design a procedure that could capture the holidays behaviors.

In analyzing results of regression methods based on days, there is not a distinguishable pattern of MAPE based on days, except from Sunday because it has different load pattern according to other days of week(Figure 4.23). Moreover, it is worth to mention here is that, linear regression method is performing remarkable worse than other methods. Because, as expected, linear regression does not have capability to capture non-linear curves as much as machine learning regression methods. In terms of bias, all models show more or less similar performances and there is not any meaningful bias based on days, except from linear regression. Linear regression shows a bit higher bias on positive side especially on Mondays(Figure 4.24).

In analyzing the model performances based on hours, it is seen that all models show similar performances(Figure 4.25). It is remarkable that regression models are performing worse at the hours when the consumption values are high. In terms of bias, the models do not show any meaningful pattern based on hours(Figure 4.26). They

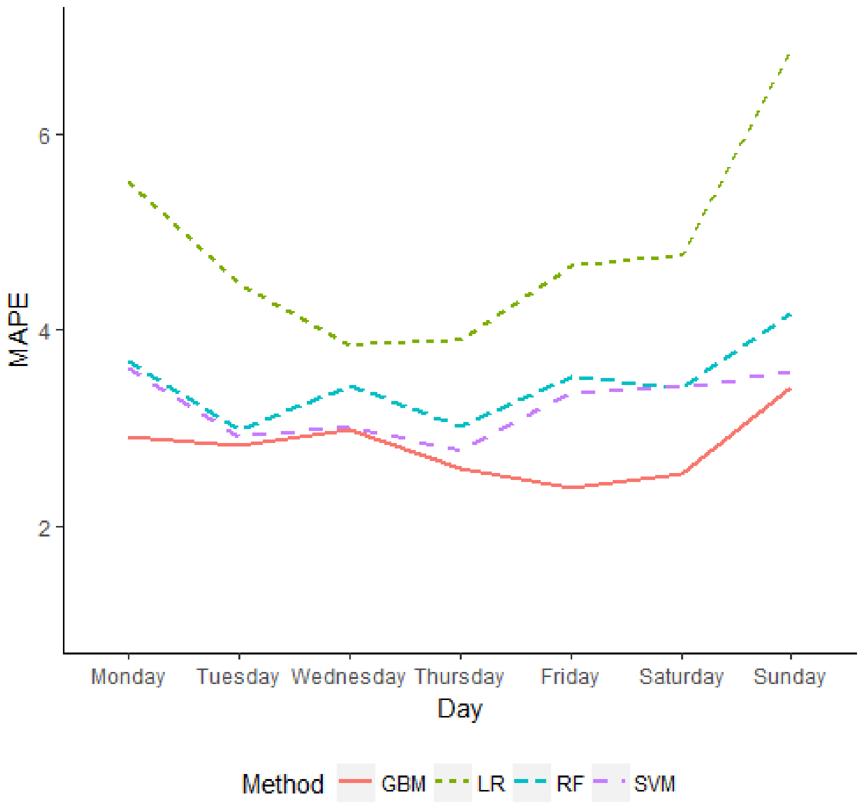

Figure 4.23. Average MAPE Results For Machine Learning Methods Based on Days

usually have positive bias. However, gradient boosting method shows remarkably netter performance then the other model

Finally, when the performances of the models are analyzed based on months, it is seen that the models are performing similarly(Figure 4.27). Like naive and time series methods, regression methods are performing worse in September and July due to the religious holidays. In terms of bias, the models do not show any meaningful pattern based on months(Figure 4.28). Linear regression shows slightly volatile bias performance. Moreover, all models have a tendency to have positive bias in September. This situation could stem from religious holidays.

According to results of all methods(Table 4.2), there several points are needed to be highlighted. Machine learning methods show significantly better performances then other methods in terms of all performance criteria. In terms of average MAPE, Naive 1 and naive 3 methods showed worst performance. Time series methods and naive

