ÇUKUROVA UNIVERSITY
INSTITUTE OF NATURAL AND APPLIED SCIENCES

MSc THESIS

Derman AKGÖL

DEVELOPMENT OF NEW MODELS USING MACHINE LEARNING METHODS COMBINED WITH DIFFERENT TIME LAGS FOR NETWORK TRAFFIC FORECASTING

DEPARTMENT OF COMPUTER ENGINEERING

ADANA, 2016

ÇUKUROVA UNIVERSITY
INSTITUTE OF NATURAL AND APPLIED SCIENCES

DEVELOPMENT OF NEW MODELS USING MACHINE LEARNING METHODS COMBINED WITH DIFFERENT TIME LAGS FOR NETWORK TRAFFIC FORECASTING

Derman AKGÖL

MSc THESIS

DEPARTMENT OF COMPUTER ENGINEERING

We certify that the thesis titled above was reviewed and approved for the award of degree of the Master of Science by the board of jury on

…….....	…….....	…….....
Assoc. Prof. Dr. M.Fatih AKAY	Asst. Prof. Dr. Zehan KESİLMİŞ	Asst. Prof. Dr. B. Melis ÖZYILDIRIM
SUPERVISOR	MEMBER	MEMBER

This MSc Thesis is written at the Department of Institute of Natural And Applied Sciences of Çukurova University.
Registration Number:

Prof. Dr. Mustafa GÖK
Director
Institute of Natural and Applied Sciences

This study was supported by Research Projects Unit Ç.Ü. Project Number: FYL-2015-5265.

ABSTRACT

MASTER THESIS

DEVELOPMENT OF NEW MODELS USING MACHINE LEARNING METHODS COMBINED WITH DIFFERENT TIME LAGS FOR NETWORK TRAFFIC FORECASTING

Derman AKGÖL

ÇUKUROVAUNIVERSITY
INSTITUTE OF NATURAL AND APPLIED SCIENCES
DEPARTMENT OF COMPUTER ENGINEERING

Supervisor : Assoc. Prof. Dr. Mehmet Fatih AKAY
Year: 2016, Pages: 109
Jury : Assoc. Prof. Dr. Mehmet Fatih AKAY
: Asst. Prof. Dr. Zehan KESİLMİŞ
: Asst. Prof. Dr. Buse Melis ÖZYILDIRIM

The purpose of this thesis is to forecast the amount of network traffic in Transmission Control Protocol/Internet Protocol (TCP/IP) -based networks by using different time lags and various machine learning methods including Support Vector Machines (SVM), Multilayer Perceptron (MLP), Radial Basis Function (RBF) Neural Network, M5P (a decision tree with linear regression functions at the nodes), Random Forest (RF), Random Tree (RT), and Reduced Error Prunning Error (REPTree), and statistical regression methods including Multiple Linear Regression (MLR) and Holt-Winters and compare the performance of statistical and machine learning methods. Two different Internet Service Providers' (ISPs) traffic data have been utilized to build traffic forecasting models. The first 66% of the data sets has been utilized as training sets and the rest has been used as test sets. The performance of the forecasting models for the data sets has been assessed using Mean Absulote Percentage Error (*MAPE*). The results show that SVM and M5P based models usually perform better than the ones obtained by the other methods.

Key Words: Machine learning, time series, traffic forecasting, time lags.

ÖZ

YÜKSEK LİSANS TEZİ

FARKLI ZAMAN GECİKMELERİ İLE BİRLEŞTİRİLMİŞ MAKİNE ÖĞRENME YÖNTEMLERİ İLE İNTERNET TRAFİK TAHMİN MODELLERİNİN GELİŞTİRİLMESİ

Derman AKGÖL

ÇUKUROVA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

Danışman : Doç. Dr. M. Fatih AKAY
Yıl: 2016, Sayfa: 109
Jüri : Doç. Dr. M. Fatih AKAY
: Yrd. Doç. Dr. Zehan KESİLMİŞ
: Yrd. Doç. Dr. Buse Melis ÖZYILDIRIM

Bu tezin amacı İletim Kontrol Protokolü/İnternet Protokolü (TCP/IP) tabanlı ağlardaki İnternet trafik miktarını Destek Vektör Makineleri (SVM), Radyal Tabanlı Fonksiyon Sinir Ağları (RBF), M5P (ağlardaki lineer fonksiyonlu karar ağacı), Rasgele Orman (RF), Rasgele Ağaç (RT), ve İndirgenmiş Hata Budama Ağacını (REP Tree) içeren makine öğrenme yöntemlerini ve Çoklu Lineer Regresyon (MLR) ve Holt-Winters'ı içeren istatiksel regresyon yöntemlerini kullanarak tahmin etmek ve istatiksel ve makine öğrenme yöntemlerinin performanslarını kıyaslamaktır. İnternet trafik tahmin modelleri üretilirken iki farklı İnternet Servis Sağlayıcısından (ISPs) sağlanılan İnternet trafiği verileri kullanılmıştır. Veri setinin ilk %66'sı eğitim kümesi olarak uygulanırken geri kalanı test kümesi olarak kullanılmıştır. Modellerin performansı Ortalama Mutlak Yüzde Hata (*MAPE*) değeri hesaplanarak değerlendirilmiştir. Sonuçlar, genel olarak SVM tabanlı ve M5P tabanlı modellerin diğer yöntemlerden daha iyi performans elde ettiğini göstermektedir.

Anahtar Kelimeler: Makine öğrenme, zaman serileri, İnternet tahmini, zaman gecikmeleri

DEDICATION

... to my mom and dad

ACKNOWLEDGMENTS

Foremost, I would like to express my sincere gratitude to my advisor Assoc. Prof. Dr. M. Fatih AKAY, for his supervision guidance, encouragements, patience, motivation, useful suggestions and his valuable time for this work.

I would like to thank members of the MSc thesis jury, Asst. Prof. Dr. Zehan KESİLMİŞ and Asst. Prof. Dr. Buse Melis ÖZYILDIRIM, for their suggestions and corrections.

I would also like to thank Cukurova University Scientific Research Projects Center for supporting this work (Project no: FYL-2015-5265)**.**

Last but not the least, I would like to thank my family for their endless support and encouragements for my life and career.

CONTENTS **PAGE**

LIST OF TABLES **PAGE**

LIST OF FIGURES **PAGE**

LIST OF ABBREVIATIONS

ANFIS	:	Adaptive Neuro-Fuzzy Inference System
ANN	:	Artificial Neural Network
AR	:	Auto-Regressive
ARIMA	:	Auto-Regressive Integrated Moving Average
ARMA	:	Auto-Regressive Moving Average
BP	:	Back Propagation
BPNN	:	Back Propagation Neural Network
CTSA	:	Chaotic Time Series Analysis
FARIMA	:	Auto-Regressive Fractionally Moving Average
FFNN	:	Feed-Forward Neural Network
FIRNN	:	Finite-Impulse-Response Neural Network
FIS	:	Fuzzy Inference System
FNT	:	Flexible Neural Tree
GA-RBF	:	Radial Basis Function optimized by Genetic Algorithm
GARCH	:	Generalized Auto-Regressive Conditional Heteroscedasticity
GM	:	Grey Model
GML	:	Gausian Maximum Likelihood
GP	:	Genetic Programming
HES	:	Holt-Exponential Smoothing
HM	:	Historical Mean
ISP	:	Internet Service Provider
KF	:	Kalman filtering
LLA	:	Local Linear Approximation
LM	:	Levenberg-Marquardt
LR	:	Linear Regression
LS-SVM	:	Least Squares Support Vector Machine

LSVM	:	Local Support Vector Machine
M5P	:	Decision Tree with Linear Regression Functions at the Nodes
MA	:	Moving Average
MAPE	:	Mean Absulote Percentage Error
MLP	:	Multi-Layer Perceptrons
MMLP	:	Multiresolution Multi-Layer Perceptrons
MMPP	:	Markov-Modulated Poison Process
MODWT	:	Maximal Overlap Discrete Wavelet Transform
NN	:	Neural Network
NNB	:	Nearest Neighborhood
NNE	:	Neural Network Ensemble
OK	:	Ordinary Kriging
OL-SVR	:	Online Support Vector Regression
OOB	:	Out-Of-Bag
RBF	:	Radial Basis Function Neural Network
REPTree	:	Reduced Error Pruning
RF	:	Random Forest
RP	:	Resilient Back Propagation
RT	:	Random Tree
SAE	:	Stacked Autoencoder
SARIMA	:	Seasonal Autoregressive Moving Average
SHW	:	Seasonal Holt-Winters
SPN	:	Spinning Network
SSVRCSA	:	Seasonal SVR with Chaotic Annealing Algorithm
SVM	:	Support Vector Machines
SVR	:	Support Vector Regression
SVRCACO	:	SVR with Continuous Ant Colony Optimization

TCP/IP	:	Transmission Control Protocol/Internet Protocol
TLFN	:	Time-Lagged Feed Forward Networks
TSF	:	Time Series Forecasting
WMRA	:	Wavelet Multi-Resolution Neural Network

1. INTRODUCTION

1.1. Internet Traffic Prediction

Internet traffic is the transfer of data across the Internet. Amount of Internet traffic is rising, and several and large number of packets are sent thorough all over the world (Hasegawa et al., 2001). Internet/network traffic analysis and modeling is an effective structure to characterize network performance; therefore, it has been a crucial point in many studies (Chen et. al., 2012).

Lately, since communication and network technologies have developed quickly, traffic characteristic is altering extremely. The research about Internet traffic analysis and modeling has varied from the large time scale to the small time scale. The researches have indicated that the traffic characteristics of the small time scale reacted differently from the large time scale's one (Chen et. al., 2012). Since the characteristic of Internet traffic is getting more and more sophisticated, this creates new difficulties to the management of the network. In order to overcome these problems, development of Internet/network traffic forecasting models have become one of the most active research areas. A traffic forecasting model needs to be able to express the characteristics of the network in the past and also, needs to be forecast the development of this network in the near future (Wang et al., 2008).

The predictability of Internet traffic is substantial point in many fields such as adaptive application, admission control, wireless and network management (Rutka and Lauk, 2015). To predict Internet traffic is very important to understand communication networks, optimize resources and have a better quality of service. As well, by comparing the real traffic with the forecast, anomalies (such as security attacks, viruses, etc.) can be detected with the help of traffic forecasting (Cortez et. al.,2007). Also, the predicted results can be used as a significant reference for the bandwidth allocation Internet traffic control and error control in management (Bai et. al.,2009).

The field of TSF considers the prediction of chronologically sorted predictors, where the aim is to identify a complicated system as a black-box and

forecasting its behavior based on historical data. The TSF approaches can be split into two parts as univariate and multivariate according to the number of variables used, i.e. one or more variables. Multivariate methods are expected to generate better results when the variables are correlated have been used (Cortez et al., 2007).

1.2. Previous Work

In (Hasegewa et. al., 2001), LLA, RBF, SVM were applied to develop Internet traffic models. First, these three methods were utilized to predict chaotic time series, and then a simple version of the LLA method was chosen because of easiness to apply and have high predictability. As a result, it has been observed that the nonlinear time series prediction method could be active for prediction of Internet traffic.

In (Sang and Li, 2002), the predictability of network traffic was assessed by using two metrics, namely the bounded error and minimum prediction error. The ARMA and MMPP methods have been used for assessment of the accuracy of the prediction models. The results have shown that proper traffic measurement and multiplexing improved accuracy for both ARMA and MMPP based prediction models.

In (Tong et. al., 2004), authors used boosting for network traffic prediction by considering it as a classical regression problem. The recommended algorithm took over FFNN as the basic learner, to acquire the non-linear characteristic of network traffic and increase using boosting to constrain overfitting. The results have shown that boosting by using FFNN is an effective method.

In (Alarcon-Aquiro and Barria, 2006), authors used multiresolution FIRNN based learning algorithm by applying the analysis into wavelet theory to predict network traffic. The MODWT and MMLP have been used in this study. The results have indicated that the network traffic prediction was improved by using FIRNN learning algorithm and MODWT.

In (Cortez et. al., 2006), authors used NNE, Naïve-Benchmark, Holt-Winters, ARIMA to develop TCP/IP network prediction models. The results have shown that NNE performed much better than other TSF methods.

In (Cortez et. al., 2007), NN, Holt-Winters, and Naïve-Benchmark have been utilized to build TCP/IP network forecasting models. It has been shown that NN-based models gave lower error rates than the ones obtained by the other methods. Also, it has been reported that Naïve-Benchmark based models gave the highest error rates TCP/IP network forecasting.

In (Wang et. al., 2008), authors have considered BP neural networks models to develop Internet traffic forecasting models by using GA-RBF and BP Neural Network. According to the results obtained, GA-RBF based models performed much better than BP Neural Networks.

In (Zhang and Liu, 2009), LS-SVM has been utilized to forecast Internet traffic. Also, authors have used five more machine learning methods including KF, ARMA, HM, RBF, and SVR for comparison. It was concluded that LS-SVM based models performed much better than the ones obtained by the other methods in different time scales such as a week, a day, and peak periods.

In (Jiang et. al., 2009), ARMA and FARIMA have been used for network traffic prediction. In this paper, the relation between time scales and time series models has not been examined. The results have shown that the performance of FARIMA based models did not give any advantage over the other method.

In (Huang and Sadek, 2009), the SPN has been utilized to forecast Internet traffic. Two more machine learning methods including BP and NNB algorithm have been used for comparison. The results have shown that the accuracy of SPN based models showed superior performance.

In (Chabaa et. al., 2009), authors have applied the ANFIS to forecast TCP/IP network. The authors concluded that ANFIS based model has been very effective to forecast Internet traffic.

In (Castro-Neto et. al., 2009), authors have used OL-SVR, which is an application of statistical regression method and three more machine learning methods including GML, HES, and ANN for comparison of Internet traffic forecasting

models. The results have shown that even though GML performed better than other methods in literature, in this study, OL-SVR has performed slightly better than GML.

In (Bermelon and Rossi, 2009), SVR has been used to forecast link load forecasting. The results have indicated that SVR based models have better result than the ones obtained by the other methods. That is, MA and AR were not good enough methods for link load forecasting.

Detailed information of the studies and methods used in each study between the years 2001 and 2009 are given in Table 1.1.

Table 1.1 Summary of studies between the years 2001 and 2009

Study	**Methods**
Hasegawa et. al. (2001)	LLA, RBF, SVM
Sang et. al. (2002)	ARMA, MMPP
Tong et. al. (2004)	Boosting with FFNN
Papagiannaki et. al. (2005)	WMRA, ARIMA
Alarcon-Aquino et. al. (2006)	FIRNN, MMLP, MODWT
Cortez et. al. (2006)	NNE, Naïve-Benchmark, Holt-Winters, ARIMA
Cortez et. al. (2007)	NN, Holt-Winters, Naïve-Benchmark
Wang et. al. (2008)	GA-RBF, BP
Zhang et. al. (2009)	LS-SVM, KF, ARMA, HM, RBF, SVR
Jiang et. al. (2009)	ARMA, FARIMA
Huang et. al. (2009)	SPN, BP, NNB
Chabaa et. al. (2009)	ANFIS
Chang et. al. (2009)	GM, ARMA, RBF, GARCH, ANFIS, ASVR-ANFIS/NGARCH
Castro-Neto et. al. (2009)	OL-SVR, GML, HES, ANN
Bermelon et. al. (2009)	SVR, MA, AR

LLA, Local Linear Approximation; **RBF**, Radial Basis Function Neural Network; **SVM**, Support Vector Machine; **ARMA**, Auto-Regressive Moving Average; **MMPP**, Markov-Modulated Poisson Process; **FFNN**, Feed-Forward Neural Network; **WMRA**, Wavelet Multiresolution Analysis; **ARIMA**, Autoregressive Integrated Moving Average; **FIRNN**, Finite-Impulse-Response Neural Network; **MMLP**, Multiresolution Multilayer Perceptron; **MODWT**, Maximal Overlap Discrete Wavelet Transform; **NNE**, Neural Network Ensemble; **NN**, Neural Networks; **GA-RBF**, Radial Basis Function optimized by Genetic Algorithm; **BP**, Back Propagation; **KF**, Kalman Filtering; **HM**, Historical Mean; **SVR**, Support Vector Regression; **FARIMA**, Autoregressive Fractionally Moving Average; **SPN**, Spinning Network; **NNB**, Nearest Neighborhood; **ANFIS**, Adaptive Neuro-Fuzzy Inference System; **GM**, Grey Model; **GARCH**, Generalized Autoregressive Conditional Heteroscedasticity; **ASVR-ANFIS/NGARCH**, ANFIS with nonlinear GARCH by adaptive SVR; **OL-SVR**, online support vector regression; **GML**, Gaussian maximum Likelihood; **HES**, Holt-Exponential Smoothing; **MA**, Moving Average; **AR**, Auto-Regressive.

In (Syed et. al., 2010), authors used the wavelet filters based SARIMA method to develop Internet traffic forecasting models. Authors also used simple SARIMA method to compare results of wavelet based SARIMA. The results showed that wavelet based SARIMA markedly improved the performance of forecasting models.

In (Chabaa et. al., 2010), MLP based models have been developed for Internet traffic forecasting. The LM and RP algorithms also have been applied for comparison purposes. It was concluded that LM and RP were very effective algorithms to prove accuracy of Internet traffic forecasting.

In (Tan et. al., 2010), the effectiveness of using ARMA based models on Internet traffic forecasting has been analyzed. The results of the study were positive and it has shown that the most accurate forecasting models have been obtained by step size of 30 seconds.

In (Hong et. al., 2011), SVRCACO based models have been developed for inter-urban traffic forecasting since SVR has been widely used in literature. The results have indicated that SVRCACO based models performed better than the ones obtained by SARIMA, which has been used for comparison.

In (Kim, 2011), authors used the AR-GARCH to develop Internet traffic forecasting models. The performance of AR-GARCH based models has been compared with the ones developed by ARIMA. The results have shown that AR-GARCH based models were more accurate than ARIMA based models.

In (Hong, 2011), authors used the SSVRCSA method to develop inter-urban traffic forecasting models. Also, SARIMA, BPNN, and SHW based models have been built for comparison purposes. The authors concluded that SSVRCSA based models yielded more accurate results than the ones obtained by the other methods.

In (Chen et. al., 2012), the FNT based models have been built for prediction of network traffic. FNT methods have been developed by using genetic programming. The results have indicated that FNT with genetic programming model was accurate to develop network traffic forecasting models and performed better than the FNT with the FFNN.

In (Miguel et. al., 2012), TLFN based models have been developed to predict long-term Internet traffic. Also, the authors used Holt-Winters to compare with TLFN. The authors concluded that TLFN based models gave more accurate results than the ones obtained by Holt-Winters.

In (Cortez et. al., 2012), the authors used Naïve-Benchmark, Holt-Winters, ARIMA, and ANN to forecast Internet traffic. It has been concluded that while ANN based models gave the best result for 5-minute and hourly data sets, Holt-Winters based models were more accurate than the ones obtained by the other method for daily data set.

In (Oliveira et. al., 2014), MLP and SAE have been utilized to build Internet traffic forecasting models. The results of MLP and SAE based models have been compared. The results have shown that even though SAE was a complex method, MLP based models performed much better than SAE based models.

In (Ratrout and Gazder, 2014), authors have used LR and ANN methods including MLP and RBF to develop Internet traffic forecasting models for daily data sets. Accuracy of the models has been examined in this paper. Authors concluded that ANN based models performed much better than LR based models. However; when ANN methods were examined separately, the performance of MLP and RBF based models were very close to each other.

In (Kamińska-Chuchmala, 2014), geostatistical estimation method called OK has been used for spatial Internet traffic load forecasting. Authors concluded that giving estimated values on whole considered area was an advantage of the OK method. On the other hand; the accuracy of OK based models was not good enough.

In (Rutka and Lauks, 2015), authors have used FFNN for prediction of network traffic. Also, the accuracy of the predicted traffic and estimated prediction interval has been examined. As a result, authors reported that numerical studies of real traffic traces to verify the prediction of real network traffic was not so easy.

In (Katris and Daskalaki, 2015), authors used FARIMA, two ANNs method including MLP and RBF, Holt-Winters, ARIMA/GARCH, FARIMA/GARCH, hybrid FARIMA+RBF, and hybrid FARIMA+MLP. It was concluded that the hybrid models performed much better than the other methods. On the other hand;

FARIMA/GARCH was more effective method than ARIMA/GARCH and Holt-Winters for Internet traffic forecasting.

In (Akgol et. al., 2015), SVM, MLP, RBF, and RT have been employed to forecast Internet traffic. The data sets in the study of (Cortez et. al., 2012) have been used in this study. Two different time lags for each time scale (5 minute, hourly, and daily) have been utilized to build Internet traffic models. The authors concluded that small time scale (5 minute) gave better results than the large time scale (daily) for all methods except RBF. Also, it was concluded that SVM and MLP based models performed better than the ones developed by the other methods while RBF based models performed worse than the ones developed by the other methods.

In (Akgol and Akay, 2016), authors have used different machine learning methods including SVM, MLP, RBF, RF, RT, and REPTree to forecast Internet traffic. The data sets in the study of (Cortez et. al., 2012) have been utilized in this study. Also, three different time lags have been used for each data set (5 minute and hourly) to develop Internet traffic forecasting models. *MAPE* has been used as a performance metric. The results have shown that SVM based models yielded lower *MAPE* values than the ones obtained by the other methods.

Detailed information of the studies and methods used in each study between the years 2010 and 2016 are given in Table 1.2.

Table 1.2 Summary of studies between the years 2010 and 2016.

Study	Methods
(Chen et. al., 2010)	BP-ANN
(Syed et. al., 2010)	Wavelet Filter based SARIMA, SARIMA
(Chabaa et. al., 2010)	MLP, LM, RP
(Tan et. al., 2010)	ARMA
(Hong et. al., 2011)	SVRCACO, SARIMA
(Liu et. al., 2011)	CTSA, LSVM
(Kim, 2011)	AR-GARCH, ARIMA
(Hong, 2011)	SSVRCSA, SARIMA, BPNN. SHW
(Chen et. al., 2012)	FNT-GP, FNT-FFNN
(Miguel et. al., 2012)	TLFN, Holt-Winters
(Cortez et. al., 2012)	Naïve-Benchmark, Holt-Winters, ARIMA, ANN
(Maurya et. al., 2012)	FIS
(Oliveira et. al., 2014)	MLP, SAE
(Ratrout et. el., 2014)	LR, MLP, RBF
(Kamińska-Chuchmała, 2014)	OK
(Rutka et. al., 2015)	FFNN
(Katris et. al., 2015)	FARIMA, MLP, RBF, Holt-Winters, ARIMA-GARCH, FARIMA-GARCH, hybrid FARIMA-RBF and FARIMA-MLP
(Akgol et. al., 2015)	SVM, MLP, RBF, RT
(Akgol et. al., 2016)	SVM, MLP, RBF, RF, RT, REPTree

BP-ANN, Back propagation Artificial Network; **SARIMA**, Seasonal Auto Regressive Moving Average; **MLP**, Multilayer Perceptron; **LM**, Levenberg-Marquard; **RP**, Resilient Back Propagation; **ARMA**, Auto Regressive Moving Average; **SVRCACO**, Support Vector Regression with Continuous Ant Colony; **CTSA**, Chaotic Time Series Analysis; **LSVM**, Local Support Vector Machine; **AR-GARCH**, Autoregressive-generalized conditional heteroscedascity; **ARIMA**, Autoregressive Integrated Moving Average; **SSVRCSA**, Seasonal Support Vector Regression with Chaotic Simulated Annealing; **BPNN**, Back-Propagation Neural Network; **SHW**, Seasonal Holt-Winters; **FNT-GP**, Flexible Neural Tree with Genetic Programming; **FNT-FFNN**, FNT with Feed Forward Neural Network; **TLFN**, Time-Lagged Feed Forward Networks; **ANN**, Artificial Neural Network; **FIS**, Fuzzy Inference System; **SAE**, Stacked Autoencoder; **LR**, Linear Regression; **RBF**, Radial Basis Function Neural Network; **OK**, Ordinary Kriging; **FARIMA**, Autoregressive Fractionally Integrated Moving Average; **RT**, Random Tree; **RF**, Random Forest; **REPTree**, Reduced Error Pruning.

1.3. Motivation, Purpose and Contributions of This Thesis

In literature, to the best of our knowledge, although there exists several studies which predict the network traffic with the help of statistical as well as

machine learning regression methods, there is no comprehensive study that compares the performance of different machine learning methods for prediction of network traffic on different data sets using several time lags.

The aim of this thesis is to extend the work of Cortez et. al. (2012) and forecast/predict the amount of traffic in TCP/IP-based networks by using various machine learning methods. The machine learning methods that have been employed are SVM, MLP, RBF, M5P, RF, RT, and REPTree and the statistical regression methods employed are MLR and Holt-Winters. The performance of statistical and machine learning regression methods has been compared in this thesis.

The main differences between this research proposal and the studies from related literature can be summarized as follows:

- This is the first study ever used trees as machine learning methods such as M5P, RF, RT, and REP tree to forecast Internet traffic.
- To the best our knowledge, there is no study that used different time lags in the same data set to develop new forecasting models. By using different time lags along with two data sets, this will be comprehensive study regarding the number of models to be developed for forecasting network traffic.
- In addition, the SVM method has not been examined according to the kernel type in any of the studies in literature. This is the first study that compares the performance of SVM with different kernel.

1.4. Overview of Data Sets

Two different ISPs traffic data (bits) have been utilized to build traffic forecasting models. Related with the time scales, the forecasting types can be explained as below (Cortez et. al., 2012):

- Real time; which includes data that not going beyond a few minutes and supposes an online forecasting system;

- Short term; from one to several hours, important to detect anomalies or control optimality;
- Middle term; from one to several days, employed to plan facilities;
- Long term; generally carried several months/years and used for strategic decisions.

Because of the characteristics of the Internet traffic collected, the data sets used in this thesis have been categorized as the first three types. Thus, three time series were formed for each data set by collecting all inputs in a given period of time. Each data set consists of different time scales including 5-minutes, 1-hour, and 1-day.

1st data set: The first data set (referred to as DS1) has been provided from a private ISP with centers in eleven European cities. The data set was saved between 07.06.2005, at 6:57 a.m. and 29.07.2005, at 11:17 a.m. The DS1 data was recorded every 30 seconds (Cortez et al., 2012).

2nd data set: The second data set (referred to as DS2) has been provided from United Kingdom education research networking association (UKERNA). The data set was saved between 19.11.2004, at 9:30 a.m. and 27.01.2005, at 11:11 a.m. The DS2 data was saved every 5 minutes (Cortez et al., 2012).

The graphic of the selected time scales are given in Figure 1.1, Figure 1.2, Figure 1.3, Figure 1.4, Figure 1.5, Figure 1.6, respectively.

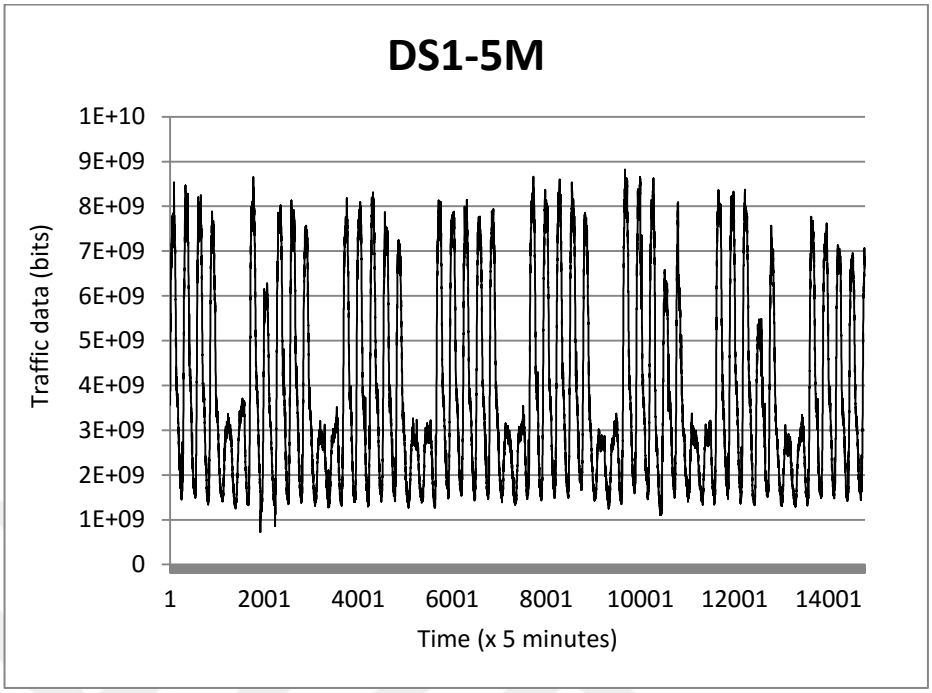

Figure 1.1. The Internet traffic time series for DS1-5M

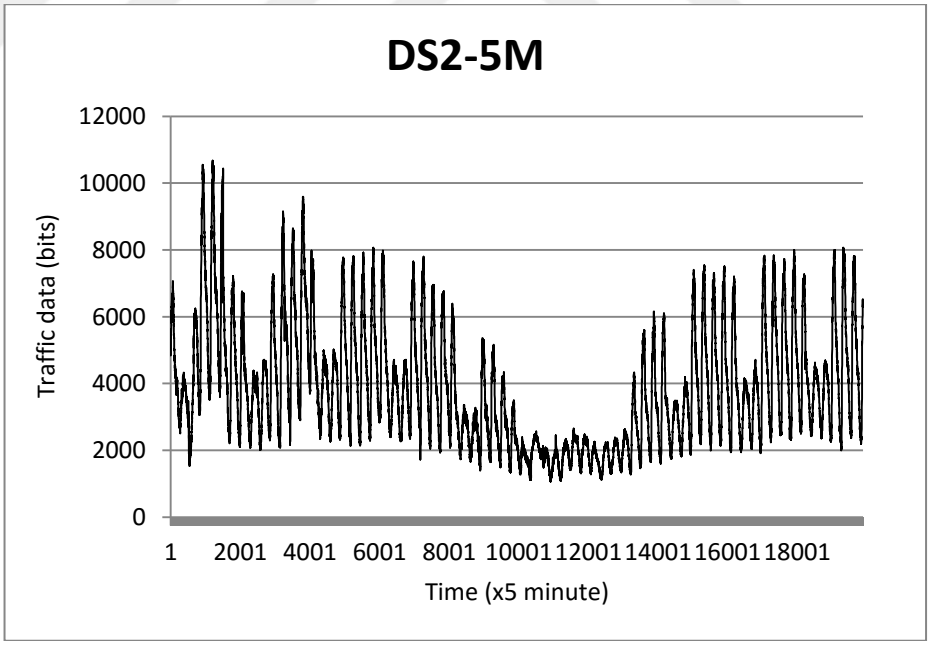

Figure 1.2. The Internet traffic time series for DS2-5M

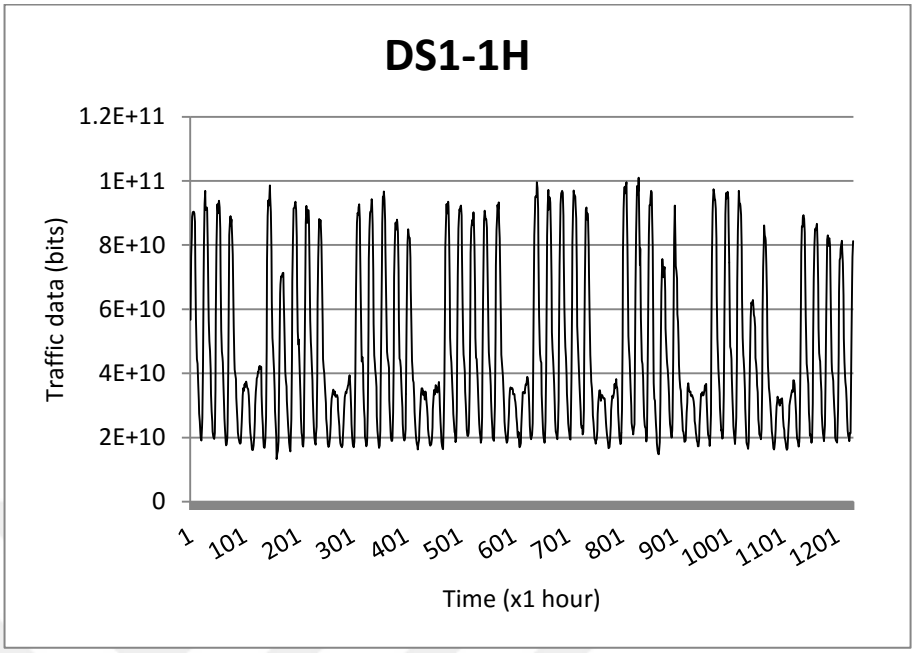

Figure 1.3. The Internet traffic time series for DS1-1H

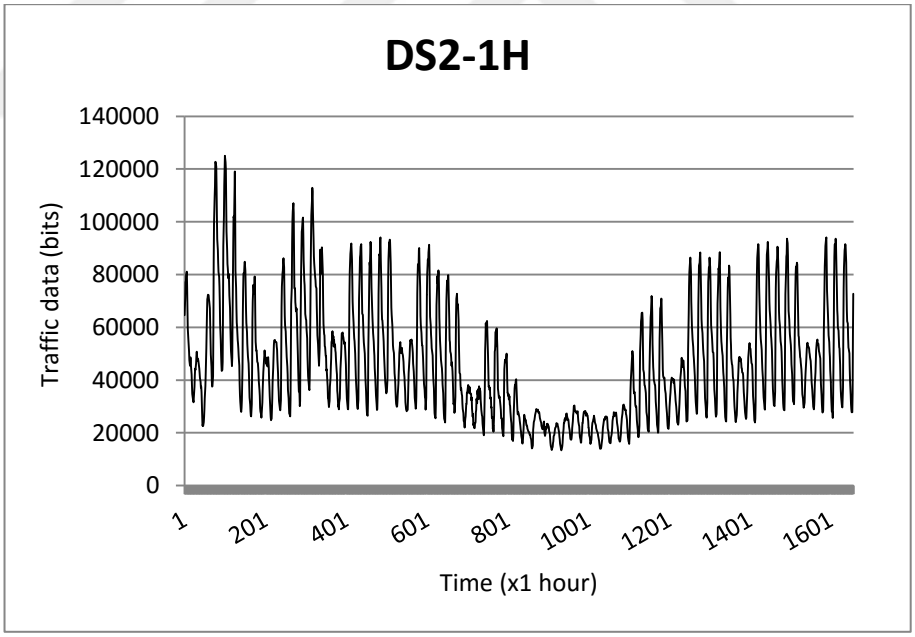

Figure 1.4. The Internet traffic time series for DS2-1H

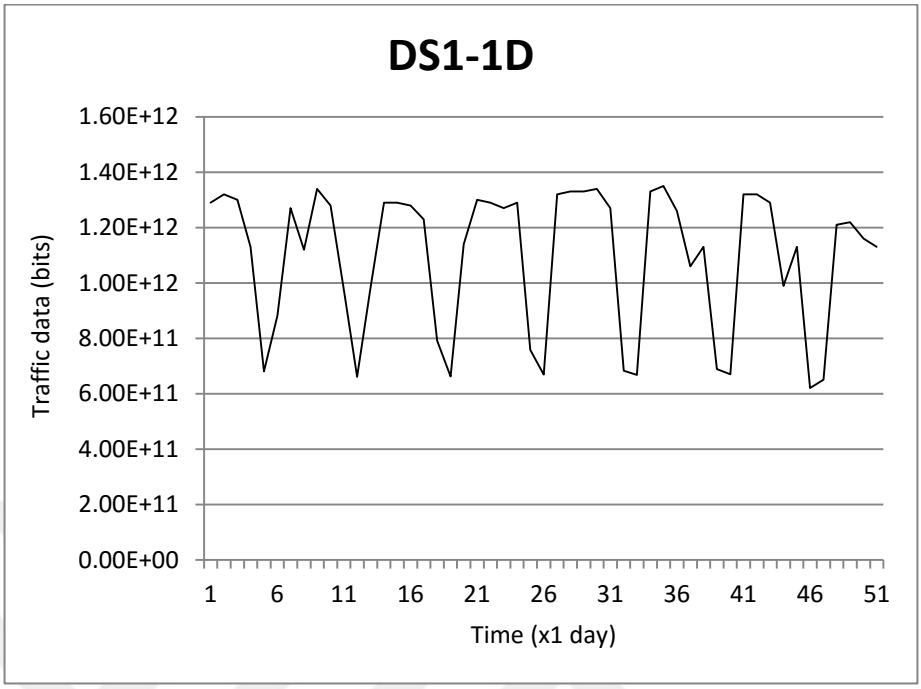

Figure 1.5. The Internet traffic time series for DS1-1D

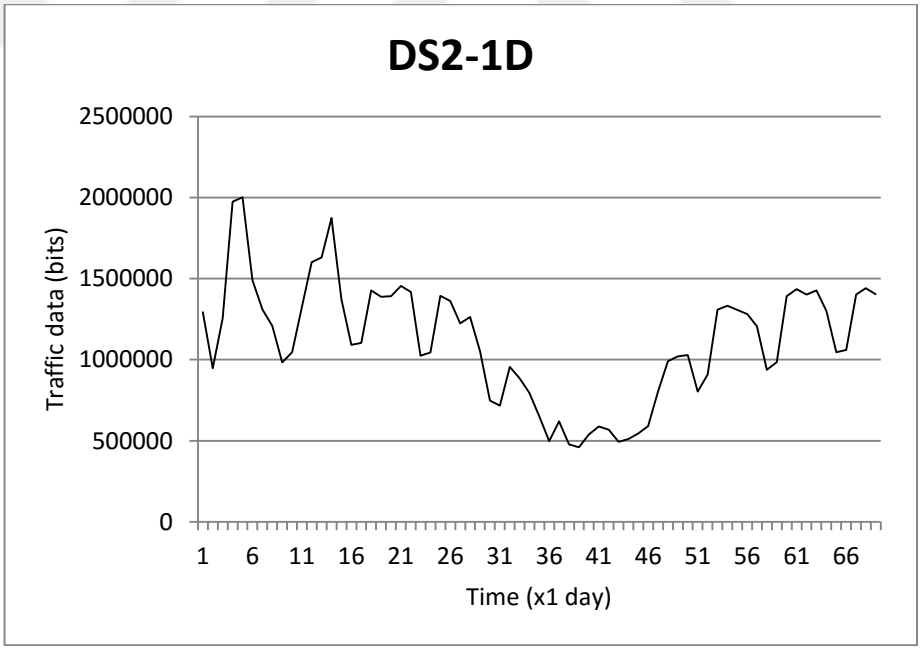

Figure 1.6. The Internet traffic time series for DS2-1D

2. OVERVIEW OF METHODS

2.1. Support Vector Machines

SVM is related to statistical learning theory (Vapnik, 1999), which was introduced in 1992 (Boser, 1992).

