DAY-AHEAD ELECTRICITY PRICE FORECASTING FOR TÜRKİYE USING AN ENSEMBLE MACHINE LEARNING TECHNIQUE

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

ÇAĞKAN ÖZBUDAK

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
INDUSTRIAL ENGINEERING

NOVEMBER 2024

Approval of the thesis:

DAY-AHEAD ELECTRICITY PRICE FORECASTING FOR TÜRKİYE USING AN ENSEMBLE MACHINE LEARNING TECHNIQUE

submitted by **ÇAĞKAN ÖZBUDAK** in partial fulfillment of the requirements for the degree of **Master of Science in Industrial Engineering Department, Middle East Technical University** by,

Prof. Dr. Naci Emre ALTUN
Dean, Graduate School of **Natural and Applied Sciences** ________________

Prof. Dr. Pelin Bayındır
Head of Department, **Industrial Engineering** ________________

Prof. Dr. Zeynep Müge Avşar
Supervisor, **Industrial Engineering, METU** ________________

Assoc. Prof. Dr. Bora Kat
Co-supervisor, **TÜBİTAK** ________________

Examining Committee Members:

Prof. Dr. Serhan Duran
Industrial Engineering, METU ________________

Prof. Dr. Zeynep Müge Avşar
Industrial Engineering, METU ________________

Assoc. Prof. Dr. Mustafa Gökçe Baydoğan
Industrial Engineering, Boğaziçi University ________________

Assist. Prof. Dr. Nader Ghaffarinasab
Industrial Engineering, METU ________________

Assoc. Prof. Dr. Gülşah Karakaya
Business Administration, METU ________________

Date:20.11.2024

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Surname: Çağkan Özbudak

Signature :

ABSTRACT

DAY-AHEAD ELECTRICITY PRICE FORECASTING FOR TÜRKİYE USING AN ENSEMBLE MACHINE LEARNING TECHNIQUE

Özbudak, Çağkan
M.S., Department of Industrial Engineering
Supervisor: Prof. Dr. Zeynep Müge Avşar
Co-Supervisor: Assoc. Prof. Dr. Bora Kat

November 2024, 86 pages

In a liberal electricity market where there is competition, accurate hourly electricity price forecasting is important. Electricity producers and consumers require methods for precise price predictions. Producers and consumers may organize their bidding strategies to maximize their benefits by using price projections, which provide important information. Due to the under-maturation and low proliferation of grid-scale storage technologies, the increasing uncertainty with the high penetration of intermittent technologies such as solar and wind makes forecasting more challenging and critical than ever before. Therefore, changes in supply or demand occur with an impact on pricing. Moreover, economic instability mainly originated from national monetary policies together with the political conjoncture in the neighbouring countries, which are also energy suppliers, in the recent decade decrease the predictability of the prices.

In this thesis, XGBoost, SVR and an ensemble of these two algorithms are used for precise for precise and reliable day-ahead electricity price forecasting in the electricity market in Türkiye. The proposed algorithms are compared with other benchmark models which are which are SARIMA and Naive Models for precise and reliable day-ahead electricity price forecasting in the electricity market in Türkiye. Different model settings and time periods for the performance metrics are investigated. The results obtained indicate that the proposed method used is promising in terms of performance metrics which shows competing values compared to the benchmark models and other studies in the literature.

Keywords: Day-Ahead Electricity Price, Price Forecasting, Machine Learning, Ensemble Learning, XGBoost, Support Vector Regression (SVR), SARIMA, Naive Models

ÖZ

BİRLEŞTİRİLMİŞ MAKİNE ÖĞRENMESİ TEKNİĞİ İLE TÜRKİYE GÜN ÖNCESİ PİYASASI ELEKTRİK FİYAT TAHMİNİ

Özbudak, Çağkan
Yüksek Lisans, Endüstri Mühendisliği Bölümü
Tez Yöneticisi: Prof. Dr. Zeynep Müge Avşar
Ortak Tez Yöneticisi: Doç. Dr. Bora Kat

Kasım 2024 , 86 sayfa

Rekabetin olduğu liberal bir elektrik piyasasında, doğru saatlik elektrik fiyatı tahmini önem arz eder. Elektrik üreticileri ve tüketicileri fiyat tahminleri için kesinliği yüksek yöntemlere ihtiyaç duyarlar. Üreticiler ve tüketiciler, önemli bilgiler sağlayan fiyat projeksiyonlarını kullanarak faydalarını en üst düzeye çıkarmak için teklif stratejilerini düzenlerler. Şebeke ölçeğindeki depolama teknolojilerinin henüz yeterince olgunlaşmamış ve yaygınlaşmamış olması nedeniyle, güneş ve rüzgar gibi süreklilik arz etmeyen teknolojilerin yüksek oranda kullanımıyla birlikte belirsizlik artmakta ve tahmin yapmak her zamankinden daha zor ve kritik hale gelmektedir. Bu nedenle, arz veya talepteki değişiklikler fiyatlandırma üzerinde etkili olmaktadır. Ayrıca, son on yılda ulusal para politikalarından kaynaklanan ekonomik istikrarsızlık ve enerji tedarikçisi de olan komşu ülkelerdeki siyasi konjonktür, fiyatların öngörülebilirliğini azaltmaktadır.

Bu tezde, Türkiye'de enerji sektöründe kesin ve güvenilir gün öncesi elektrik fiyatı tahmini için Extreme Gradient Boosting (XGBoost) ve Destek Vektör Regresyonu

(SVR) algoritmaları ve bu algoritmaların birleşimi kullanılıp SARIMA ve Naif modellerle kıyaslaması yapılmıştır. Performans metrikleri için farklı model ayarları ve zaman periyotları incelenmiştir. Elde edilen sonuçlar, önerilen yöntemin performans metrikleri açısından umut verici olduğunu ve kıyaslama yapılan modeller ile literatürdeki diğer çalışmalarla karşılaştırıldığında rekabetçi değerlere sahip olduğunu göstermektedir.

Anahtar Kelimeler: Gün Öncesi Electrik Fiyatı, Fiyat Tahmini, Makine Öğrenmesi, Topluluk Öğrenimi, XGBoost, Destek Vektör Regresyonu (SVR), SARIMA, Naif Modeller

To my family

ACKNOWLEDGMENTS

I would like to express my deepest gratitude to my advisors, Prof. Dr. Zeynep Müge Avşar and Assoc. Prof. Dr. Bora Kat, for their invaluable guidance, support, and encouragement throughout the course of this thesis. Their knowledge and experience have greatly influenced my study and helped me complete my study.

I am also grateful to the members of examining committee; Prof. Dr. Serhan Duran, Dr. Nader Ghaffarinasab, Dr. Gülşah Karakaya and especially Assoc. Prof. Dr. Mustafa Gökçe Baydoğan for their time, effort, and constructive feedback, which have significantly enhanced the quality of this thesis.

A heartfelt thanks to my family, Şebnem Özbudak and Fehmi Özbudak for their unwavering love, support, and belief in me. Their encouragement has been my anchor throughout this journey.

I owe a special debt of gratitude to Ezgi Özkürkçü for her endless emotional support, patience, and understanding during the most challenging times. Your presence has been a constant source of strength for me.

Lastly, I would like to thank my friends Yusuf Berker Demirci and Mert Özdemir for always being by my side. Your companionship, encouragement, and unwavering support have made this journey much more enjoyable and fulfilling.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

LIST OF ABBREVIATIONS

ANN	Artificial Neural Network
ARMA	Auto-Regressive Moving Average
ARIMA	Auto-Regressive Integrated Moving Average
ARFIMA	Autoregressive Fractional Integral Moving Average
BNN	Bayesian Neural Network
CNN	Convolutional Neural Network
EPF	Energy Price Forecasting
FARIMA	Fractionally Differenced Autoregressive Integrated Moving Average
GARCH	Generalized AutoRegressive Conditional Heteroskedasticity
GRU	Gated Recurrent Unit
LSTM	Long Short-Term Memory
MARS	Multivariate Adaptive Regression Splines
MLP	Multilayer Perceptron
RNN	Recurrent Neural Network
SVM	Support Vector Machines
TEDSE	Transformer Encoder-Decoder with Self Attention

CHAPTER 1

INTRODUCTION

The Turkish electricity market is a critical component of the overall national energy system where electricity is produced, transmitted, distributed, and exchanged among its participants. In this market, commitment to market-oriented conduct is characterized by the departure of the governmental authority. It was established as a result of extensive changes coordinated by the Energy Market Regulatory Authority (EMRA/EPDK in Turkish). Encompassing multiple divisions, such as the Day-Ahead Market (DAM), Intraday Market and the Balancing Market, the system serves a wide range of users, including traders, consumers, and generators.

This study aims to fulfill the essential demand for accurate and reliable price forecasts in the energy sector by developing an innovative methodology for day-ahead electricity price forecasting in Türkiye. Furthermore, the goal of this research is to provide a basis for future studies that will improve and broaden this technique, ultimately advancing data science and machine learning applications in the region for forecasting on energy related issues.

This thesis study is primarily driven by the novel use of an ensemble machine learning technique that combines Support Vector Regression (SVR) and Extreme Gradient Boosting (XGBoost) to Turkish data for the first time in an academic setting. This innovative method makes use of the complementing advantages of both algorithms to improve regression models' robustness and predictive accuracy. Through the integration of these two approaches, the study seeks to offer a detailed analysis of their combined effectiveness by comparing with the results of benchmark models, SARIMA and Naive Models through the use of an extensive set of performance metrics developed for regression tasks.