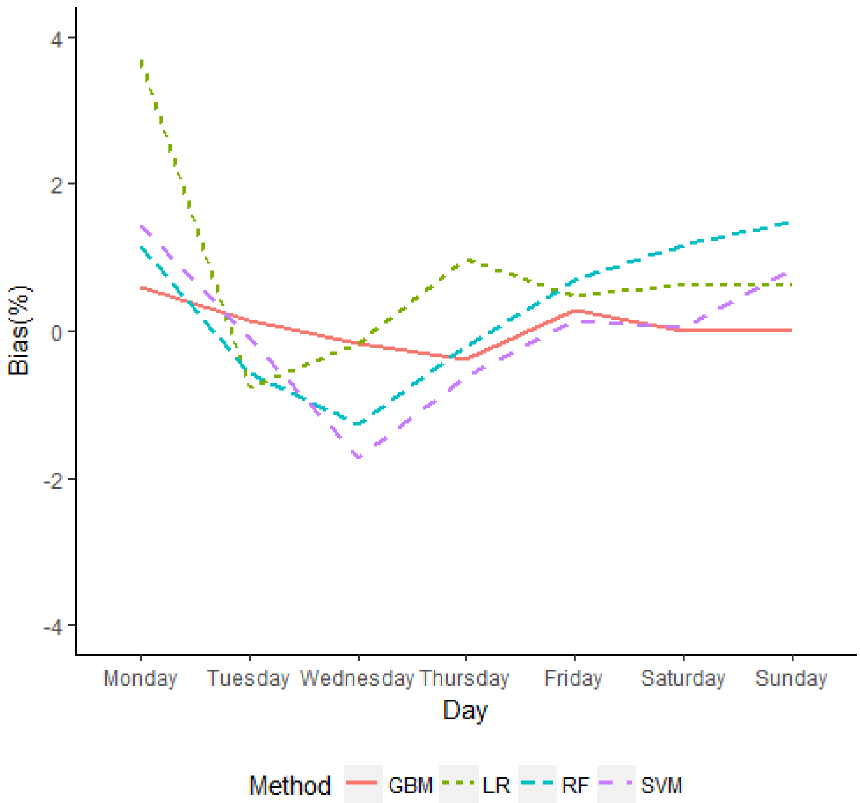

Figure 4.24. Average Bias Results For Machine Learning Methods Based on Days

3 methods show better performance which are around 7. Linear regression is better than time series and gave 4.87 MAPE value. Machine learning methods performed best and gave MAPE value around 3. This situation is same for maximum MAPE results. In terms of minimum MAPE, all methods have 0 value. It means that, all methods forecasted at least one instance with 0 error. MAPE variance values show similar patterns. Machine learning methods again outperformed the other methods. In terms of bias, naive methods gave negative and relatively higher bias values. This results are probably due to the incapability of naive methods to capture trend which is explained above. Machine learning methods, time series methods and linear regression show positive bias. However, these bias values are close to zero. Then, it is hard to say bias is a significant problem for these methods in this study except from naive methods.

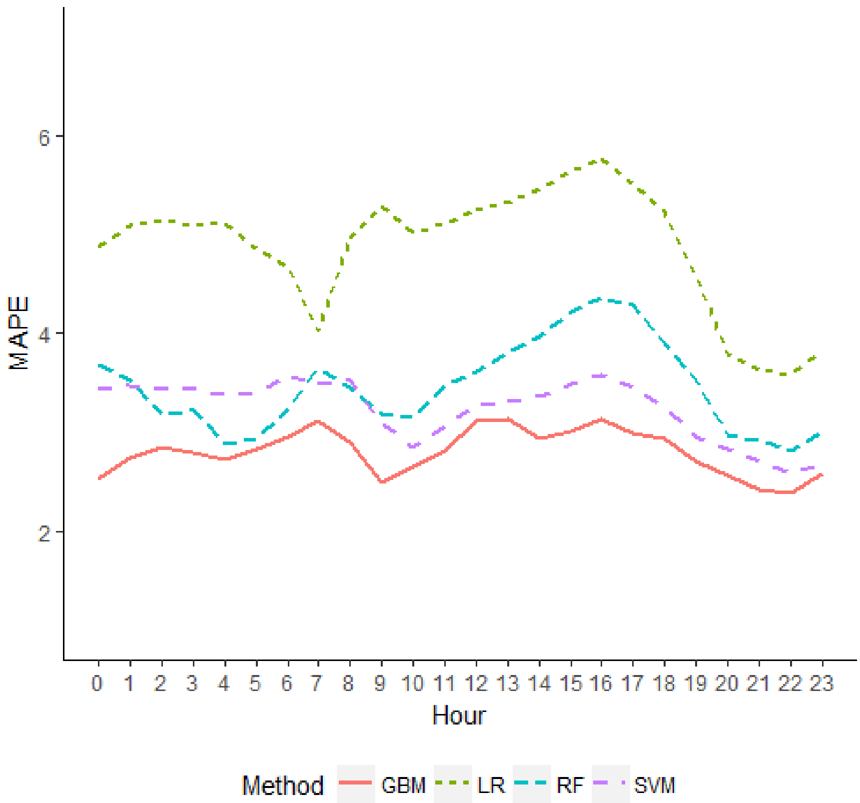

Figure 4.25. Average MAPE Results For Machine Learning Methods Based on Hours

Table 4.2. Training Period Lengths, MAPE and Bias Results of All Methods

	Training Period Length	**Minimum MAPE**	**Maximum MAPE**	**Average MAPE**	**MAPE Variance**	**Average Bias**
Naive1	Complete Past Data	0.00	81.18	10.51	81.64	-1.59
Naive2	Complete Past Data	0.00	77.15	6.52	54.78	-0.45
Naive3	Complete Past Data	0.00	81.18	10.29	80.24	-2.32
ARIMA Complete	Complete Past Data	0.00	58.30	6.82	47.44	0.64
ARIMA Twoweeks	Two weeks	0.00	58.43	7.20	49.33	0.70
ARIMA Regressor	Two weeks	0.00	57.85	6.81	45.93	0.49
Exponential Smoothing	Two weeks	0.00	56.17	6.61	44.40	0.34
Tbats	Two weeks	0.00	50.98	6.73	43.02	0.34
RF	Complete Past Data	0.00	41.99	3.46	10.98	0.35
SVM	Complete Past Data	0.00	34.34	3.24	10.44	0.00
GBM	Complete Past Data	0.00	19.13	2.81	6.47	0.07
LR	Complete Past Data	0.00	42.37	4.87	17.51	0.78