A. Linear SVM

Assume a training set of N data points, $S = \{x_k, y_k\}$ where $x_k \in R^n$ is an input vector and $y_k \in R$ is an output vector. The SVM problems are related with hyperplanes which separate the data. The equation of the hyperplanes are given using a vector v and a bias c. Decision function for the hyperplanes is $v^t x + c = 0$. The margin of separation ρ can be made maximum by constructing the optimal hyperplane. The support vector method technique aims at building a classifier

$$f(x) = sign(v^t.x + c). \quad (2.1.)$$

The v and c parameters are limited with

$$\min_i |v.x_i + c| \geq 1. \quad (2.2.)$$

After the vectors are divided without any problem, and as a result of this dividing process, if the space between the nearest vector and the hyperplane is maximum, it is called to be divided by a hyperplane. Consequently, a dividing hyperplane in standard form has to fulfill the constraints given in (2.3),

$$y_i (v.x_i + c) \geq 1, \ i = 1, 2, ..., n. \quad (2.3.)$$

A point x_i having the distance d from the hyperplane (v,c) is,

$$d\left((v,c),x_i\right)=\frac{y_i\left(x_i.v+c\right)}{\|v\|}\geq\frac{1}{\|v\|} \tag{2.4.}$$

ρ can be calculated as

$$\rho=\frac{2}{\|v\|}. \tag{2.5.}$$

Hence, SVM searches for a separating hyperplane by minimizing

$$\Phi(v)=\frac{1}{2}(v.v). \tag{2.6.}$$

$\Phi(v)$ in (2.6) can be minimized by performing the structural risk minimization principle,

$$\|v\|^2\leq c. \tag{2.7.}$$

h is the series of standard hyperplanes in space that has *n*-dimension and is limited by,

$$h\leq\min\left[\left(R^2c\right),d\right]+1, \tag{2.8.}$$

in which *R* is a hypersphere's radius surrounding all training vectors. As a result of this, minimizing (2.6) is equal to minimization of the upper bound.

The limitations of (2.3) can be reduced by presenting slack numbers $\zeta_i\geq 0, i=1,2,...,n$, therefore (2.3) can be rewritten as

$$y_i\left(vx_i+c\right)\geq 1-\zeta_i,\ i=1,2,...,n. \tag{2.9.}$$

Under these circumstances, the problem of optimization becomes

$$\Phi\left(v,\zeta\right)=\frac{1}{2}\left(v.v\right)+C\sum_{i=1}^{n}\zeta_i\ . \tag{2.10.}$$

In (2.10.) C is a user specified positive fixed constant. The saddle point of Lagrangian function is utilized in the solution of the problem given in (2.10).

$$L\left(v,c,\alpha,\zeta,\gamma\right)=\frac{1}{2}\left(v.v\right)C\sum_{i=1}^{n}\zeta_i-\sum_{i=1}^{n}\alpha_i\left|y_i\left(v.x_i+c\right)-1+\zeta_i\right|-\sum_{i=1}^{n}\gamma_i\zeta_i\ . \tag{2.11.}$$

In (2.11), $\alpha_i\geq 0,\zeta_i\geq 0,i=1,2,...,n$ are Lagrange multipliers. (2.11) must be solved in terms of v, c, and ζ_i. Classical Lagrangian duality empowers the first issue, turning (2.11) into a dual problem of it and this makes the solution easier. (2.12) shows the dual problem to be solved

$$\max_{\alpha}\left[\sum_{i=1}^{n}\alpha_i-\frac{1}{2}\sum_{i=1}^{n}\alpha_i\alpha_j y_i y_j\left(x_i x_j\right)\right] \tag{2.12.}$$

with constraints

$$\sum_{i=1}^{n}\alpha_i y_i=0,\ 0\leq\alpha_i\leq B,\ i=1,2,...,n \tag{2.13.}$$

There is a unique solution for this classic quadratic optimization problem. In respect of the optimization theory which is called Kuhn-Tucker theorem (2.16.),

$$\alpha_i \left[y_i \left(v.x_i + c \right) - 1 \right] = 0,\ i = 1,2,...,n \qquad (2.14.)$$

If x_i satisfy (2.15), then (2.14) will have non-zero Lagrange multipliers

$$y_i \left(v.x_i + c \right) = 1. \qquad (2.15.)$$

The subjects in (2.15.) are called support vectors (SV's). The hyperplane is determined with a small subset of the training vectors of the SV's. Therefore, if the optimal solution α_i^* does not take a value of zero, the classifier function is states by

$$f\left(x\right) = \operatorname{sgn}\left\{ \sum_{i=1}^{n} \alpha_i^* y_i \left(x_i.x \right) + c^* \right\}. \qquad (2.16.)$$

In (2.16.) c^* is the answer of (14) for any $\alpha_i^* \neq 0$.

B. Non-linear SVM

The majority of the data sets cannot be decently divided by a linear separating hyperplane. However, they can be linearly divided if mapped into a higher dimensional field by utilizing a nonlinear mapping. Therefore, $z = \phi(x)$ that converts the input vector x having a dimension *d* into a vector z having a dimension d is defined and $\phi(\ )$ is selected so that $\{\phi(x_i, y_i)\}$ (new training data) is divisible with a hyperplane.

The data points from the input space into some space of higher dimension are mapped by using the function

$$\vartheta(.) : R^n \to R^{nh}. \qquad (2.17.)$$

Optimal function (2.9) transforms (2.18) using the same constraints,

$$\max_{\alpha}\left[\sum_{i=1}^{n}\alpha_i-\frac{1}{2}\sum_{i,j=1}^{n}\alpha_i\alpha_j y_i y_j K\left(x_i,x_j\right)\right] \tag{2.18.}$$

where

$$K\left(x_i,x_j\right)=\left\{\vartheta\left(x_i\right).\vartheta\left(x_j\right)\right\} \tag{2.19.}$$

is the kernel function.

The RBF kernel function is given by (2.20.)

$$K\left(x_i,x_j\right)=\exp\left\{-\mu\left\|x_i-x_j\right\|^2\right\}, \tag{2.20.}$$

whereas the polynomial kernel function is given by (2.21)

$$K\left(x_i,x_j\right)=\left(x_i.x_j+1\right)^p,\ p=1,2,... \tag{2.21.}$$

where the values σ and p in (2.20) and (2.21) have to be prearranged.

Then, the classifier function is specified as

$$f\left(x\right)=\operatorname{sgn}\left\{\sum_{i=1}^{n}\alpha_i^* y_i K\left(x_i,x\right)+b^*\right\} \tag{2.22.}$$

and b^* is the answer of (2.22) for any $\alpha_i^*\neq 0$.

2.2. Multi-Layer Perceptron

An MLP is a type of artificial neural network model and also feed-forward process. This model maps a set of input data over a set of convenient outputs. An MLP includes multiple layers of nodes in a coordinated graph and each layer is completely connected to the following one. Each node is a neuron with a nonlinear activation function except the input nodes. This method uses a handled learning technique. This technique is named as back propagation in order to train the network. MLP is an alteration of the standard linear perceptron and can recognize inseparable data (Imrie and Durucan, 2000). An MLP has a form as given Figure 2.1.

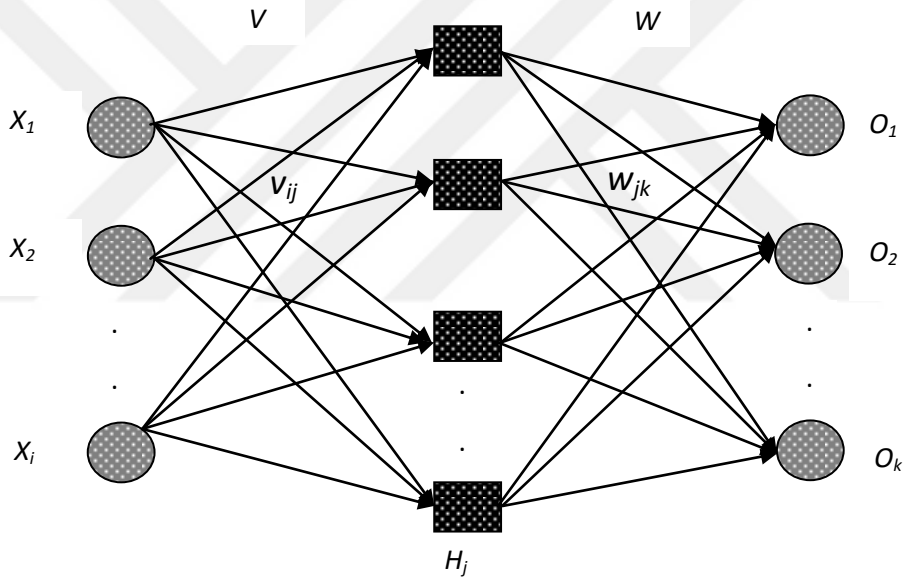

Figure 2.1. A typical MLP Structure

A data model consisting of the x_i values used in the input layer i^{th} is transmitted through the network towards j in the first hidden layer. The weighted outputs $w_{ij}x_i$ are received of the previous layer's units per hidden unit. The outputs from this process are summed. After that, these values are to be turned into an output value using an activation function.

Activation function is $f^{(m)}(x)$ at layer m. The output unit has two-layers.

$$out_k^{(2)} = f^{(2)}\left(\sum_j out_j^{(1)}.w_{jk}^{(2)}\right) = f^{(2)}\left(\sum_j f^{(1)}\left(\sum_i in_i w_{ij}^{(1)}\right).w_{jk}^{(2)}\right) \qquad (2.23.)$$

If the activations at hidden layer are linear, (2.23.) is reduced to

$$out_k^{(2)} = f^{(2)}\left(\sum_j out_j^{(1)}.w_{jk}^{(2)}\right) = f^{(2)}\left(\sum_j f^{(1)}\left(\sum_i in_i w_{ij}^{(2)}\right)\right). \qquad (2.24.)$$

Nevertheless, (2.24) is equal to a network having one layer with weight $w_{ik} = \sum_j w_{ij}^{(1)} w_{jk}^{(2)}$. This network cannot be used on non-linearly separable problems.

A. Non-Linear Activation/Transfer Function

The values of the logistic sigmoid function range between 0 and 1. The standard sigmoid is basically used with the hyperbolic tangent. It has the feature,

$$f(x) = \tanh(x) = 2sigmoid(2x) - 1, \quad f(-x) = -f(x), \qquad (2.25.)$$

its derivative is given by (2.26.)

$$f'(-x) = 1 - f(x)^2. \qquad (2.26.)$$

B. Learning

The same steps are used for training $N-layer$ neural networks as the networks having a single layer. The network weights $w_{ij}^{(m)}$ are set to make the output cost function given in (2.27) minimum,

$$E_{SSE} = \frac{1}{2}\sum_{p}\sum_{j}\left(t\arg_j^p - out_j^{(N)p}\right)^2, \qquad (2.27.)$$

or

$$E_{CE} = -\sum_{p}\sum_{j}\left[t\arg_j^p .\log\left(out_j^{(n)p}\right) + \left(1 - t\arg_j^p\right).\log\left(1 - out_j^{(N)p}\right)\right] \qquad (2.28.)$$

and once again the changes of a group of gradient descent weight can be done by using (2.29.)

$$\Delta w_{kl}^{(m)} = -\eta \frac{\partial E\left(\left\{w_{ij}^{(n)}\right\}\right)}{\partial w_{kl}^{(m)}}. \qquad (2.29.)$$

$out_j^{(N)}$ is the only output of the last layer. This becomes apparent in the error function E of the output. However, outputs of the final layer are related with all the weights of the previous layers. This learning algorithm automatically sets $out_j^{(n)}$ of the previous layers.

C. Training

Training for multi-layer networks performs in same way with networks having a single layer:

1. Assume that we have the series of training patterns given below in (2.30)

$$\left\{in_i^q, out_j^q : i = 1...m\,inputs,\ j = 1...m\,outputs, q = 1...m\ patterns\right\} \qquad (2.30.)$$

2. Network is set up with *M* inputs, *M-1* hidden layers of $m hidden^{(m)}$ nonlinear numbers, and *M* outputs in layer *M*. Each layer is completely connected with the previous layer by weight $w_{ij}^{(n)}$.
3. First weights are generated randomly in the range $[-smwt, +smwt]$.
4. The learning rate η and a convenient error function $E\left(w_{jk}^{(n)}\right)$ are selected.
5. For each training pattern p, the equation for weight $\Delta w_{jk}^{(n)} = -\eta \partial E\left(w_{jk}^{(n)}\right) \Big/ \partial w_{jk}^{(n)}$ is applied to every weight $w_{jk}^{(n)}$.
6. Step 5 is repeated until the network's error function gives negligible error rates.

Algebraical statements need to be reproduced for the weight updates in order to be practical.

2.3. Radial Basis Function Network

The RBF's appeared as a variant of artificial neural network in the last years of 1980. Nonetheless, their roots are settled in much more pattern recognition techniques. These techniques can be clustering, mixture models, potential functions, sp line interpolation and functional approximation (Park and Sandberg, 1991).

RBF network includes several layers and the first layer has input neurons. These neurons feed the feature vectors into the network. The second one is the hidden layer that calculates the result of the basic functions. The last layer is the output layer that calculates a linear union of the basic functions. These kinds of networks have the general estimation property (Park and Sandberg, 1991). Simple structures of these networks give a decreasing for training time and make possible learning in stages.

RBF has the feed-forward structure. It has one layer that consists of hidden units. These units are completely adjusted with the output units. (2.31) indicates the

output values (μ_k) from a linear union of the basic functions (Ghosh-Dastidar et al., 2008).

$$\mu_k(x) = K\left(\frac{\|x - c_k\|}{\sigma_k^2}\right) \tag{2.31.}$$

$f: R^n \to R^1$ (a function) is estimated by using an RBF network. $x \in R^n$ (a vector) is an input, $\mu(x, c_k, \sigma_k)$ is the j^{th} function with central $c_k \in R^n$, width σ_j, and $v = (v_1, v_2, ..., v_m) \in R^M$ is the vector of linear output gravities. *M* represents the basic numeral function. The *M* centrals $c_k \in R^n$ are connected to obtain $c = (c_1, c_2, ..., c_m) \in R^{nM}$. Lastly, the widths are connected to obtain $\sigma = (\sigma_1, \sigma_2, ..., \sigma_m) \in R^M$. The function of output where $x \in R^n$ and $\sigma \in R^M$ is given by (2.32)

$$F(x, c, \sigma, \omega) = \sum_{k=1}^{M} \left(\omega_k \psi(x, c_k, \sigma_k)\right). \tag{2.32.}$$

Suppose that $y = (y_1, y_2, ..., y_n)$ is the weighted output vector and $(x_j, y_j): j = 1, 2, ..., N$ is a series of training pairs. For each $c \in R^{nM}$, $\omega \in R^M$, $\sigma \in R^M$ and for random gravities $\lambda_j, j = 1, 2, ..., N$, which are taken as positive digits to point out importance of definite domains of the input space, write (2.33) (Wright et al., 2013).

$$E(c, \sigma, \omega) = \frac{1}{2}\sum_{j=1}^{N}\left[\lambda_j\left(y_j - F(x_j, c, \sigma, \omega)\right)\right]^2 \tag{2.33.}$$

2.4. M5P

M5 tree is developed by Quinlan (1992) as a new type of tree to predict continuous variables. Multivariate linear models can be formed by the trees developed by M5 method which provides more adjustable predictions.

Three major steps which are tree construction, tree pruning, and tree smoothing have been developing in M5 tree. The flow chart of M5 tree is shown in Figure 2.2.

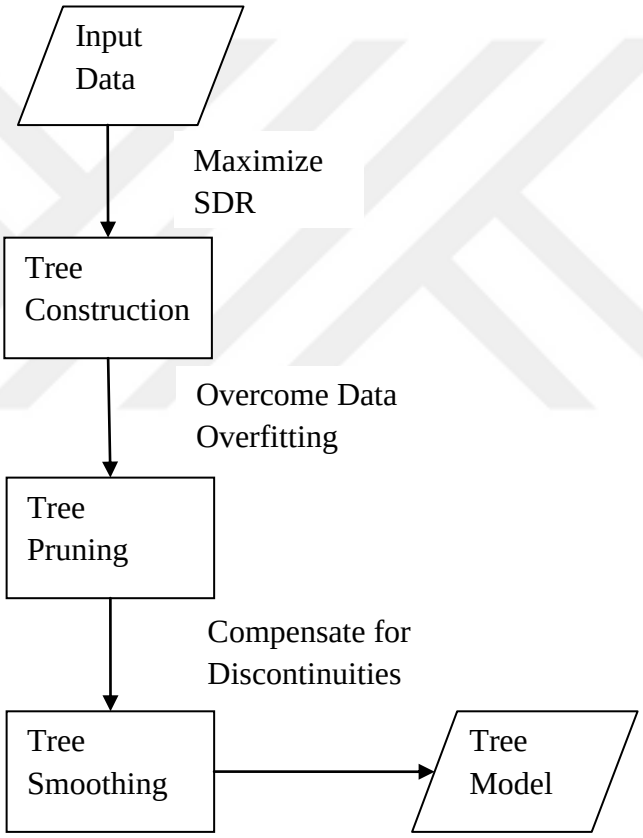

Figure 2.2. M5 tree flowchart

Process of M5 tree structure aims to increase value of the standard deviation reduction (SDR) which formula is given below by (2.34.)

$$SDR = sd(T) - \sum_i \frac{|T_i|}{|T|} \times sd(T_i) \quad (2.34.)$$

In (2.34.), T represents the set of conditions, T_i represents the i^{th} subset of conditions that outcome from tree splitting based on a set of attributes, $sd(T)$ is T's standard deviation, and $sd(T_i)$ is T_i's standard deviation as an error.

Thereafter, M tree has been built first, tree pruning is done to eliminate or aggregate the subtrees that not needed to result data with minimizing error which was occurred during building the tree, and the algorithm of M5 tree assesses a linear regression model for every inner unpruned tree's loop which used standardized regression and by assigning attributes in the subtree. During the tree pruning, variables have been reduced one after another to reduce the predicted error till the root loop is obtained. Tree has been pruned in M5 tree algorithm to fix discontinuities which is formed between contiguous linear models at the pruned tree's leaves, and then a smoothing process has been employed.

The original M5 tree algorithm was improved by Wang and Witten (1997) to deal with countable attributes and the attributes that have missing values. This new tree algorithm was called as M5P algorithm. Each and every countable variable is mapped into binary attributes before tree has been constructed in the M5P tree algorithm. As well the formula of SDR value for improved algorithm considering missing values is given below by (2.35.):

$$SDR = \frac{m}{|T|} \times \beta(i) \times \left[sd(T) - \sum_{j \in \{L,R\}} \frac{|T_j|}{|T|} \times sd(T_j) \right] \quad (2.35.)$$

such that m represents the amount of training situations that do not include missing costs for a variable, T represents the training set for a node, $\beta(i)$ is the correction factor for countable variables, and T_L and T_R determine outcomes from the division of a variable's subsets (Zhan et. al., 2001).

2.5. Random Forest

RF was built by Leo Breiman (2001). A group of unpruned classification or regression trees which are formed by the random choice of samples of the training data is created with the RF (Ali et. al., 2012).

RF makes use of bagging and random feature selection, which are two important machine learning methods. In bagging, every tree is practiced on the training set's bootstrap case and estimations are generated by a large number of trees. RF is a method that is an extension of bagging. While growing a tree, a subset of features is selected randomly by RF instead of using all features to divide into each node. To evaluate the prediction performance of the RF algorithm, RF employs a sort of cross-validation correspondingly to training process by utilizing OOB cases. Particularly, at the training, a certain bootstrap sample has been used during growth of each tree. Since bootstrapping is exemplifying by replacing with the training set, a certain set of the sequence is repeated in the case when others is "left out" of the samples. The "left out" ranks generate the OOB case. In the mean, while each tree is grown, $1-e^{-1} \cong 2/3$ of the training sequences has been used and $e^{-1} \cong 1/3$ has been left as OOB. Since OOB ranks haven't been employed during construction of the tree, they can be employed to assess the estimation performance (Jiang et. al., 2007).

RF algorithm is shown below;

1. Choose m_{tree} bootstrap cases by using the main data.
2. For every case, raise an *unpruned* regression or classification tree by regulating sequent change: in every node, instead of selecting the best division between all predictors, say n_{try} of the predictors, select the best divison through those attributes.
3. Guess new data by collecting predictions of the m_{tree} trees (that is, if it is a classification problem, then choose majority, if it is a regression problem, then choose average).

The estimated error rate could be attained, according to the training set as below;

1. At every bootstrap repetition, guess the data in the OOB using the tree risen with the bootstrap sample.
2. Collect the OOB estimations. Figure out the error rate, and define it as the OOB estimate ones (Liaw and Wiener, 2002).

2.6. Random Tree

The RT algorithm makes a random motion designing method that is developed for nonholonomic robots. Nonholonomic means that the vector space of the feasible motion ways at a specific structure is constrained where the restraint is not transformed to an algebraic relation among structural variables. Since the direction of the automobile displacement is stated by its position, distinct kinematic types of an automobile are ordinary instances of a nonholonomic system.

The algorithm of RT generates a graph where nodes of it are distinct structures of the robot. The sides of the graph are equal to the applicable traces between the structures. The robot's control entries are kept for each side which determines the way to get on structure to its neighbor. The motion designing's outcome is a trace in this graph between beginning and the aim structure.

The algorithms of RT are developed to configure trace designing problems.

The process is to determine a trace in the structure space S from a specific initial configuration refers as s_{start} to a target structure refers as s_{target} (or to target area such that $S_{target} \subset S$). Drawbacks also have been in the outboard, they describe the stable obstacle region $S_{obs} \subset S, \left(s_{start}, s_{target}\right) \in S \setminus S_{obs}$. The trace ought to abstain them; it has to appear in the free structure space S_{free} that is the complement of S_{obs} .

A case equation of the form $\dot{y} = g(y,u)$ is dedicated to state the nonholonomic restraints where y refers the cases, u determine the control entry, and dot symbol implies the derivative with respect to time.

The RT algorithm is given in Figure 2.3. In every stage of the repetition, a new edge has been used to enlarge the tree *K*. The new edge is called as s_{new} on the base of a structure s_{rand} which is chosen randomly. The structure s_{rand} is an elective structure in S_{free}. The first stage, the closest edge s_{near} to s_{rand} be chosen in *K* which is aimed to indicate the closest structure a metric is needed. After that, the control entry is chosen from a finite set of acceptable entries expressed by U. Then the chosen control entry which is expressed with $u_s\,(u_s \in U)$ the tract from s_{near} to the way of s_{rand}. The new structure s_{new} is built by applying the input u_s for some time increment Δt from s_{near} or different point of the trace from s_{near}. If a clash happens at s_{new} (that is $s_{near} \in S_{obs}$), or different point of the trace from s_{near} to s_{new} is a point in S_{obs}, then another u_s has to be chosen. Thus, the edge that linked s_{new} to the tree is (s_{near}, s_{new}, u_c). If the aim area or the aim structure is attained $(s_{new} \in S_{target})$, then the algorithm have been finished and trace from s_{start} to s_{target} s_{target} (or S_{target}) is determined.

If nodes' number in the tree attains the maximum boundary (B) before obtained s_{target} or S_{target}, the algorithm drops. The trace between the initial structure and the target structure cannot be found.

If the number of nodes in the tree reaches the maximum limit (N) without finding s_{target} or S_{target}, the algorithm fails, and it cannot find the path connecting the initial configuration to the goal (Szadeczky—Kardoss and Kiss, 2006).

T=GENERATE_RT $(s_{start,}, s_{target}, N, \Delta t)$

(1) Initialize of the tree *K* with s_{start}
(2) **For** $i = 1$ to *N* **do** (until the maximum number of nodes is reached)
(3) Select a random configuration: s_{rand}
(4) Select the nearest neighbor of $s_{rand} : s_{near}$
(5) Select the input from s_{near} to the direction of $s_{rand} : u_s$
(6) Generate a new state by using the input u_s at configuration s_{near} for $\Delta t : s_{new}$
(7) Add s_{new} to the tree *K*
(8) Add the new edge (s_{near}, s_{new}, u_c) to the tree *K*
(9) **if** the goal configuration s_{target} or the target region S_{target} is reached
then exit: $i = N$
End
(10) End
(11) return *K*

Figure 2.3. The steps of the basic RT algorithm.

2.7. Reduced Error Pruning Tree:

REPTree was developed by Quinlan (1987). REPTree algorithm is established upon the form of estimating the information gain with entropy. Also, it decreases the error that is occurred by variance.

REPTree is a tree which is a data structure that uses divide and conquer approach. Supervised learning has been used in REPTree. It is a tree such that model in which the local region is existed consecutively is built with a set of division in a few steps. It includes inner decision node and outer leaf. Each decision node confirms $f_m(x)$ test function whose pure value is connected to branches. Test function is applied in every node for an input and one of the branches is chosen with respect to outcome. This action begins in a root and goes on consecutively till a leaf node is obtained; the output is generated by the value written on the leaf.

REPTree is one of the most common used classifiers on classifying problems. It is an easy way to constituted compared to the other methods.

Let Y and X be the discrete variables such that $\{x_1,...,x_n\}$ and $\{y_1,...,y_n\}$. The calculation of entropy and conditional entropy of Y is given by (2.36.) and (2.37.). Also, the formula for information gain of X is given by (2.38.):

$$H(Y) = -\sum_{i=1}^{k} P(Y = y_i) \log P(Y = y_i) \tag{2.36.}$$

$$H(Y \setminus X) = -\sum_{i=1}^{l} P(X = x_i) H(Y \setminus X = x_i) \tag{2.37.}$$

$$IG(Y; X) = H(Y) - H(Y \setminus X) \tag{2.38.}$$

In REPTree, pruning is performed in two ways such that prepruning and postpruning. If the number of instances attained a node whose percentage is lower than the training percentage ones, then that node is not splitted. Thus the variance of the model is formed by training data with a small number of instances and this causes increment in the generalization error. Therefore during the growing the tree, extension of the tree is terminated. This process is called prepruning.

The second way building tree is postpruning as said before. In general, this way perform better than prepruning since the tree does not make progress in backward, expansion of tree is kept to go on during a tree which causes obtaining high value of variance. In postpruning, first the subtrees not used are discovered and pruned which leads to keep away the situation occurred in prepruning.

In postpruning, the tree is enlarged until all the leaves are discrete and error in training set is not obtained. Then subtrees that stored are found and pruned. To do that, first, majority of the training set is used as growing set and the rest of them is used as a pruning set. After that each subtree is changed with a leaf which is trained by the instanced in the training set of that subtree and these two results are compared. If the leaf does not reveal high error's value on the pruning set, then subtree is pruned and that leaf is used, else subtree is stored. By comparing prepruning and postpruning, it is observed even though prepruning produces tree faster, postpruning generates better trees (Zontul et. al., 2013).

2.8. Multiple Linear Regressions:

The simple linear regression model is expanded to associate two or more illustrative attribute in an estimation equation for an output attribute. Nowadays, MLR is the backbone of statistical analysis in most fields since the MLR is very impressive and flexible method. Also, the MLR spends not too much effort to predict complex models which have too many attributes.

Generally, the MLR has been used to give a form for a continuous respond attribute. There exists only one continuous entry attributes for simple linear regression. The MLR method has been made general by using at least two attributes which can be continuous or categorical. The MLR is a general form of the evaluation of variance and covariance.

In MLR, the basic model is the following equation given by (2.39.):

$$z_i = \lambda_0 + \lambda_1 Y_{j1} + \lambda_2 Y_{j2} + ... + \lambda_m Y_{jm} + \varepsilon_j \qquad (2.39.)$$

Assume that ε_j, the error rate, is normally scattered where a mean 0 and standard deviation σ . In equation (2.39.), z_j refers the respond for j and $Y_{j1}, Y_{j2}, ..., Y_{jm}$ determine m entry attributes. The respond attribute z_j is dependent to the entry attributes $Y_{j1}, Y_{j2}, ..., Y_{jm}$ which are independent attributes. The following is up to be continuous or nominal. Since the entry variables, Y 's cannot be independent of one another other, the use of "independent" is wrong. Every entry attributes are relevant to a regression parameter $\lambda_1, \lambda_2, ..., \lambda_m$. The λ_0 is an additive stable parameter. The λ 's are the model terms.

The equation (2.39.) can be written as given by (2.40.):

$$LP_j = \lambda_0 + \lambda_1 Y_{j1} + \lambda_2 Y_{j2} + ... + \lambda_m Y_{jm} \qquad (2.40.)$$

where LP_j is known as the linear predictor and is the value of z_j predicted by the input variables. The difference $z_j - LP_j = \varepsilon_j$ is the error term.

The estimates $\lambda_0, \lambda_1, ..., \lambda_m$ have been selected as being compatible with the models to minimize the sum of squares of the predicted error. These estimates are termed ordinary least squares estimates. The fitted variables z_j^{fit} can be calculated by employed these estimates and the residuals is remarked as $e_j = z_j - z_j^{fit}$. Here it is clear that the residuals estimate the error term (Draper and Smith, 2014).

Uses of MLR can be summarized as follows;

1. To set impact of an entry attributes on a continuous respond attribute. For instance, to observe the influence of a diet on weight by considering the smoking uses. In these, the results come from a clinical test will be the dependent attributes. Smoking (can be selected the numbers of packs per week for a continuous attributes) and baseline weight can be the independent attributes as two groups. The results of these two groups are compared by using MLR method. The differences between weight and smoking uses can be set by MLR method. Doing this is also called as analysis of covariance.
2. To examine synchronal influence of categorical attributes on a respond attribute.
3. To estimate a value of a respond, for given entries.

2.9. Holt-Winters:

The Holt-Winters is a crucial forecasting method such that the prediction model consists of the trended and seasonable samples. Seasonable and trended samples are separated from the unnecessary attributes by meaning the historical rates. It has some benefits like easiness of using it, having less computation, and more accurate results for seasonal series. The equation of Holt-Winters-based models is shown by (2.41.):

$$\text{Level} \qquad N_m = \delta \frac{z_m}{C_{m-L_1}} + (1-\delta)(N_{m-1} - W_{m-1})$$

$$\text{Trend} \qquad W_m = \lambda (N_m - N_{m-1}) + (1-\lambda) W_{m-1} \tag{2.41.}$$

$$\text{Seasonality} \qquad C_m = \zeta \frac{z_m}{N_m} + (1-\zeta) C_{m-L_1}$$

$$\hat{z}_{m+h,m} = (N_m + hW_m) \times C_{m-L_1+h}$$

such that h is the lead time period, N_m, W_m and C_m determine for the level, trend and seasonal calculations, L_1 is the seasonal period, and δ, λ, and ζ refer to the model values. If a seasonal component does not exist, then the ζ is dismissed and the C_{m-L_1+h} is switched by unity.

Nowadays, Holt-Winters has been developed to include two seasonal cycles. The formula for the extended case is given by (2.42.):

$$\text{Level} \qquad N_m = \delta \frac{z_t}{C_{m-L_1} V_{m-L_2}} + (1-\delta)(N_{m-1} - W_{m-1})$$

$$\text{Trend} \qquad W_m = \lambda (N_m - N_{m-1}) + (1-\lambda) W_{m-1}$$

$$\text{Seasonality 1} \qquad C_m = \zeta \frac{z_m}{N_m V_{m-L_2}} + (1-\zeta) C_{m-L_1} \tag{2.42.}$$

$$\text{Seasonality 2} \qquad V_m = \phi \frac{z_m}{N_m C_{m-L_1}} + (1-\phi) V_{m-L_2}$$

$$\hat{z}_{m+h,m} = \left(N_m + hW_m\right) \times C_{m-L_1+h} V_{m-L_2+h}$$

such that V_m determines the second seasonal calculation, L_1 and L_2 refer the first and second seasonal intervals, and ϕ is the second seasonal value.

The initial values for the level, trend, and seasonal predictions are adjusted by considering the former investigations. The values of the parameters δ, λ, ζ and ϕ are optimized by using grid search. The aim of this method is to get minimum error rate (Cortez et. al., 2012).

3. DEVELOPMENT OF PREDICTION MODELS

3.1. SVM Model for Forecasting Internet Traffic

The accuracy of an SVM regression model depends on the value of C, type of the kernel function and parameters of this function. There is a tradeoff between reducing the error related to training and reducing complexity of model and this is stated by parameter C. A small value of C will increase the quantity of training error sand a large value of C will exhibit an attitude similar to that of a hard-margin SVM. The parameter ε controls the width of ε-insensitive zone. ε states the number of support vectors. The kernel parameters are important in the sense that they act as a bridge between the input space and the high-dimensional feature space (Ji and Wang, 2007). In this thesis, the polynomial and RBF kernel have been utilized separately during developing SVM models.