The rest of this chapter explores the development and significant developments in the Turkish electricity market's history. This is followed by a summary of the Turkish day-ahead market's operations. Chapter 2 examines the related studies in the literature which focus on forecasting electricity prices with a specific emphasis on the studies for the Turkish electricity market. In Chapter 3, the methodology, i.e., the data collection process, data preprocessing steps and data modeling steps, introducing the algorithms used in the study, namely Extreme Gradient Boosting (XGBoost) and Support Vector Regression (SVR), are presented in detail. Numerical Analysis and Results presents the outputs of the model trials, including results with different feature combinations, and provides a complete analysis of the performance metrics to evaluate the model results. Finally, Chapter 5 summarizes the findings and suggests directions for future work. Supplementary material, such as detailed tables and plots related to the model results, is included in the appendix to provide additional context and to support the main text.

1.1 History of the Turkish Electricity Market

Turkish Electricity Authority Act No. 1312 in 1970 marked the country's initial institutionalization of the electricity market. Increased energy efficiency was the result of the First Five Year Development Plan. Transmission and distribution of electricity produced under the monopoly are the responsibility of Turkish Electricity Authority, a state economic entity established by law.

Due to rising global costs resulted by the Oil Crisis in 1973, privatization began to gain importance in the publicly owned electricity sectors all around the world. Türkiye experienced it likewise. Privatization was considered a tool to overcome inefficiencies found in the public sector. Law No. 2705 in 1982, together with the monopoly held by the Turkish Electricity Authority in the production, transmission, and distribution of electricity, offers the means for the private sector to enter the market. Trade, transmission, and distribution of electricity for both local and foreign private enterprises were liberalized with Law No. 3096 in 1984. In 1994, Turkish Electricity Authority established Turkish Electricity Generation-Transmission Corporation (TEAŞ) and Turkish Electricity Distribution Company (TEDAŞ) as two distinct economic state companies in accordance with employment and privatization plans.

By 2001, the electricity market underwent a shift towards a free and competitive market. TEAŞ transmission was split into three distinct economic state enterprises which are Turkish Electricity Transmission Company (TEİAŞ), Electricity Generation Inc. (EÜAŞ) and Turkish Electricity Trading and Contracting Co. (TETAŞ).

With the enactment of the Electricity Market Law No. 4628 in 2001, the framework for a competitive market was established; both domestic and foreign investors are leading the charge for implementing regulations regarding the production, transmission, distribution, and providing with electricity energy to guarantee the efficient operation of market operations. Enacted by legislation, TEAŞ is the sole governing body for transmitting energy. It has the term licenses necessary for both domestic and foreign enterprises seeking to engage in the market. Furthermore, since its establishment under the Electricity Market Law No. 4628, Energy Market Regulatory Authority (EMRA) has been in charge of license distribution, market monitoring, and the assessment and examination of pricing principles.

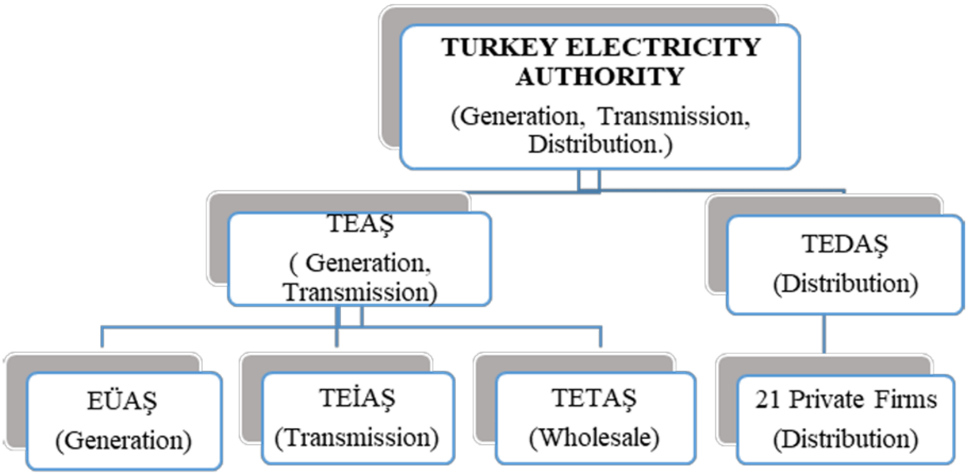


Figure 1.1: Transformation in the Turkish Electricity Market given in [3]

As it can be seen in Figure 1.1, the Turkish Electricity Authority splits into two branches. While TEAŞ is in charge of production and transmission, TEDAŞ is mostly in charge of distribution. TEAŞ splits into three suborganizations, namely EÜAŞ, TEİAŞ, and TETAŞ, where respectively production, transmission, and wholesale are handled independently. Simultaneously, TEDAŞ divides its electricity distribution business into 21 local regions and transfers it to separate distributors for each region.

On July 1, 2006, the electricity market underwent the first phase of change, moving from a single buyer and single seller model to a liberal and competitive model. This involved switching to a monthly three-period financial settlement system. Subsequently, the Day-Ahead Planning system began giving service on December 1, 2009. These transitional phases are crucial for the electricity market's development into a more robust and dynamic structure. The establishment of the DAM system, which is presently in operation, is considered to be the largest step towards the development of the intended electrical market structure. The establishment of the DAM gave the Turkish Electricity Market a new beginning and a competitive framework, enabling the development of a competitive market structure.

Energy Exchange Istanbul (EXIST) was founded on March 18, 2015, and is also known by its Turkish name, Enerji Piyasaları İşletme A.Ş. (EPİAŞ). EXIST is a company that was formally established in accordance with the Turkish Electricity Market Law and is governed by the Energy Markets Operation License that is obtained from EMRA. Gas, electricity, and environmental commodities are among the energy markets that EXIST is in charge of managing and operating. EXIST operates Day-Ahead and Intraday Spot Power Market, Spot Natural Gas Market, Power Futures Market, Natural Gas Futures Market and Renewable Energy Guarantees of Origin System & Organized YEK-G Market as stated in [6].

1.2 An Overview of the Turkish Day-Ahead Market

The DAM in Türkiye is one of the main components of the national electricity market, offering a mechanism for anticipating and allocating electricity usage and generation for the next day. This market allows market participants, including producers and consumers, to make bids and offers based on projected supply and demand circumstances. The idea is to establish a fair and competitive marketplace where prices are determined by the equilibrium of supply and demand offers. This approach enables more efficient resource utilization and better planning for both producers and consumers in order to meet Türkiye's dynamic and rising electricity demand.

The capability of users to adjust their usage in response to price changes is an important part of the DAM's entrance in to the electricity market. As a result, the demand side has begun to engage in the market more actively, with the ability of responding to fluctuating prices. Another innovation is DAM portfolios for market players, which enable them to balance their own portfolios. Attending DAM is not mandatory. The DAM has made a substantial improvement by allowing daily financial settlement and clearing of receivables and payables resulting from business transactions on the day following the transaction date. Market participants can continue to invest without worrying about their financial situation because of the possibility to earn revenue from the sale of generated electricity on a daily rather than a monthly basis. The final advantage to be adopted is the collateral mechanism, which guarantees market players' receivables in the electricity business against potential cash-flow concerns, thereby diminishing their influence on the market.

The general principles of the DAM are presented below.

- Legal companies with licenses are eligible to participate in the DAM by signing the DAM Participation Agreement, which outlines the obligations for the market participants in the DAM.
- Each day starts from 00:00 and ends at 00:00 of following day and consists of hourly time periods.
- Offer submissions can can be submitted from the next day to 5 days later by participants.
- Every hour, prices and volumes that are applied for clearing of the daily DAM are determined.
- Market Operator (EXIST) announces payables to the Market Operator and receivables from the Market Operator for individual market participants through advance payment notifications, which are the outcome of clearing calculations for the market participants based on their day-ahead balancing activities. Market Operator notifies market participants on a daily basis through Central Clearing House.

- Every day until 10:30 am, market participants present letters of guarantee to the market operator. Every day until 11:00 am, market players present their collateral, amount of money paid by the electricity producer to guarantee a successful trade, to the Central Clearing House, aside from letters of guarantee. The amount of the collateral follows the market price of electricity and is returned to the producer after a successful delivery of the electricity
- In order for the market participants to go on with the DAM operations on the weekends and public holidays, letters of guarantee must be submitted by 10:30 am, and collateral other than letters of guarantee must be given by 11:00 am on the preceding work day.

Processes of the DAM consists of the following steps.

- Firstly, DAM participants send the Market Operator their DAM offers for the following day until 12:30 pm.
- Collaterals are checked from 12:30 pm to 1:00 pm to see if DAM offerings qualify.
- Offers for the DAM that are submitted to the Market Operator are checked between 12:30 and 3:00 pm.
- Verified offers are evaluated using an optimization tool between 13:00 and 13:30 pm; market clearing prices and volumes are calculated for each hour of the day.
- Approved sales-bid volumes and commercial transaction approvals are communicated daily at 13:30 pm to the relevant market participants. Market participants may reject those notifications 13:30 and 13:50 pm if they believe there are inaccuracies in any transactions that occurred between.
- At 13:50-14:00, objections are assessed, and the market participants who raised the objection are informed of any pertinent findings. Finalized pricing and matched volumes for the next day's 24 hours are announced at 14:00.
- 0:00 am to 17:00 pm market players send their bilateral agreement notifications, a private trade between two parties, to the Market Operator.