4.2.3.6. Data Replication and Temperature Separation. Machine learning methods performances are dependent on several conditions such as the type of features, number

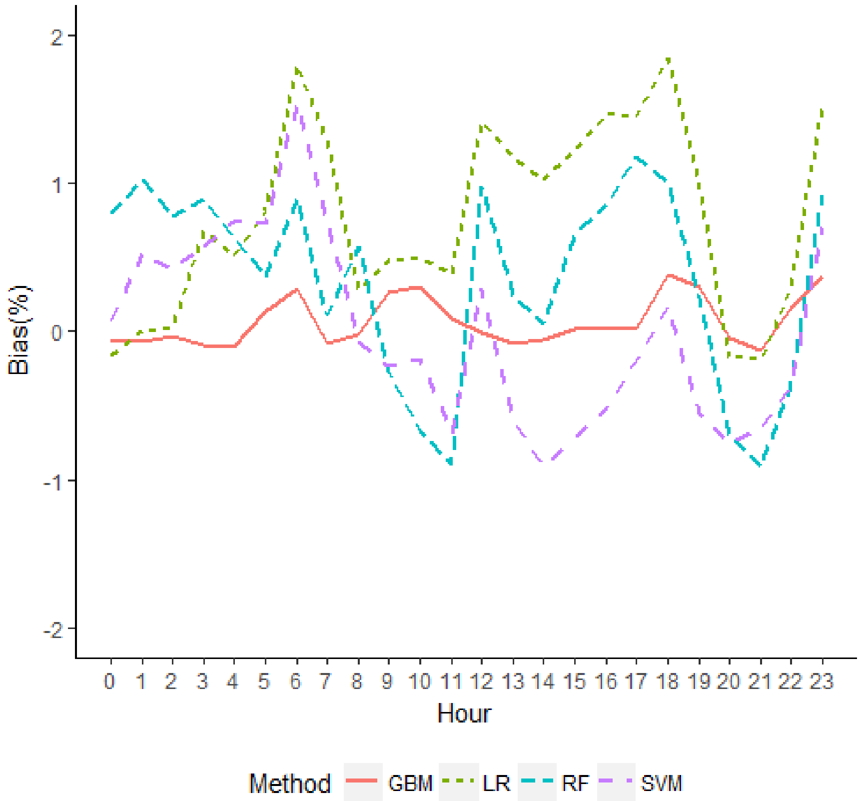

Figure 4.26. Average Bias Results For Machine Learning Methods Based on Hours

of features, the composition of the training data and etc.. According to the results of machine learning methods, model performances show difference between forecast of instances. In order to increase model performances on those particular instances and/or increase overall model performances, two approach was applied.

First one was replicating the data for religious and national holidays. All models' worst performances are on national and religious holidays. Because the number of observation of those days is less than to non-holiday days. Moreover, religious holiday dates differ from year to year, then, models could not catch the behaviour of those days properly. In order to make model fit better on those days, it is a logical way to increase the weights of those days or replicate the data of those days in training set. By this way, the models are expected to fit better on those days.

Second one was separating the temperature of the cities. Because the population of the cities are almost equal, their temperatures are aggregated for the sake of