There is no way to determine in advance which C, ε, and γ are the ideal for a regression problem. Hence some kind of parameter search has to be manipulated. The aim is to achieve optimal values of the parameters (C, ε and γ) so that the testing data can be estimated with minimum error by the regression model. There exist many methods that can be used for deciding the values of the optimal parameters. The grid search is the most effective one for achieving the optimum values of C, ε, and γ on median-sized problems. In this method, the parameters are differentiated by constant step-sizes through an interval of values, and the success of each value of these parameters is evaluated. A cross validation within the grid search can be utilized in order to develop the generalization capability of the SVM regression model. In the process of k-fold cross validation, the cross validation is recurred k times (this means k folds), with each of the k subsets utilized precisely once as the validation data. The k results from the folds then ought to be associated to generate a single prediction.

The intervals for values of the parameters for the SVM models are given in Table 3.1.

Table 3.1. The intervals for values of the parameters for the SVM model

Parameter	**Value**
Cost (*C*)	[1, 300]
Gamma (γ)	[0, 1]
Epsilon (ε)	[0, 1]

3.2. MLP Model for Forecasting Internet Traffic

Back-propagation is utilized to create MLP models for prediction of Internet traffic. The success of an MLP is related with the number of hidden layers and neuron numbers in each hidden layer. If the number of neuron that is utilized is not sufficient, less information will be acquired. On the contrary, the local minimum might enhance and the network might come close to a local minimum, hence the network's sensitive will decrease. Nevertheless, there is not a regulation for detecting the number of neurons in a hidden layer. Generally, the optimal number is selected with try and error based on the difficulty of the problem.

In the hidden layer of the MLP models, the tansigmoid function is utilized, whereas in the output layer of MLP models, the pure-linear function is used. LM algorithm is implemented to train the network. Other important parameters of the MLP based models and their values are given in Table 3.2.

Table 3.2 The intervals for values of the parameters for the MLP models

Parameter	**Value**
Number of neurons in the hidden layer	[1-10]
Learning rate	[0-1]
Momentum	[0-1]
Maximum iteration	10000
Number of epochs	500

3.3. RBF Network Model for Forecasting Internet Traffic

The success of RBF network relies on numerous factors. The neuron number of the hidden layer has to be specified before the parameter selection for the RBF network. After the neuron number of the hidden layer is selected, the success of the RBF depends on the choosing of the network parameters. The efficiency of training an RBF network is subjected to a form of functions, their quantity and locations, and the method utilized for learning the input-output mapping. The most common learning strategies (Karayiannis and Randolph-Gips, 2003) for RBF networks could be categorized as follows;

- The first one is selecting the RBF centers randomly from the training data.
- The second one is using uncontrolled techniques for selecting the RBF centers.
- The last one is using controlled techniques for selecting the RBF centers.

Some approaches have been presented, which include orthogonal least squares (Gomm and Yu, 2000), resource allocating in genetic algorithm in gradient decent. These approaches involve searching for a sub-optimal network in a lower dimensional space. However, one common feature of the above methods is that they often lead to over-fitting, with a negative influence on the success of trained RBF neural networks. The biggest obstacle is iterative strategy that often cannot reach an optimal result. To overcome these computational difficulties, the complexity of the network must be decreased, which needs an approach to the arranged network. The design of a controlled neural network may be maintained in a diversity of ways.

Other important parameters of the RBF models are the number of clusters and the clustering seed, and their values are given in Table 3.3.

Table 3.3. The intervals for values of the parameters for the RBF models

Parameter	Value
Number of cluster	[4 - 30]
Clustering seed	[2 - 24]

3.4. M5P Model for Forecasting Internet Traffic

M5P is a reconstruction of M5 with some improvement. M5P integrates a conventional decision tree with the possibility of linear regression functions at the nodes. The M5 rules algorithm uses separate-and-conquer technique to produce a decision list for regression problems. Separate-and-conquer technique is a rule that includes instances in the class, then separates them out, and proceeds on those that are left (Harms et al., 2009).

The minimum number of instance to allow at a leaf node is the only parameter that affects the performance of a M5P model. Table 3.4. shows its interval for the M5P based models.

Table 3.4. The intervals for values of the parameters for the M5P models

Parameter	Value
Minimum number of instances	[4, 24]

3.5. RF Model for Forecasting Internet Traffic

RF is an ensemble classifier using many decision tree models in order to improve the error rate for classification and regression analysis. There are many advantages of random forest method such as generating a highly accurate classifier, running efficiently on large database, giving prediction about the variable that are important in the classification, having an effective method for estimating missing data, etc. (Ali et al., 2012).

The important parameters that affect the performance of an RF model are the number of attributes to be used in random selection, the number of trees, and the

random number seed. The intervals for values of the parameters for the RF models are given in Table 3.5.

Table 3.5. The intervals for values of the parameters for the RF models

Parameter	Value
Number of features	[0, 25]
Max number of tree	[100, 800]
Random number seed	[1, 25]

3.6. RT Model for Forecasting Internet Traffic

RT is a tree that is formed randomly by using a cluster of trees having N random features at every node (Ali et. al., 2012). RT is a simple greedy heuristic that tries to connect two trees. One of them is chosen from the initial configuration and the other one is selected from the goal. The aim of establishing search trees from the initial to goal configurations issues classical artificial intelligence bidirectional search (Kuffner and LaValle, 2000). Random trees can be produced effectively and random trees consisting of large sets give more accurate models (Ali et al., 2012).

The parameter of number of folds (i.e. the amount of data used for back-fitting), the minimum total weight of the instances in a leaf, and the random number seed used for selecting attributes affect the performance of an RT model. The important parameters for RT-based models and their values are given in Table 3.6.

Table 3.6. The intervals for values of the parameters for the RT models

Parameter	Value
Number of folds	[0, 16]
Minimum total weight of the instances	[1, 17]
Random number seed	[1, 30]

3.7. REPTree Model for Forecasting Internet Traffic

REPTree is a fast decision tree. Information gain as the splitting criterion has been used to build a decision/regression tree and REP has been used to prune the decision/regression tree. It only sorts values for numeric attributes once (Zhao and Zhang, 2008). REPTree reveals from the decision tree pruning algorithm as said above. To get best voted performance on the pruning set, the set of M classifiers is aimed to choose.

Greedy algorithm could be used for this. However, backfitting which is more developed algorithm has been chosen to use. The process of backfitting is as follows. The first step of it is like greedy algorithm. That is, first, a set of U of classifiers by building U one classifier at a time has been constructed. Second step, which is also the same with greedy algorithm, is that U has been initialized to include one classifier h_i that has the lowest error on the pruning set. Then the classifier h_j has been added where the voted combination of h_i and h_j has the lowest pruning set error.

In the third step, the classifier h_k which is the lowest pruning error in the combination of all classifiers of U has been added to U by using backfitting. But it checks each of its earlier decisions. First, h_i, h_j and h_k are the lowest pruning set error. Then some process has been done for h_j, after that for h_k. This process has been terminated when none of the classifiers changes or a limit on the number of iterations is attained.

The number of folds (i.e. the amount of data used for pruning), the minimum total weight of the instance in a leaf, and the seed used for randomizing the data are the parameters that affect the performance of an REPTree model. The important parameters for REPTree based models and their intervals are given in Table 3.7.

Table 3.7. The intervals for values of the parameters for the REPTree models

Parameter	**Value**
Number of folds	[3, 12]
Minimum total weight of the instances	[2, 7]
Seed	[1, 25]

3.8. Holt-Winters Model for Forecasting Internet Traffic

Holt-Winters method uses at most 3 variables for the data components' Seasonable, Level, and Trend to make estimation. Thus, Holt-Winters method can be called as 3-parameter method. Also, the beginning parameters for the Seasonal, Level, and Trend can be predicted (Hyndman and Kostenko, 2007).

Even though Holt-winters methods generate precise forecasts, there are no acceptable estimation intervals in Holt-Winters methods. That is, there is no bound on the forecast error. An estimation interval of the form

$$forecast \pm k \times \sqrt{\text{var}\, iance\, of\ the\ forecast\, error} \qquad (3.1.)$$

has been found to determine a bound on the forecast error. In the (3.1.), *k* value is related with probability distribution of the forecast error. To derive the variance of the forecast error, a statistical model needs to be used. Thus, a forecast method includes an essential statistical model because of the prediction intervals. In brief, the predictive model depends on some essential forms like a seasonal cycle (K_1) or a trend. Nowadays, this method has been enlarged with involving two seasonal cycles (K_1 and K_2) (Kohler et. al., 2001).

The important parameters that affect the performance of a Holt-Winters model are the length of the seasonal cycle, the smoothing factor for the seasonal component (ζ), the smoothing factor for the trend (λ), and the smoothing factor for the series values (δ). These parameters and their intervals are given in Table 3.8.

Table 3.8. The intervals for values of the parameters for the Holt-Winters models

Parameter	**Value**
Seasonal cycle length	[1, 3]
Seasonal smoothing factor (ζ)	[0, 1]
Trend smoothing factor (λ)	[0, 1]
Value smoothing factor (δ)	[0, 1]

3.9. Autocorrelation Function

The autocorrelation is defined as the correlation a time series attributes at times *n* among *n-k* where *k=1, …, K=N-1*. It measures the sign correlated between a time shift and itself where the map of the number of the time shift determined also as a time lag. The autocorrelation function helps to determine the rate of dependence in a data. Also, it helps to state stability of time series, utilize likely time series model and sorting sample that occurs again in time series. Further, time series with distinct scale can be compared by using the autocorrelation function.

A time series can be determined by (3.2.)

$$X=\left\{x_{in}:i=1,...,I;n=1,...,N\right\}=\left\{x_i:i=1,...,I\right\}=\begin{bmatrix} & & & & \\ x_{11} & \cdots & x_{i1} & \cdots & x_{I1} \\ \cdots & & \cdots & & \cdots \\ x_{1n} & \cdots & x_{in} & \cdots & x_{In} \\ \cdots & & \cdots & & \cdots \\ x_{1N} & \cdots & x_{iN} & \cdots & x_{IN} \\ & & i^{th}\ time\ series & & \end{bmatrix} \quad (3.2.)$$

such that $x_n=\left\{x_{in}:n=1,...,N\right\}$ refers the i^{th} time series $\left(i=1,...,I\right)$, x_{in} expresses the n^{th} investigation $\left(n=1,...,N\right)$ of the i^{th} time series $\left(i=1,...,I\right)$.

For a specific i^{th} time series, $x_n=\left\{x_{in}:n=1,...,N\right\}$ the autocorrelation coefficient at lag $k\left(k=1,...,N-1=K\right)$ is shown by (3.3.)

$$\lambda_{ik} = \frac{\sum_{n=k+1}^{N}\left(x_{in} - \overline{x}_i\right)\left(x_{i(n-k)} - \overline{x}_i\right)}{\sum_{n=1}^{N}\left(x_{in} - \overline{x}_i\right)^2} \qquad (3.3.)$$

such that $\overline{x}_i$ is the average of the time i^{th} series. The autocorrelation function of a time series is built by the autocorrelation coefficients at the distinct time lags. The interval of the autocorrelation is [-1, 1]. Figure 3.1. gives the relation between time series and estimated correlation coefficients (D'Urso and Maharaj, 2009).

$$x_i = \left(x_{i1},...,x_{in},...,x_{iN}\right) \quad i^{th}\ time\ series \longrightarrow \hat{\lambda}_i = \left(\hat{\lambda}_{i1},...,\hat{\lambda}_{ik},...,\hat{\lambda}_{iK}\right)$$
$$\left(estimated\ autocorrelation\ coefficients\ of\ i^{th}\ time\ series\ for\ distinct\ time\ lags\ k = 1,...,K\right)$$

Figure 3.1. Time series and estimated autocorrelation coefficients

The autocorrelation for the each data set is given by Figure 3.2 through 3.7.

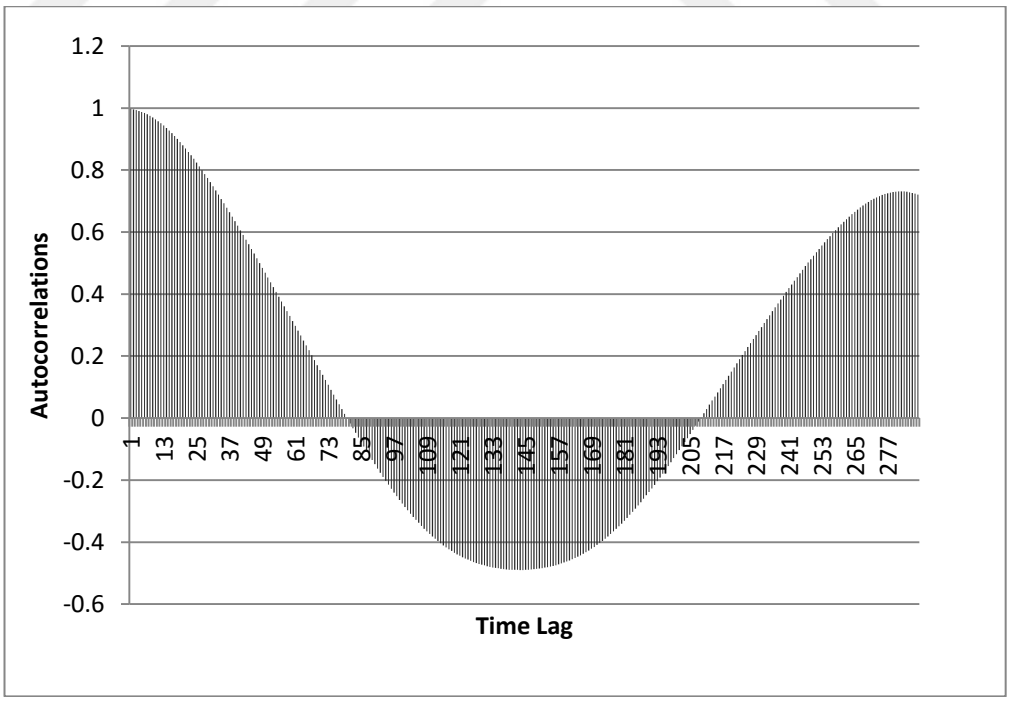

Figure 3.2. The autocorrelations for the DS1-5M data set

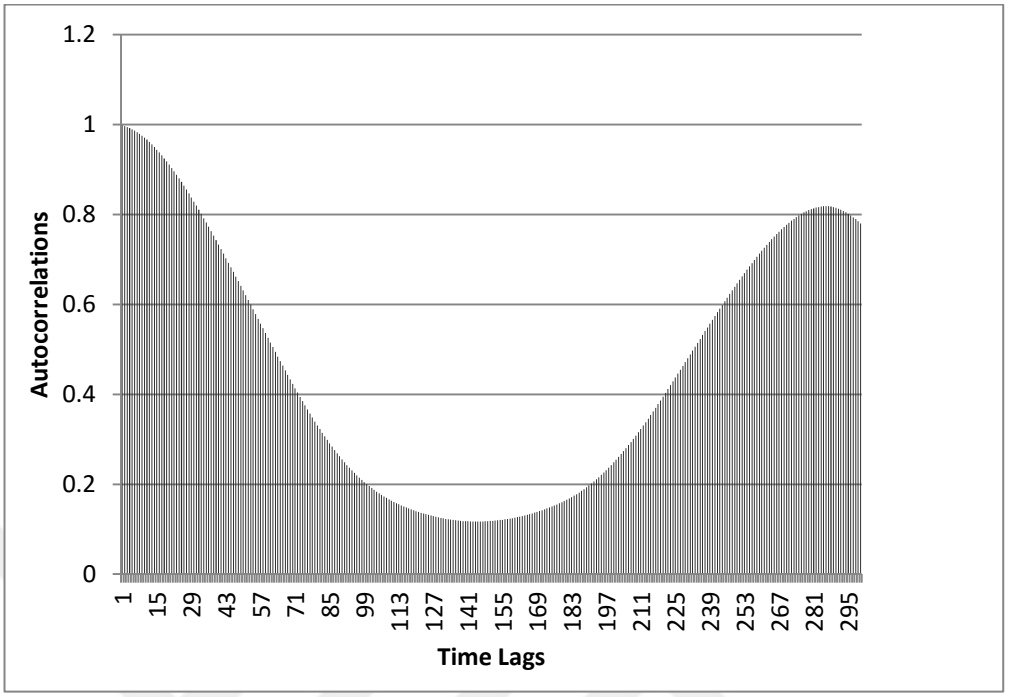

Figure 3.3. The autocorrelations for the DS2-5M data set

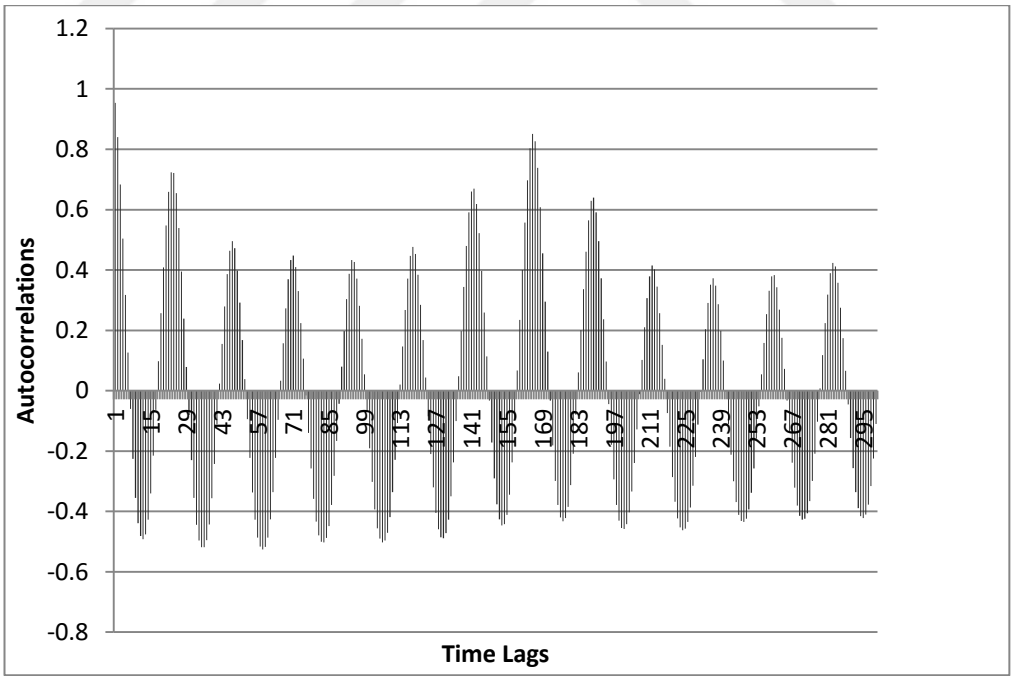

Figure 3.4. The autocorrelations for the DS1-1H data set

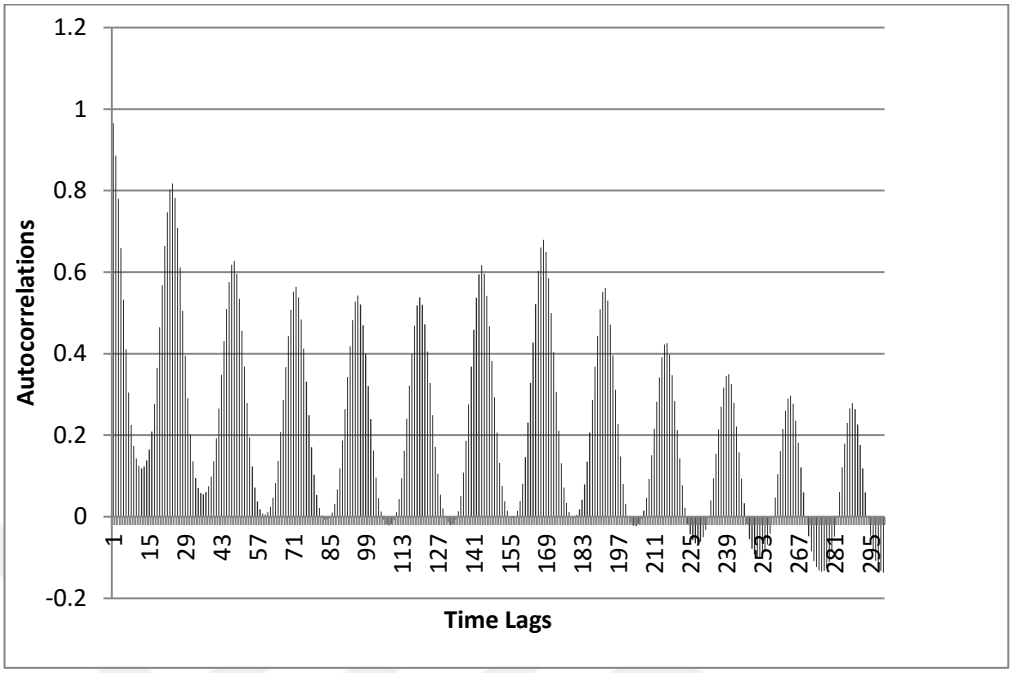

Figure 3.5. The autocorrelations for the DS2-1H data set

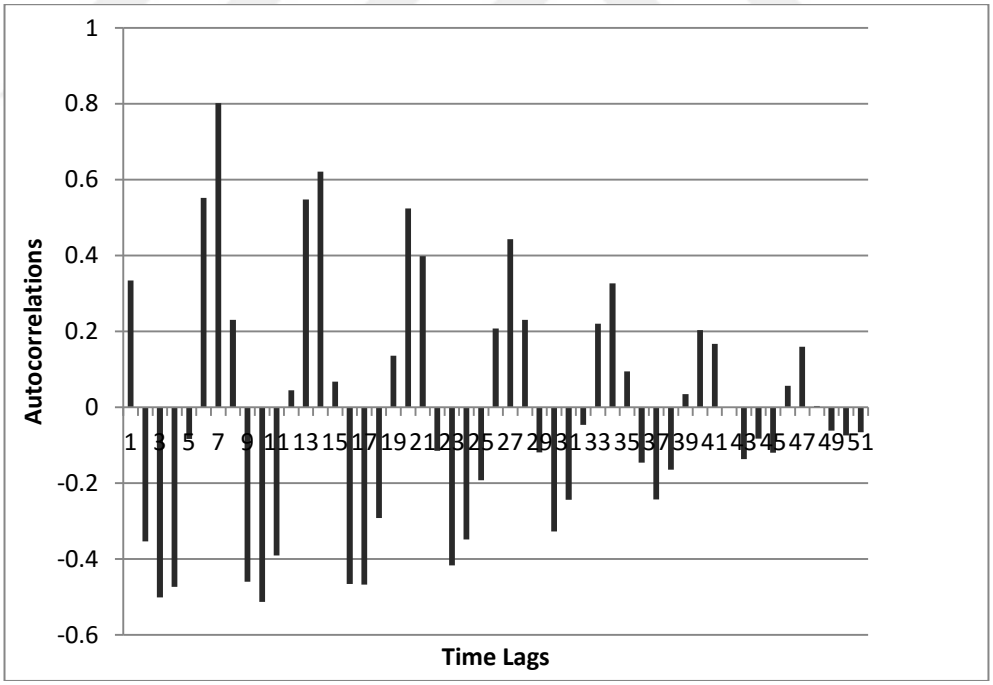

Figure 3.6. The autocorrelations for the DS1-1D data set

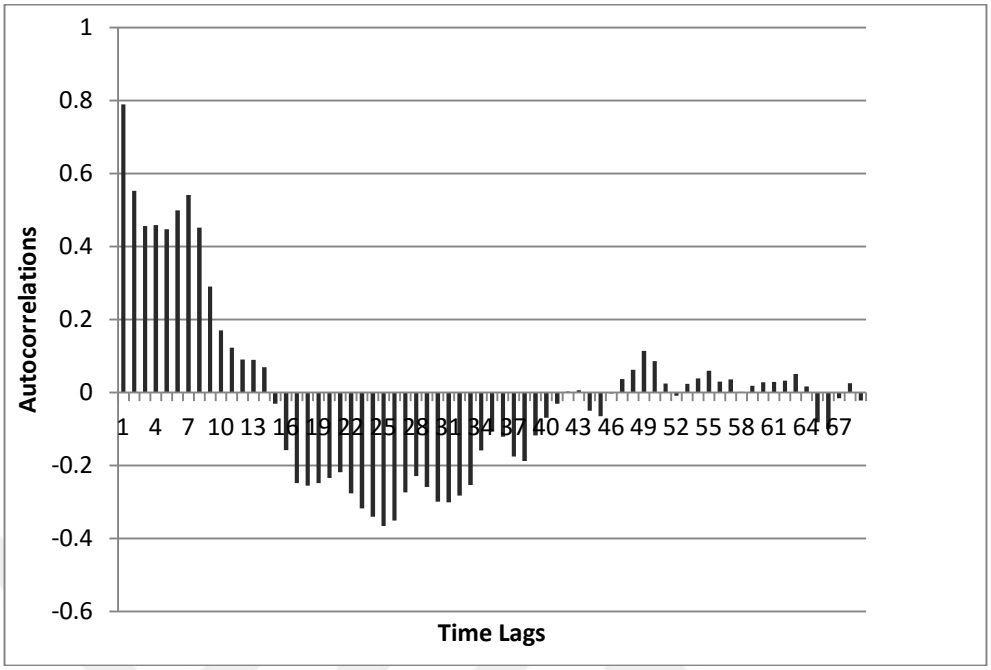

Figure 3.7. The autocorrelations for the DS2-1D data set

3.10. Model Selection

Model selection is a vital problem to model time series. That is, finding the best sliding time window for a time specific time series is very important issue. A sliding time window means a group of time lags $\{I_1, I_2, ..., I_m\}$ which employ to utilize a forecast. Let the entries be $x_{n-l_m}, ..., x_{n-l_2}, x_{n-l_1}$ and the aimed output be x_n for a specific time period *n*. For instance, the series $6_1, 11_2, 15_3, 20_4, 24_5, 29_6$ (x_n values) have been used. If $\{1, 4\}$ has been employed as a sliding window, then two training samples can be constituted as $5, 20 \rightarrow 24$ and $11, 24 \rightarrow 29$. The length of the sliding windows is important issue in the forecasting performance. A small window gives limited information to the network while a large number of time lags can enhance the entropy which has an impact on learning.

Distinct possible models which are considered by using a generalization estimate have been employed for the statistical approach to model selection. Many complex predictive that are hard to compute have been built. Model complexity can

be overcome by using a simple statistic like Bayesian Information Criterion (*BIC*) which formula's is given by (3.4.)

$$BIC = N \ln\left(\frac{SSE}{N}\right) + P \ln(N) \tag{3.4.}$$

such that *N* defines the number of training, *P* refers the number of parameters, *SSE* means sum squared error. Even though *BIC* has been used to build linear model, it has also been used to build neural estimation model. When employed to neural networks, *P* is the same value with the number of connection weights (Cortez et. al., 2005).

In this thesis, three rules have been utilized to build forecasting models. The rules of the sliding windows are given below;

1. Use all time lags from 1 to a given maximum m: <1, 2 , …, m>
2. Use all lags the autocorrelation values of which are above a given threshold
3. Use the four time lags with the highest autocorrelations.

By using the heuristic rules for time lag selection, discussed above, five different sliding windows have been generated for each data set. The first three models have built according to the first rule given above. The fourth one has been generated by choosing the lags the autocorrelation values of which are above 0.9, 0.7, and 0.5 for the time scales, 5 minute, 1 hour, and 1 day, respectively. The last one has been generated by using four time lags with the highest autocorrelations. Table 3.9. through Table 3.14. show the time lags used in each model.

Table 3.9. List of the chosen time lags for DS1-5M

Models	Chosen Lag
DS1-5M-lag1	{1, 2, …, 7}
DS1-5M-lag2	{1, 2, …, 13}
DS1-5M-lag3	{1, 2, …, 25}
DS1-5M-auto	{1, 2, …, 18}
DS1-5M-h-auto	{1, 2, 3, 4}

Table 3.10. List of the chosen time lags for DS2-5M

Models	Chosen Lag
DS2-5M-lag1	{1, 2, …, 7}
DS2-5M-lag2	{1, 2, …, 13}
DS2-5M-lag3	{1, 2, …, 25}
DS2-5M-auto	{1, 2, …, 21}
DS2-5M-h-auto	{1, 2, 3, 4}

Table 3.11. List of the chosen time lags for DS1-1H

Models	Chosen Lag
DS1-1H-lag1	{1, 2, …, 13}
DS1-1H-lag2	{1, 2, …, 25}
DS1-1H-lag3	{1, 2, …, 36}
DS1-1H-auto	{1, 2, 23, 24, 164, 165, 166, 167, 330, 331}
DS1-1H-h-auto	{1, 2, 165, 166}

Table 3.12. List of the chosen time lags for DS2-1H

Models	Chosen Lag
DS2-1H-lag1	{1, 2, …, 13}
DS2-1H-lag2	{1, 2, …, 25}
DS2-1H-lag3	{1, 2, …, 36}
DS2-1H-auto	{1, 2, 22, 23, 24, 25, 26}
DS2-1H-h-auto	{1, 2, 23, 24}

Table 3.13. List of the chosen time lags for DS1-1D

Models	Chosen Lag
DS1-1D-lag1	{1, 2, 3, 4, 5}
DS1-1D-lag2	{1, 2, …, 8}
DS1-1D-lag3	{1, 2, …, 12}
DS1-1D-auto	{6, 7, 13, 14, 20}
DS1-1D-h-auto	{6, 7, 13, 14}

Table 3.14. List of the chosen time lags for DS2-1D

Models	**Chosen Lag**
DS2-1D-lag1	{1, 2, 3, 4, 5}
DS2-1D-lag2	{1, 2, …, 8}
DS2-1D-lag3	{1, 2, …, 12}
DS2-1D-auto	{1, 2, 7}
DS2-1D-h-auto	{1, 2, 6, 7}

For model testing the first 2/3 of each data set has been utilized as a training set and the rest has been used as a test set. The *MAPE* has been utilized to assess the forecasting models' performance. The formula for the *MAPE* is given by (3.5.)

$$MAPE = \frac{1}{n}\sum_{t=1}^{n}\frac{|A_t - F_t|}{A_t} \tag{3.5.}$$

where n is the number of forecast samples, A_t is the actual value, and F_t is the forecast value (Benzer and Benzer, 2015). *MAPE*, which is an accuracy measure, is a widespread metric in forecasting error and the actual value.

4. RESULTS AND DISCUSSION

In this thesis, the prediction methods are categorized in four groups as statistical, ANN, SVM, and decision trees. The results of the forecasting models are examined according to these four categories. The computed *MAPE*'s for each traffic forecasting model on each data set are shown in Table 4.1. through Table 4.24.

Table 4.1. *MAPE* values for statistical Internet traffic forecasting models DS1-5M data set.

Models	**MLR**	**Holt-Winters**
	***MAPE* (%)**	***MAPE* (%)**
DS1-5M-lag1	3.03	3.23
DS1-5M-lag2	2.91	3.11
DS1-5M-lag3	2.89	3.02
DS1-5M-auto	2.91	3.04
DS1-5M-h-auto	3.05	3.48

Table 4.2. *MAPE* values for ANN Internet traffic forecasting models on DS1-5M data set.

Models	**MLP**	**RBF**
	***MAPE* (%)**	***MAPE* (%)**
DS1-5M-lag1	3.06	9.94
DS1-5M-lag2	2.98	10.29
DS1-5M-lag3	2.88	9.91
DS1-5M-auto	2.95	10.71
DS1-5M-h-auto	3.03	9.64

Table 4.3. *MAPE* values for SVM Internet traffic forecasting models on DS1-5M data set.

Models	**SVM-Poly**	**SVM-RBF**
	***MAPE* (%)**	***MAPE* (%)**
DS1-5M-lag1	3.02	3.12
DS1-5M-lag2	2.90	2.87
DS1-5M-lag3	2.87	2.85
DS1-5M-auto	2.89	2.89
DS1-5M-h-auto	3.04	3.05

Table 4.4. *MAPE* values for decision trees Internet traffic forecasting models on DS1-5M data set.

Models	**M5P**	**RF**	**RT**	**REPTree**
	***MAPE* (%)**	***MAPE* (%)**	***MAPE* (%)**	***MAPE* (%)**
DS1-5M-lag1	3.05	3.07	3.38	3.23
DS1-5M-lag2	2.92	2.97	3.70	3.21
DS1-5M-lag3	2.93	2.95	4.04	3.20
DS1-5M-auto	2.91	2.96	3.92	3.22
DS1-5M-h-	3.03	3.17	3.30	3.23

Table 4.5. *MAPE* values for statistical Internet traffic forecasting models on DS2-5M data set.

Models	**MLR**	**Holt-Winters**
	***MAPE* (%)**	***MAPE* (%)**
DS2-5M-lag1	1.47	1.47
DS2-5M-lag2	1.46	1.47
DS2-5M-lag3	1.50	1.46
DS2-5M-auto	1.48	1.47
DS2-5M-h-auto	1.49	1.58

Table 4.6. *MAPE* values for ANN Internet traffic forecasting models on DS2-5M data set.

Models	**MLP**	**RBF**
	***MAPE* (%)**	***MAPE* (%)**
DS2-5M-lag1	1.57	9.64
DS2-5M-lag2	1.50	7.49
DS2-5M-lag3	1.49	8.69
DS2-5M-auto	1.51	8.30
DS2-5M-h-auto	1.52	9.86

Table 4.7. *MAPE* values for SVM Internet traffic forecasting models on DS2-5M data set.

Models	**SVM-Poly**	**SVM-RBF**
	***MAPE* (%)**	***MAPE* (%)**
DS2-5M-lag1	1.46	1.59
DS2-5M-lag2	1.45	1.49
DS2-5M-lag3	1.48	1.59
DS2-5M-auto	1.46	1.52
DS2-5M-h-auto	1.49	1.51

Table 4.8. *MAPE* values for decision trees Internet traffic forecasting models on DS2-5M data set.

Models	**M5P**	**RF**	**RT**	**REPTree**
	***MAPE* (%)**	***MAPE* (%)**	***MAPE* (%)**	***MAPE* (%)**
DS2-5M-lag1	1.47	1.51	1.82	1.62
DS2-5M-lag2	1.44	1.47	1.99	1.61
DS2-5M-lag3	1.45	1.46	2.20	1.60
DS2-5M-auto	1.45	1.46	2.05	1.63
DS2-5M-h-auto	1.49	1.55	1.71	1.62

Table 4.9. *MAPE* values for statistical Internet traffic forecasting models on DS1-1H data set.

Models	**MLR**	**Holt-Winters**
	***MAPE* (%)**	***MAPE* (%)**
DS1-1H-lag1	34.69	20.27
DS1-1H-lag2	28.25	16.59
DS1-1H-lag3	27.60	9.44
DS1-1H-auto	4.95	8.53
DS1-1H-h-auto	4.86	9.85

Table 4.10. *MAPE* values for ANN Internet traffic forecasting models on DS1-1H data set.

Models	**MLP**	**RBF**
	***MAPE* (%)**	***MAPE* (%)**
DS1-1H-lag1	5.68	17.28
DS1-1H-lag2	5.90	13.76
DS1-1H-lag3	5.99	18.05
DS1-1H-auto	4.99	11.45
DS1-1H-h-auto	4.81	12.95

Table 4.11. *MAPE* values for SVM Internet traffic forecasting models on DS1-1H data set.