- "Karşılığı Olmayan Piyasa İşlemleri (KOPİ)" in Turkish, Market Transactions without Compensation, is in charge of the bilateral contracts from 17:00 to 17:05 pm every day.

- When KOPİ cancels bilateral contracts, market players have the option to object to their transactions between 17:05 and 17:15 pm every day.

Having given the general principles and processes in DAM, the offer types and bid types are discussed next. Hourly offers, block offers and flexible bids are some of the offer types. Detailed explanations are given below.

Participants can enter the DAM by submitting bids for specific hours or periods, either on an hourly or daily basis, along with flexible proposals. These proposals consist of both quantity and price details, which may vary across different hours. The prices quoted in the offers are sensitive to centesimal changes. Also, participants have the option of expressing their offers in Turkish Lira, US Dollar, or Euro. If one of the currencies other than Turkish Lira is used, then the submitted prices are converted based on the daily Central Bank of the Republic of Türkiye (CBRT) bid rate. The offer quantity is provided in Lots, where 1 Lot equals 0.1 MWh.

Offers can be given in the form of buying or selling proposals. Whether an offer is for buying or selling is determined by the sign accompanying the offer quantity. For example, a quantity of 100 Lots signifies a buying offer, whereas -100 Lots indicates a selling offer. The Market Operator establishes the minimum offer quantities as 0 Lot, while the maximum offer quantities are determined by organizational capacity and KOPİ. According to the DAM's structure and the procedures and principles for evaluating offers, the quantity for flexible bids submitted to the DAM cannot surpass 100 MWh (equivalent to 1000 lots), and the quantity for block bids is capped at 600 MWh (equivalent to 6,000 lots). Offers submitted for the same delivery date are recorded in the system as a new version in case they are updates.

Hourly offers consist of up to 64 levels, divided equally into a maximum of 32 buying levels and 32 selling levels. Prices assigned to each level must follow an ascending order. A particular price level cannot simultaneously feature both buying and selling directions. When constructing the supply-demand curve, the linear interpolation method is applied to estimate values between two successive price/quantity levels.

Block offers provide details concerning price, quantity, and the duration they cover. These offers can span a minimum of 3 hours to a maximum of 24 hours, with block offer hours defined as consecutive and whole hours. Block offers are treated as entirely indivisible entities, and each block offer is subject to either acceptance or rejection for the specified time period. Participants in the market have the option of using the existing block offer structures or give their own. The maximum number of block offers that can be submitted daily is limited to 50. Furthermore, block offers may feature varying quantities for each hour, with the flexibility of an increase or decrease in the quantity up to three times in each consecutive hour.

Flexible bids involve quantities that can be adjusted during a specified order time within an offer period, with a single price designated for each hour. The order time interval spans a minimum of 8 hours and a maximum of 24 hours, while the offer period for flexible orders must not exceed 4 hours. Flexible orders are open for both buying and selling, allowing participants to submit up to a maximum of 6 different flexible orders on a delivery day. It is important to note that within the same offer period, both buying and selling quantities cannot coexist.

As for Bilateral Agreements, it encompasses information for a 24-hour duration, with positive values indicating buying and negative values indicating selling. Values submitted by agreement parties are considered reciprocal, with one value representing buying and its corresponding reciprocal value indicating selling. Bilateral agreements are deemed valid when both parties submit the same absolute values. These agreements can be submitted up to a maximum of 60 days in advance.

CHAPTER 2

RELATED STUDIES IN THE LITERATURE

Price prediction in DAM has gathered widespread attention in academic and industrial research, leading to a diverse range of methodologies which aimed at enhancing the forecasting accuracy. Researchers have explored different kinds of approaches, spanning the traditional time series models and cutting-edge machine learning techniques [7]. The global nature of these investigations reflects the diverse characteristics of worldwide energy markets. Each region faces unique challenges and dynamics. As the field continues to evolve, the integration of explainable Artificial Intelligence (AI), real-time adaptation strategies, and the exploration of new data sources exhibit the dynamic nature of research aiming at refining DAM price prediction methodologies.

In this chapter, studies in the related literature on the DAM prices are reviewed in two subsections. Section 2.1 is devoted to the studies conducted for the energy market around the world and Section 2.2 presents the studies carried out in Türkiye. Besides, while presenting the related studies in the literature, the contribution of the study in this thesis to the literature is presented by comparing it with the existing studies in the literature.

2.1 Related Studies around the World

In order to forecast the DAM prices accurately and help companies optimize their electricity production processes and increase their profit, there are different types of studies conducted all over the world with a variety of methodologies and various data used [7]. Following two subsections present time series and machine learning approaches respectively conducted around the world.

2.1.1 Time Series and Regression Approaches

Models that are using statistical methods mostly rely on linear regression. The dependent variable, i.e. the price, for a the specific day and hour is represented by a linear combination of independent variables, which are also called regressors, inputs or features. Several significant advancements for statistical approaches for Energy Price Forecasting (EPF) have been observed in recent years; the linear regression models with a high number of input features that use regularization techniques have been one of the effective approaches [8].

[9] propose using an Auto-Regressive Moving Average (ARMA) model in conjunction with Generalized AutoRegressive Conditional Heteroskedasticity (GARCH) to estimate price changes on the DAM using extended ARMA models. [10] employs wavelet decomposition in conjunction with multiple regression. Specifically, they compute the regression coefficients by using the wavelet decomposition detail series and the forecasted demand. The DAM prediction is subsequently derived from the low-frequency component of the previous day and the forecasted high-frequency components. [11] examines the day-ahead electricity price of the EPEX Spot for Germany and Austria. They setup a model which is considered as an AR24-X model where X stands for the external regressors, e.g. futures, weekday dummies and periodic B-splines. [12] explores comprehensive seasonal periodic regression models incorporating Auto-Regressive Integrated Moving Average (ARIMA), Autoregressive Fractional Integral Moving Average (ARFIMA), and GARCH disturbances to analyze daily electricity spot prices. The included regressors account for annual cycles, holiday impacts, and potential interventions in both mean and variance. The findings in this study indicate that, specifically for the Nord Pool market (unlike other European markets), an effective modeling of daily spot prices necessitates a long-memory model with periodic coefficients. It is important to note, however, that the performance of these models in [12] for forecasting is not assessed by the authors of the study. [13] employs diverse autoregression methods to model and predict prices in the California market. They note that an Auto-Regressive (AR) model incorporating lags of 24, 48, and 168 hours, with each hour of the day modeled individually, outperforms the unified (S)ARIMA specification suggested by [14] for all hours.

2.1.2 Machine Learning Approaches

Recently, in line with widespread implementation of AI in many areas, there have been an increasing trend in using the AI techniques for forecasting the day-ahead electricity prices. Machine learning has grown in popularity due to its ability to handle complicated and nonlinear relationships within datasets, adapt to shifting patterns, and catch delicate nuances that conventional statistical models may miss. Algorithms like neural networks, Support Vector Machines (SVM), and ensemble approaches have gained popularity for their capacity to uncover patterns in vast datasets, allowing more accurate day-ahead price forecasting. The dynamic nature of electricity markets, combined with the rising availability of varied data sources, has driven the adoption of machine learning techniques, which provide a flexible and robust framework for modeling and predicting the complex dynamics of energy prices. This trend reflects a larger recognition of machine learning's strengths in improving forecasting accuracy and responding to the complexities inherent in the electricity market. These are some of the reasons to use machine learning approaches in this study.

SVMs are commonly used in the EPF applications. For example, in [15] SVM is applied to predict the value of the spot price. [16] proposes a novel machine learning method which uses linear regression, Automatic Relevance Determination (ARD) and Extra Tree Regression (ETR) models. In the study it is seen that more accurate prediction results and overcoming the limitations of individual models can be acquired by combining several models. Experimental results show that proposed method gives lower prediction errors than other individual models. They show that the model can outperform several other models in the literature. [17] proposes Support Vector Regression-Auto-Regressive Integrated Moving Average (SVRARIMA), which is a hybrid model that combines Support Vector Regression (SVR; to capture nonlinear patterns) and ARIMA models. The results show that SVRARIMA model surpasses certain existing Artificial Neural Network (ANN) approaches and conventional ARIMA models. [18] is on the development of a novel hybrid deep learning-based model named convolutional neural network+stacked sparse denoising auto-encoders; the authors suggest a decomposition method to enhance the model performance. In [18] the Australian national electricity market is considered as a case study. Their results show that the proposed model gives successful prediction results in terms of accuracy and stability and shows outstanding prediction performance for price spikes. Moreover, the proposed model can reduce the training time for neural networks during the prediction process due to its quicker convergence speed. [19] introduces three techniques, namely Feed-Forward Neural Network (FFNN), Cascade-Forward Neural Network (CFNN), and Generalized Regression Neural Network (GRNN), to forecast the day-ahead prices in the Spanish OMEL market for the period of January to December in 2002, as well as in the New York electricity market for the period of January to December in 2010. Rather than forecasting the price value, the authors chose to classify the level of the electricity prices since they support the idea that all market participants do not know the exact value of future prices in their decision-making process. [20] applies LSTM deep neural networks combined with feature selection algorithms for EPF. [21] implements a Bayesian Neural Network (BNN) approach to predict the electricity prices in Italy. A probabilistic price forecast methodology is implemented. By this method different results coming from a specific distribution for the same instance is acquired. The methodology gives competing results with the deterministic approaches.