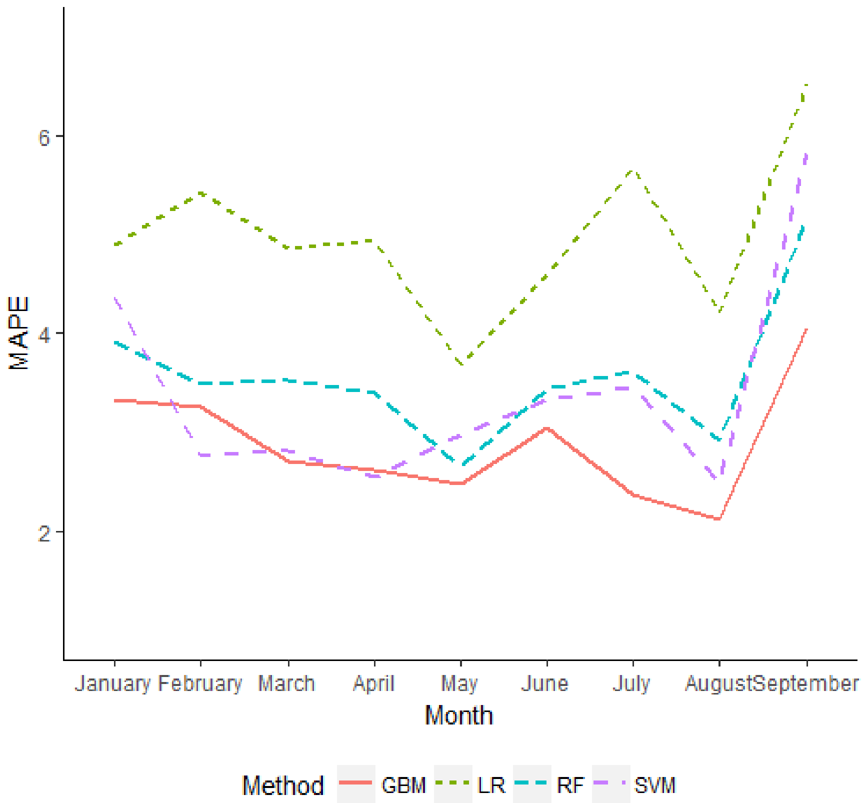

Figure 4.27. Average MAPE Results For Machine Learning Methods Based on Months

decreasing number features. However, the touristic population is changing differently in each city especially in summer. Thus, their contribution to electricity consumption may be different. Then, it might be meaningful to separate their temperature features. When this separation is applied, the lagged temperature features are also needed to be separated. However, separating all lagged temperature features will increase number of features too much. This will increase the model complexity and may decrease the model performances. Hence, lagged temperature values are separated up to 3 hours and the lagged values more than 3 hours kept as aggregated.

According to results, both replicating the holiday instances and separating the temperature features seem not effective or negatively effective on model performances. Hence, it is concluded that these two approaches are not proper approaches to improve the performance in this study.

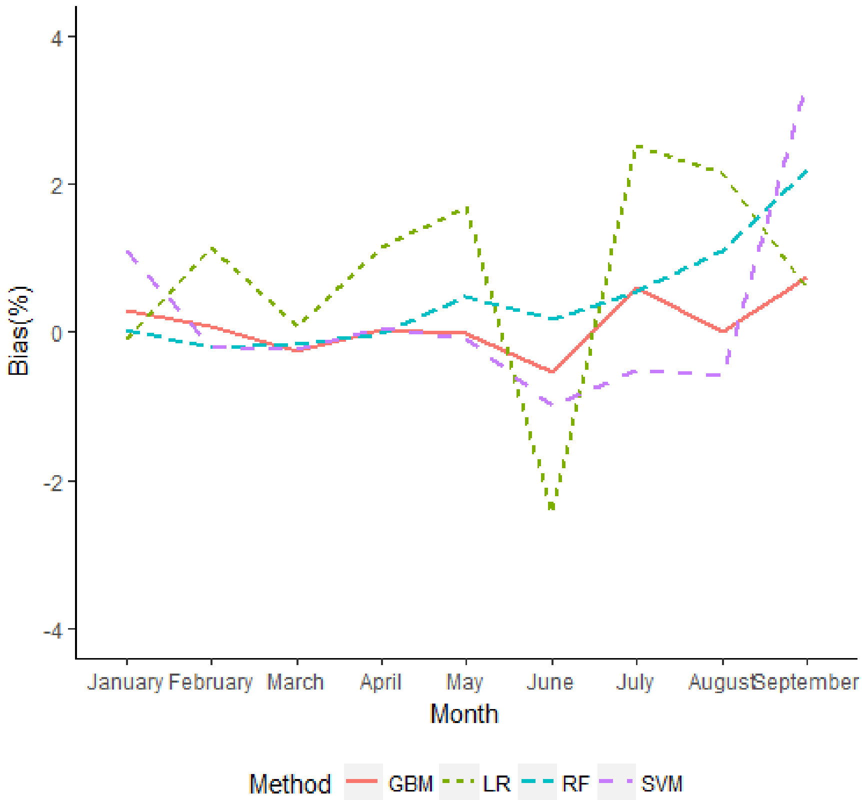

Figure 4.28. Average Bias Results For Machine Learning Methods Based on Months

5. CONCLUSION

Short term load forecasting is a difficult task. There are several methods proposed to perform this task up to this time. Machine learning methods are most common and popular methods used in STLF recent years. The purpose of this study is to analyze the performances of machine learning methods in STLF. Several generic machine learning methods such as support vector machines, gradient boosting method and random forest are applied to an hourly electricity consumption data of a region. These methods are compared with linear regression, with generic time series methods such as seasonal ARIMA, exponential smoothing, tbats and with 3 naive methods. The feature list used in this study is constructed mostly based on literature.

It is observed that machine learning methods significantly outperformed the time series methods, linear regression and naive methods. The reason behind this difference is the fitting capability of machine learning methods to nonlinear curves. Among the machine learning methods, gradient boosting method showed slightly better performance than support vector machine and random forest.

Secondly, all methods including machine learning methods have difficulty to forecast the instances which have small number of data. Especially in religious and national holidays, all methods show relatively worse performance because there are not so much data to capture their behaviours more accurately. To overcome this situation, instances of national and religious holidays are replicated to increase their number in training set, however, it does not provide any improvement.

Thirdly, it is observed that for the time series methods used in this study, training the models by using two weeks of data is quite enough to forecast without loosing significant accuracy. Moreover, as in the case of naive methods, it is observed that including any variable which captures trend of the data improved the performance.

As future works, it is suggested to try different approaches to capture the behavior of the instances at which models perform relatively worse such as religious and national holidays. Moreover, it is observed that performance of the models may vary depend on the instances. For example, for different days, the best performing model could be changed. Hence, a module could be integrated which selects the best forecasting model for every particular instances. Finally, separate models could be constructed for different hours, days or months to capture their load behavior more uniquely.

REFERENCES

1. Jones, Z. and F. Linder, “Exploratory Data Analysis using Random Forests”, *Prepared for the 73rd annual MPSA conference*, 2015.