Models	**SVM-Poly**	**SVM-RBF**
	***MAPE* (%)**	***MAPE* (%)**
DS1-1H-lag1	7.07	6.07
DS1-1H-lag2	5.92	5.26
DS1-1H-lag3	6.12	5.42
DS1-1H-auto	4.51	4.29
DS1-1H-h-auto	4.60	4.46

Table 4.12. *MAPE* values for decision trees Internet traffic forecasting models on DS1-1H data set.

Models	**M5P**	**RF**	**RT**	**REPTree**
	***MAPE* (%)**	***MAPE* (%)**	***MAPE* (%)**	***MAPE* (%)**
DS1-1H-lag1	6.64	5.42	8.44	6.64
DS1-1H-lag2	5.07	5.97	8.84	7.24
DS1-1H-lag3	5.23	5.92	10.27	7.11
DS1-1H-auto	4.92	5.04	7.37	6.94
DS1-1H-h-auto	4.69	5.25	6.59	7.47

Table 4.13. *MAPE* values for statistical Internet traffic forecasting models on DS2-1H data set.

Models	**MLR**	**Holt-Winters**
	***MAPE* (%)**	***MAPE* (%)**
DS2-1H-lag1	66.83	9.01
DS2-1H-lag2	4.42	5.79
DS2-1H-lag3	4.32	5.68
DS2-1H-auto	4.35	5.70
DS2-1H-h-auto	4.42	5.68

Table 4.14. *MAPE* values for ANN Internet traffic forecasting models on DS2-1H data set.

Models	**MLP**	**RBF**
	***MAPE* (%)**	***MAPE* (%)**
DS2-1H-lag1	5.54	16.90
DS2-1H-lag2	3.78	14.09
DS2-1H-lag3	3.79	16.61
DS2-1H-auto	4.03	13.63
DS2-1H-h-auto	4.40	12.84

Table 4.15. *MAPE* values for SVM Internet traffic forecasting models on DS2-1H data set.

Models	**SVM-Poly**	**SVM-RBF**
	***MAPE* (%)**	***MAPE* (%)**
DS2-1H-lag1	6.23	6.35
DS2-1H-lag2	3.88	3.80
DS2-1H-lag3	3.77	3.77
DS2-1H-auto	3.90	4.35
DS2-1H-h-auto	3.96	4.08

Table 4.16. *MAPE* values for decision trees Internet traffic forecasting models on DS2-1H data set.

Models	**M5P**	**RF**	**RT**	**REPTree**
	***MAPE* (%)**	***MAPE* (%)**	***MAPE* (%)**	***MAPE* (%)**
DS2-1H-lag1	6.85	4.63	6.58	5.46
DS2-1H-lag2	4.41	4.10	7.97	5.63
DS2-1H-lag3	4.30	4.26	8.44	5.83
DS2-1H-auto	4.31	4.23	6.33	5.73
DS2-1H-h-auto	4.42	4.70	6.38	5.72

Table 4.17. *MAPE* values for statistical Internet traffic forecasting models on DS1-1D data set.

Models	**MLR**	**Holt-Winters**
	***MAPE* (%)**	***MAPE* (%)**
DS1-1D-lag1	21.31	37.43
DS1-1D-lag2	12.02	20.03
DS1-1D-lag3	12.89	22.13
DS1-1D-auto	9.82	22.10
DS1-1D-h-auto	10.11	22.28

Table 4.18. *MAPE* values for ANN Internet traffic forecasting models on DS1-1D data set.

Models	**MLP**	**RBF**
	***MAPE* (%)**	***MAPE* (%)**
DS1-1D-lag1	10.45	13.25
DS1-1D-lag2	10.85	13.49
DS1-1D-lag3	10.66	6.60
DS1-1D-auto	8.36	5.85
DS1-1D-h-auto	7.56	6.13

Table 4.19. *MAPE* values for SVM Internet traffic forecasting models on DS1-1D data set.

Models	**SVM-Poly**	**SVM-RBF**
	***MAPE* (%)**	***MAPE* (%)**
DS1-1D-lag1	17.91	16.74
DS1-1D-lag2	8.66	5.59
DS1-1D-lag3	8.51	8.68
DS1-1D-auto	8.02	7.60
DS1-1D-h-auto	8.01	7.84

Table 4.20. *MAPE* values for decision trees Internet traffic forecasting models on DS1-1D data set.

Models	**M5P**	**RF**	**RT**	**REPTree**
	***MAPE* (%)**	***MAPE* (%)**	***MAPE* (%)**	***MAPE* (%)**
DS1-1D-lag1	19.88	7.38	13.82	7.09
DS1-1D-lag2	7.05	7.38	8.88	7.97
DS1-1D-lag3	7.05	7.89	8.45	7.97
DS1-1D-auto	7.05	7.74	6.03	7.97
DS1-1D-h-auto	7.05	16.77	6.50	7.97

Table 4.21. *MAPE* values for statistical Internet traffic forecasting models on DS2-1D data set.

Models	**MLR**	**Holt-Winters**
	***MAPE* (%)**	***MAPE* (%)**
DS2-1D-lag1	37.20	15.99
DS2-1D-lag2	38.72	12.32
DS2-1D-lag3	39.18	12.96
DS2-1D-auto	41.57	15.28
DS2-1D-h-auto	40.28	15.31

Table 4.22. *MAPE* values for ANN Internet traffic forecasting models on DS2-1D data set.

Models	**MLP**	**RBF**
	***MAPE* (%)**	***MAPE* (%)**
DS2-1D-lag1	16.82	16.62
DS2-1D-lag2	16.71	15.51
DS2-1D-lag3	16.70	16.02
DS2-1D-auto	17.01	16.29
DS2-1D-h-auto	16.75	18.16

Table 4.23. *MAPE* values for SVM Internet traffic forecasting models on DS2-1D data set.

Models	**SVM-Poly**	**SVM-RBF**
	***MAPE* (%)**	***MAPE* (%)**
DS2-1D-lag1	15.79	17.01
DS2-1D-lag2	13.85	16.69
DS2-1D-lag3	14.32	16.14
DS2-1D-auto	17.58	16.87
DS2-1D-h-auto	16.22	16.78

Table 4.24. *MAPE* values for decision trees Internet traffic forecasting models on DS2-1D data set.

Models	**M5P**	**RF**	**RT**	**REPTree**
	***MAPE* (%)**	***MAPE* (%)**	***MAPE* (%)**	***MAPE* (%)**
DS2-1D-lag1	40.75	16.41	14.65	17.39
DS2-1D-lag2	35.46	15.88	16.81	17.39
DS2-1D-lag3	35.46	14.61	15.06	17.39
DS2-1D-auto	37.01	23.41	16.70	17.39
DS2-1D-h-auto	37.01	19.20	13.71	17.39

Figure 4.1. through Figure 4.12 show the average of the *MAPE*'s of all categories separately and the percentage decrements rates in *MAPE*'s between the category having the lowest *MAPE*'s on the average and the other categories for each data set.

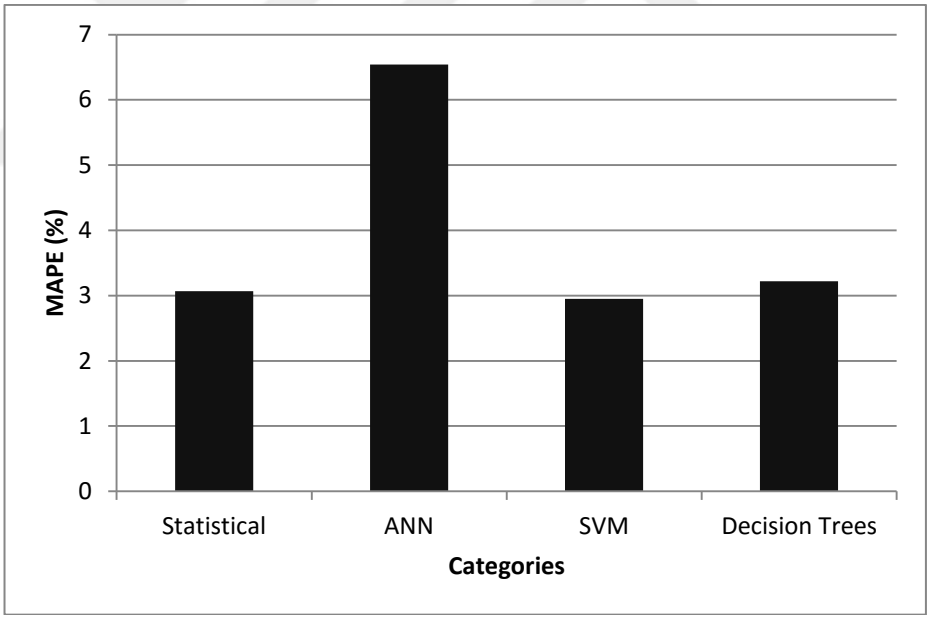

Figure 4.1. Illustration of average *MAPE*'s of all categories (DS1-5M)

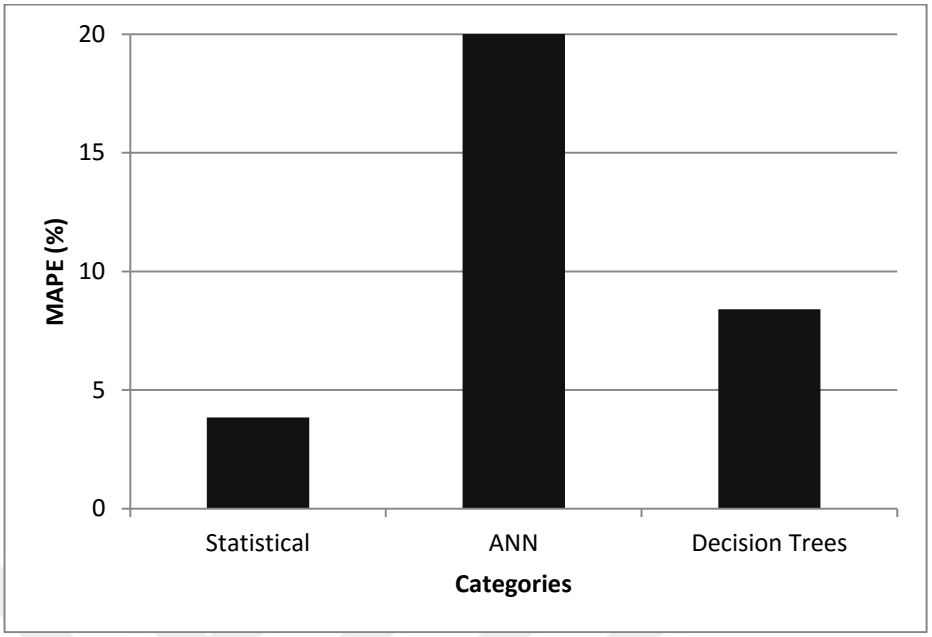

Figure 4.2. Percentage decrease rates in *MAPE*'s SVM compared to the ones obtained by statistical, ANN, and Decision Trees (DS1-5M)

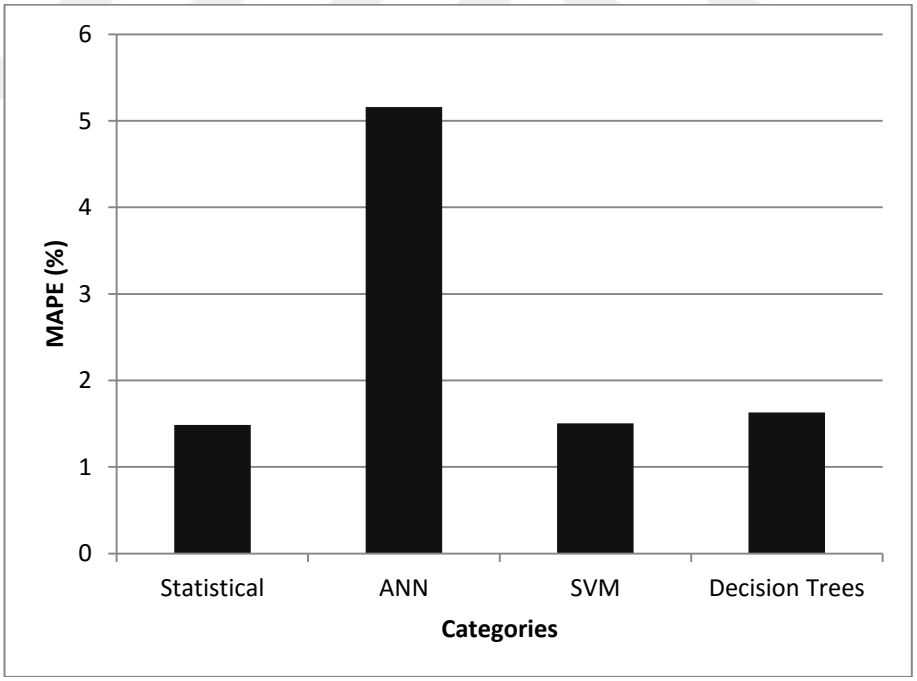

Figure 4.3. Illustration of average *MAPE*'s of all categories (DS2-5M)

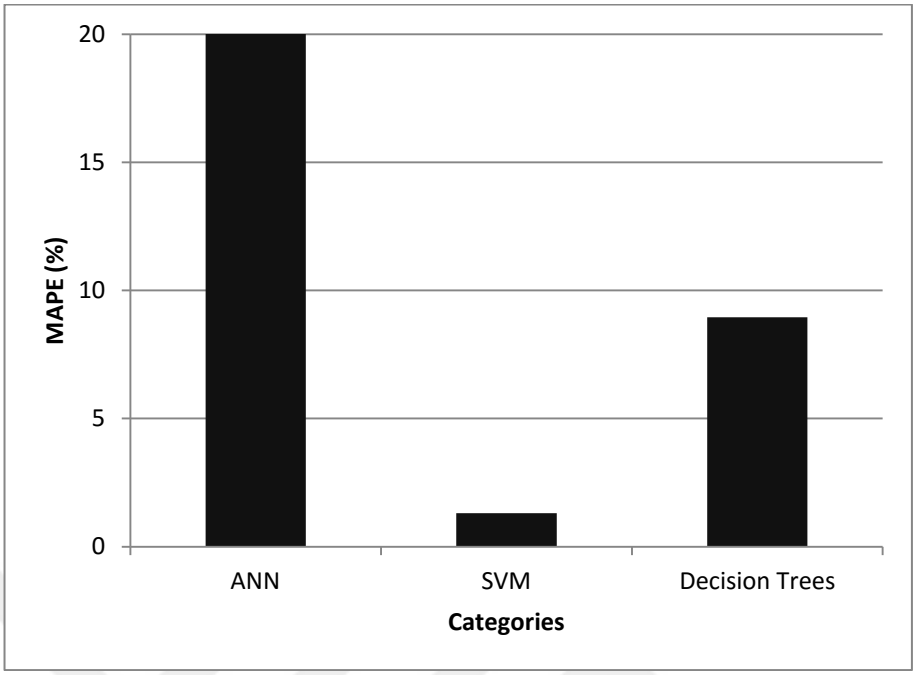

Figure 4.4. Percentage decrease rates in *MAPE*'s statistical compared to the ones obtained by ANN, SVM, and Decision Trees (DS2-5M)

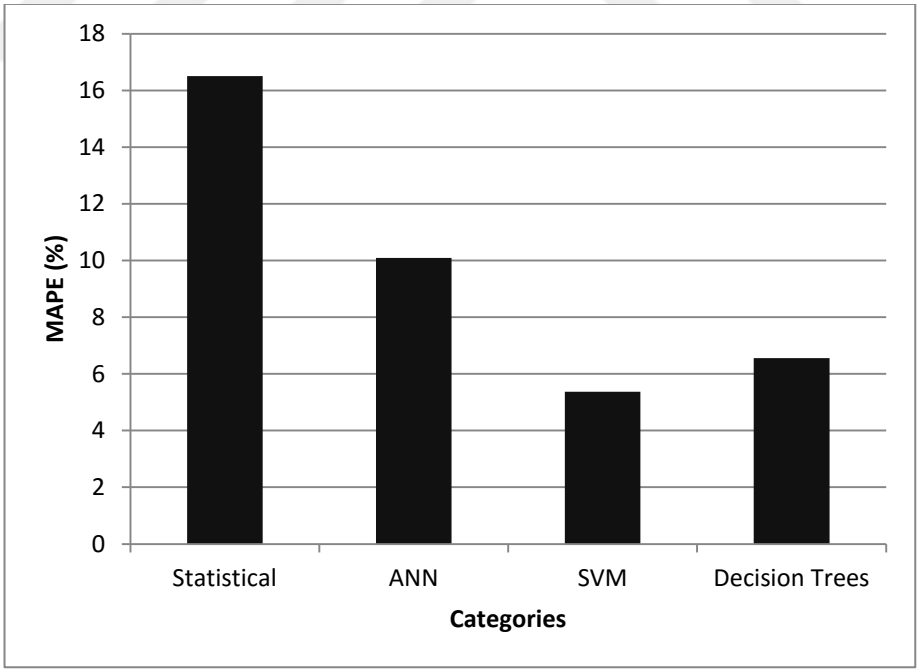

Figure 4.5. Illustration of average *MAPE*'s of all categories (DS1-1H)

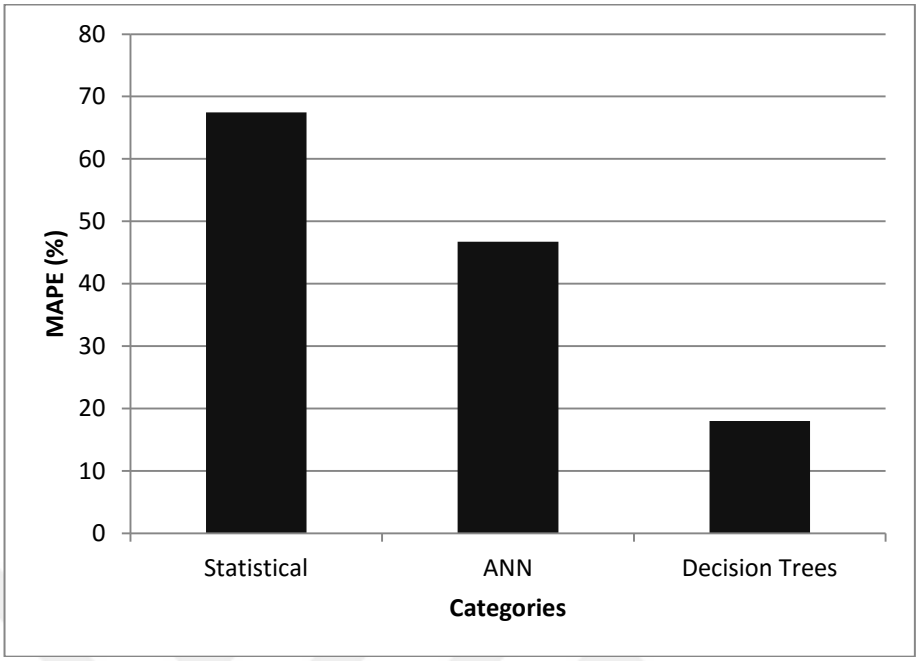

Figure 4.6. Percentage decrease rates in *MAPE*'s SVM compared to the ones obtained by statistical, ANN, and Decision Trees (DS1-1H)

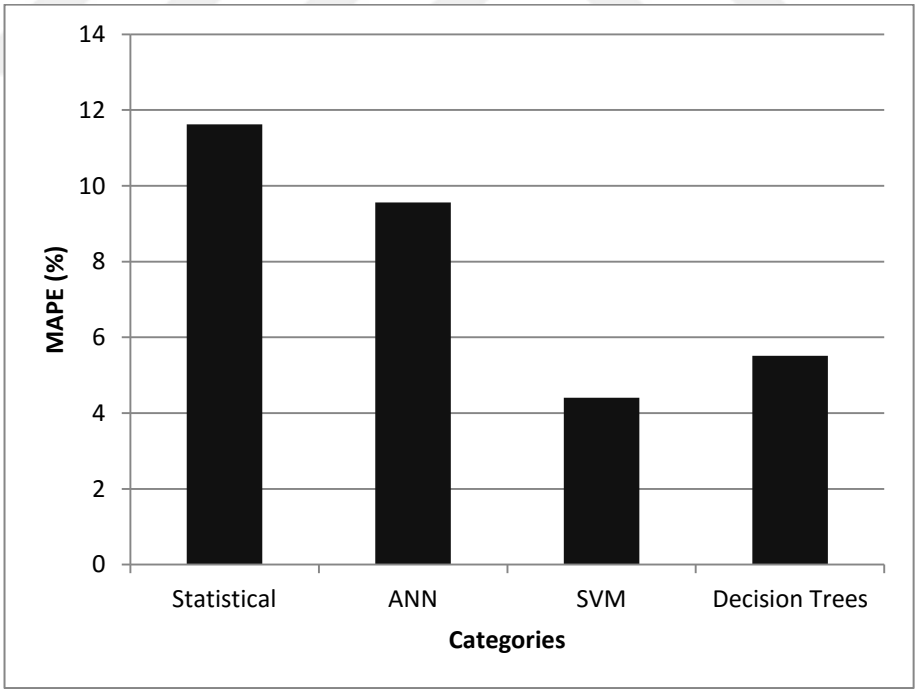

Figure 4.7. Illustration of average *MAPE*'s of all categories (DS2-1H)

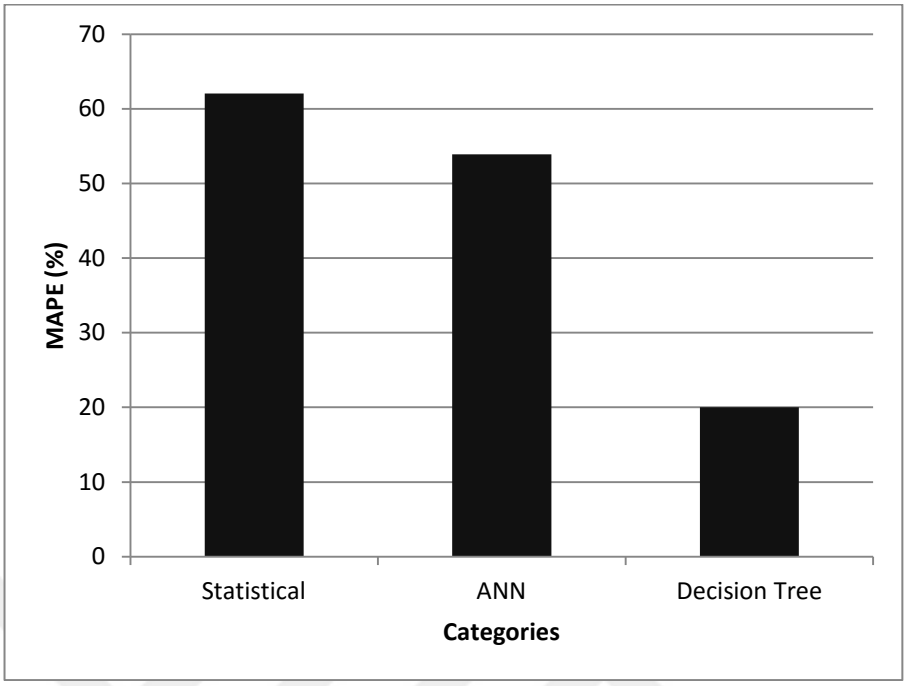

Figure 4.8. Percentage decrease rates in *MAPE*'s SVM compared to the ones obtained by statistical, ANN, and Decision Trees (DS2-1H)

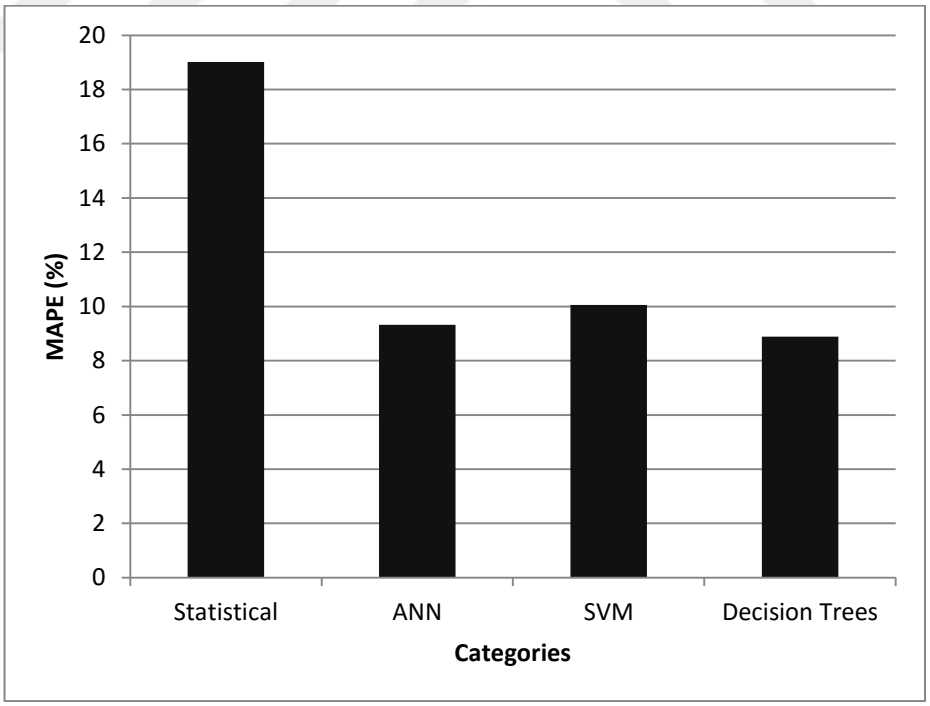

Figure 4.9. Illustration of average *MAPE*'s of all categories (DS1-1D)

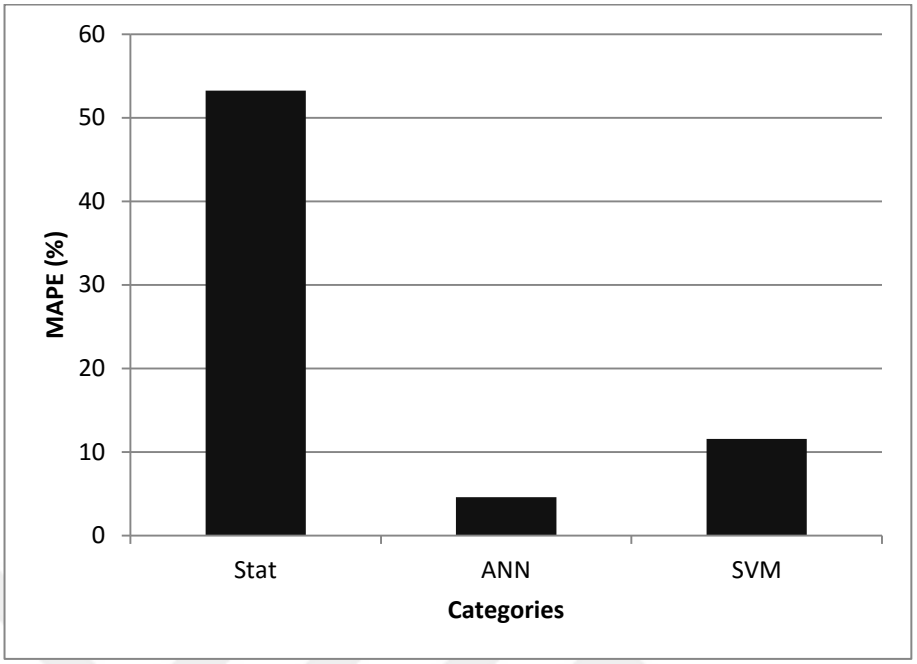

Figure 4.10. Percentage decrease rates in *MAPE*'s Decision Trees compared to the ones obtained by statistical, ANN, and SVM (DS1-1D)

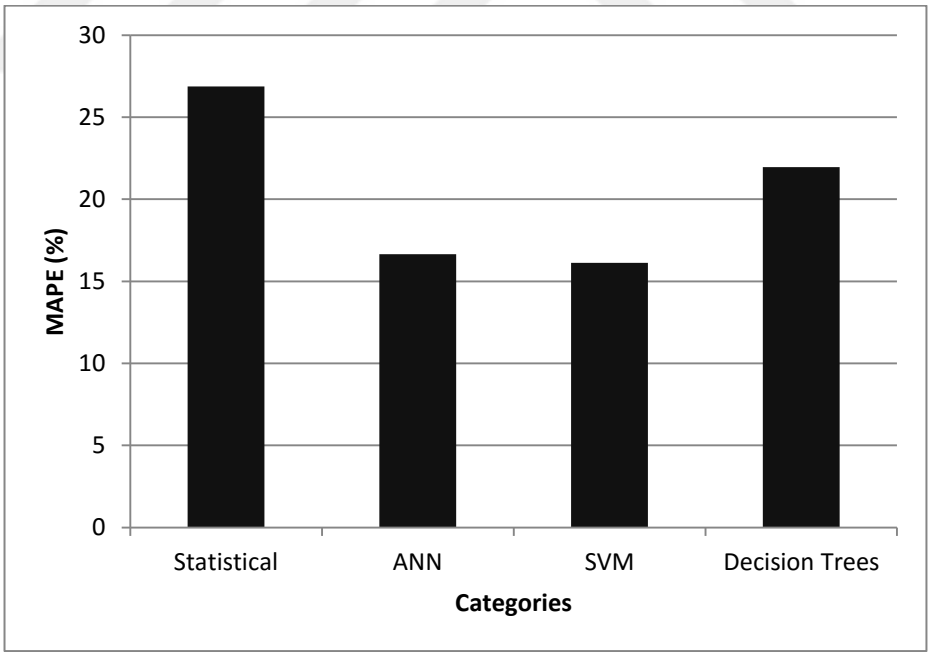

Figure 4.11. Illustration of average *MAPE*'s of all categories (DS2-1D)

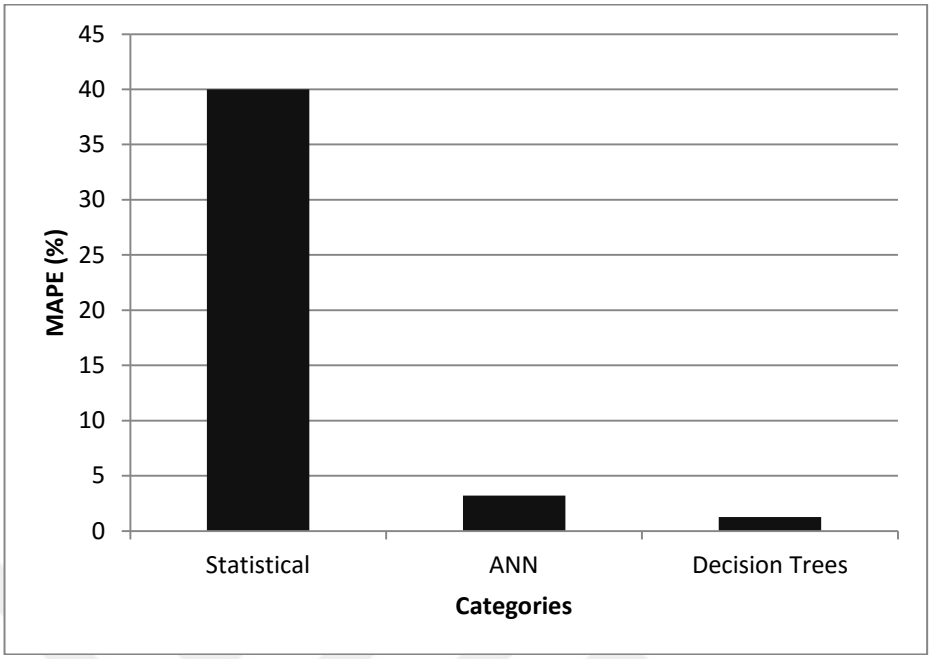

Figure 4.12. Percentage decrease rates in *MAPE*'s SVM compared to the ones obtained by statistical, ANN, and Decision Trees (DS2-1D)

Figure 4.13. through Figure 4.36 show the average of the *MAPE*'s for the methods in each category for each data set, separately.