2.2 Related Studies in Türkiye

In Türkiye, DAM price forecasting is crucial for the power market to function effectively and reliably. Accurate forecasting is critical for market players to make better actions in a rapidly changing energy landscape defined by increased renewable energy integration and unpredictable market circumstances. Various forecasting approaches are used to predict electricity prices for the following day. These methods include a range of techniques, such as statistical models, machine learning algorithms, and time series analysis. By using historical data, weather patterns, market trends, and other relevant factors, the precision of the forecasts are endeavoured to be enhanced. Because of the complex interaction of various variables, an entire plan is required to maintain the durability of forecasting models, allowing stakeholders to make informed decisions in the dynamic Turkish energy market.

[22] aim to develop a price forecasting tool based on ANN. A short-term price fore-

casting model is developed by studying the data that most accurately affect the price. The forecasting model relies on supply and demand curves. At the end of the study, it is seen that the proposed ANN model scientifically supports the market participants to make short-term decisions. Another research is presented in [23]. In their study, multiple linear regression method on electricity price forecasting is examined. They analyze various predictors in order to reduce mean absolute percentage error (MAPE). The lagged electricity prices such as the previous one day, one week, and lagged moving average prices are proven to be the effective factors in electricity price estimation. Moreover, they investigate whether there would be a difference if a regular regression method or a dynamic regression method is used. It is seen that there is no dramatic difference regarding the error rates. The research presented in [24] seeks to predict the hourly market clearing price using deep learning techniques, which are Multilayer Perceptron (MLP), Convolutional Neural Network (CNN), Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU). Among these methodologies, LSTM gives the best average forecasting performance in their study. [25] presents customized models to forecast short-term electricity prices. They mainly use the time series models. Results are compared with dynamic regression. [26] also presents the deep learning approaches for performing the prediction of electricity prices. In order to assess the resilience and reliability of the model, twelve Recurrent Neural Network (RNN)-based models are re-estimated using the identical dataset. While all models demonstrate proficiency in price prediction, it is noteworthy that the model which is named as the Transformer Encoder-Decoder with Self Attention (TEDSE) model, that is used for the first time to estimate the electricity prices outperforms its counterparts. [27] mentions how data frequency and different estimation methodologies affect performance of the electricity price forecasting. In this study, different kinds of machine learning and statistical analysis techniques are used parallel with the distinct data periods which belong to COVID-19 pre-pandemic and pandemic. The forecasting frequency is also separated as weekly or daily. At the end of the study, it is shown that the role of the data frequency and method selection can not be ignored in electricity price estimation.

Table 2.1 is given to summarize the aforementioned studies for Türkiye. In these tables, methodologies to forecast the electricity prices, the features used in the models, the metrics that are used to measure the performance of the models etc. can be seen. Some studies in the tables may give multiple results for different settings. Therefore, results shown in the tables for the studies are given such a way that they presents the best or average metric results shown in the related studies.

Table 2.1: Summary of the studies in Türkiye

Study	Methodology	Inputs	Evaluation Metrics	Data Period and Frequency	Metric Results	Key Findings
[25]	• CMARS • RCMARS	• Electricity prices	• MAE • RMSE • Correlation Coefficient • Varience	March 2011 Only day ahead prices are forecasted.	MAE:0.53, RMSE:0.86, Var:0.34, r:0.73 (CMARS)	• One additional and two new approaches proposed. • CMARS and RCMARS performs better comparing dynamic regression.
[23]	• Multiple Linear Regression	• Lagged Electricity Prices • Oil Prices • Natural Gas Prices • Coal Prices	• MAPE • RMSE • Mean Signed Deviation (MSD)	Training set: 03.09.2018 - 02.09.2019, Test set: 02.09.2019 - 14.09.2019, Forecasted for 12 days.	MAPE: 4.80%, RMSE:17.30, MSD:-0.24 (For Dynamic Forecast Error)	• Historical electricity prices reduce error rate. • Reduced MAPE and RMSE when natural gas and oil prices utilized. • Dynamic and regular forecast method giving similar errors.
[22]	• ANN	• The production of wind power • Production by hydroelectric • Natural gas capacity • Coal capacity • Total of consumption	• MAE • RMSE	Train and Test set is taken as whole year of 2017, 24 hours day ahead prices forecasted.	MAE:8.37, RMSE:2.42 for 12.10.2017	• Low error rates. • Similar error rates to other studies with large number of inputs.

Table 2.1: cont'd

Study	Methodology	Inputs	Evaluation Metrics	Data Period and Frequency	Metric Results	Key Findings
[28]	• ANN • CNN • XGBoost • Catboost • Adaboost	• Electricity Prices	• MSE • RMSE • MAE	Dataset period: 01.07.2009 - 01.03.2020, nearly 800 days are predicted.	MSE:0.0045, RMSE:0.0664, MAE:0.0568 (CNN)	• The least CPU time for Adaboost. • Close results for all methods in terms of RMSE, MSE and MAE . • Catboost for the best MSE and MAE. • All methods are successful for bidding and manage electricity usage.
[24]	• MLP • CNN • LSTM • GRU	• Real-time consumption • Load forecast plan • System marginal price (SMP) • Hydroelectric production • Temperature • Wind speed	• MAPE	Training set: 13.06.2016 - 04.10.2020, Test set:01.03.2021-28.03.2021,24 hours day ahead prices forecast.	MAPE:8.15% for LSTM.	• Models' results are close. • Average 8.15% - 9.27% deviations.
[29]	• ARIMA • Holt Winter's • Double Exponential Smoothing	• Electricity prices	• MAPE • MAD • MSD	Different time periods are taken between the years 2007 - 2015, one year ahead forecast.	MAPE:1.05, MAD:0.00174 for ARIMA model; MSD:1.803E-05 for Holt-Winter's Smoothing Model	• For the time period of 2011-2014 Holts-Winter's is the best. • For the time period of 2007-2014 ARIMA is the best. • The longer the time period less the error value.

Table 2.1: cont'd

Study	Methodology	Inputs	Evaluation Metrics	Data Period and Frequency	Metric Results	Key Findings
[30]	• SARIMA • ANN	• Calendar effect • Consumption • Electricity price • Weather • Currency	• MAPE • APE	Data period is between 01.01.2013 - 31.12.2014,weekly forecast is conducted.	MAPE: 0.98%	• ANN fits better than SARIMA to Turkish market. • the more number of features, the less model performance.
[31]	• Wavelet-SVM	• Electricity price	• MAPE	Data for 2014 is used. Weekly forecast is conducted.	MAPE:10.20 and 7.41 for SVM and WT+SVM on the average.	• Wavelet transformation increases pre-processing performance for SVM. • Lower error value for hybrid method.
[26]	• RNN • TEDSE • LSTM • GRU • CNN	• Hourly time • Price of electricity • production • consumption • production-consumption difference	• RMSE • MSE • MAE • r-squared	Time period is between 01.01.2017-31.12.2021.24 hours day ahead forecast.	RMSE: 3.14, MSE: 9.87, MAE: 2.82, r-squared: 0.94	• TEDSE model is successful. • TEDSE model outperforms comparing to other models.

According to the studies in Table 2.1, it can be said that there is still a gap in terms of new methodologies, test period, data frequency and performance metrics for DAM price forecasting although there are numerous studies related with this topic.

In this thesis, SVR and XGBoost algorithms are implemented to predict the electricity prices. There are also some studies which also implement similar algorithms. For example, [31] uses only SVM with wavelet transformation. [28] uses the boosting algorithms and compares their performances. XGBoost is one of the methodologies that is adopted in [28]. However, these studies propose an analysis over daily or weekly results instead of an hourly resolution.

In this thesis, the main difference is the modeling approach. Even tough the similar studies in the literature use only one algorithm for modeling, SVR and XGBoost are both implemented while modeling the data in this thesis. By this way, the overall performance of the model is tried to be improved by ensembling two models whose performance may be worse for each using the weighted average procedure. While these algorithms are being ensembled, an optimization problem is used to determine the weight of each algorithm in the model. In the literature, a similar approach is not encountered to the best of our knowledge for Türkiye. Also, by comparing over an extensive set of performance metrics with the benchmark approaches like SARIMA and Naive methodolgy, the effectiveness of the new methodolgy is shown. In addition, it is seen that most of the existing studies in the literature adopt deep learning approaches. Deep learning models are chosen for their high performances as mentioned before. Although the deep learning approaches have high performances, similar results are thought to be achieved using machine learning approaches with less complex structures compared to deep learning structures with less training times as shown in [32]. In [32], K Nearest-Neighbors (KNN) model produces forecasts that are more accurate than any of the Deep Learning models examined. Another study conducted in [26] shows similar result by using hybrid CNN_LSTM model which performs slightly weaker than some of the ensemble models generated in this thesis. Therefore, machine learning approaches are implemented in this thesis.