2. Koehrsen, W., *Random Forest Simple Explanation*, `https://medium.com/@williamkoehrsen/random-forest-simple-explanation-377895a60d2d`, accessed at July 2018.

3. Arora, U., *Gradient Boosting Part1–Visual Conceptualization*, `https://dimensionless.in/gradient-boosting/`, accessed at July 2018.

4. Grover, P., *Gradient Boosting from scratch*, `https://medium.com/mlreview/gradient-boosting-from-scratch-1e317ae4587d`, accessed at July 2018.

5. *Support Vector Machine - Regression (SVR)*, `http://www.saedsayad.com/support_vector_machine_reg.htm`, accessed at July 2018.

6. *Support Vector Machine Regression*, `http://kernelsvm.tripod.com/`, accessed at July 2018.

7. Al-Hamadi, H. and S. Soliman, “Long-term/mid-term electric load forecasting based on short-term correlation and annual growth”, *Electric Power Systems Research*, Vol. 74, 2005.

8. Khatoon, S., Ibraheem, A. K. Singh and Priti, “Effects of Various Factors on Electric Load Forecasting: An Overview”, *2014 6th IEEE Power India International Conference (PIICON)*, 2014.

9. Lahouar, A. and J. B. H. Slama, “Day-ahead load forecast using random forest and expert input selection”, *Energy Conversion and Management*, Vol. 103, 2015.

10. Javed, F., N. Arshad, F. Wallin, I. Vassileva and E. Dahlquist, "Forecasting for demand response in smart grids: an analysis on use of anthropologic and structural data and short term multiple loads forecasting", *Applied Energy*, Vol. 96, 2012.

11. Dudek, G., "Short-Term Load Forecasting Using Random Forests", *Intelligent Systems'2014*, pp. 821–828, Springer International Publishing, Cham, 2015.

12. Samsher, K. S. and M. G. Unde, "Short Term Load Forecasting Using ANN Technique", *International Journal of Engineering Sciences and Emerging Technologies*, Vol. 1(2), 2012.

13. Hambali, M. A., Akinyemi, Oladunjoye and Y. N., "Electric Power Load Forecast Using Decision Tree Algorithms", *Computing, Information Systems, Development Informatics Allied Research Journal*, Vol. 7, 2016.

14. Pappas, S., L. Ekonomou, D. Karamousantas, G. Chatzarakis, S. Katsikas and P. Liatsis, "Electricity demand loads modeling using AutoRegressive Moving Average (ARMA) models", *Energy*, Vol. 33, 2008.

15. Cho, M., J. Hwang and C. Chen, "Customer Short Term Load Forecasting By Using Arima Transfer Function Model", *Energy Management and Power Delivery, 1995. Proceedings of EMPD '95., 1995 International Conference on*, Vol. 1, 1995.

16. Lee, C.-M. and C.-N. Ko, "Short-term load forecasting using lifting scheme and ARIMA models", *Expert Systems with Applications*, Vol. 38, 2011.

17. Taylor, J. W., L. M. de Menezes and P. E. McSharry, "A comparison of univariate methods for forecasting electricity demand up to a day ahead", *International Journal of Forecasting*, Vol. 22, 2006.

18. Taylor, J. W., "Triple seasonal methods for short-term electricity demand forecasting", *European Journal of Operational Research*, Vol. 204, 2010.

19. Al-Hamadi, H. and S. Soliman, “Short-term electric load forecasting based on Kalman filtering algorithm with moving window weather and load model”, *Electric Power Systems Research*, Vol. 68, 2004.

20. Amral, N., C. Özveren and D. King, “Short Term Electricity Load Forecasting Using Multiple Linear Regression”, *UPEC*, 2007.

21. Gladysz, B. and D. Kuchta, “Application Of Regression Trees In The Analysis Of Electricity Load”, *Badania Operacyjne I Decyzje*, Vol. 4, 2008.

22. Taieb, S. B. and R. J. Hyndman, “A gradient boosting approach to the Kaggle load forecasting competition”, *International Journal of Forecasting*, Vol. 30, 2014.

23. Papadopoulos, S. and I. Karakatsanis, “Short-term Electricity Load Forecasting using Time Series and Ensemble Learning Methods”, *2015 IEEE Power and Energy Conference at Illinois (PECI)*, 2015.

24. Mohandes, M., “Support vector machines for short-term electrical load forecasting”, *International Journal Of Energy Research*, Vol. 26, 2002.

25. Pereira, J. J. F., R. M. Salgado, T. Ohishi and R. Ballini, “Short-Term Load Forecasting Using Support Vector Machines”, *2006 IEEE PES Transmission and Distribution Conference and Exposition Latin America, Venezuela*, 2006.

26. Lee, K. Y. and J. H. Park, “Short-Term Load Forecasting Using An Artificial Neural Network”, *Transactions on Power Systems*, Vol. 7, 1992.

27. Livera, A. M. D., R. J. Hyndman and R. D. Snyder, “Forecasting Time Series With Complex Seasonal Patterns Using Exponential Smoothing”, *Journal of the American Statistical Association*, Vol. 106, pp. 1513–1527, 2011.