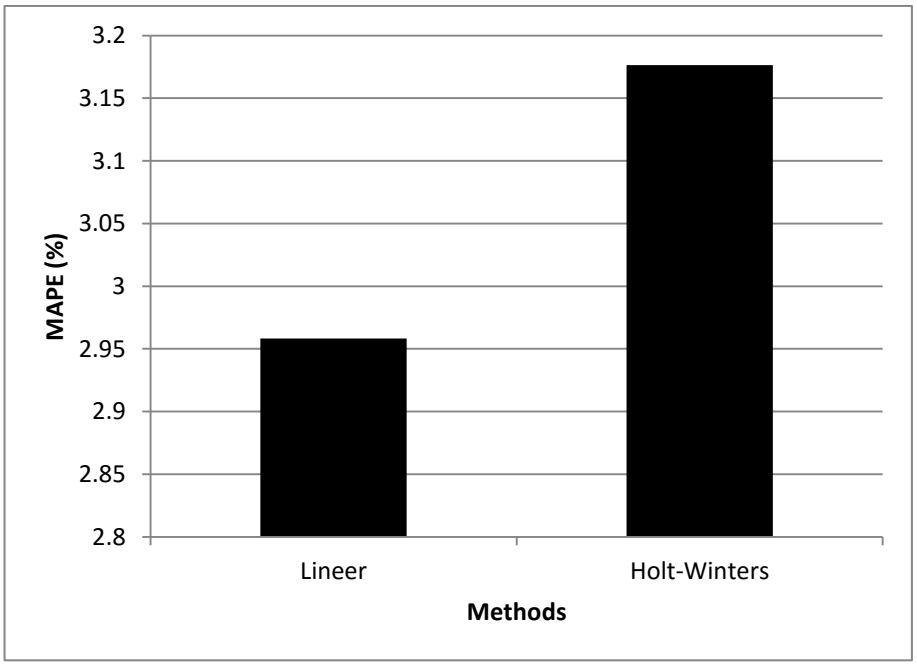

Figure 4.13. Illustration of average *MAPE*'s of Linear and Holt-Winters based models in the statistical category (DS1-5M)

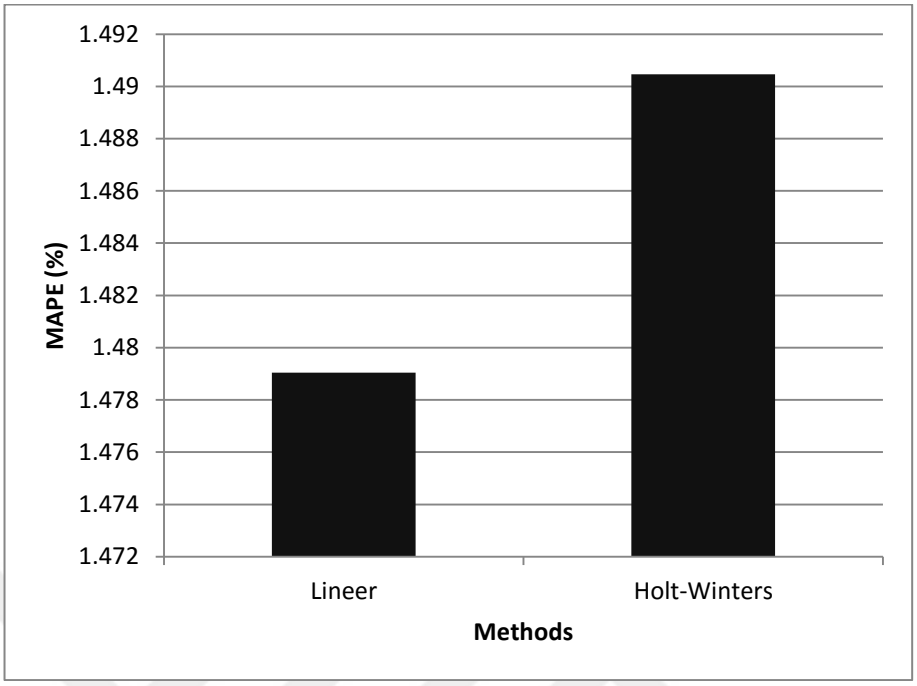

Figure 4.14. Illustration of average *MAPE*'s of Linear and Holt-Winters based models in the statistical category (DS2-5M)

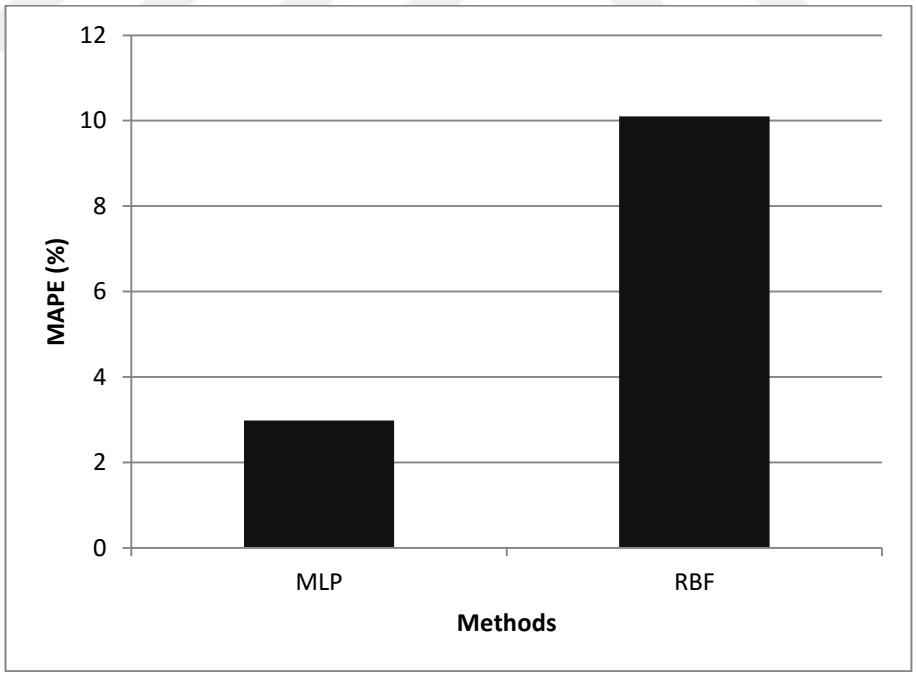

Figure 4.15. Illustration of average *MAPE*'s of MLP and RBF based models in the ANN category (DS1-5M)

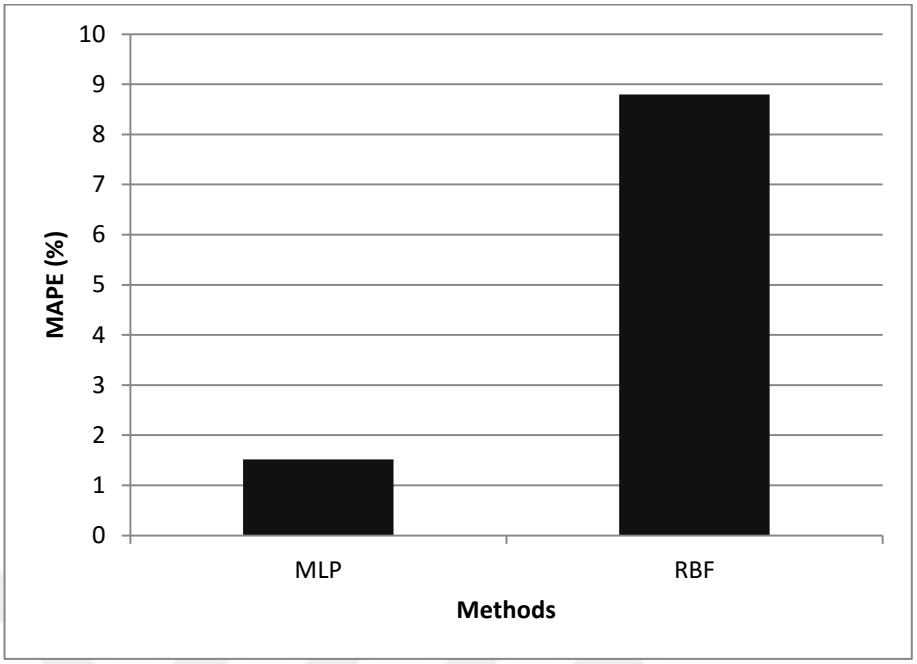

Figure 4.16. The Illustration of average *MAPE*'s of MLP and RBF based models in the ANN category (DS2-5M)

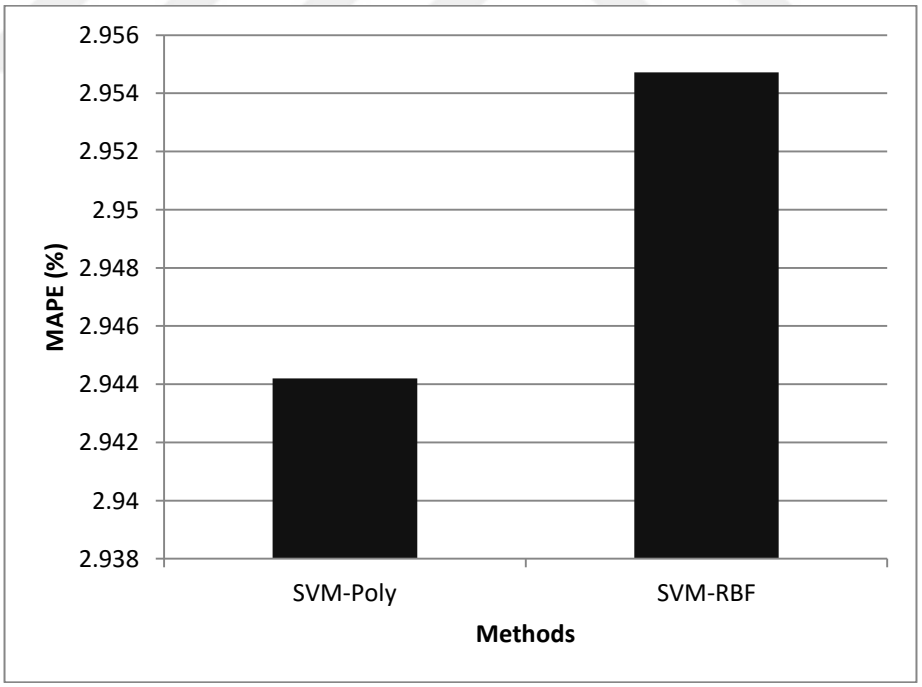

Figure 4.17. Illustration of average *MAPE*'s of SVM-poly and SVM-RBF based models in the SVM category (DS1-5M)

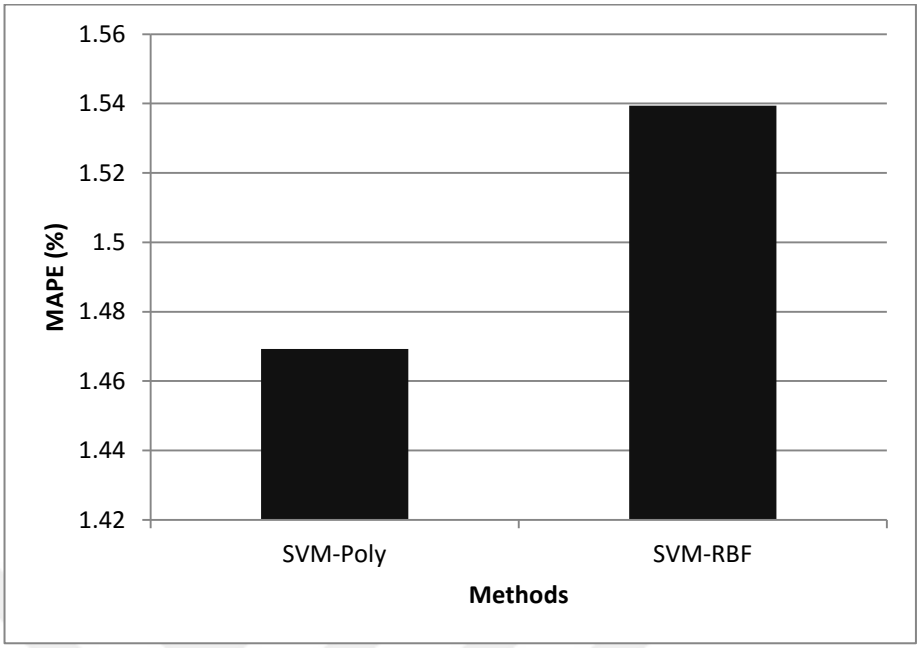

Figure 4.18. Illustration of average *MAPE*'s of SVM-poly and SVM-RBF based models in the SVM category (DS2-5M)

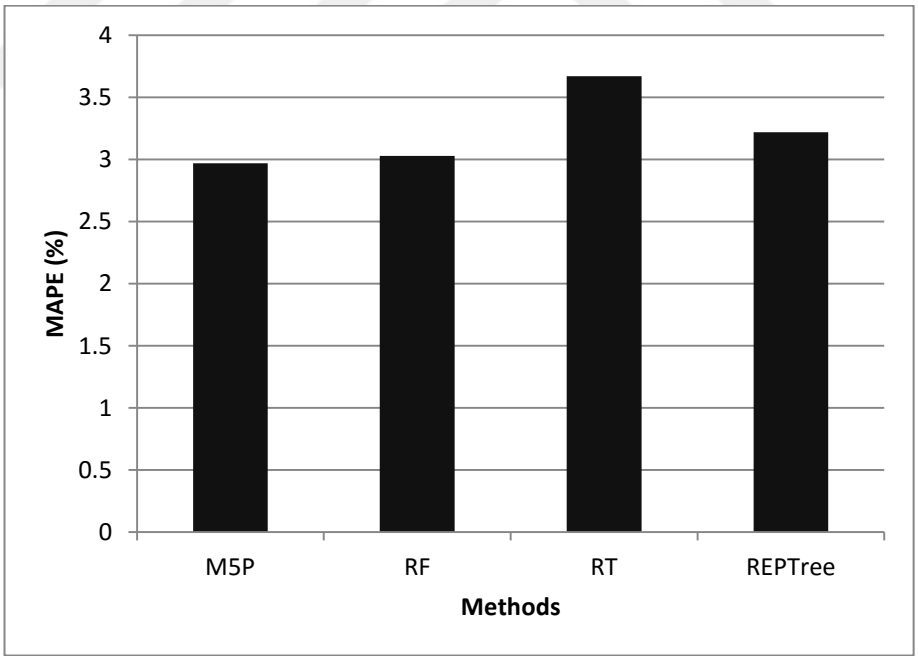

Figure 4.19. Illustration of average *MAPE*'s of M5P, RF, RT, and REPTree based models in the Decision Trees category (DS1-5M)

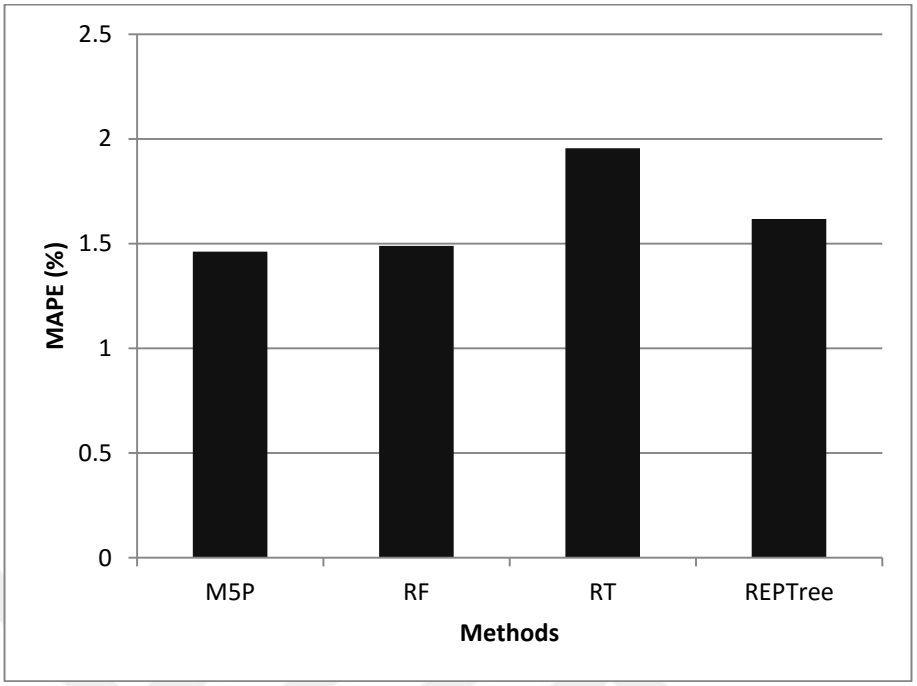

Figure 4.20. Illustration of average *MAPE*'s of M5P, RF, RT, and REPTree based models in the Decision Trees category (DS2-5M)

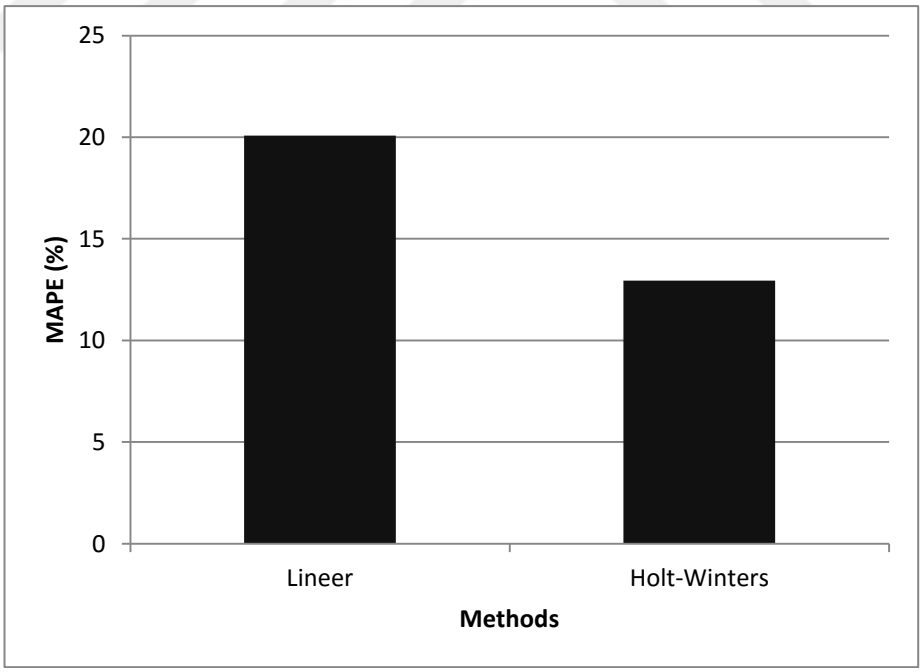

Figure 4.21. Illustration of average *MAPE*'s of Linear and Holt-Winters based models in the statistical category (DS1-1H)

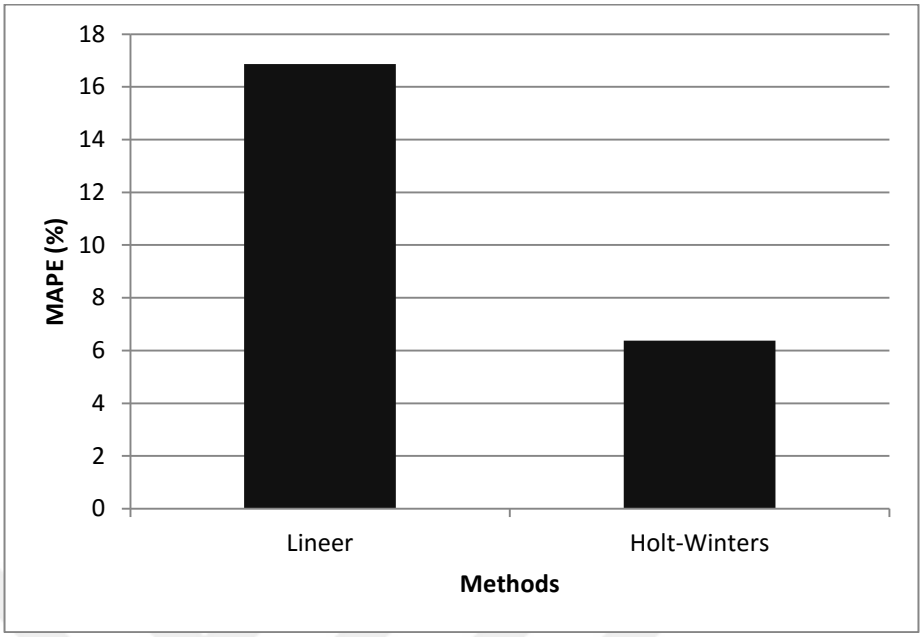

Figure 4.22. Illustration of average *MAPE*'s of Linear and Holt-Winters based models in the statistical category (DS2-1H)

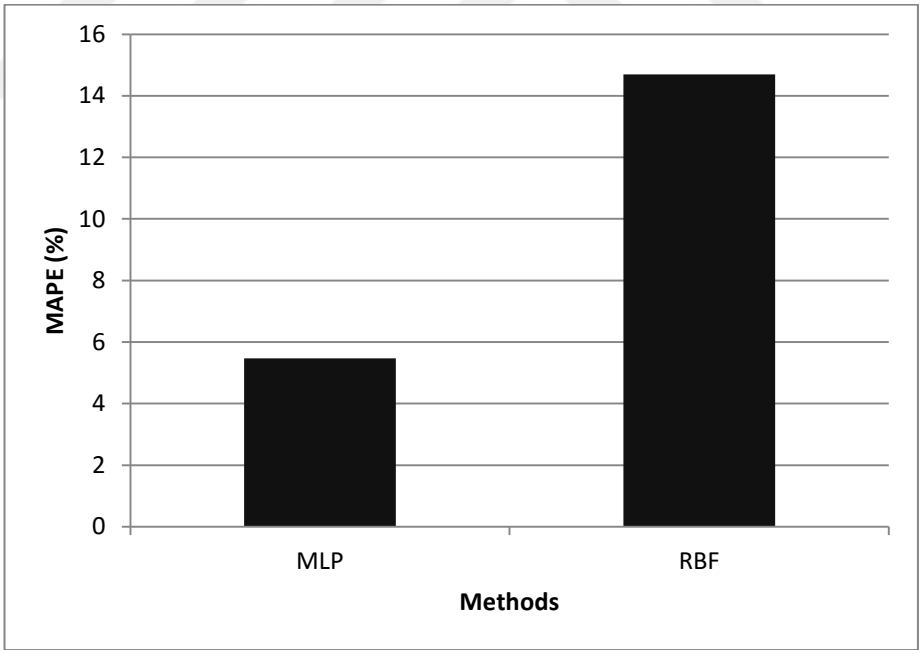

Figure 4.23. Illustration of average *MAPE*'s of MLP and RBF based models in the ANN category (DS1-1H)

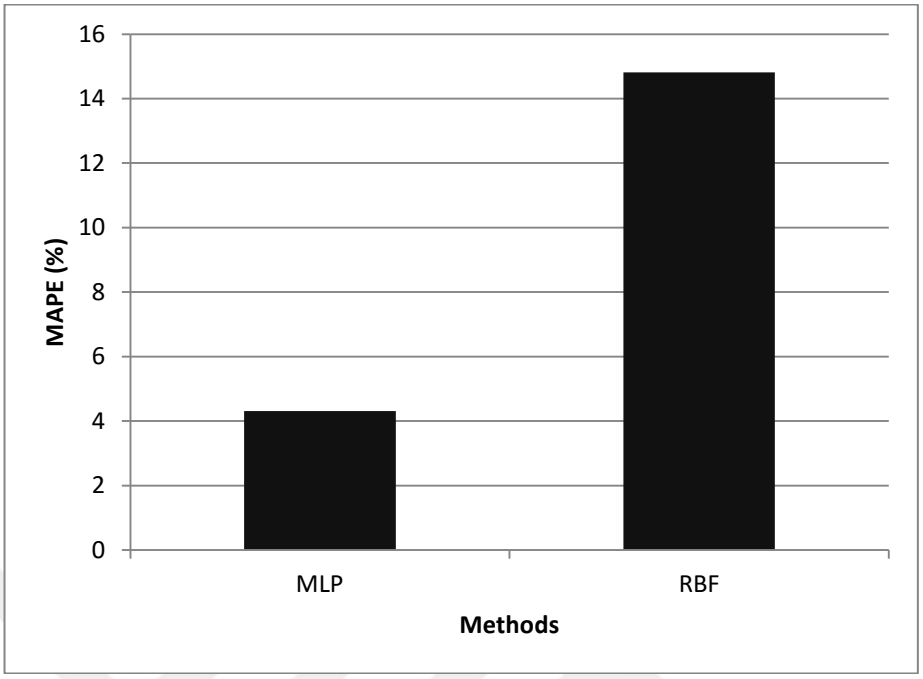

Figure 4.24. Illustration of average *MAPE*'s of MLP and RBF based models in the ANN category (DS2-1H)

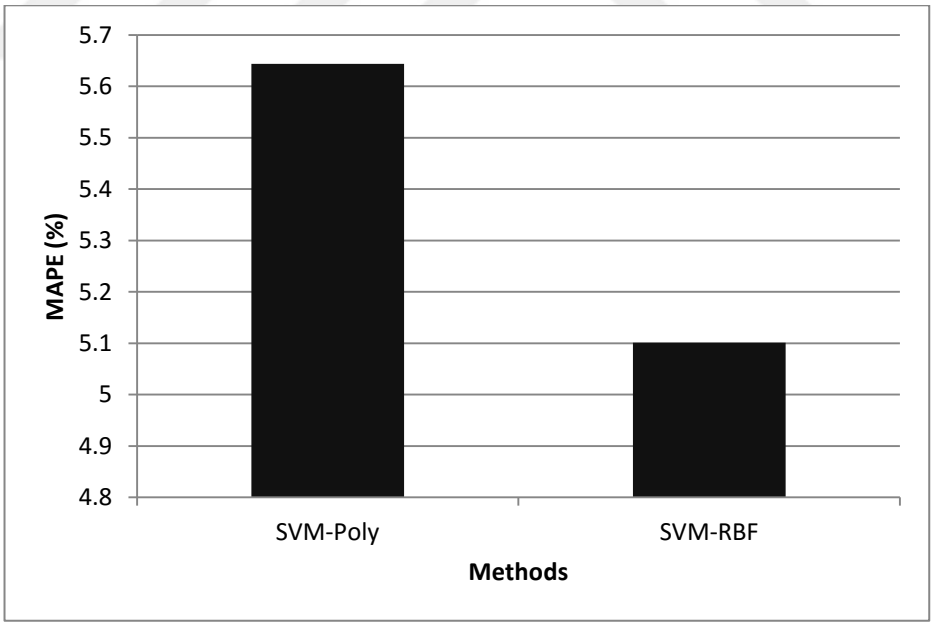

Figure 4.25. Illustration of average *MAPE*'s of SVM-poly and SVM-RBF based models in the SVM category (DS1-1H)

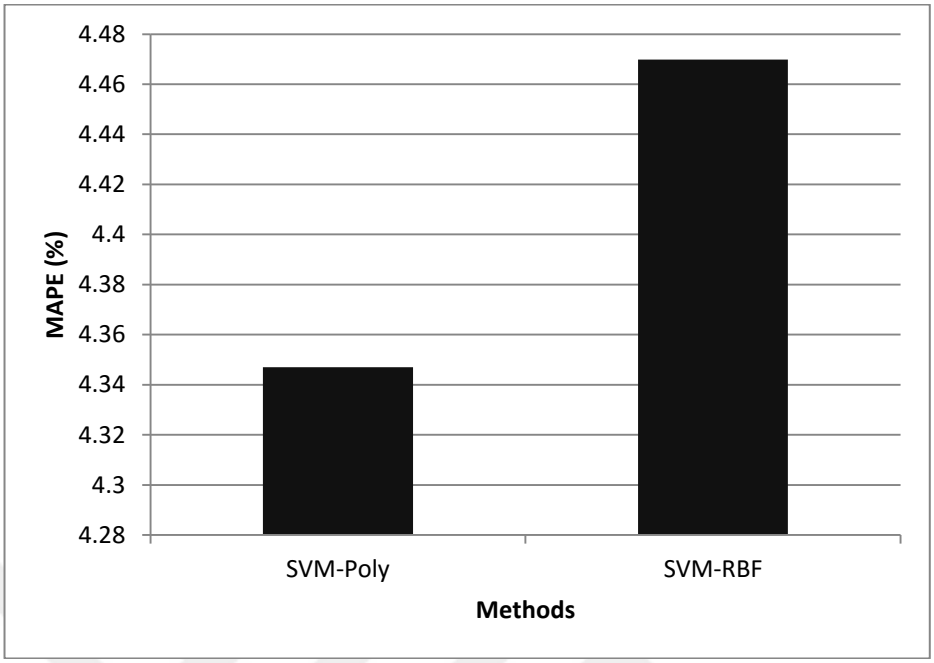

Figure 4.26. Illustration of average *MAPE*'s of SVM-poly and SVM-RBF based models in the SVM category (DS2-1H)

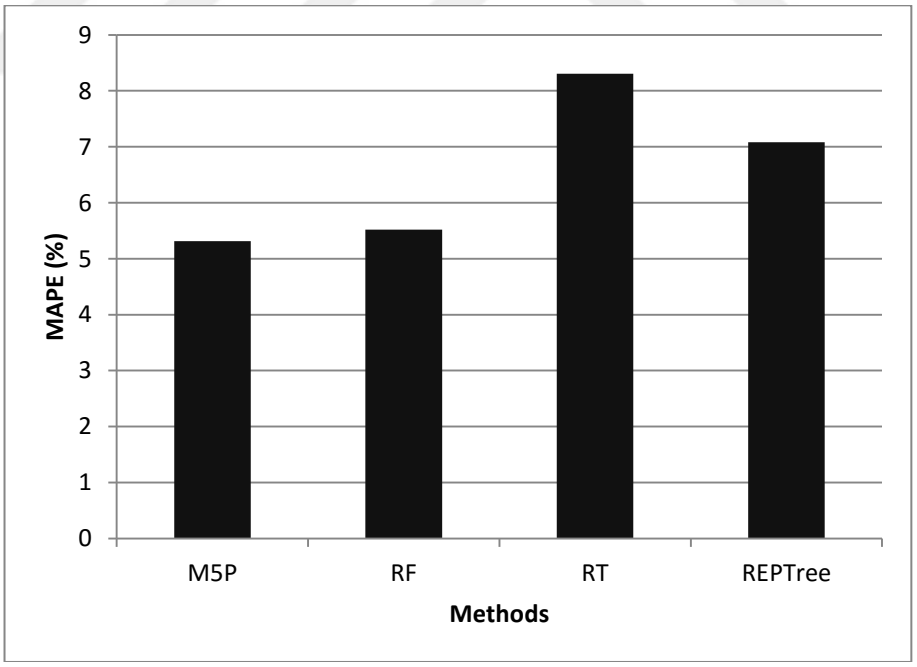

Figure 4.27. Illustration of average *MAPE*'s of M5P, RF, RT, and REPTree based models in the Decision Trees category (DS1-1H)

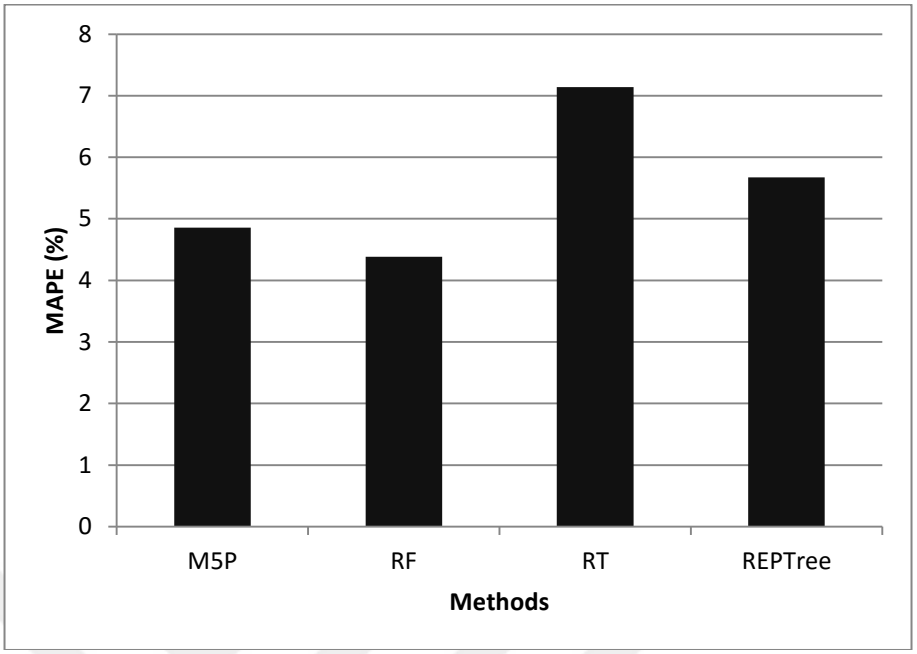

Figure 4.28. Illustration of average *MAPE*'s of M5P, RF, RT, and REPTree based models in the Decision Trees category (DS2-1H)

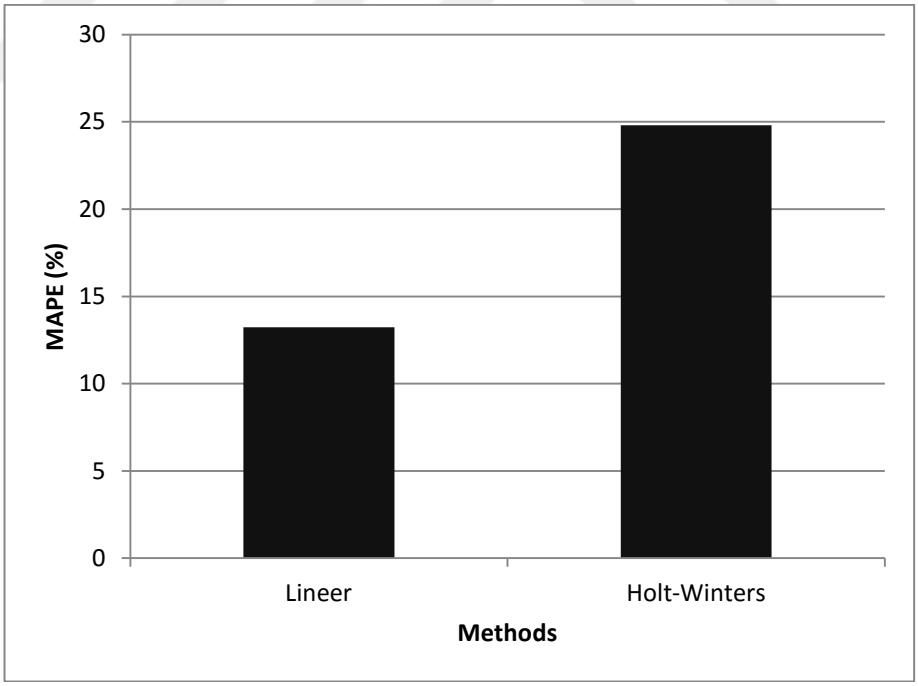

Figure 4.29. Illustration of average *MAPE*'s of Linear and Holt-Winters based models in the statistical category (DS1-1D)

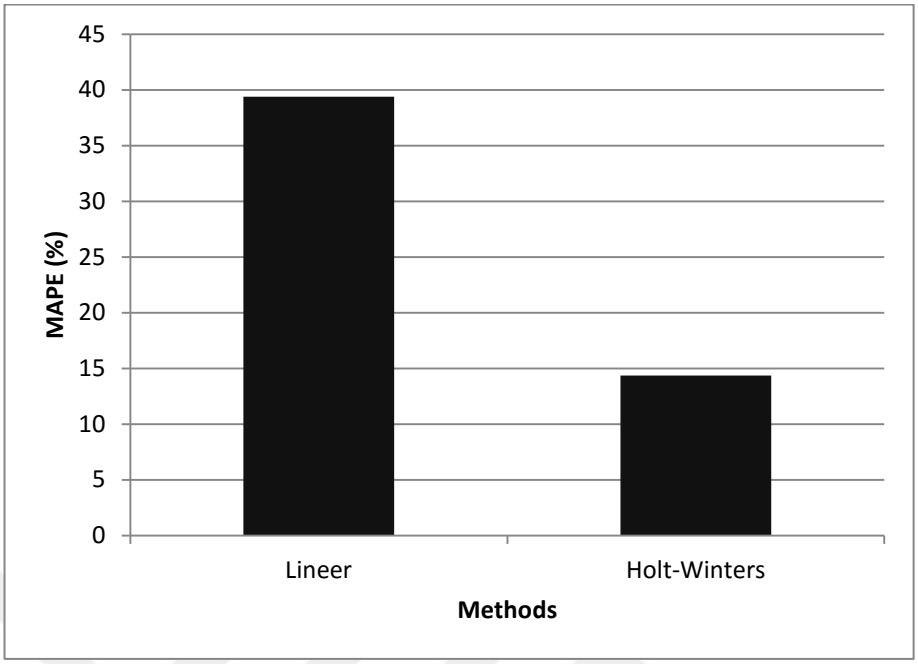

Figure 4.30. Illustration of average *MAPE*'s of Linear and Holt-Winters based models in the statistical category (DS2-1D)

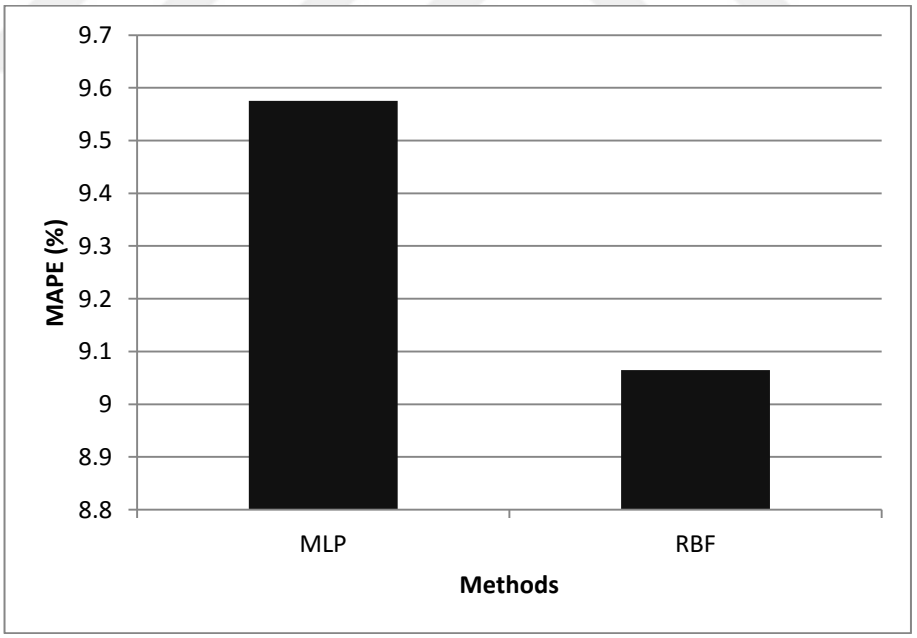

Figure 4.31. Illustration of average *MAPE*'s of MLP and RBF based models in the ANN category (DS1-1D)