Another contribution of this study is regarding the performance evaluation. In the literature, studies regarding the energy price forecasting present performance values based on a time period. In most of the studies, the performance of the proposed algorithms are mostly given for short time periods or in aggregate time intervals such as days or weeks. For example, [33] applies SVM algorithm and obtains successful results over a specific time period such as consecutive two days in a month. However, in this thesis, performance metrics are given for a longer period which spans a whole year. Moreover, the performance metrics are compared for different time periods, such as hours, days and weeks in order to put forward an idea about how the model performs in different time periods. In addition to the aforementioned contributions, the training data in this thesis is from 09.2021 to 06.2023. Also, the electricity sector has changed with the increase in the renewable energy share. Therefore, the up-to-date data of the Türkiye's energy is used in this thesis unlike the case in the other studies listed in Tables 2.1. Besides, an extensive set of performance metrics are taken into account. In addition to $MAPE$, MAE, MSE and $RMSE$; $SMAPE$ and $MASE$ are used for more precise analysis.

CHAPTER 3

METHODOLOGY

The machine learning techniques are capable of revealing the hidden relationships between the features [34]. As it is given in the result of the study [35], machine learning techniques shows superior performance compared to the traditional time series techniques. Therefore in this study, machine learning techniques will be used for forecasting the time series data.

Developing models with machine learning requires common processes. Figure 3.1 shows steps of the common processes described in study [36] for a machine learning problem.

Figure 3.1: Basic Flow of a Machine Learning Process

At the data collection step, the data required for the problem is collected from the necessary data sources. The selection of the data period for the time series analysis is important for the model performance. A period when the data shows regular behavior and does not have too much outlier can be preferred to make the model learn more effectively. In data preprocessing, data is undergone some operations to make it suitable to feed into the model. Scaling, transformation, feature determination, etc. can be given as examples of the data preprocessing procedures. It usually takes most of the time in the model development flow. At the model learning step, the necessary machine learning algorithm and model are determined. The hyperparameter tuning for the related model is also conducted here. As a last step, created model is evaluated according to the related evaluation metrics.

In this chapter, a detailed explanation of the model development stages will be provided for Turkish DAM analysis in this thesis study.

3.1 Data Collection

The raw data is gathered from [6] which is the day-ahead electricity market operator. Price data between 01.09.2021 and 01.06.2024 is collected for this study. One of the reasons for selecting that period is obtaining the recent past data. In this way, the new characteristics, which can be the increased prices due to the inflation etc, of the price values can be observed. Also, the effect of the covid pandemic started to decrease in the last quarters of 2021. Therefore, price fluctuations due to the pandemic are thought to reduce in that period. Besides, the time period should be sufficiently large in order to make the model capture necessary patterns. Thus, the aforementioned approximately three year period is considered to be long enough for developing a powerful model. By this way, a data period which is more up-to-date and has a longer monitoring period, i.e. 1 year test data can be acquired compared to most of the studies in Table 2.1. Using a complete year as the test data allows gaining insights on the performance of the algorithms under possible seasonality impacts.

The price value is a target variable that is going to be predicted. However, there may be other variables that describe the target variable. In the literature, features that may best define the electricity prices are used. According to [18] total electricity generation amount, temperature, wind speed, natural gas prices, electricity demand, the electricity generated by the renewable sources could be candidates for describing the target variable. [37] use renewable energy production amount, gas prices and date features as their inputs. In the studies conducted for Turkish data, similar approaches are observed. Crude oil prices, volatility index, USD/TRY rates, electricity generated from renewable sources, and stock market index are considered as the feature variables in the studies due to [27], [23], [30]. As it can be seen in Table 2.1, the input features are similar to each other in different studies. Besides, day of the week information can be effective in determining electricity prices according to [38].

After investigating the example studies, hourly electricity demand amount, daily average temperature and hourly electricity amount, which is produced by the renewable sources, weekday number information and flag for whether a day is a work day or not are determined to be the initial features that are going to be used in the model since it is seen that these features are commonly used in the studies that are mentioned before. In this thesis, in feature extraction processes, features will be introduced into the data for increasing prediction performance and suitability to be used in the model. These topics are explained in detail in Section 3.2.2 and 3.2.3 respectively. While hourly electricity demand and hourly electricity production data can be gathered from [6], daily average temperature data are gathered through [39]. The same temperature data values are used for each hour of the related day because in this study, hourly analysis are considered.

After predictions are gathered, the results are modified according to the maximum electricity prices defined by EPIAS for the specific time periods. All price thresholds are collected from [6]. Collecting the price thresholds, they transformed to USD amounts according to the corresponding exchange rate at the respective time since the predictions are done in USD amounts in order to decrease the inflation effect in the Turkish economy. The USD exchange rates are collected from [40].

The collected features and the target variable can be observed in Table 3.1 given below.

3.2 Data Preprocessing and Feature Extraction

After collecting the necessary data for modelling, the data should be preprocessed and, if necessary, new features can be added. Descriptive statistics for the collected data are presented in Table 3.2.

In the Table 3.2, minimum, maximum, percentiles for 25%, 50% and 75%, mean

Table 3.1: Features Collected and Descriptions

Feature	**Description**
DATE	Day / month / year Date is not used as an input for the model.
WORKDAY	Flag for whether the day is holiday or not.
WEEKDAY_NO	Weekday number of the day. Takes values between 1 and 7.
HOUR	Hour of the day between 0 am and 11 pm
DEMAND	Hourly electricity demand information in MWh
TAVG	Daily temperature information in celcius
RENEW_PERC	Energy supply percentage from the renewable sources: ratio of the energy acquired from rivers, dams, wind, sun and jeo termal to the total energy production amount
EP	Day-ahead electricity price in USD.

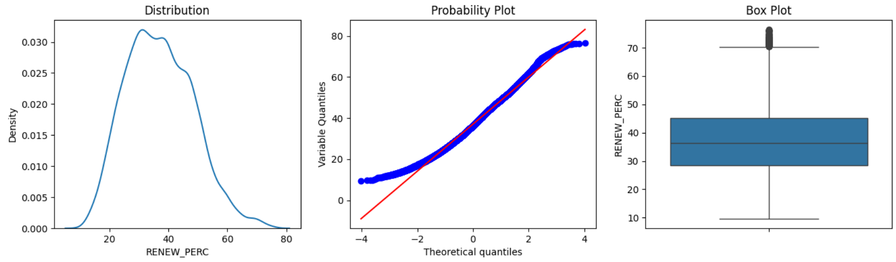
Table 3.2: Descriptive Statistics for the Collected Data

STATISTIC	**DEMAND (MWh)**	**WORKDAY**	**WEEKDAY_NO**	**TAVG (C^o)**	**RENEW_PERC (%)**	**EP ($)**
Count	24120	24120	24120	24000	24120	24120
Mean	26946.83	0.69	4	12.92	37.09	107.04
Std	4141.48	0.46	2	8.12	11.51	55.46
Minimum	13663.90	0	1	-5.50	9.61	0.00
Quantile 25%	23922.90	0	2	6.80	28.43	70.85
Quantile 50%	26922.15	1	4	12.80	36.40	90.26
Quantile 75%	30069.65	1	6	19.13	45.20	134.70
Maximum	39322.20	1	7	33.30	76.45	264.17

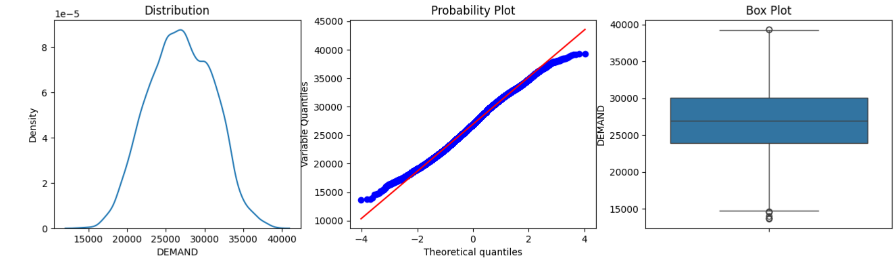
values and standard deviation value, which is given as "Std", can be seen.

Figure 3.2 shows the diagnostic plots of the continuous input features used in the modeling.

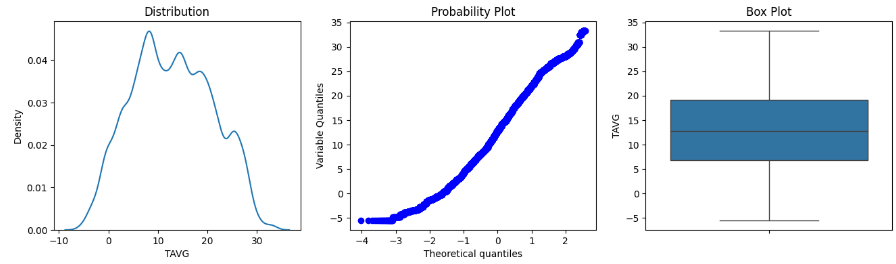
According to the plots, it can be said that all the features distributed normally according to the probability plots and possible outllier analysis is not needed according to the box plots since the number of outliers are negligible.



(a) Diognastic Plots for RENEW_PERC



(b) Diognastic Plots for DEMAND



(c) Diognastic Plots for TAVG

Figure 3.2: Diagnostic Plots for Features

3.2.1 Imputing Missing Data

The data may have some missing values in machine learning problems. Imputing the missing data is the part of the preprocessing. According to Table 3.2 count values, feature TAVG has missing data. Since TAVG has a normal distribution and there is no skewness according to Figure 3.2c, mean value can be used to fill the missing data. Monthly mean temperature values are used to fill the missing values in a respective month.