28. Hipper, H. S., C. E. Pedreira and R. C. Souza, “Neural networks for short-term load forecasting: a review and evaluation”, *IEEE Transactions on Power Systems*,

Vol. 16(1), 2001.

29. Martín, M. and M. Acera, "Electricity Load Forecasting Using Machine Learning Techniques", *Business Intelligence in Economic Forecasting: Technologies and Techniques*, 2016.

30. Cannady, J., "Artificial neural networks for misuse detection", *National information systems security conference*, Vol. 26, 1998.

31. Tu, J. V., "Advantages and Disadvantages of Using Artificial Neural Networks versus Logistic Regression for Predicting Medical Outcomes", *Journal of Clinical Epidemiology*, Vol. 49, pp. 1225–1231, 1996.

32. Breiman, L., J. H. Friedman, R. A. Olshen and C. J. Stone, *Classification and Regression Trees*, Wadsworth, 1984.

33. Jiang, W., "On Weak Base Hypotheses and Their Implications for Boosting Regression and Classification", *The Annals of Statistics*, Vol. 30, 2002.

34. Bergstra, J. and Y. Bengio, "Random Search for Hyper-Parameter Optimization", *Journal of Machine Learning Research*, Vol. 13, 2012.

35. Snoek, J., H. Larochelle and R. P. Adams, "Practical Bayesian Optimization of Machine Learning Algorithms", *Advances in neural information processing systems*, 2012.

APPENDIX A: PARAMETER TUNING

A.1. ARIMA

For the first version of ARIMA, autoarima function forecast package is used firstly. Then, according to autoarima parameter results, it is observed that model usually selects the small number as parameter during its optimization process. Then the range of 0-3 is searched over for basic arima parameters and 0-2 for seasonal arima parameters. According to preruns results difference parameter range is choosed as 0-1 and seasonal difference level is choosed as 0. Period is selected as 24. Parameter search is applied on random 10 weeks and whole days of those weeks(70 days). After parameter analysis it is concluded that (p,d,q,P,D,Q) = (0,0,2,2,0,2) is the best parameter set for this dataset.

For the second version of ARIMA, same procedure of two weeks data version is applied. Only difference is, parameter search is applied on 5 random weeks(35 days) data because of the time requirements is high than the previous version. After parameter analysis it is concluded that (p,d,q,P,D,Q) = (0,0,0,2,0,2) is the best parameter set for this dataset.

For the third version of ARIMA, the parameters of the second version is used.

A.2. Exponential Smoothing

In order to decide the parameter values of exponential smoothing, firstly the internal optimization process of ets function is used to see that what are the possible parameter values. It is observed that ets function is selecting the parameters like these:

Alpha = around 0.999

Beta = 0.001 – 0.03

Gamma = 0.0001 – 1200

Then, according to these values the candidate parameter values to try is designed like these:

Alpha = (0.8, 0.85, 0.9, 0.95, 0.99)

Beta = (0.001, 0.005, 0.01, 0.02)

Gamma= (0.001, 1, 100, 500, 1000)

According to test results over 10 random weeks(70 days), it is observed that ets function gives best results fort he parameters (Alpha, Beta) = (0.8, 0.01). Different Gamma values did not affect the results. Then, in order to get more precise parameter values, second step of the parameter search is applied. The candidate parameter values are selected as:

Alpha = (0.77, 0.8, 0.83)

Beta = (0.008, 0.01, 0.012)

According to test results over 10 random weeks(70 days), it is observed that parameter values of (Alpha, Beta) = (0.8, 0.01) are again the best values.

A.3. Random Forest

In order to decide the parameter values of the random forest, first two year of the data is used. This parameter estimation is done by 10-fold cross validation. Parameters are selected based on preruns and default values of the package itself(random forest package, randomForest function). Parameter estimation is applied in two stage for random forest.

In parameter estimation stage three parameters are estimated, number of tree, number of predictors for each tree and node size. These parameters are searched over these values:

Table A.1. First Stage Parameter Search of Random Forest

Number Of Tree	250	500	1000
Number Of Predictors For Each Tree	10	15	20
Node Size	3	5	10

After first stage of parameter estimation these values are determined as the best performing parameter values: (Number of tree, Number of predictors for each tree, Node size) = (500,10,3) By using these values, in order to find better parameter values new parameter values around these values are tested:

Table A.2. First Stage Parameter Search of Random Forest

Number Of Tree	450	500	550
Number Of Predictors For Each Tree	8	10	12
Node Size	2	3	4

After second stage these values are determined as best performing parameter values and these values are used as final parameter values in model training:

(Number of tree, Number of predictors for each tree, Node size) = (450,12,2)

A.4. Gradient Boosting Method

In order to decide the parameter values of the gradient boosting method, first two year of the data is used. This parameter estimation is done by 10-fold cross validation, the folds are same with random forest. Parameters are selected based on preruns and default values of the package itself(gbm package,gbm function). Parameter estimation is applied in two stage for gradient boosting method.