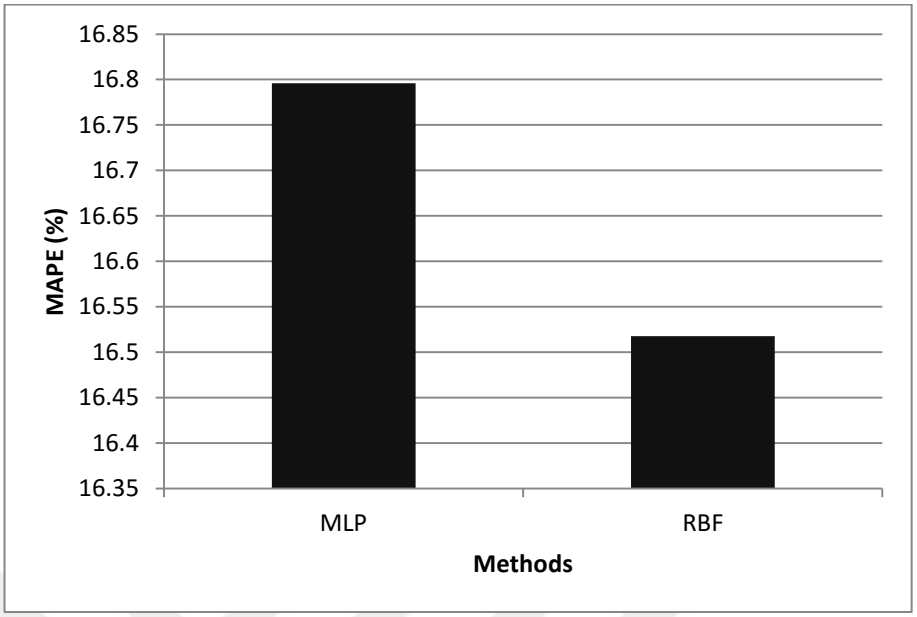

Figure 4.32. Illustration of average *MAPE*'s of MLP and RBF based models in the ANN category (DS2-1D)

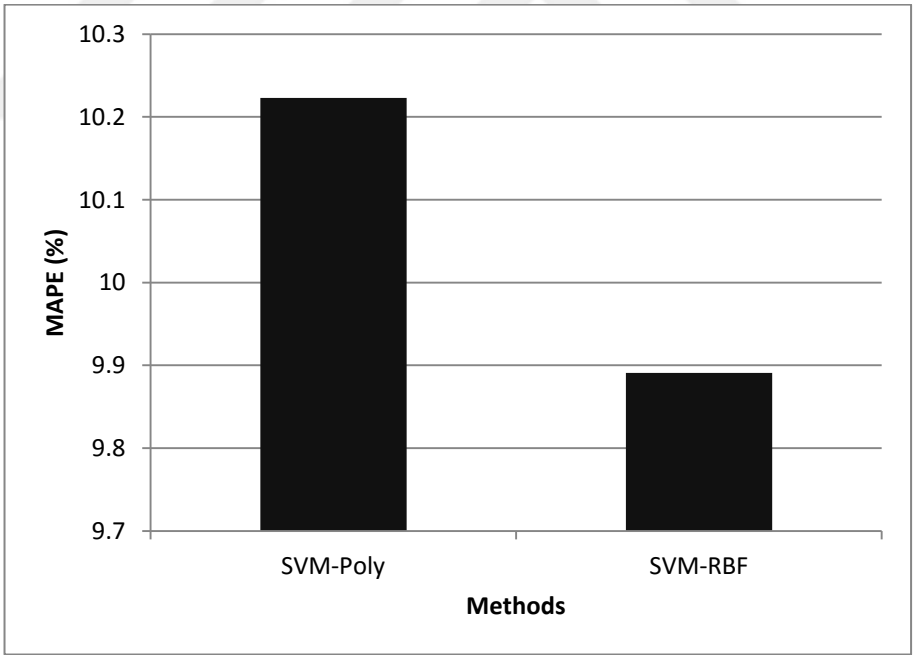

Figure 4.33. Illustration of average *MAPE*'s of SVM-poly and SVM-RBF based models in the SVM category (DS1-1D)

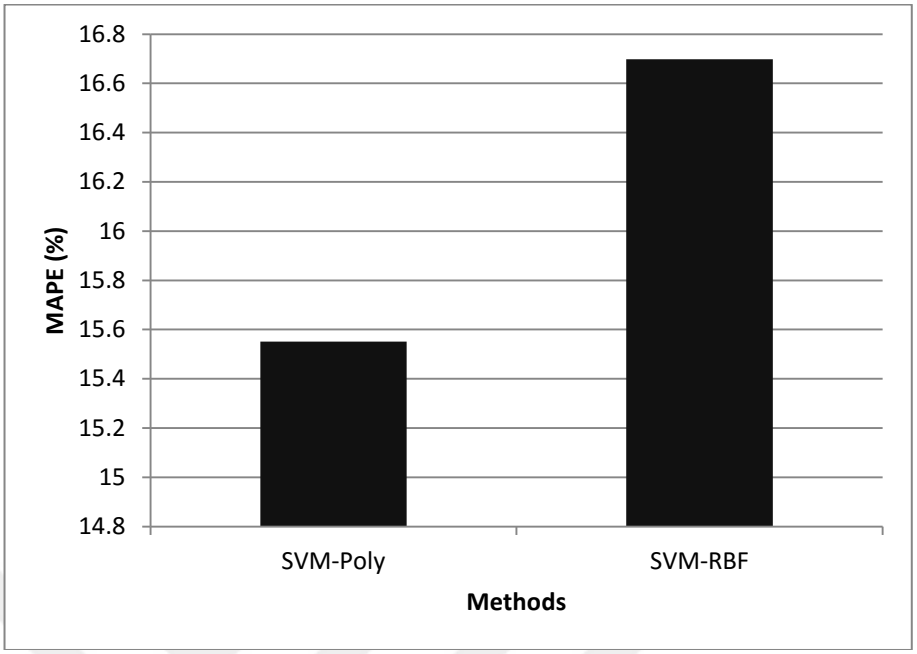

Figure 4.34. Illustration of average *MAPE*'s of SVM-poly and SVM-RBF based models in the SVM category (DS2-1D)

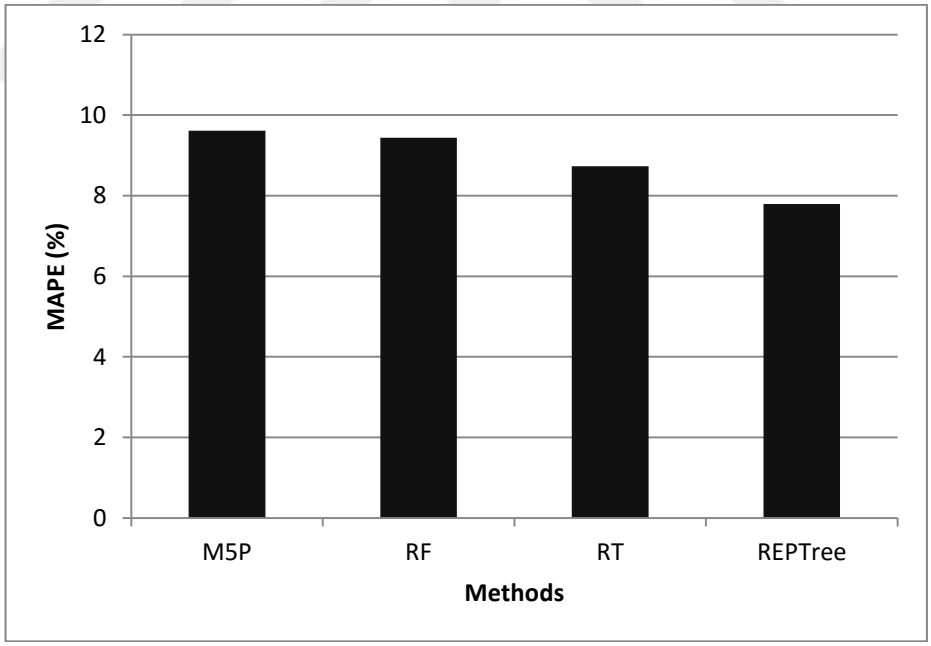

Figure 4.35. Illustration of average *MAPE*'s of M5P, RF, RT, and REPTree based models in the Decision Trees category (DS1-1D)

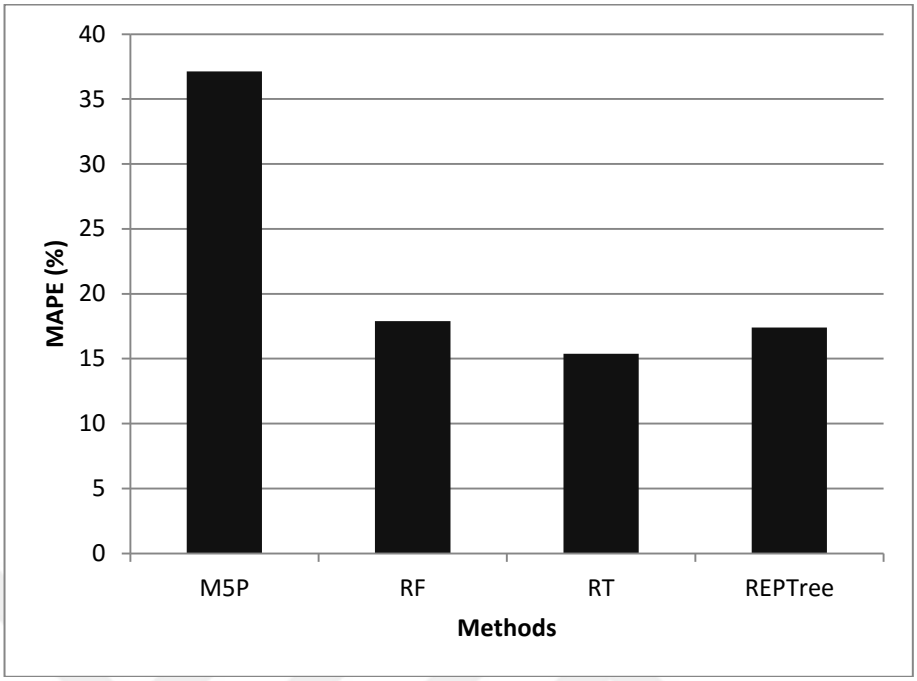

Figure 4.36. Illustration of average *MAPE*'s of M5P, RF, RT, and REPTree based models in the Decision Trees category (DS2-1D)

Figure 4.37. through Figure 4.42 show the average of the *MAPE*'s of the forecasting models in each category for each data set, separately.

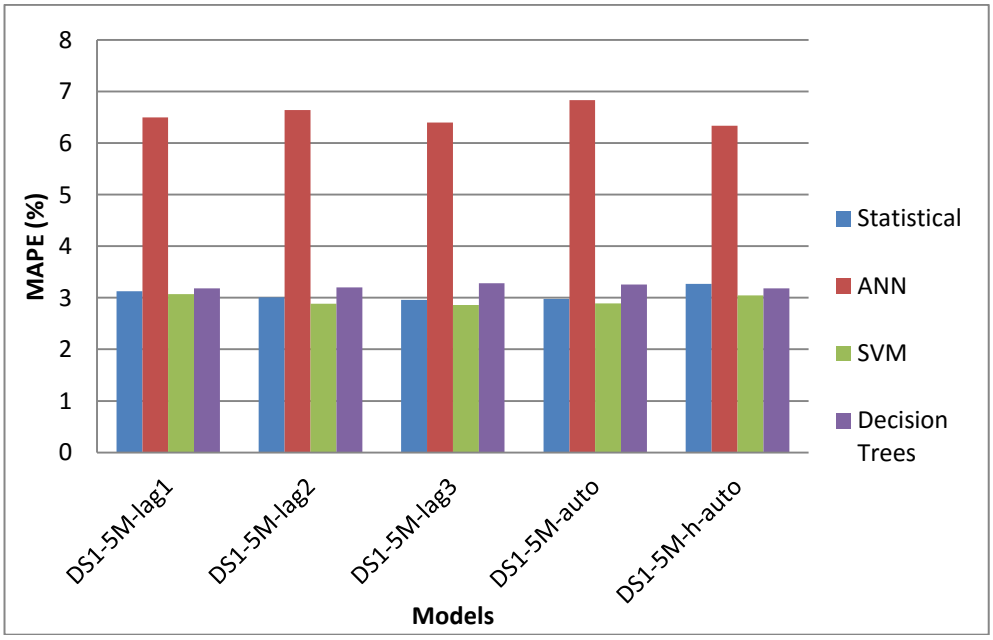

Figure 4.37. Illustration of average *MAPE*'s of forecasting models in each category (DS1-5M)

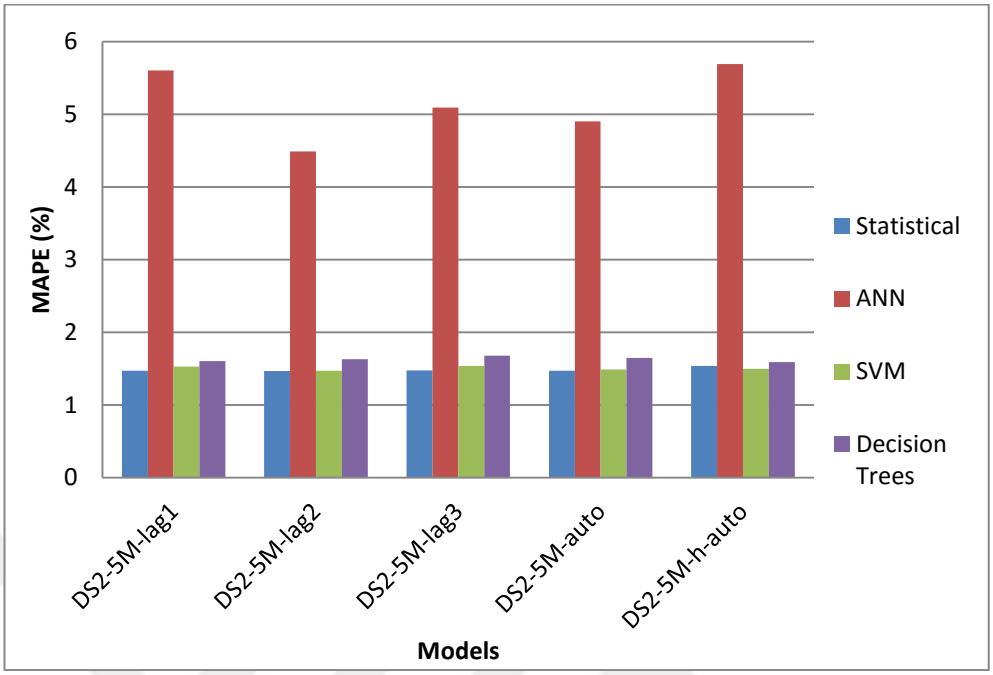

Figure 4.38. Illustration of average *MAPE*'s of forecasting models in each category (DS2-5M)

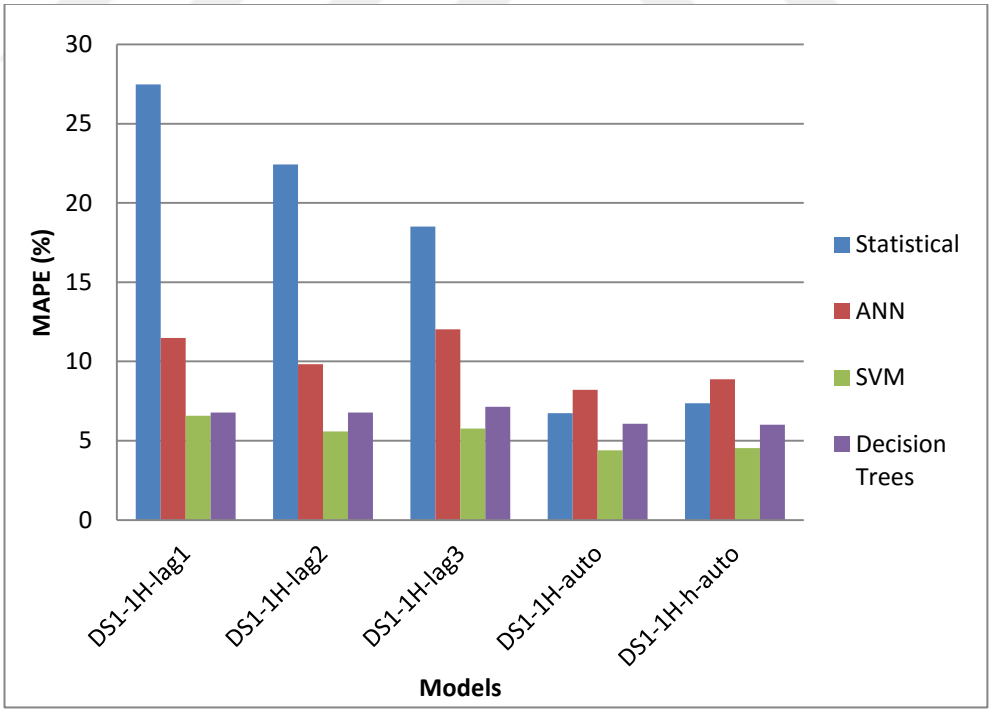

Figure 4.39. Illustration of average *MAPE*'s of forecasting models in each category (DS1-1H)

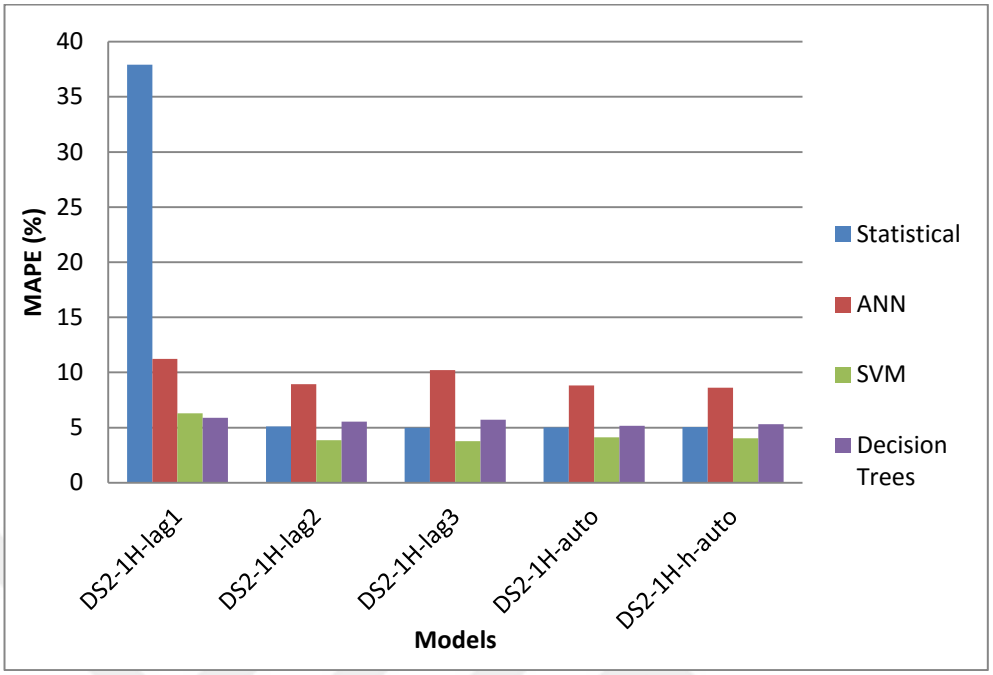

Figure 4.40. Illustration of average *MAPE*'s of forecasting models in each category (DS2-1H)

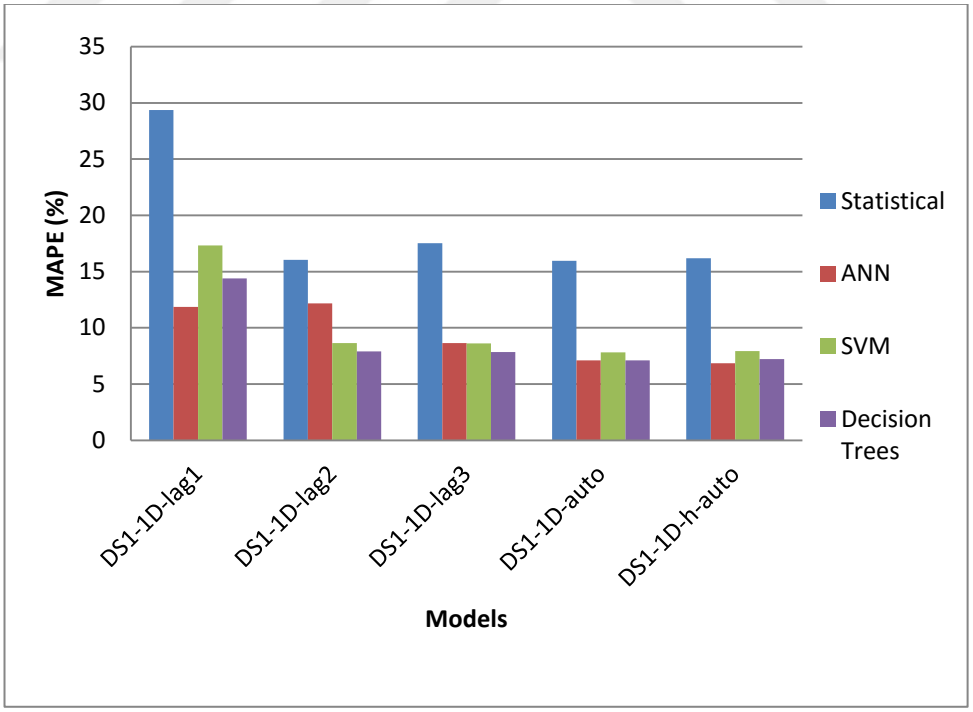

Figure 4.41. Illustration of average *MAPE*'s of forecasting models in each category (DS1-1D)

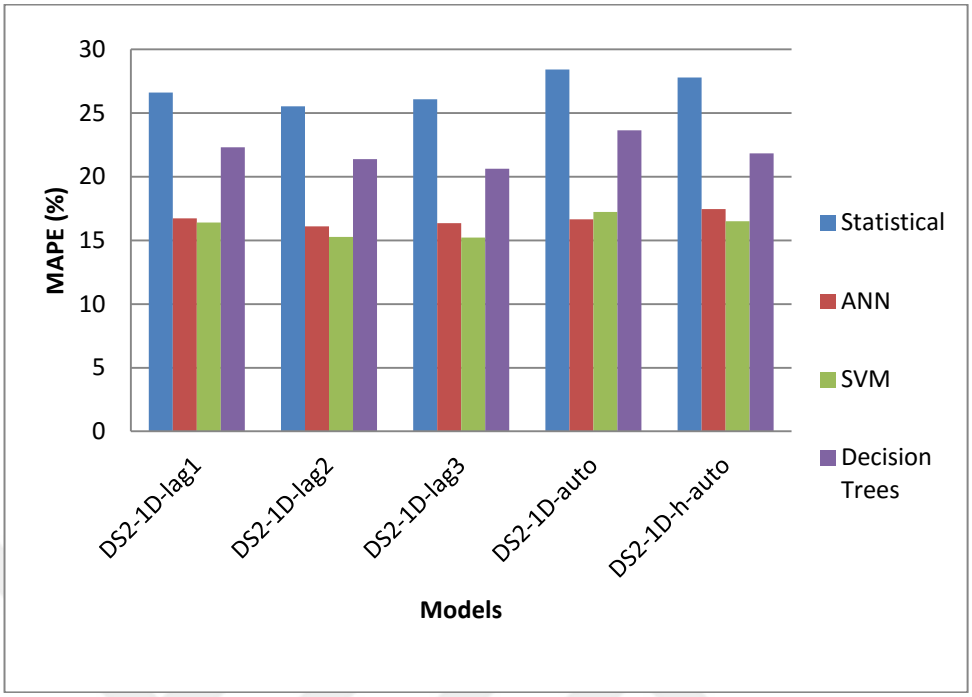

Figure 4.42. Illustration of average *MAPE*'s of forecasting models in each category (DS2-1D)

Figure 4.43. through Figure 4.66. show the percentage decrement rates in *MAPE*'s between the forecasting model having the lowest *MAPE*'s on the average and the other forecasting models in each categories for each data set.

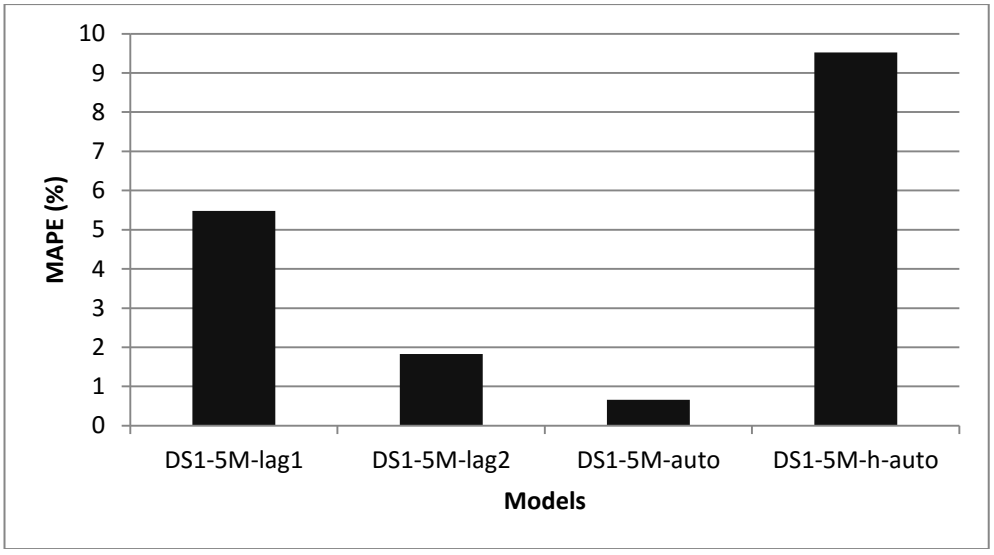

Figure 4.43. Percentage decrease rates in average *MAPE* of the forecasting model of DS1-5M-lag3 compared to ones obtained by the rest of forecasting models in the statistical category (DS1-5M)

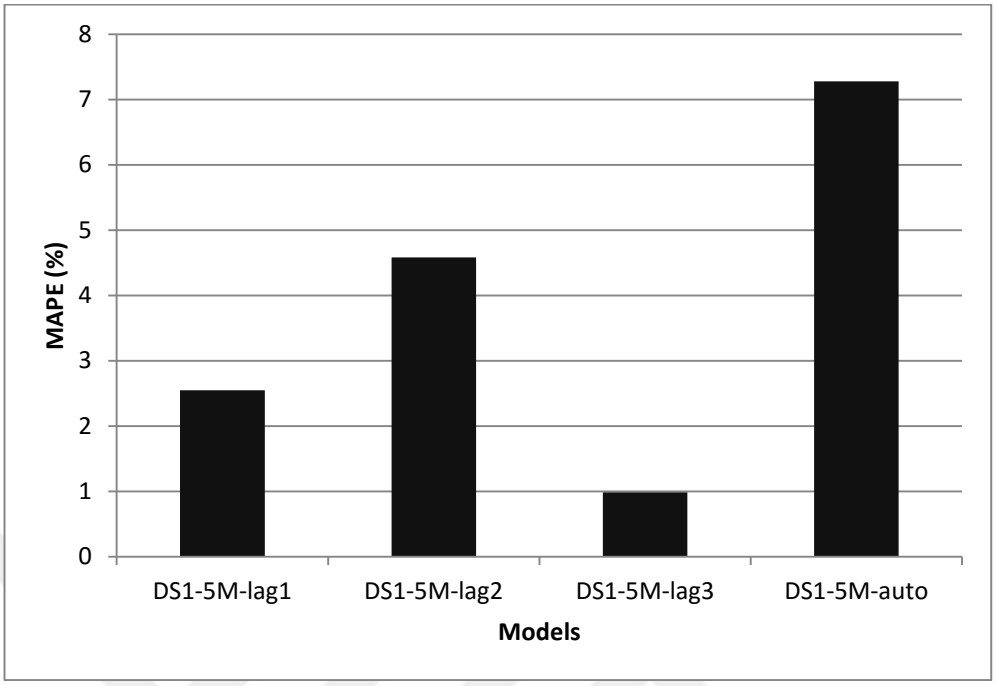

Figure 4.44. Percentage decrease rates in average *MAPE* of the forecasting model of DS1-5M-h-auto compared to ones obtained by the rest of forecasting models in the ANN category (DS1-5M)

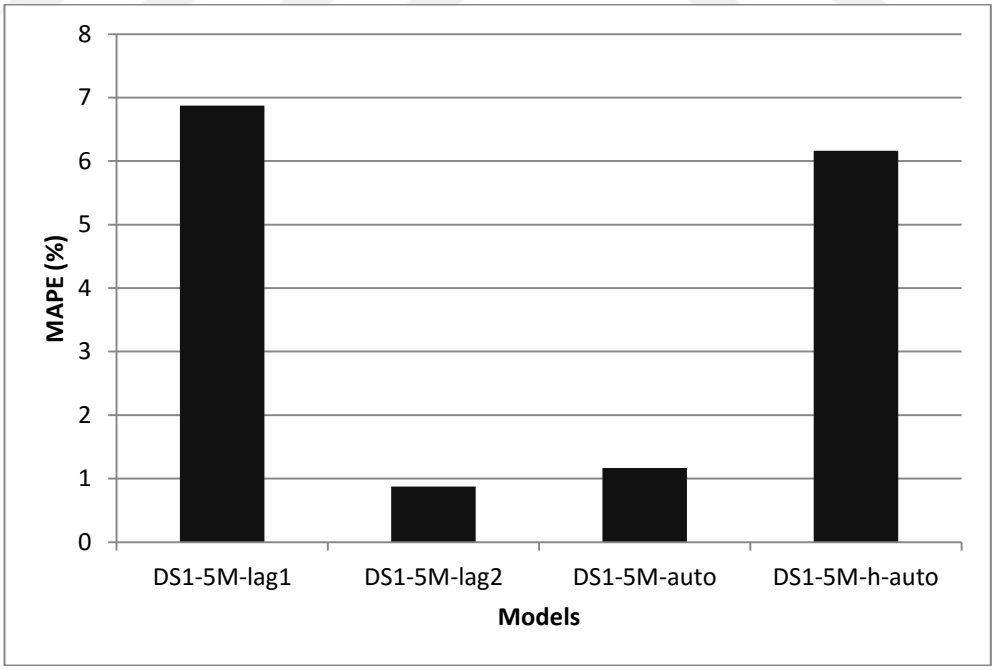

Figure 4.45. Percentage decrease rates in average *MAPE* of the forecasting model of DS1-5M-lag3 compared to ones obtained by the rest of forecasting models in the SVM category (DS1-5M)

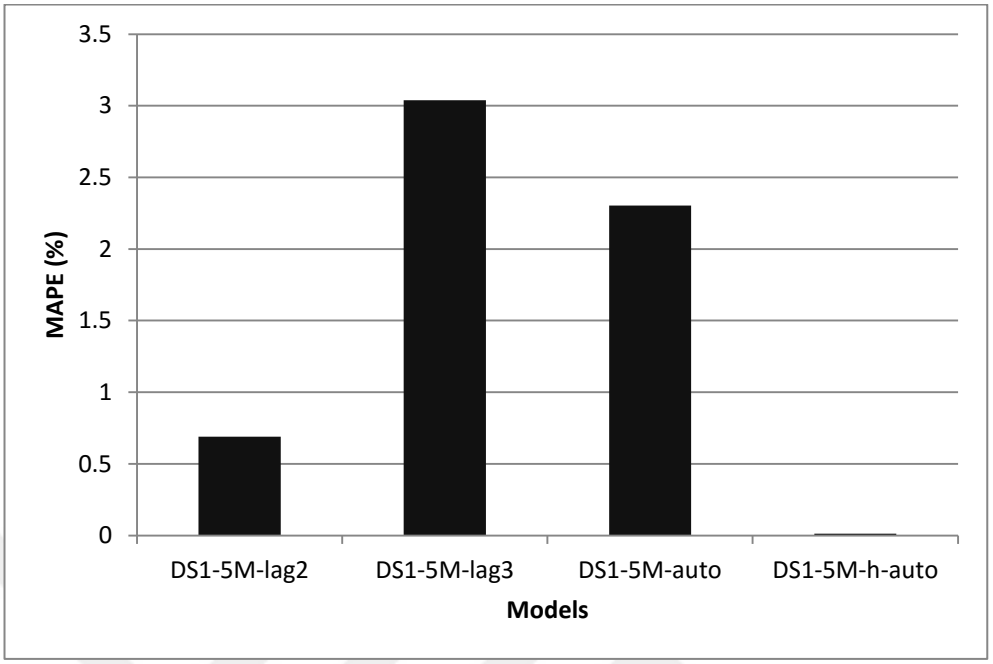

Figure 4.46. Percentage decrease rates in average *MAPE* of the forecasting model of DS1-5M-lag1 compared to ones obtained by the rest of forecasting models in the Decision Trees category (DS1-5M)

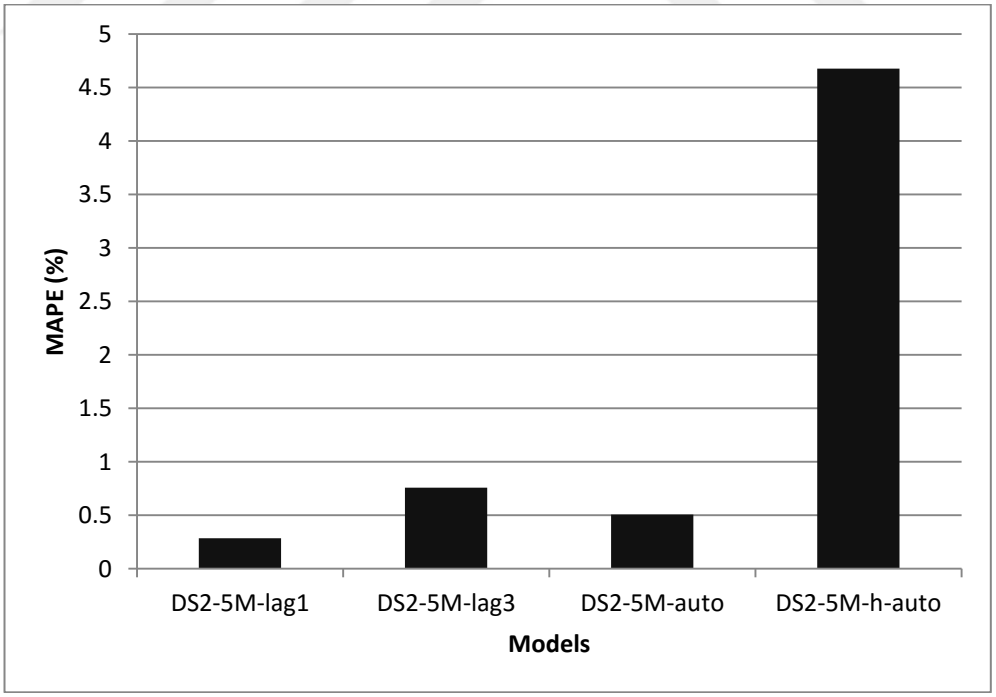

Figure 4.47. Percentage decrease rates in average *MAPE* of the forecasting model of DS2-5M-lag2 compared to ones obtained by the rest of forecasting models in the statistical category (DS2-5M)