3.2.2 Feature Extraction

Since the data is on hourly basis, the information about the hour may reveal some patterns for the model. Therefore, hour can be included into the input data in a proper way. Hour data should not directly be included to the input data because it is an ordinal number and it may cause the model to learn the patterns in an inappropriate way. Also, weekday number information can be used in the model. Since it is an ordinal number it may mislead the pattern recognition in the model. Thus, applying the sin and cosine functions may be helpful like in [41]. Features named as "HOUR_SIN", "HOUR_COS", "DAY_SIN" and "DAY_COS" are added to the data as shown below,

$$\hat{y}_i = sin(\frac{2\pi y_{i,hour}}{24}), \quad \hat{y}_i = cos(\frac{2\pi y_{i,hour}}{24})$$
$$\hat{y}_i = sin(\frac{2\pi y_{i,day}}{7}), \quad \hat{y}_i = cos(\frac{2\pi y_{i,day}}{7})$$

where $\hat{y}_i$ is the calculated value for the i^{th} instance. The hour and weekday number values, $y_{i,hour}$, $y_{i,day}$ are transformed into the values that is more proper to make the model understand in modeling phase. The new hour and day values transformed by the trigonometric functions are shown in Tables 3.3 and 3.4 for each hour and day.

Table 3.3: Hour Trigonometric Features

HOUR	HOUR_SIN	HOUR_COS	HOUR	HOUR_SIN	HOUR_COS
0:00	0.000	1.000	12:00	0.000	-1.000
1:00	0.259	0.966	13:00	-0.259	-0.966
2:00	0.500	0.866	14:00	-0.500	-0.866
3:00	0.707	0.707	15:00	-0.707	-0.707
4:00	0.866	0.500	16:00	-0.866	-0.500
5:00	0.966	0.259	17:00	-0.966	-0.259
6:00	1.000	0.000	18:00	-1.000	-0.000
7:00	0.966	-0.259	19:00	-0.966	0.259
8:00	0.866	-0.500	20:00	-0.866	0.500
9:00	0.707	-0.707	21:00	-0.707	0.707
10:00	0.500	-0.866	22:00	-0.500	0.866
11:00	0.259	-0.966	23:00	-0.259	0.966

Table 3.4: Weekday Trigonometric Features

DAY	DAY_SIN	DAY_COS
1	0.782	0.623
2	0.975	-0.223
3	0.434	-0.901
4	-0.434	-0.901
5	-0.975	-0.223
6	-0.782	0.623
7	0.000	1.000

In addition to trigonometric features, lag features are introduced to the model. In the literature, when similar studies are investigated, lag features are usually used by the researchers as in [42], [43]. By introducing lag features, a correlation between the near past and current values of each feature is tried to be identified. Actually, in order to make the time series data to be useful for predicting purposes in supervised learning algorithms, the data should be transformed to a format comprising of the lag values. By doing this, the past values of the features are used to predict the feature value. For deciding the lag value, partial autocorrelation plots of input and target variables are examined.

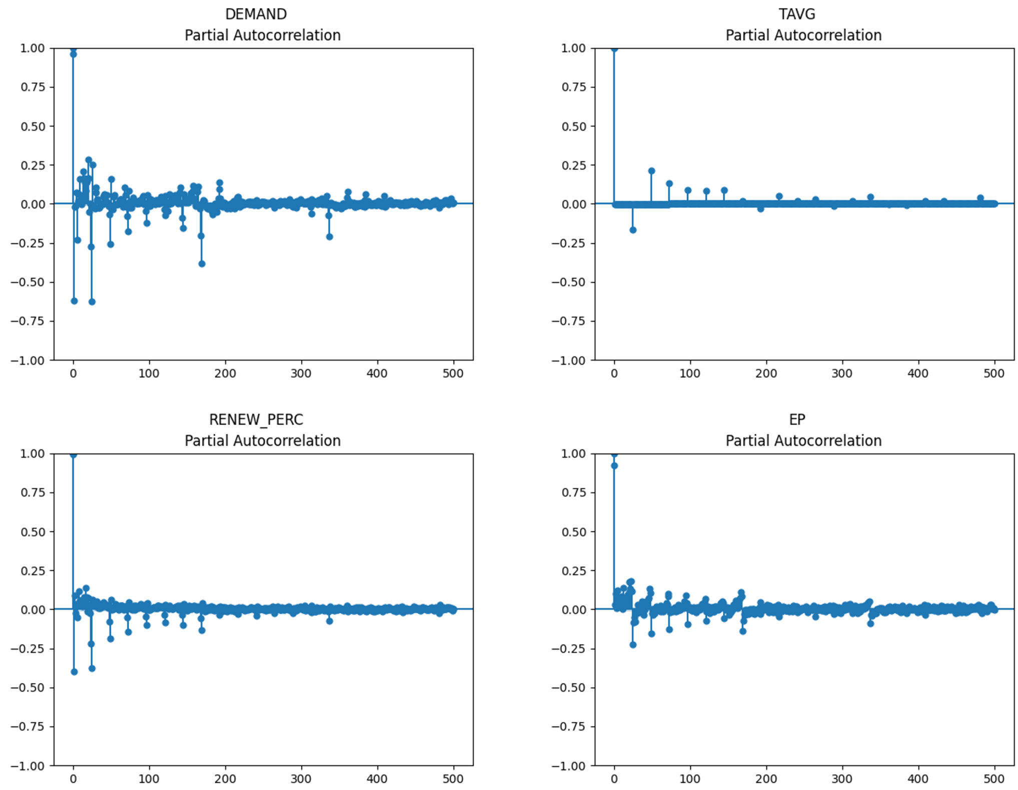


Figure 3.3: Partial Autocorrelation Plots of the Features and Target Variable

In Figure 3.3, partial autocorrelation plots for input features and target variable can be seen. Trigonometric features are not investigated in autocorrelation analysis. According to [42], the lag value can be determined as a value at which the partial autocorrelation value falls below a significant level. Also, in [44], it is stated that if there is a repeated pattern in partial autocorrelation plots for a specific value, that value can be taken as the lag value. According to these, lag value of 24 and 168 are determined to be used in analysis. The lag values for trigonometric features are not introduced in to the model since the lag values will be the same.

3.2.3 Preparing Train and Test Data

The data should be split into train and test sets before modeling. The train data set is used for determining parameters for the model. To train a model, different methodologies can be followed. In this study, time series cross-validation method is used. The test data set is separated from the modeling procedure and it is used for the unbiased evaluation of the model.

Splitting ratios can be different. In [45], ratios used in practice are 80:20, 70:30, 60:40, and even 50:50. It is also stated that there does not appear to be any clear recommendation on what ratio is appropriate or ideal for a certain data set. Since, 1 year period is desired to be investigated in test data set, split ratio of 65:35 is implemented. Thus, approximately one year period is reserved for test dataset. After data is transformed to a data of supervised learning data which is discussed next, split procedure is conducted.

Table 3.5: Example Data Table

t	A	B	C	D	T
1	A_1	B_1	C_1	D_1	T_1
2	A_2	B_2	C_2	D_2	T_2
3	A_3	B_3	C_3	D_3	T_3
4	A_4	B_4	C_4	D_4	T_4
. . .	. . .	. . .	. . .	. . .	. . .
n	A_n	B_n	C_n	D_n	T_n

As it mentioned before, time series data should be transformed to a data of supervised learning problem. The procedure is explained by an example. Let the input data be as shown in Table 3.5, where A, B, C, D are the input features and T is the target that is going to be predicted and t denotes the time.

If the lag value is 24, the past 24 hours of data is used to predict the next 24 hours. After the transformation of the data, the input would seem as shown in Table 3.6.

Table 3.6: Example Data Table after Transformation

A_1	B_1	C_1	D_1	T_1	T_{25}
A_2	B_2	C_2	D_2	T_2	T_{26}
. . .	. . .	. . .	. . .	. . .	. . .
A_n	B_n	C_n	D_n	T_n	T_{n+24}

3.3 Modeling the Data

After data is split into train and test data sets, training takes place. Training the time series data requires a cross - validation procedure. Also, data should be scaled since the features may not be in the same scale which might lead to deviate from an optimally trained model. Therefore, the data scaling is important while training. In the following Sections 3.3.1 and 3.3.2, feature scaling and parameter optimization is explained.

3.3.1 Feature Scaling

In the literature, there are plenty of scaling techniques [46]. Normalization and standardization are the most widely used ones. When the features are close to normal distribution, the standardization is more appropriate for the scaling [47]. Therefore, standardization is conducted in this study. The formulas for normalization and standardization are given below.

$$x_{new} = \frac{x_i - \mu}{max_i(x_i) - min_(x_i)}$$

Here, x_{new} is the new x value after the normalization, x_i is the value of instance i before the scaling, μ is mean of the feature in training data set, $max_i(x_i)$ and $min_i(x_i)$ are maximum and minimum values of the feature, respectively,

$$z = \frac{x_i - \mu}{\sigma}$$

Here, z is the standardized value and σ is the standard deviation of the feature value. It is important to specify that mean of the training data, μ, should be used for scaling the validation and test data sets. Validation data set is used for parameter optimization while conducting the cross-validation. In Section 3.3.2, parameter optimization with the cross-validation techniques is explained.