In parameter estimation stage three parameters are estimated, number of tree, learning rate(shrinkage) and interaction depth. These parameters are searched over these values:

Table A.3. First Stage Parameter Search of Gradient Boosting Method

Number Of Tree	10000	15000	20000
Learning Rate	0.001	0.005	0.01
Interaction Depth	1	2	

After first stage of parameter estimation these values are determined as the best performing parameter values:

(Number of tree, Learning rate, interaction depth) = (20000,0.01,2)

By using these values, in order to find better parameter values new parameter values around these values are tested:

Table A.4. Second Stage Parameter Search of Gradient Boosting Method

Number Of Tree	18000	20000	22000
Learning Rate	0.008	0.01	0.012
Interaction Depth	2		

After second stage these values are determined as best performing parameter values and these values are used as final parameter values in model training:

(Number of tree, Learning rate, interaction depth) = (22000,0.012,2)

A.5. Support Vector Machines

In order to decide the parameter values of the support vector machine, first two year of the data is used. This parameter estimation is done by 10-fold cross validation, the folds are same with random forest. Parameters are selected based on preruns and default values of the package itself(e1071 package, svm function).Parameter estimation is applied in three stage for support vector machine.

In parameter estimation stage three parameters are estimated, kernel type, tolerance and epsilon. These parameters are searched over these values:

Table A.5. First Stage Parameter Search of Support Vector Machine

Kernel Type	Linear	Polynomial	Sigmoid	Radial
Tolerance	0.0005	0.001	0.0015	
Epsilon	0.05	0.1	0.15	

After first stage of parameter estimation these values are determined as the best performing parameter values:

(Kernel, Tolerance, Epsilon) = (Radial , 0.001 ,0.05)

It is observed that tolerance values are not effective on results in this study hence it is used as its default value 0.001. By using these values, in order to find better parameter values new parameter values around these values are tested:

Table A.6. Second Stage Parameter Search of Support Vector Machine

Kernel Type	Radial		
Tolerance	0.001		
Epsilon	0.03	0.05	0.07

After second stage these values are determined as best performing parameter values:

(Kernel, Tolerance, Epsilon) = (Radial , 0.001 ,0.03)

In parameter estimation one more step is applied. In this step only epsilon value is 0.01 is analyzed.

Table A.7. Third Stage Parameter Search of Support Vector Machine

Kernel Type	Radial
Tolerance	0.001
Epsilon	0.01

It seems that it gives better result. Hence final parameter set is chosen as:

(Kernel, Tolerance, Epsilon) = (Radial , 0.001 ,0.01)

APPENDIX B: TABLES

Table B.1. Average MAPE Results For Naive Methods Based on Days

Days	**Naive1**	**Naive2**	**Naive3**
Monday	11.40	7.23	10.89
Tuesday	10.26	5.98	9.86
Wednesday	8.89	5.39	8.89
Thursday	9.58	5.24	9.60
Friday	10.43	6.77	10.68
Saturday	11.15	7.31	10.89
Sunday	11.84	7.70	11.26

Table B.2. Average MAPE Results For Naive Methods Based on Hours

Hours	**Naive1**	**Naive2**	**Naive3**
0	11.19	6.24	10.90
1	9.93	5.90	9.44
2	9.74	5.76	9.24
3	9.80	5.73	9.45
4	8.84	5.39	8.73
5	8.90	5.18	8.73
6	9.15	5.84	9.09
7	9.61	6.23	9.52
8	9.23	6.49	9.20
9	9.71	6.79	9.43
10	9.69	7.00	9.82
11	9.96	7.26	9.76
12	11.45	7.46	11.41
13	11.90	7.39	11.61
14	11.10	7.63	10.53
15	11.49	7.72	11.91
16	12.05	7.47	12.07
17	12.27	7.44	12.34
18	12.70	7.25	12.09
19	12.63	6.50	12.01
20	11.27	5.49	10.85
21	9.77	5.98	9.62
22	9.20	5.89	9.05
23	10.59	6.39	10.23

Table B.3. Average MAPE Results of Naive Methods Based on Months

Months	**Naive1**	**Naive2**	**Naive3**
January	10.79	6.46	9.73
February	11.07	4.85	10.61
March	8.42	3.98	8.16
April	7.09	3.58	6.77
May	12.08	4.27	11.02
June	14.34	6.38	13.75
July	9.68	11.47	11.52
August	7.55	5.36	7.64
September	17.90	19.71	17.86

Table B.4. Average MAPE Results of Time Series Methods Based on Days

Days	**ARIMA Complete**	**ARIMA Twoweeks**	**ARIMA Regressor**	**Exponential Smoothing**	**Tbats**
Monday	6.06	6.42	5.89	5.32	5.58
Tuesday	6.09	7.73	7.00	10.37	9.52
Wednesday	5.55	5.69	5.59	4.01	4.84
Thursday	5.42	5.75	5.46	4.28	4.59
Friday	5.53	6.06	5.98	4.40	4.53
Saturday	4.92	4.90	4.42	4.22	4.62
Sunday	14.04	13.75	13.19	13.52	13.23