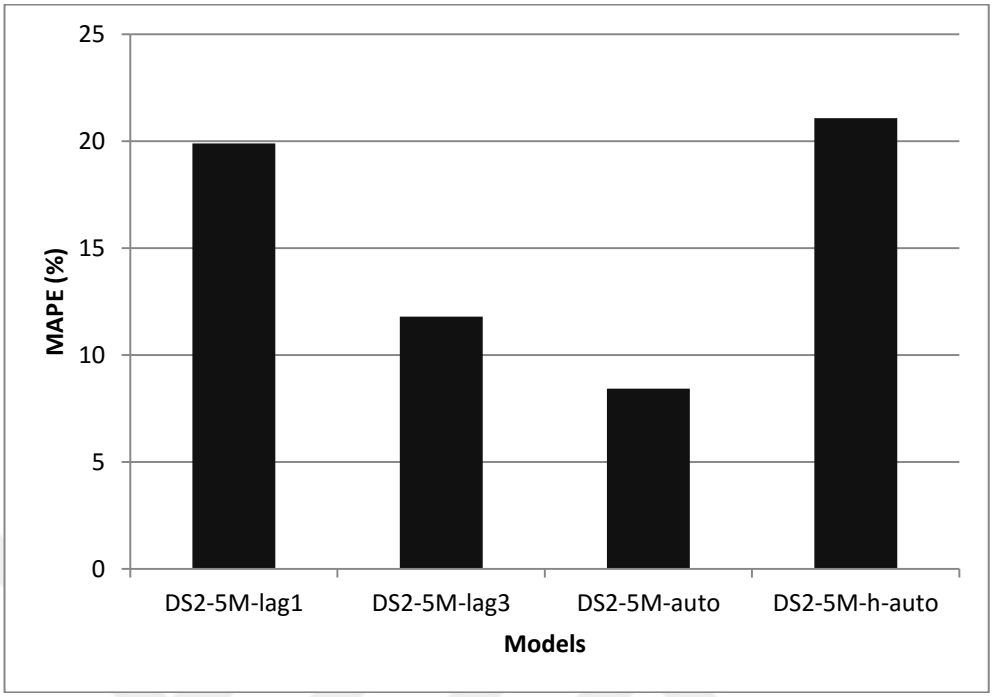

Figure 4.48. Percentage decrease rates in average *MAPE* of the forecasting model of DS2-5M-lag2 compared to ones obtained by the rest of forecasting models in the ANN category (DS2-5M)

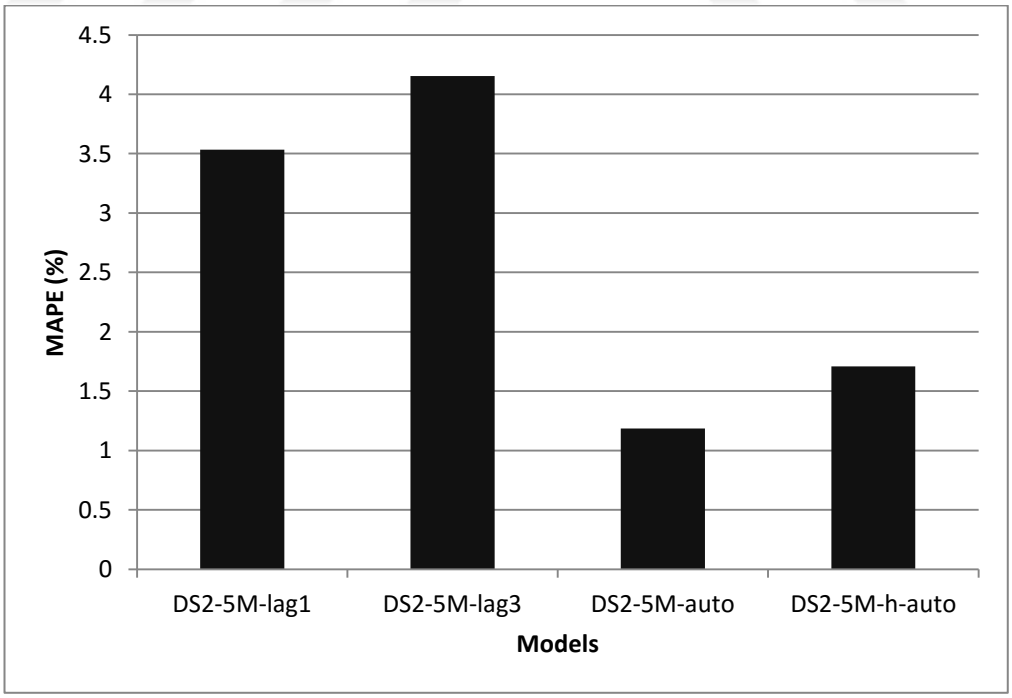

Figure 4.49. Percentage decrease rates in average *MAPE* of the forecasting model of DS2-5M-lag2 compared to ones obtained by the rest of forecasting models in the SVM category (DS2-5M)

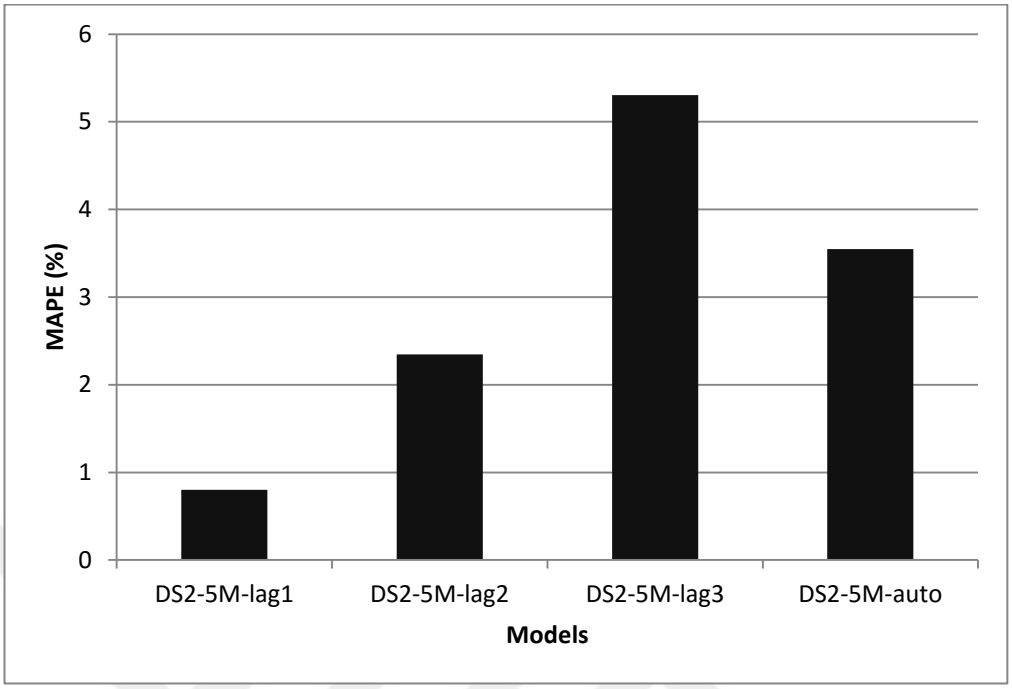

Figure 4.50. Percentage decrease rates in average *MAPE* of the forecasting model of DS2-5M-h-auto compared to ones obtained by the rest of forecasting models in the Decision Trees category (DS2-5M)

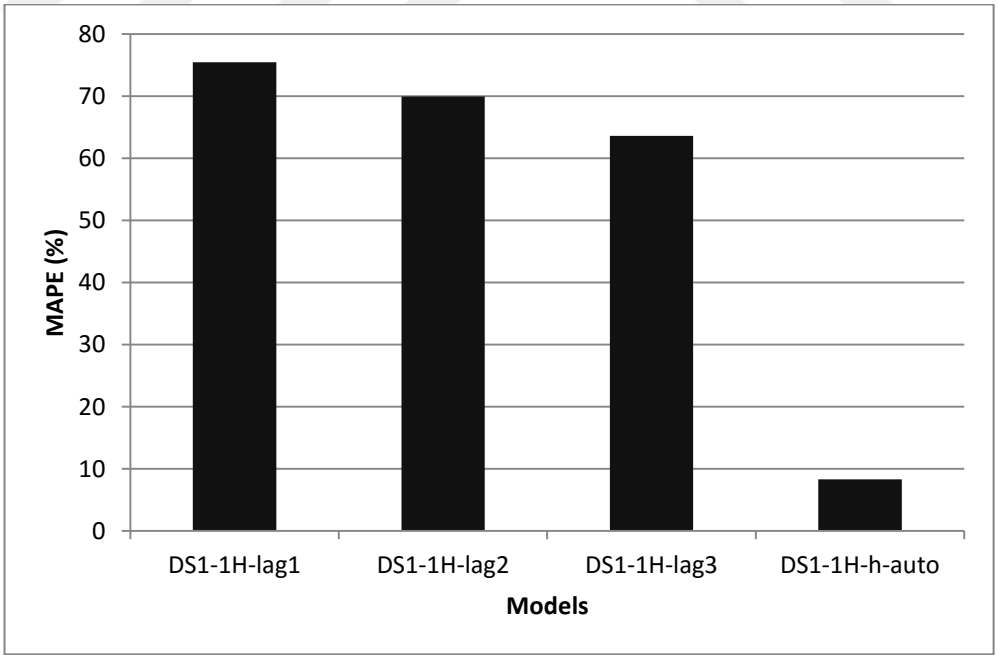

Figure 4.51. Percentage decrease rates in average *MAPE* of the forecasting model of DS1-1H-auto compared to ones obtained by the rest of forecasting models in the statistical category (DS1-1H)

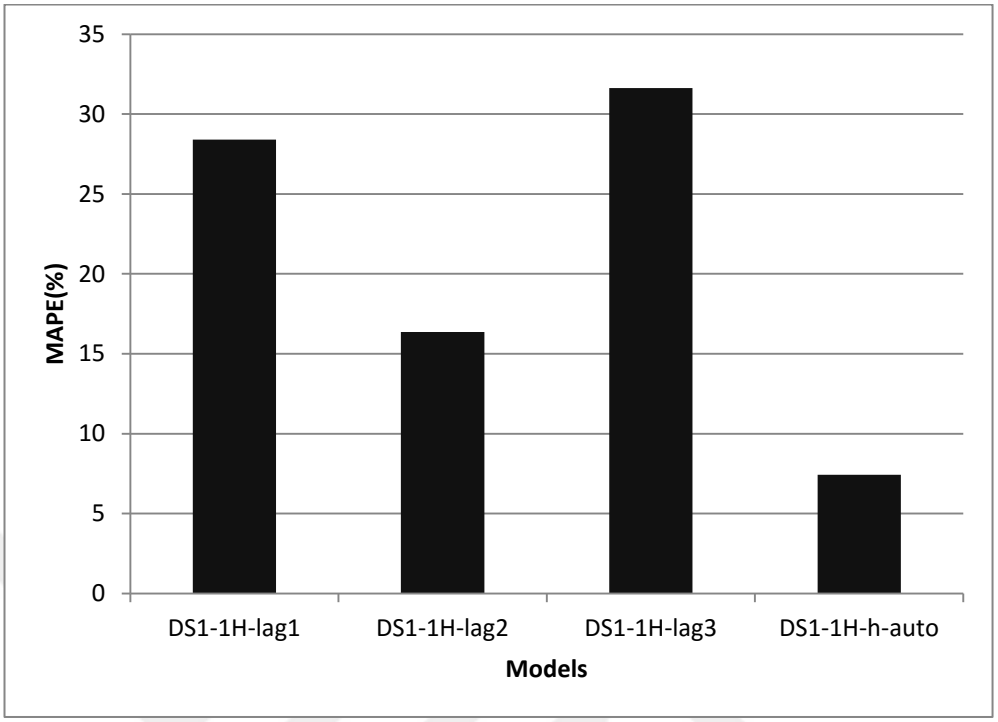

Figure 4.52. Percentage decrease rates in average *MAPE* of the forecasting model of DS1-1H-auto compared to ones obtained by the rest of forecasting models in the ANN category (DS1-1H)

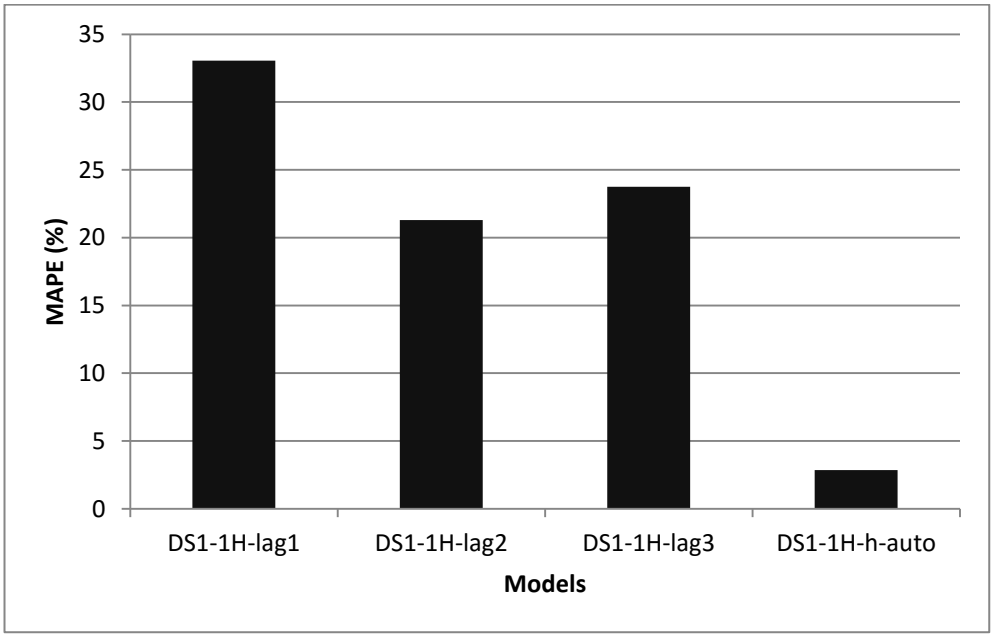

Figure 4.53. Percentage decrease rates in average *MAPE* of the forecasting model of DS1-1H-auto compared to ones obtained by the rest of forecasting models in the SVM category (DS1-1H)

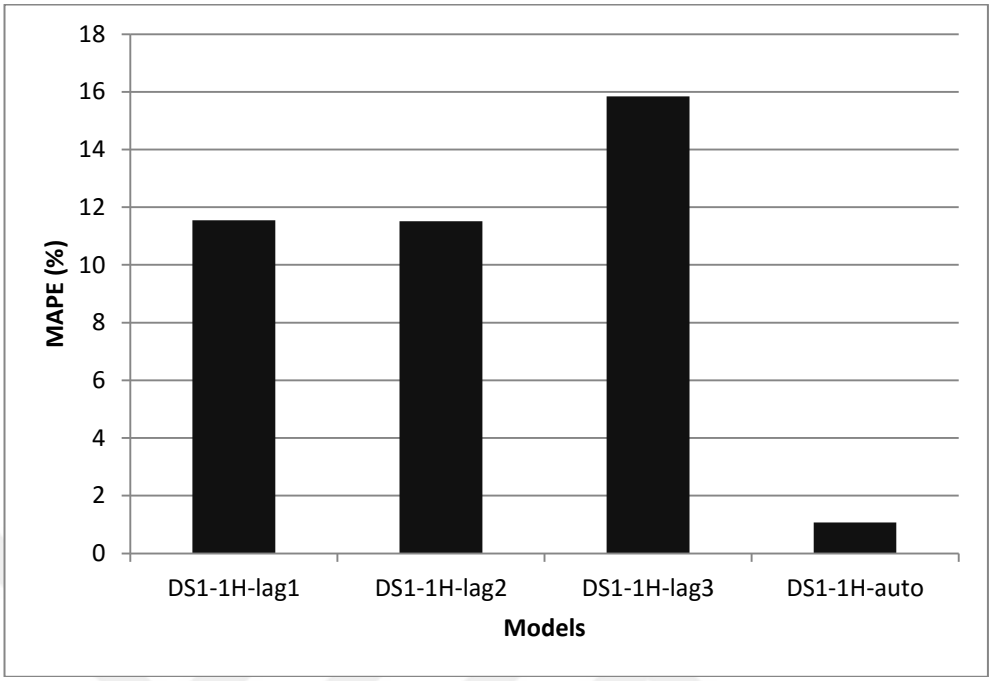

Figure 4.54. Percentage decrease rates in average *MAPE* of the forecasting model of DS1-1H-h-auto compared to ones obtained by the rest of forecasting models in the Decısıon Trees category (DS1-1H)

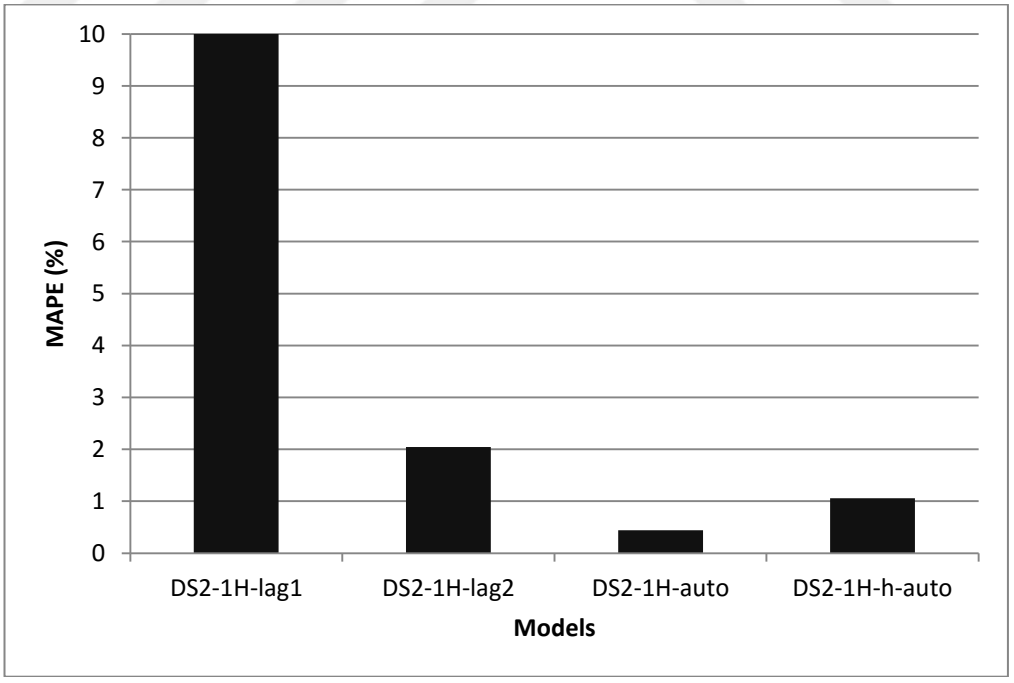

Figure 4.55. Percentage decrease rates in average *MAPE* of the forecasting model of DS2-1H-lag3 compared to ones obtained by the rest of forecasting models in the statistical category (DS2-1H)

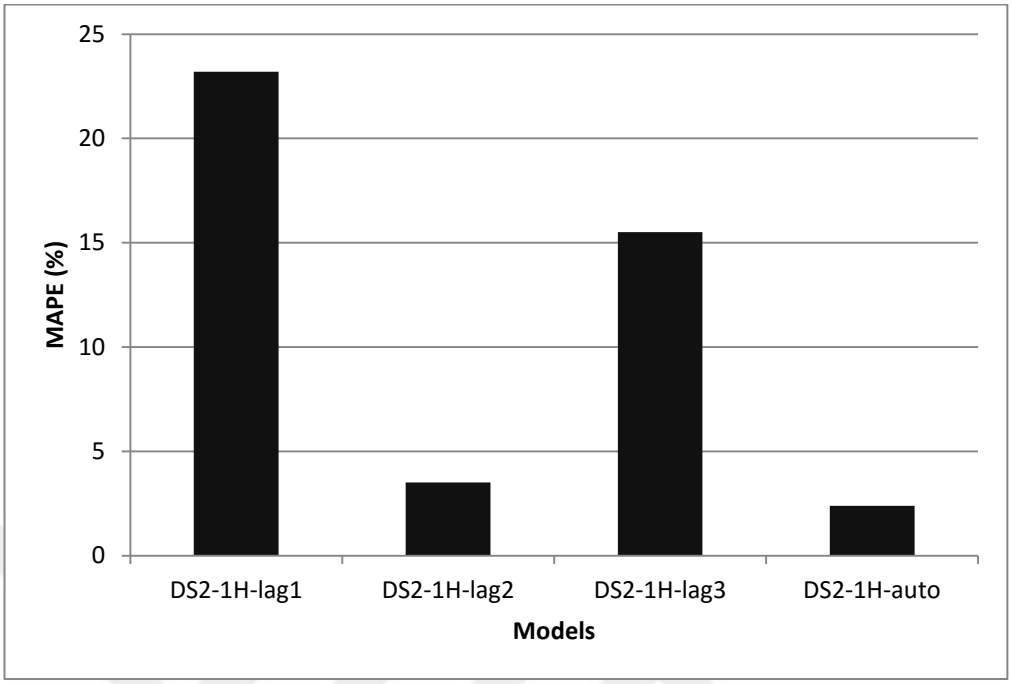

Figure 4.56. Percentage decrease rates in average *MAPE* of the forecasting model of DS2-1H-h-auto compared to ones obtained by the rest of forecasting models in the ANN category (DS2-1H)

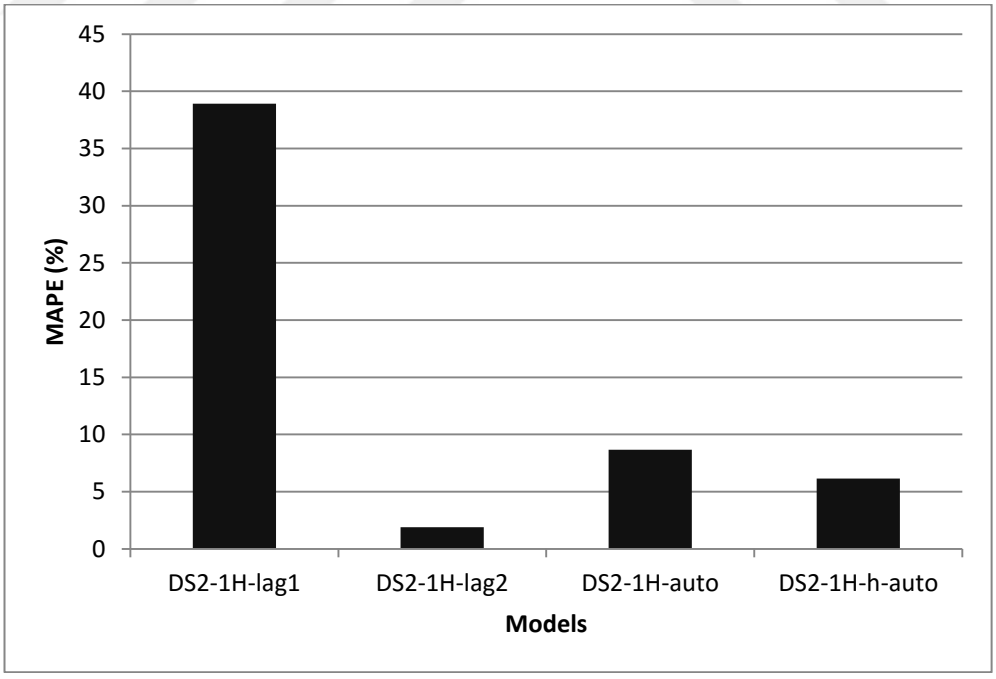

Figure 4.57. Percentage decrease rates in average *MAPE* of the forecasting model of DS2-1H-lag3 compared to ones obtained by the rest of forecasting models in the SVM category (DS2-1H)

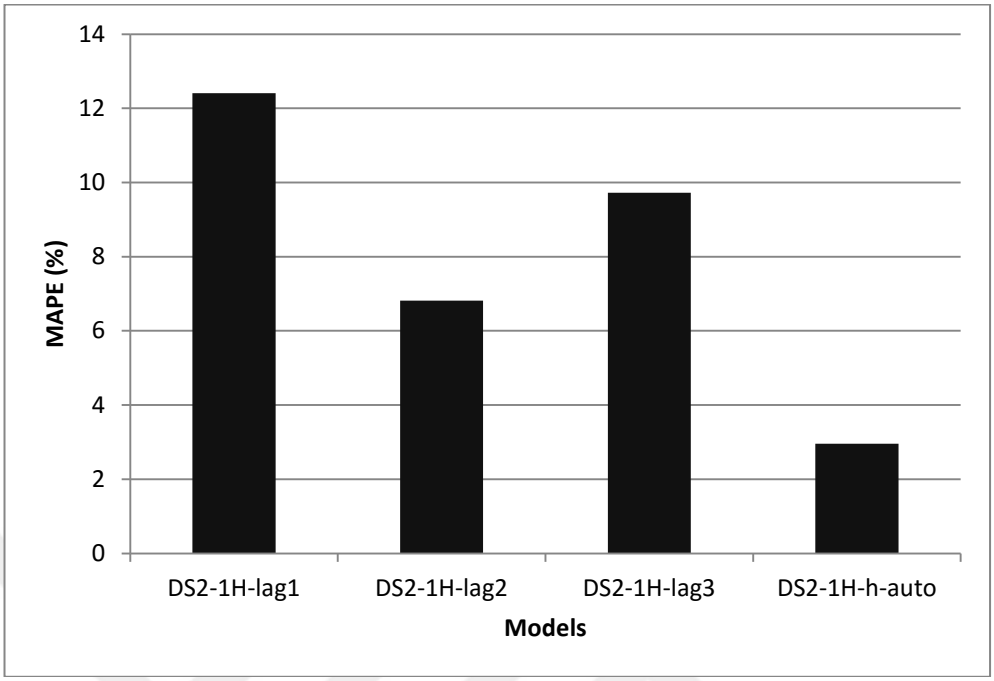

Figure 4.58. Percentage decrease rates in average *MAPE* of the forecasting model of DS2-1H-auto compared to ones obtained by the rest of forecasting models in the Decision Trees category (DS2-1H)

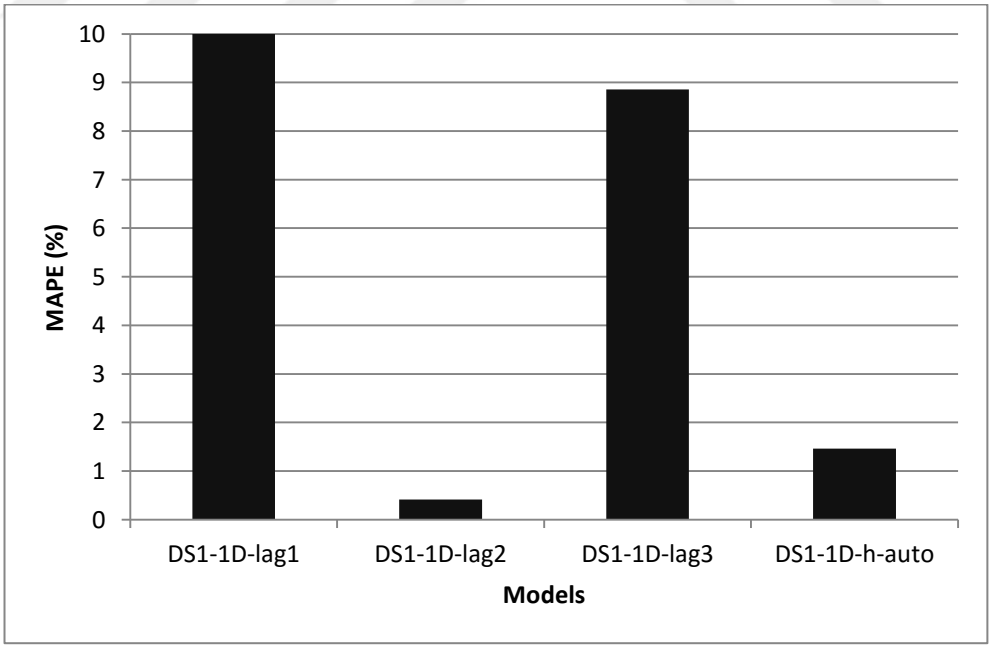

Figure 4.59. Percentage decrease rates in average *MAPE* of the forecasting model of DS1-1D-auto compared to ones obtained by the rest of forecasting models in the statistical category (DS1-1D)

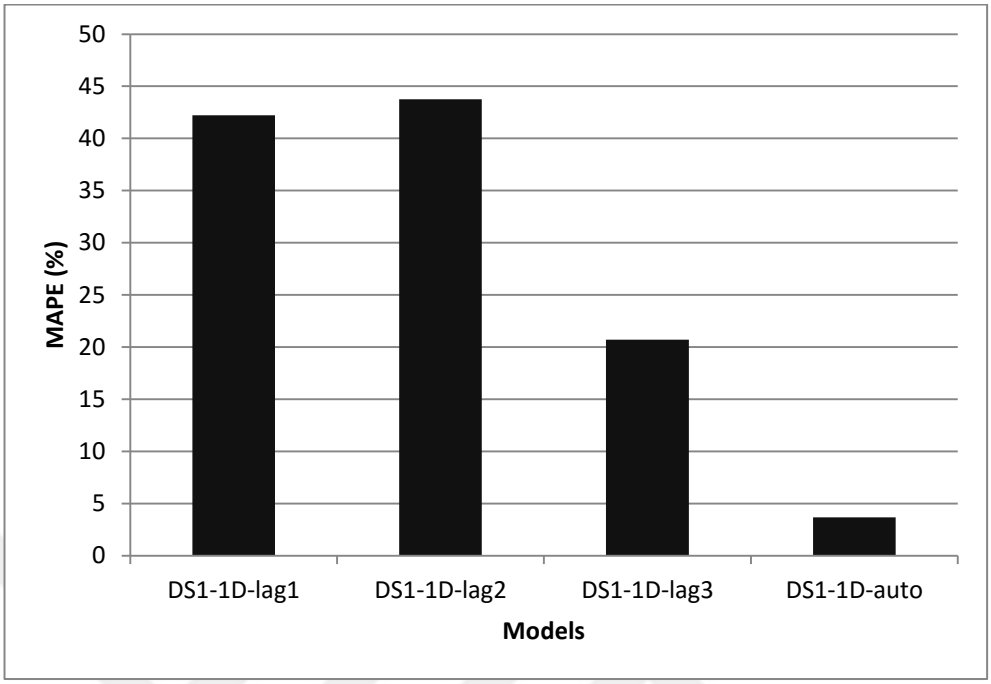

Figure 4.60. Percentage decrease rates in average *MAPE* of the forecasting model of DS1-1D-h-auto compared to ones obtained by the rest of forecasting models in the ANN category (DS1-1D)

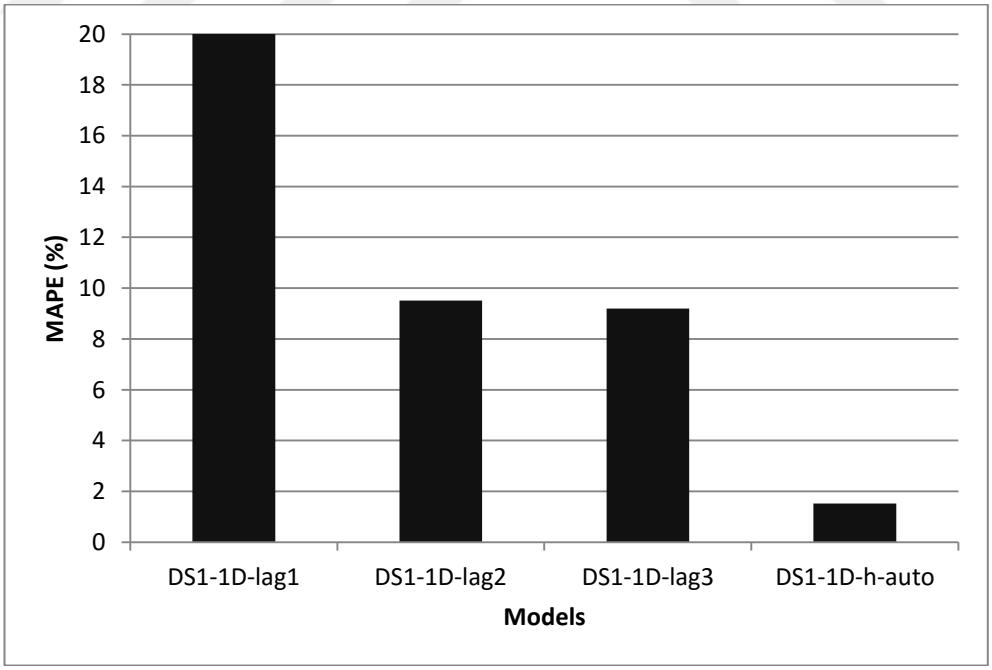

Figure 4.61. Percentage decrease rates in average *MAPE* of the forecasting model of DS1-1D-auto compared to ones obtained by the rest of forecasting models in the SVM category (DS1-1D)

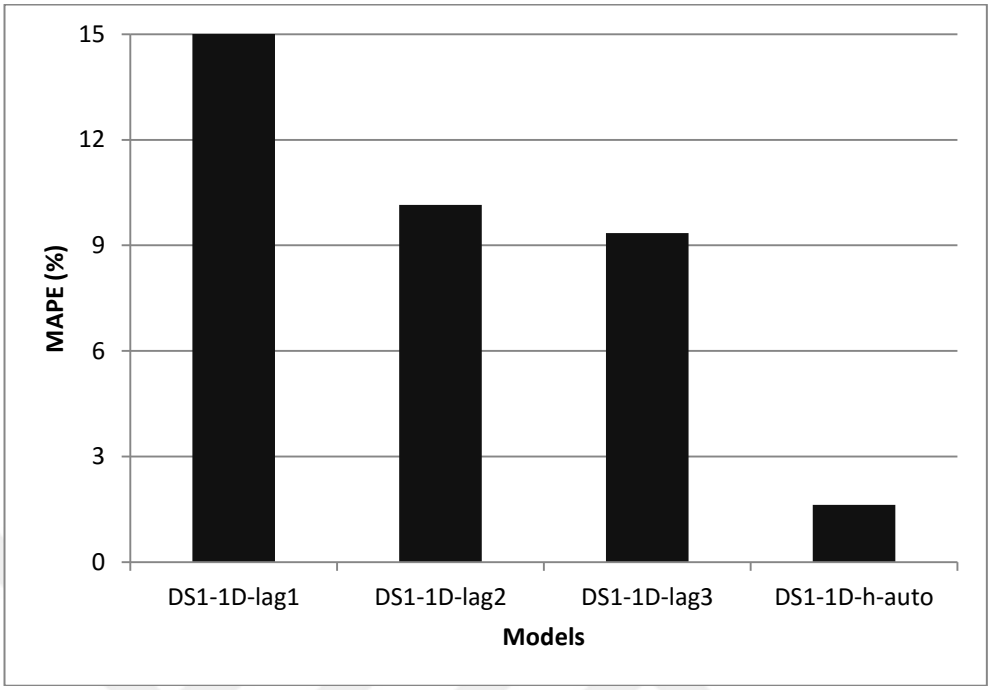

Figure 4.62. Percentage decrease rates in average *MAPE* of the forecasting model of DS1-1D-auto compared to ones obtained by the rest of forecasting models in the Decision Trees category (DS1-1D)