3.3.2 Optimizing the Model Parameters

Parameter optimization is a fundamental stage of the modeling procedure. Then, the model would be ready to be used in the prediction and evaluating the model metrics. In order to find the optimal parameters, parameter search and cross-validation procedures are conducted together.

In the literature, there are different parameter search and cross-validation techniques for machine learning and time series [48], [49], [50], [51]. Grid search is one of the popular parameter search algorithms used widely in machine learning problems. It searches for all possible hyperparameter sets, and gives the best combination among those sets. For example, in [52], cross-validation and grid search are used to improve performance of study of a multi-class classification problem by determining the optimal parameters. Also, in [53] grid search and cross-validation techniques are used to increase the performance of their machine learning models. Since the grid search with cross validation is used by the studies mentioned, it is also adopted in this thesis. These two procedures form the hyperparameter tuning process for the model.

In classification problems, k-fold cross validation is one of the most widely used method for machine learning problems [50]. k is the number of splits and is defined by the user. Generally, k is defined as 3, 5, 7 which is stated in [54].

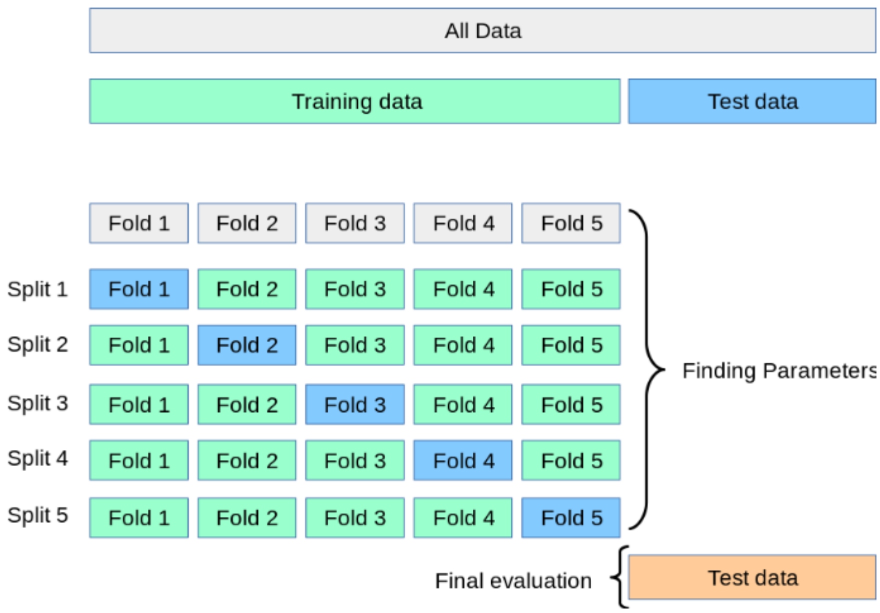


Figure 3.4: Cross-validation Process given in [4]

The cross validation procedure is shown in Figure 3.4. As it is shown, the train data is split into again train and validation data sets. The model is trained using k-1 of the folds as train data. The generated model is verified on the remaining data (i.e., it serves as a test set for calculating a performance metric like accuracy). The performance metric given by k-fold cross-validation is the average of the values calculated for each fold. If k is 5, number of parameters to be optimized is 3 and there are 3 candidate values for each parameter, then there will be $5\text{x}3\text{x}3 = 45$ models generated.

The connection between nearby observations (autocorrelation) characterizes time series data. Classical cross-validation methods, shown in Figure 3.4, assume that samples are independent and identically distributed, resulting in an unjustifiable correlation between training and testing instances (and inaccurate estimates of generalization error) for time series data. As a result, it is critical to evaluate the model for time series data on "future" observations that differ significantly from those used to train the model.

Figure 3.5 shows the time series cross validation procedure conducted in this study. It is a version of k-fold that returns the first fold as the train set and the last fold as the test set. It is worth noting that, unlike typical cross-validation approaches, consecutive

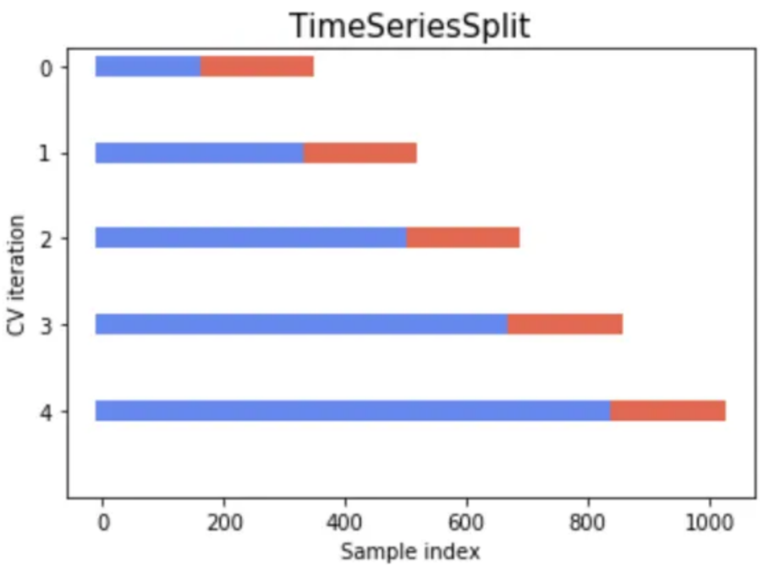


Figure 3.5: Cross validation process for time series data given in [5]

training sets are super sets of the previous ones. Additionally, it adds all surplus data to the initial training partition, which is always used to train the model. In this thesis, the number folds are set to be 3 for the sake of simplicity and considering the model training times.

3.4 Machine Learning Algorithms used in the Study

In this study, two of the machine learning algorithms are used. One of them is SVM. Since the target variable, namely the day-ahead electricity price, is a continuous variable, it can be highlighted that SVM is used for regression, which can also be named as Support Vector Regression (SVR) instead of its general use for classification. One of the reasons behind why SVM is chosen is that it offers a solution to bypass the complexities involved in using linear functions within feature spaces characterized by a high dimensionality as stated in [55]. Therefore, SVMs are more effective in high dimensional spaces.

The other machine learning algorithm, which takes a part of predicting the day-ahead energy prices in this thesis is Extreme Gradient Boosting, which is called generally XGBoost. It is an ensemble learning algorithm. The name ensemble comes from the

Table 3.7: Advantages and Disadvantages of SVMs given in [1]

Advantages of SVMs	**Disadvantages of SVMs**
Effective in high-dimensional spaces	Computationally expensive during training
Handles large feature spaces and datasets	Careful selection of Kernel function and hyperparameter tuning is required
Robust against overfitting	Difficult to interpret results
Complex decision boundaries are captured via different Kernel functions	Sensitive to noisy or mislabeled data
Linearly separable or non-linearly separable data is handled well	Imbalanced datasets may lead to struggle

fact that the algorithm ensembles weak learning algorithms and gives an output. This weak learning algorithm in XGBoost is decision trees.

In Table 3.7, advantages and disadvantages of SVM are given. As it is seen in the table, SVM is effective in high dimensional spaces and it is a flexible algorithm in terms of overfitting. The Kernel function options are used to struggle with the complex decision boundaries. However, SVMs are sensitive to the noisy and mislabeled data. Also, imbalanced datasets and rigorous selection process of Kernel functions and hyperparameters create different challenges to fit a proper model.

Table 3.8 gives advantages and disadvantages of XGBoost algorithm. It can be said that while improvements are included to the original gradient boosting models to increase the performance and accuracy of the results, overfitting and slower training process may be observed in XGBoost models.

Consequently, although both SVM and XGBoost algorithms have some drawbacks like any other algorithms, they are used in this thesis because of their high level benefits mentioned in advantages columns. The explanations for the aforementioned algorithms are provided in Sections 3.4.1 and 3.4.2.

Table 3.8: Advantages and Disadvantages of XGBoost given in [2]

Advantages of XGBoost	**Disadvantages of XGBoost**
Effective in small number of samples with a large number of features	Overfitting may be observed unless hyperparameters are adjusted correctly
Explainability capabilities that can help validate the correctness of the model, i.e., the most significant features can be checked	Applicable for numeric features only
Improvements for increasing the performance and accuracy of the results are included to the original gradient boosting models	Slower to train

3.4.1 Support Vector Machine for Regression (SVR)

The fundamental concept of SVMs was initially introduced in 1960s by Vapnik and his colleagues [56], [57]. Subsequently, significant advancements were made in the following decades. The comprehensive framework of SVM was formally documented in 1992 for classification by [58], and later extended to regression, referred to as the ϵ-SVR model, in [59] and [60].

Conventional statistical regression methods are typically described as processes generating a function that minimizes the difference between predicted and actual responses across all training instances. A distinguishing feature of SVR is its focus on minimizing the generalized error bound rather than solely targeting the training error. This bound encompasses both the training error and a regularization term, which governs the complexity of the hypothesis space, thereby aiming for improved generalization performance [61].

3.4.2 Extreme Gradient Boosting (XGBoost)

Extreme gradient boosting (XGBoost) is a machine learning technique initially developed in [62]. Gradient tree boosting is a machine learning method that is widely utilized in a range of applications. Tree boosting has been shown to achieve superior performance on various traditional classification standards. XGBoost is comparable to other gradient boosting methods, but its effectiveness stems mostly from its ability to scale across all scenarios. The system surpasses earlier approaches on a single computer and is capable of handling billions of samples in distributed or memory-constrained environments. XGBoost's scalability is accomplished by a variety of system and algorithmic improvements. These improvements include an innovative tree learning technique for sparse data and a theoretically justified weighted quantile sketch procedure for handling instance weights in approximation tree learning. Parallel and distributed computers also enhance learning, allowing fast model exploration. The approach is based on the study of [63]. This research now includes minor improvements.