Table B.5. Average MAPE Results of Time Series Methods Based on Hours

Hours	ARIMA Complete	ARIMA Twoweeks	ARIMA Regressor	Exponential Smoothing	Tbats
0	5.02	4.69	4.30	4.26	4.50
1	4.97	4.99	4.51	4.35	4.60
2	4.86	5.30	4.75	4.49	4.71
3	4.89	5.40	5.00	4.70	4.90
4	4.90	5.47	4.97	4.72	4.94
5	5.07	5.58	5.11	4.86	5.06
6	5.64	6.13	5.84	5.46	5.58
7	7.11	7.39	7.33	6.82	6.86
8	9.74	9.84	9.73	9.01	9.05
9	9.53	9.76	9.63	8.73	8.74
10	8.64	8.98	8.83	7.91	7.97
11	8.49	8.93	8.68	7.80	8.05
12	7.75	8.19	7.90	7.31	7.53
13	8.20	8.64	8.40	8.01	8.19
14	8.73	9.27	8.90	8.49	8.66
15	8.89	9.49	8.93	8.66	8.80
16	8.87	9.47	8.82	8.69	8.85
17	8.22	8.87	8.22	8.26	8.39
18	6.91	7.44	6.78	7.20	7.19
19	5.93	6.34	5.81	6.28	6.28
20	5.35	5.70	5.29	5.75	5.73
21	5.38	5.75	5.33	5.71	5.69
22	5.32	5.68	5.25	5.70	5.70
23	5.29	5.59	5.11	5.59	5.54

Table B.6. Average MAPE Results of Time Series Methods Based on Months

Months	**ARIMA Complete**	**ARIMA Twoweeks**	**ARIMA Regressor**	**Exponential Smoothing**	**Tbats**
January	6.90	7.14	6.88	7.36	7.40
February	7.54	8.57	7.13	6.45	7.72
March	6.42	7.20	5.14	6.54	6.47
April	6.01	6.28	6.93	6.36	5.92
May	5.95	6.07	6.92	6.22	5.93
June	7.29	7.62	7.14	6.80	6.83
July	6.62	6.34	6.08	5.69	5.89
August	6.08	6.71	6.03	6.10	5.91
September	11.15	11.55	12.22	9.92	11.13

Table B.7. Average MAPE Results of Regression Methods Based on Days

Days	**RF**	**SVM**	**GBM**	**LR**
Monday	3.68	3.61	2.92	5.52
Tuesday	2.99	2.90	2.82	4.48
Wednesday	3.43	3.03	2.98	3.85
Thursday	3.02	2.76	2.59	3.91
Friday	3.51	3.36	2.39	4.65
Saturday	3.41	3.42	2.53	4.76
Sunday	4.17	3.57	3.41	6.87

Table B.8. Average MAPE Results of Regression Methods Based on Hours

Hours	**RF**	**SVM**	**GBM**	**LR**
0	3.68	3.45	2.53	4.87
1	3.54	3.48	2.75	5.10
2	3.19	3.43	2.84	5.15
3	3.23	3.44	2.81	5.09
4	2.89	3.39	2.73	5.11
5	2.95	3.39	2.83	4.86
6	3.25	3.58	2.96	4.67
7	3.66	3.50	3.13	4.02
8	3.45	3.52	2.91	4.97
9	3.18	3.10	2.51	5.29
10	3.15	2.85	2.65	5.02
11	3.48	3.06	2.81	5.11
12	3.61	3.28	3.12	5.25
13	3.81	3.32	3.14	5.32
14	3.98	3.36	2.94	5.46
15	4.23	3.48	3.01	5.65
16	4.37	3.58	3.14	5.76
17	4.29	3.46	3.00	5.51
18	3.90	3.26	2.94	5.23
19	3.53	2.96	2.70	4.59
20	2.97	2.83	2.57	3.80
21	2.92	2.71	2.43	3.63
22	2.82	2.58	2.39	3.58
23	3.01	2.68	2.58	3.82

Table B.9. Average MAPE Results of Regression Methods Based on Months

Months	RF	SVM	GBM	LR
January	3.92	4.37	3.34	4.90
February	3.50	2.76	3.27	5.42
March	3.53	2.82	2.72	4.87
April	3.40	2.55	2.63	4.93
May	2.65	2.97	2.48	3.68
June	3.44	3.33	3.06	4.59
July	3.62	3.45	2.37	5.68
August	2.92	2.50	2.13	4.23
September	5.16	5.82	4.06	6.52

Table B.10. Average MAPE Results For Regression Methods on Replicated and Temperature Separated Data

	Min	Mean	Max	Variance	Religious	Nonreligious	National	Nonnational
RForiginal	0.00	3.46	41.99	10.98	7.61	3.31	8.08	3.37
SVMoriginal	0.00	3.24	34.38	10.44	8.70	3.04	8.51	3.13
GBMoriginal	0.00	2.81	19.13	6.47	5.96	2.69	5.09	2.76
RFrep	0.00	3.45	41.84	10.53	7.66	3.34	8.22	3.40
SVMrep	0.00	3.19	29.80	9.67	8.86	2.98	7.69	3.10
GBMrep	0.00	2.88	27.75	6.63	5.73	2.75	5.67	2.80
RF3temp	0.00	3.49	44.81	11.35	7.07	3.32	7.88	3.37
SVM3temp	0.00	3.19	30.95	9.82	8.20	3.01	7.52	3.11
GBM3temp	0.00	2.85	24.35	6.69	6.04	2.76	5.88	2.82