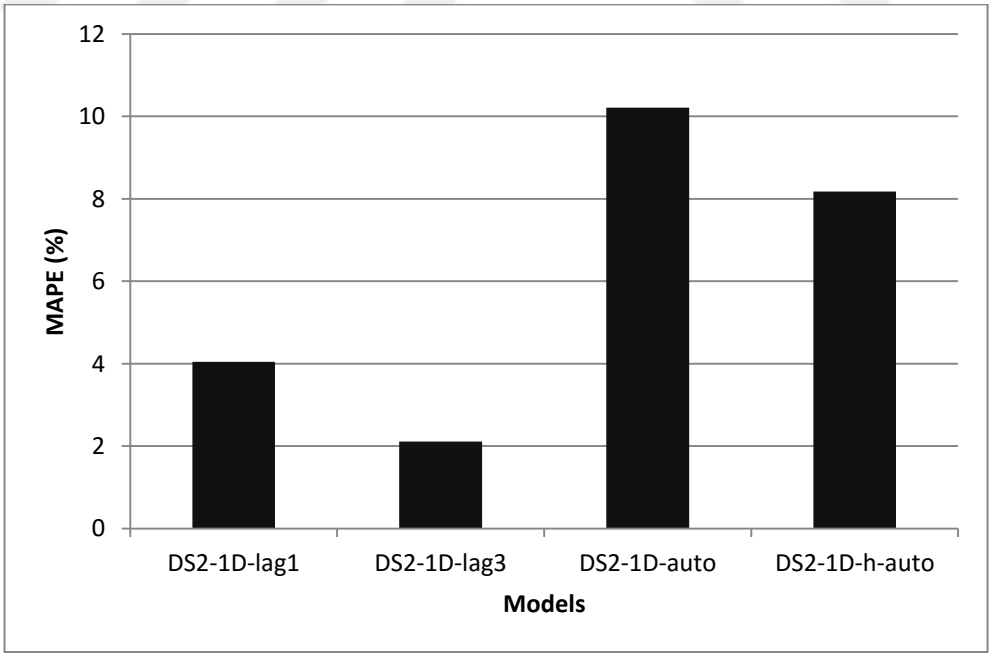

Figure 4.63. Percentage decrease rates in average *MAPE* of the forecasting model of DS2-1D-lag2 compared to ones obtained by the rest of forecasting models in the statistical category (DS2-1D)

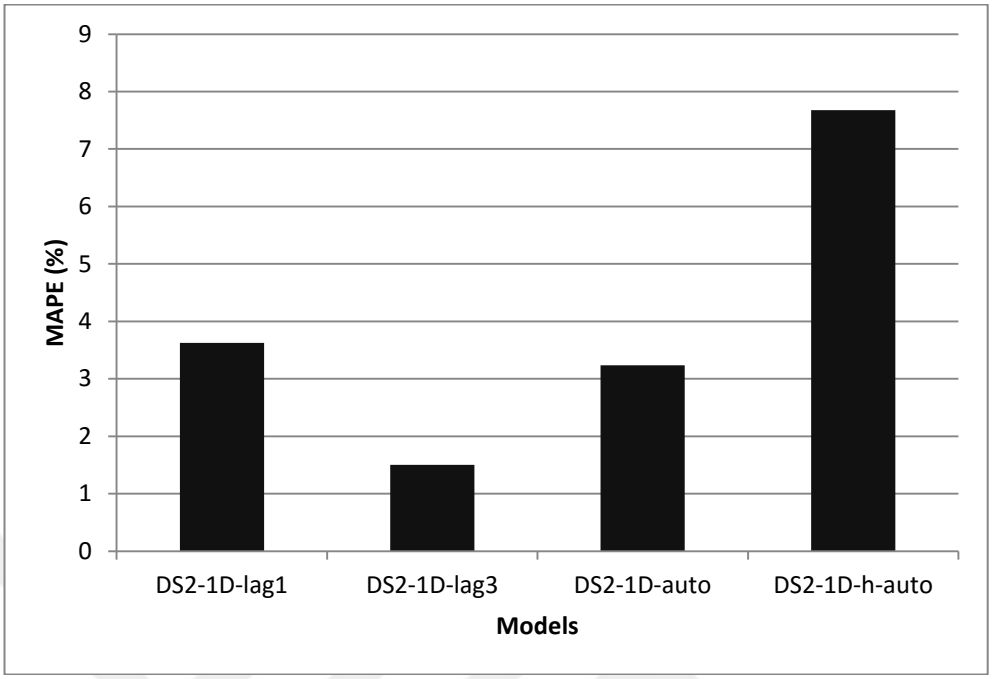

Figure 4.64. Percentage decrease rates in average *MAPE* of the forecasting model of DS2-1D-lag2 compared to ones obtained by the rest of forecasting models in the ANN category (DS2-1D)

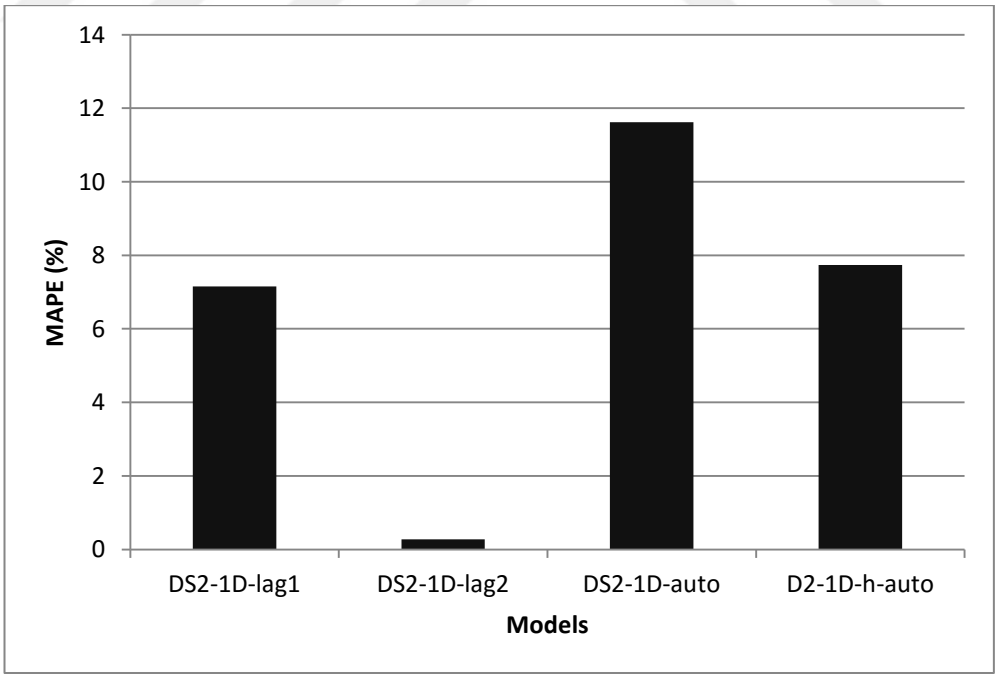

Figure 4.65. Percentage decrease rates in average *MAPE* of the forecasting model of DS2-1D-lag3 compared to ones obtained by the rest of forecasting models in the SVM category (DS2-1D)

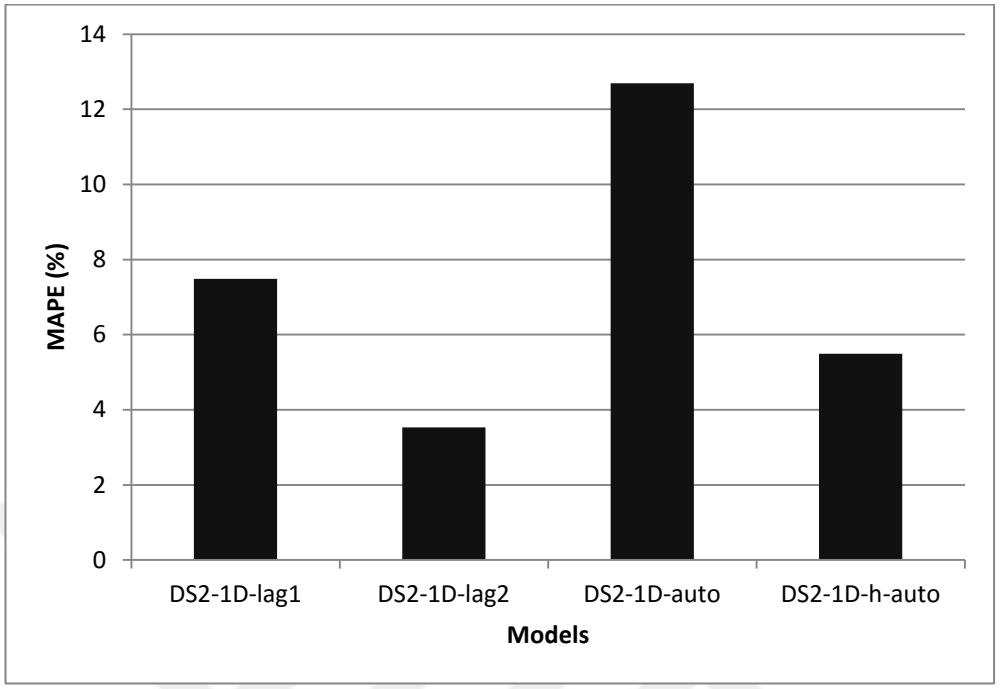

Figure 4.66. Percentage decrease rates in average *MAPE* of the forecasting model of DS2-1D-lag3 compared to ones obtained by the rest of forecasting models in the Decision Trees category (DS2-1D)

4.1. General Discussions on the Results

- For all data sets, in general, SVM and decision trees based prediction models show the highest performance regardless of the selection of time lags. Among the categories, the general ranking of the methods in terms of their prediction performance based on the *MAPE*'s is SVM, decision trees, ANN, and statistical.
- Generally, the results show that prediction models based on the small time scale (5 minute) exhibit higher performance than the models based on the other time scales (hourly and daily) while the prediction models based on larger time scale (hourly) perform better than the largest time scale (daily).
- For all data sets, in general, the time lags having shorter scale yield higher *MAPE*'s. On the other hands, the time lags having longer scales and building by using autocorrelation function outperform.

- When the performance of the prediction models based on *MAPE*'s is examined, the *MAPE*'s of the prediction models obtained on DS2-5M and DS2-1H, which were recorded during the winter and Christmas, are lower than the ones obtained on DS1-5M and DS1-1H, which were saved during the summer while the *MAPE*'s of the prediction models obtained on DS1-1D is much better than the ones obtained on DS2-1D.
- In statistical category, in general, MLR based prediction models perform either comparably or much better than Holt-Winters based prediction models.
- When ANN category is examined, MLP based prediction models outperform in small time scales (5 minute and hourly) while RBF based prediction models perform better than MLP based prediction models in large time scale (daily).
- In SVM category, the performance of the prediction models is comparable for both kernel types.
- When decision trees category is analyzed, the *MAPE*'s of M5P and RF based prediction models are comparable and in general, are lower than the ones obtained by the other methods. Especially, the accuracy of REPTree based models is negotiable since the *MAPE*'s of REPTree based prediction models is equal on the large time scales.

4.2. Discussions on DS1-5M and DS2-5M Results.

- In general, the *MAPE*'s of the prediction models range from 2.83% to 10.71% for DS1-5M and from 1.44% to 9.86% for DS2-5M. Specifically, the prediction models for DS2-5M give 39.95% lower *MAPE*'s on the average than the *MAPE*'s of the prediction model for DS1-5M.
- Among the categories formed, the general ranking of the categories in terms of their prediction performance based on the average *MAPE*'s is SVM, statistical, decision trees, and ANN for DS1-5M; statistical, SVM, decision trees, and ANN for DS2-5M. More specifically, SVM based

prediction models give 3.84%, 54.90%, and 8.41% lower *MAPE*'s on the average than the *MAPE*'s of statistical, ANN, and decision trees for DS1-5M, respectively; statistical based prediction models yield 71.21%, 1.30%, and 8.95% lower *MAPE*'s on the average than the *MAPE*'s of ANN, SVM, and decision trees for DS2-5M, in the order given. It can be said that the performance of statistical and SVM based prediction models are comparable.

- For DS1-5M, MLR based prediction models yield 6.86% lower *MAPE*'s on the average than Holt-Winters based prediction models.
- For DS1-5M, MLP based prediction models give 70.48% lower *MAPE*'s on the average than RBF based prediction models.
- The *MAPE*'s of the SVM prediction models on the average are very close to each other when analyzed according to the kernel type in DS1-5M.
- Even though the *MAPE*'s of M5P and RF based prediction models are comparable, the M5P based prediction models yield 19.10% and 7.78% lower *MAPE*'s on the average than RT and REPTree based prediction models in DS1-5M, respectively.
- When the categories are analyzed according to the methods for DS2-5M, it is observed that the *MAPE*'s of MLR and Holt-Winters based prediction models are comparable.
- MLP based prediction models yield 82.73% lower *MAPE*'s on the average than RBF based prediction models for DS2-5M. That explains the reason of having high *MAPE*'s of the ANN category on the average.
- For DS2-5M, SVM based prediction models with polynomial kernel yield 4.55% lower *MAPE*'s on the average than SVM based prediction models with RBF kernel.
- As is the same case with decision trees category in DS1-5M, in DS2-5M, the *MAPE*'s of the M5P and RF based prediction models is very close to each other while M5P based prediction models give 25.26% and 9.68% lower *MAPE*'s than RT and REPTree based prediction models, in the order given.

- When the lengths of lags are examined, the time lags having longer scale or built by selecting four of the highest autocorrelations yield lower *MAPE*'s for DS1-5M. More specifically, DS1-5M-lag3 gives 1.50%, 0.88%, 2.28%, and 1.25% lower *MAPE*'s on the average than the *MAPE*'s of DS1-5M-lag1, DS1-5M-lag2, DS1-5M-auto, and DS1-5M-h-auto, respectively.
- Among the lengths of time lags, the lags having longer scale or generated by selecting all autocorrelations above a given threshold give lower *MAPE*'s for DS2-5M. Particularly, DS2-5M-lag2 yield 9.50%, 6.76%, 4.29%, and 10.25% lower *MAPE*'s on the average than the *MAPE*'s of DS2-5M-lag1, DS2-5M-lag3, DS2-5M-auto, and DS2-5M-h-auto, respectively.

4.3. Discussions on DS1-1H and DS2-1H Results

- In general, the interval of the *MAPE*'s of the prediction models for DS1-1H is between 4.29% and 34.69% while the interval of the *MAPE*'s of the prediction models for DS2-1H is between 3.77% and 66.83%. Specifically, the prediction model for DS2-1H yield 18.76% lower *MAPE*'s on the average than the *MAPE*'s of the prediction models for DS1-1H.
- When the categories created are examined, the general ranking of the categories in terms of their prediction performance based on the average of *MAPE*'s is SVM, decision trees, ANN, and Statistical for both data sets. Particularly, the decrements rates in *MAPE*'s of the prediction models between the SVM category and the other categories statistical, ANN, and decision trees are on the average 67.45%, 46.74%, and 18.02% for DS1-1H, in the order given while the decrements rates in *MAPE*'s of the prediction models on the average between the SVM category and the other categories statistical, ANN, and decision trees are on the average 62.06%, 53.90%, and 20.04% for DS2-1H, respectively.

- For DS1-1H, Holt-Winters based prediction models give 35.55% lower *MAPE*'s on the average than the *MAPE*'s of MLR based prediction models.
- MLP based prediction models yield on the average 62.76% lower *MAPE*'s than RBF based prediction models for DS1-1H.
- For DS1-1H, SVM based prediction models with RBF kernel yield 9.61% lower *MAPE*'s on the average than the *MAPE*'s of SVM based prediction models with polynomial kernel.
- M5P based prediction models give 3.81%, 36.04%, and 24.99% lower *MAPE*'s on the average than the ones obtained by RF, RT, and REPTree for DS1-1H, in the order given.
- For DS2-1H, even though Holt-Winters based prediction models give 62.22% lower *MAPE*'s on the average than the ones obtained by MLR, when the *MAPE*'s of DS2-1H-lag1 is omitted since it has incredible high *MAPE*'s value, the MLR based prediction models yield 23.42% lower *MAPE*'s on the average than the ones obtained by Holt-Winters.
- For DS2-1H, the decrement rate in *MAPE*'s of the prediction models between MLP and RBF is 70.92% on the average.
- The *MAPE*'s of SVM based prediction models with polynomial kernel is 2.75% lower on the average than the *MAPE*'s of the SVM prediction models with RBF kernel in DS2-1H.
- For DS2-1H, RF based prediction models yield 9.76%, 38.59% and 22.73% lower *MAPE*'s on the average than the ones obtained by M5P, RT, and REPTree, respectively.
- Among the lengths of time lags, the time lags generated by selecting all autocorrelations above a given threshold outperform for DS1-1H. Particularly, the prediction model DS1-1H-auto yields 46.70%, 38.73%, 37.72%, and 3.88% lower *MAPE*'s on the average than the *MAPE*'s of DS1-1H-lag1, DS1-1H-lag2, DS1-1H-lag3, and DS1-1H-h-auto, respectively.

- Among the lengths of time lags, the time lags generated by using all autocorrelations above a given threshold or selecting four of the highest autocorrelations perform better than the other prediction models for DS2-1H. More specifically, the *MAPE*'s of the prediction models DS2-1H-auto and DS2-1H-h-auto are comparable while DS2-1H-auto gives 57.91%, 2.26%, and 6.91% lower *MAPE*'s than the ones obtained by DS2-1H-lag1, DS2-1H-lag2, and DS2-1H-lag3, respectively.

4.4. Discussions on DS1-1D and DS2-1D Results

- The smallest and highest *MAPE*'s vary in the range of 5.59% - 37.43% and 13.71% - 41.57% for DS1-1D and DS2-1D, respectively. Particularly, the prediction models on DS1-1D yield 45.76% lower *MAPE*'s on the average than the *MAPE*'s of the prediction models on DS2-1D.
- When the categories formed are analyzed, the general ranking of the categories in terms of their prediction performance based on the average of *MAPE*'s is decision trees, ANN, SVM, and statistical for DS1-1D; SVM, decision trees, ANN, and statistical for DS2-1D. More specifically, decision trees based prediction models yield 53.23%, 4.58%, 11.57% lower *MAPE*'s on the average than the MAPE's of statistical, ANN, and SVM for DS1-1D, respectively; the *MAPE*'s of SVM and decision trees based prediction models are comparable while SVM based prediction models give 40.01% and 3.20% lower *MAPE*'s on the average than the ones obtained by statistical and ANN for DS2-1D, in the order given.
- For DS1-1D, the *MAPE*'s of MLR based prediction models are 46.65% lower on the average than the *MAPE*'s of Holt-Winters based prediction models.
- In contrast to 5-minute and hourly data sets, RBF based prediction models perform better than MLP based prediction models. The *MAPE*'s

of RBF based prediction models are 5.33% lower on the average than the ones obtained by MLP for DS1-1D.

- SVM based prediction models with RBF kernel give 3.25% lower *MAPE*'s on the average than SVM based prediction models with polynomial kernel for DS1-1D.
- When the results of the models in decision tree category are considered, it can be seen that REPTree based prediction models outperforms the others. REPTree based prediction models yield 18.95%, 17.42%, and 10.79% lower *MAPE*'s on the average than the ones obtained by M5P, RF, and RT for DS1-1D, respectively.
- For DS2-1D, Holt-Winters based prediction models outperform by giving 63.52% lower *MAPE*'s on the average than the ones obtained by MLR.
- Since RBF based prediction models yield 1.66% lower *MAPE*'s on the average than MLP based prediction models, it can be said that the *MAPE*'s of RBF and MLP based prediction models are comparable for DS2-1D.
- The *MAPE*'s of SVM based prediction models with polynomial kernel is 6.87% lower on the average than the *MAPE*'s of SVM based prediction models with RBF kernel for DS2-1D.
- RT based prediction models give 58.57%, 14.06%, and 11.53% lower *MAPE*'s on the average than the ones obtained by M5P, RF, and REPTree for DS2-1D, respectively.
- For DS1-1D and DS2-1D, it is clearly seen that REPTree based prediction models give equal *MAPE*'s with an exception. Even though REPTree based prediction model perform better than the other methods for both data sets, except RT in DS2-1D, the comparison cannot reflect the real. Thus, it is concluded that REPTree is not an effective method on the large time scale (daily).
- When the lengths of lags are analyzed, the time lags generated by using all autocorrelations above a given threshold or selecting four of the highest autocorrelations yield lower *MAPE*'s than the other prediction

models for DS1-1D. Particularly, the prediction model DS1-1D-auto gives 48.37%, 14.35%, and 10.58% lower *MAPE*'s on the average than *MAPE*'s of DS1-1D-lag1, DS1-1D-lag2, and DS1-1D-lag3, respectively while the *MAPE*'s of DS1-1D-auto and DS1-1D-h-auto are very close to each other on the average.

- The prediction models having longer scales perform better than the other prediction models for DS2-1D. More specifically, the *MAPE*'s of the prediction model DS2-1D-lag3 are 5.18%, 9.72%, and 6.16% lower on the average than the other prediction models DS2-1D-lag1, DS2-1D-auto, and DS2-1D-h-auto, respectively while the *MAPE*'s of the prediction models DS2-1D-lag2 and DS2-1D-lag3 are comparable.

5. CONCLUSION

In this thesis, several Internet traffic forecasting models have been developed using various machine learning methods including SVM, MLP, RBF, M5P, RF, RT, and REPTree and statistical regression methods which are MLR and Holt-Winters. Experiments have been conducted on six different data sets which have been formed by different time scales. Several time lags have been utilized for each data set to develop Internet traffic forecasting models. The first 2/3 of each data set has been used as a training set and the last 1/3 of each data set has been used as a test set for model testing. The performance of models has been evaluated by calculating *MAPE*'s.

Considering the results obtained, various conclusion can be deduced. First of all, among the categories formed, the prediction models in SVM and decision trees categories show better performance than the ones obtained in other categories. The order of the categories for Internet traffic forecasting in terms of their prediction performance based on the *MAPE*'s, from the best to the worst, is SVM, decision trees, ANN, and statistical. Secondly, the forecasting models on the small time scale (that is, 5 minute) indicate much better performance than the forecasting models based on the other time scales (that is, hourly and daily). Thirdly, when the lengths of the time lags are compared, the time lags having longer scales or generated by using autocorrelations yield lower *MAPE*'s on the average while the time lags having shorter scales yield higher *MAPE*'s on the average for forecasting.

MLR based forecasting models perform better than Holt-Winters based prediction models in statistical category. MLP based forecasting models yield lower *MAPE*'s on the average than the ones obtained by RBF for small time scales while MLP based prediction models yield higher *MAPE*'s on the average than RBF based forecasting models for larger time scales (daily) in ANN category. In general, M5P based prediction models give lower *MAPE*'s on the average than RF, RT, and REPTree based forecasting models in decision trees category. Since REPTree based forecasting model produce same *MAPE* value in large time scale (daily), it can be said that REPTree is not useful method to forecast Internet traffic.

Future work can be performed in a number of different areas. Different machine learning methods with different time lags can be applied to forecast the amount of traffic in TCP/IP based network since it can help management operations performed by Internet Service Providers. Additional research on new data sets can be carried out to forecast Internet traffic.

REFERENCES

AKGOL, D., AKAY, M. F., and YUR, Y., 2015. Performance Comparison of Machine Learning Methods for Network Traffic Forecasting. Third International Symposium on Engineering, Artificial Intelligence & Applications (ISEAIA2015), North Cyprus, 12 – 13.

AKGOL, D., and AKAY, M. F., 2016. Performance Comparison of Machine Learning Methods and Different Time Lags for Network Traffic Forecasting, International Conference on Natural Science and Engineering (ICNASE'16), Kilis, Turkey, 177 – 183.

ALARCON-AQUINO, V., and BARRIA, J. A., 2006. Multiresolution FIR neural-network based learning algorithm applied to network traffic prediction. IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews, 36(2), 208 - 220.

ALI, J., KHAN, R., AHMAD, N., and MAQSOOD, I., 2012. Random Forests and Decision Trees. International Journal of Computer Science Issues (IJCSI), 9(5), 272 – 278.

BAI, Y., MA, K., and MA., G., 2009. An Analysis of the Combined Wavelet-GM(1,1) Model for Network Traffic Forecasting. 2009 IEEE International Conference on Network Infrastructure and Digital Content (IC-NIDC 2009), Beijing, 155 - 158.

BENZER, R., and BENZER, S., 2015. Application of artificial neural network into the freshwater fish caught in Turkey. International Journal of Fisheries and Aquatic Studies (IJFAS), 2(5), 341 – 346.

BERMELON P., and ROSSI, D., 2009. Support vector regression for link load prediction. Computer networks, 53(2), 191 – 201.

BOSER, B. E., 1992. A Training Algorithm for Optimal Margin Classifiers. In Proceedings of the Fifth Annual Workshop on Computational Learning Theory, Pittsburgh, 144-152.

BREIMAN, L., 2001. Random Forests. Machine Learning, 45(1), 5 – 32.

CASTRO-NETO, M., JEONG, Y., JEONG, M., and HAN, L. D., 2009. Online-SVR

for short-term traffic flow prediction under typical and atypical traffic conditions. Experts Systems With Applications, 36(3), 6164 – 6173.

CHABAA, S., ZEROUAL, A., and ANTARI, J., 2009. ANFIS method for forecasting Internet traffic time series. 2009 Microwave Symposium (MMS), Tangiers, 1 – 4.

CHABAA, S., ZEROUAL, A., and ANTARI, J., 2010. Identification and Prediction of Internet Traffic Using Neural Networks. Journal of Intelligent Learning Systems and Applications, 2(3), 147 – 155.

CHANG, B. R., and TSAI, H. F., 2009. Improving network traffic analysis by foreseeing data-packet-flow with hybrid fuzzy-based model prediction. Expert Systems with Applications, 36(3), 6960 – 6965.

CHEN, X. T., ZHANG, S. Y., and TIAN, T. T., 2010. Internet Traffic Forecasting Based on BP Neural Network [J]. Journal of Nanjing University of Posts and Telecommunications (Natural Science), 2, 004.

CHEN, Y., YANG, B., and MENG, Q., 2012. Small-time scale network traffic prediction based on flexible neural tree. Applied Soft Computing, 12(1), 274 - 279.

CORTEZ, P., ROCHA, M., and NEVES, J., 2005. Time series forecasting by evolutionary neural networks, Chapter III: Artificial Neural Networks in Real-Life Applications, Hersey, PA, USA: Idea Group Publishing, pp. 47 – 70.

CORTEZ, P., RIO, M., ROCHA, M., and SOUSA, P., 2006, Internet Traffic Forecasting using Neural Networks. The 2006 IEEE International Joint Conference on Neural Networks (IJCNN), Vancouver, BC, 2635 – 2642.

CORTEZ, P., RIO, M., ROCHA, M., and SOUSA, P., 2007. Topology Aware Internet Traffic Forecasting Using Neural Networks. Artificial Neural Networks, Porto, Portugal: Springer Berlin Heidelberg, 445 - 454.

CORTEZ, P., RIO, M., ROCHA, M., and SOUSA P., 2012. Multi-scale Internet traffic forecasting using neural networks and time series methods. Expert Systems, 29(2), 143 – 155.

DRAPER, N. R., and SMITH, H., 2014. Applied regression analysis. John Wiley &

Sons.

D'URSO, P., and MAHARAJ., E., A., 2009. Autocorrelation-based fuzzy clustering of time series. Fuzzy Sets and Systems, 160(24), 3565 – 3589.

GHOSH-DASTİDAR, S., ADELI, H., and DADHMER, N., 2008. Principal Component Analysis Enhanced Cosine Radial Basis Function Neural Network for Robust Epilepsy and Seizure Detection. IEEE Transactions on Biomedical Engineering, 55(2), 512 – 518.

GOMM, J. B., and YU, D. L., 2000. Selecting Radial Basis Function Network Centers with Recursive Orthogonal Least Squares Training. IEEE Transactions on Neural Networks, 2, 306 – 314.

HARMS, S., TADESSE, T., and WARDLOW, B., 2009. Algorithm and Feature Selection for VegOut: AVegetation Condition Prediction Tool. Discovery Science: 12th International Conference, Porto, Portugal, 107 – 120.

HASEGAWA, M., GANG, W., and MIZUNI, M., 2001. Application of nonlinear prediction methods to the Internet traffic. The 2001 IEE International Symposium on Curciuts and Systems (ISCAS 2001), Sydney, NSW, 169 - 172.

HONG, W. C., 2011. Traffic flow forecasting by seasonal SVR with chaotic simulated annealing algorithm. Neurocomputing, 74(12), 2096 – 2107.

HONG, W. C., DONG, Y., ZHENG, F., and LAI, C. Y., 2011. Forecasting urban traffic flow by SVR with continuous ACO. Applied Mathematical Modelling, 35(3), 1282 – 1291.

HUANG, S., and SADEK, A. W., 2009. A novel forecasting approach inspired by human memory: the example of short term traffic volume forecasting. Transportation Research Part C: Emerging Technologies, 17(5). 510 – 525.

HYNDMAN, R. J., and KOSTENKO, A. V., 2007. Minimum Sample Size Requirements for Seasonal Forecasting Models. Foresight, 6(Spring), 12 – 15.

IMRIE, C. E., DURUCAN, S., 2000. A River flow prediction using artificial neural networks: generalization beyond the calibration range. Journal of Hydrology, 233(1), 138-153.

JI, L., and WANG, B., 2007. Parameters selection for SVR based on the SCEM-UA algorithm and its application on monthly runoff prediction. Proceedings of the 2007 International Conference on Computational Intelligence and Security, Harbin, 48-51.

JIANG M., WU, C. M., ZHANG, M., and HU, D. M., 2009. Research on the Comparison of Time Series Models for Network Traffic Prediction. Acta Electronica Sinica, 37(11), 2353 – 2358.

JIANG, P., WU, H., WANG, W., MA, W., SUN, X., and LU, Z., 2007. MiPred: classification of real and pseudo microRNA precursors using random forest prediction model with combined features. Nucleic Acids Research, 35, 339 – 344.

KAMİŃSKA-CHUCHMALA, A., 2014. Spatial Internet traffic load forecasting with using estimation method. Procedia Computer Science, 35, 290 – 298.

KARAYIANNIS, N. B., and RANDOLPH–GIPS, M. M., 2003. On the Construction and Training of Reformulated Radial Basis Function Neural Networks. IEEE Transactions on Neural Networks, 14(4), 835-844.

KATRIS, C., and DASKALAKI, S., 2015. Comparing forecasting approaches for Internet traffic. Expert Systems with Applications, 42(21), 8172 – 8183.

KIM, S., 2011. Forecasting Internet traffic by using seasonal GARCH models. Journal of Communications and Networks, 13(6), 621 – 624.

KOEHLER, A. B., SNYDER, R. D., and ORD, J. K., 2001. Forecasting models and prediction intervals for the multiplicative Holt-Winters method. International Journal of Forecasting 17(2), 269 – 286.

KUFFNER, J. J., and LAVALLE, S. M., 2000. RRT-Connect: An Efficient Approach to Single-Query Path Planning. IEEE Transactions on International Conference on Robotics and Automation, 2, 995 – 1001.

LIAW, A., and WIENER, M., 2002. Classification and Regression by Random Forest. R News, 2(3), 18 – 22.

LIU, X., FANG, X., QIN, Z., YE, C., and XIE, M., 2011. A Short-Term Forecasting Algorithm for Network Traffic Based on Chaos Theory and SVM. Journal of Network and Systems Management, 19(4), 427 – 447.

MAURYA, C. K., and MINZ, S., 2012. Fuzzy inference system for Internet traffic load forecasting. 2012 National Conference on Computing and Communication Systems (NCCCS), Durgapur, 1 – 4.

MIGUEL, M. L., PENNA, M. C., NIEVOLA, J. C., and PELLENZ, M. E., 2012. New models for long-term Internet traffic forecasting using artificial neural networks and flow based information. 2012 IEEE Network Operations and Management Symposium (NOMS), Maui, HI, 1082 – 1088.

OLIVEIRA, T. P., BARBAR, J. S., and SOARES, A. S., 2014. Multilayer Perceptron and Stacked Autoencoder for Internet Traffic Prediction. Network and Parallel Computing, Springer Berlin Heidelberg, 61 – 71.

PAPAGIANNAKI, K., TAFT, N., ZHANG, Z., and DIOT, C., 2005. Long-Term Forecasting of Internet Backbone Traffic: Observations and Initial Models. Neural Networks, 16(5), 1110 – 1124.

PARK, J., and SANDBERG, I. W., 1991. Universal approximation using radial basis function networks. Neural Computation, 3(2), 246–257.

QUINLAN, J., 1987. Simplifiying Decision Trees. International journal of man-machine studies, 27(3), 221 – 234.

QUINLANN, J., R., 1992. Learning with continuous classes. 5th Australian joint conference on artificial intelligence, Singapore, 343 – 348.

RATROUT, N. T., and GAZDER, U., 2014. Factors Affecting Performance of Parametric and Non-Parametric Models for Daily Traffic Forecasting. Procedia Computer Science, 32, 285 – 292.

RUTKA, G., and LAUKS, G., 2015. Study on Internet Traffic Prediction Models. Elektronika ir Elekrotechnika, 78(6), 47 - 50.

SANG, A., and LI, S., 2002. A Predictability Analysis of Network Traffic. Computer Networks, 39(4), 329 - 345.

SYED, A. R., BURNEY, A. S. M., and SAMI, B., 2010. Forecasting Network Traffic Load Using Wavelet Filters and Seasonal Autoregressive Moving Average Model. International Journal of Computer and Electrical Engineering, 2(6), 979.

SZADECZKY-KARDOSS, E., and KISS, B., 2006. Extension of the Rapidly

Exploring Random Tree Algorithm with Key Configurations for Nonholonomic Motion Planning. IEEE Transaction on 3rd International Conference on Mechatronics, Budapest, 363 – 368.

TAN, I. K., HOONG, P. K., and KEONG, C. Y., 2010. Towards Forecasting low Network Traffic for Software Patch Downloads-An ARMA Model Forecast Using CRONOS. Second International Conference on Computer and Network Technology (ICCNT), Bangkok, 88 – 92.

TONG, H., H., LI, C., R., and HE, J., R., 2004. Boosting feed-forward neural network for Internet traffic. Proceedings of 2004 International Conference on Machine Learning and Cybernetics, 5, 3129 - 3134.

VAPNİK, V., 1999. The Nature of Statistical Learning Theory. 2nd edition, Springer, 1999.

WANG, C., ZHANG, X., YAN, H., and ZHENG, L., 2008. An Internet Traffic Forecasting Model Adopting Radical Based on Function Neural Network Optimized by Genetic Algorithm. First International Workshop on Knowledge Discovery and Data Mining (WKDD), Adelaide, SA, 367 – 370.

WANG, Y., and WITTEN, H., I., 1997. Inducing model trees for continuous classes. 9th European Conference Machine Learning, Prague, Czech Republic.

WRIGHT, G. B., FLYER, N., and FORNBERG, B., 2013. Radial Basis Function-generated Finite Differences: A Mesh-free Method for Computational Geosciences, Handbook of Geomathematics. Springer, Berlin, 1-30.

ZHAN, C., GAN, A., and HADI, M., 2011. Prediction of Lane Clearance Time of Freeway Incidents Using the M5P Tree Algorithm. IEEE Transactions on Intelligent Transportation Systems, 12(4), 1549 – 1557.

ZHANG, Y., and LIU, Y., 2009. Traffic forecasting Using least squares support vector machines. Transportmetrica, 5(3), 193 – 213.

ZHAO, Y., and ZHANG, Y., 2008. Comparison of decision tree methods for finding active objects. Advances in Space Research, 41(12), 1955 – 1959.

ZONTUL, M., AYDIN, F., DOGAN, G., SENER, S., and KAYNAR, O., 2013. Wind Speed Forecasting Using REPTree and Bagging Methods in Kirklareli-Turkey. Journal of Theoretical and Applied Information Tech., 56, 17 – 29.

CURRICULUM VITAE

Derman AKGÖL was born in Antakya-Turkey, in 1986. She completed the elementary education at İskenderun, Hatay. She went to İbn-i Sina Anatolian High School. She graduated from the Department of Mathematics, Çukurova University, Adana, at 2008 and started to MSc program at Department of Mathematics, University Cincinnati, Ohio, USA at 2010 and graduated at 2012. She completed MSc program at the Department of Computer Engineering in 2016.