3.5 Ensemble of Two Algorithms

Ensemble of the machine learning algorithms are used in the literature to get increased final model performance rather than less powerful individual models. Several ensembling techniques, including bagging, boosting, dagging, stacking/blending and model averaging are used in the literature and practice [64]. While boosting and bagging techniques concentrate more on lowering bias and variance, respectively, stacking approaches aim to reduce both by determining the best way to mix base learners. Ensembles are formed by stacking the weighted averages of many basic learners together. It is well known that improving each base learner's hyperparameter throughout the ensemble weight optimization procedure might result in higher-performing ensembles [65]. The procedure in [65] which is stated as Generalized Ensemble Model (GEM) algorithm is implemented while producing ensemle of the algorithms.

In this thesis, SVM and XGBoost is combined in order to achieve the output. The weighted average of the outputs coming from both algorithms gives the final result. The weights of the algorithms are determined according to the average performance metric of the cross-validation scores. An optimization problem is solved to minimize the error metric and the final weights are determined. Those weights are used in predicting the test output. Next, the performance metrics used in this thesis are described. Then, the procedure is given in this section.

It is known that there are plenty of performance metrics in order to examine the performance of the regression and classification problems. Since the problem in this study is a regression problem, the metrics for regression problems are used. After the suitable metrics for the study are searched in the literature, four performance metrics are determined to be used in this study as these metrics are commonly used in the literature [66], [66], [28]. The first performance metric is called Mean Absolute Error (MAE). The calculation of the metric is given as follows:

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|z_i - \hat{z}_i| \tag{3.1}$$

where N is the number of instances, z_i is the real value and $\hat{z}_i$ is the predicted value of instance i. The second performance metric is Mean Absolute Percentage Error ($MAPE$). It is computed by the following formulation:

$$MAPE = \frac{100}{N}\sum_{i=1}^{N}\frac{|z_i - \hat{z}_i|}{z_i} \tag{3.2}$$

Mean Squared Error (MSE) is the third performance metric used in this study. Its calculation is given below.

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(z_i - \hat{z}_i)^2 \tag{3.3}$$

The fourth performance metric is Root Mean Squared Error ($RMSE$) which is calculated by taking the squared root of MSE and the calculation is shown below.

$$RMSE = \sqrt{MSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(z_i - \hat{z}_i)^2} \tag{3.4}$$

The other metric is Mean Error (ME). It is used how the model deviates from actual values on the average. It shows whether the model overestimates or underestimates the actual values on the average. The calculation is shown below.

$$ME = \frac{1}{N}\sum_{i=1}^{N}(z_i - \hat{z}_i) \tag{3.5}$$

The sixth performance metric used in the thesis is Symmetric Mean Absolute Percentage Error ($SMAPE$). The $SMAPE$ metric is designed to account for the relative difference between predicted and actual values, considering their magnitudes in a balanced manner. Unlike other error measures, $SMAPE$ assigns equal weight to overestimations and underestimations [67], [68]. Calculation is shown below. Also, since the values closer to zero distract the $MAPE$ value [69], $SMAPE$ can be an alternative.

$$SMAPE = \frac{100}{N}\sum_{i=1}^{N}\frac{|z_i - \hat{z}_i|}{\frac{|z_i|+|\hat{z}_i|}{2}} \tag{3.6}$$

The last performance metric used in this thesis is Mean Absolute Scaled Error ($MASE$). It is the mean absolute error of the forecast values, divided by the mean absolute error of the in-sample seasonal naive forecast. It is used in this study because $MASE$ is independent of the scale of the data, so can be used to compare forecasts across data sets with different scales. Also, $MASE$ penalizes positive and negative forecast errors equally, and penalizes errors in large forecasts and small forecasts equally [70], [71]. The formula of $MASE$ is shown below.

$$MASE = \frac{1}{N}\sum_{i=1}^{N}\frac{\frac{1}{N}\sum_{i=1}^{N}|z_i - \hat{z}_i|}{\frac{1}{T-m}\sum_{i=m+1}^{T}|z_i - \hat{z}_{i-m}|} \tag{3.7}$$

where T is the size of training data, m is seasonal period, which is taken as 24 in this study.

Having given the performance metrics used in the study, how the two algorithm is combined is explained next. Since the aim here is to determine the weight of each algorithm which becomes a decision variable, let w_j be the weight of algorithm j. When j is 1, it denotes SVR algorithm and when it is 2, it denotes XGBoost algorithm. Note that the weight should satisfy the following equation that assures that the resulting measure is a convex combination of individual measures, i.e., sum of weights is equal to unity.

$$\sum_{j=1}^{J} w_j = 1, \qquad w_j \geq 0, \qquad \text{for } \forall j, j = 1, ...J. \tag{3.8}$$

Parameters of the optimization problem are provided below.

$\hat{y}_{ijk}$: Day-ahead electricity price predicted by algorithm j for the k^{th} cross-validation split and for instance i.
y_{ik}: Real day-ahead electricity price value for instance i and k^{th} cross-validation split.
u_{ik}: Upper limit price value obtained from EPIAS after transforming it to USD for instance i in k^{th} cross-validation split.
l_{ik}: Lowest price value, which is 1, after transforming TL price values to USD for instance i in k^{th} cross-validation split.
u_i: Upper limit price value obtained from EPIAS after transforming TL price values to USD for instance i in the test set.
l_i: Lowest price value, which is 1, after transforming TL price values to USD for instance i in the test set.

$\hat{y}^*_{ijk}$ values are obtained after modifying the model outputs according to the real maximum or minimum price levels. Let $\hat{y}_{ijk}$ be the first version of the output from machine learning model. $\hat{y}^*_{ijk}$ value is obtained according to the relation below.

$$\hat{y}^*_{ijk} = \begin{cases} l_{ik}, & \text{if } \hat{y}_{ijk} \leq 0, \\ \hat{y}_{ijk}, & \text{if } 0 < \hat{y}_{ijk} < u_{ik} \\ u_{ik}, & \text{otherwise.} \end{cases} \tag{3.9}$$

The similar procedure is followed for the test dataset after the validation set.

For this study, MAE and $MAPE$ values are chosen to be in the objective function. For each cross-validation split (fold) k, they can be calculated by the expression below,

$$MAE_k = \frac{1}{N}\sum_{i=1}^{N} |y_{ik} - \sum_{j=1}^{J} w_j \hat{y}^*_{ijk}| \text{ for } \forall k \tag{3.10}$$

$$MAPE_k = \frac{100}{N}\sum_{i=1}^{N} \frac{|y_{ik} - \sum_{j=1}^{J} w_j \hat{y}^*_{ijk}|}{y_{ik}} \text{ for } \forall k \tag{3.11}$$

Then, by putting the expressions found above into the optimization problems for Error Metric Minimization, objective functions of problems are formulated below.

Problem EMMmae: Minimize $\frac{1}{K}\sum_{k=1}^{K} MAE_k$

subject to $(3.8), (3.10)$

Problem EMMmape: Minimize $\frac{1}{K}\sum_{k=1}^{K} MAPE_k$

subject to $(3.8), (3.11)$

Problem EMMmae and Problem EMMmape above give the weight values for each algorithm to minimize the average of the error metrics. The weight values are expected to be changed when any property of each algorithm is changed. Using the weight values found, the final predictions for the test data set are found. Denoting the final prediction result by $\hat{y}_i^{test}$,

$$\hat{y}_i^{test} = \begin{cases} l_i, & \text{if } \sum_{j=1}^{J} w_j^* \hat{y}_{ij}^{test} \leq 0, \\ \sum_{j=1}^{J} w_j^* \hat{y}_{ij}^{test}, & \text{if } 0 < \sum_{j=1}^{J} w_j^* \hat{y}_{ij}^{test} < u_i \\ u_i, & \text{otherwise.} \end{cases} \tag{3.12}$$

gives the final prediction result. In this formula, w_j^* shows the weight values found by solving Problem EMM, $\hat{y}_{ij}^{test}$ shows the i^{th} predicted electricity price value in the test dataset by algorithm j.

After finding the final values, $\hat{y_i}^{test}$, for test data set, the performance metrics to present model performances are calculated. Next chapter presents the results of Problem EMM, different model settings and performance metrics for those models. Also, several analysis for the performance metrics are presented.

3.6 Benchmark Models

To compare the results of new methodology, models which can be used as benchmark are investigated. In this study, two modeling approaches are used as benchmark and their performance are compared with the new approach mentioned before.

3.6.1 Naive Model

Although there are many studies focusing on precise forecasting of energy related issues, the decision makers would rather prefer traditional and simple methods they have been using for a long time in practice. One of them in DAM price forecasting using recent data as they were observed. Thus, a naive algorithm that employs specific lag values of 24, 48 and 168 hours is proposed. In this model, these specific lag values of the electricity price are taken as the prediction values. The prediction is found as shown in the calculation below.

$$\hat{y}_t = y_{t-m}$$

where $\hat{y}_t$ is the predicted value for time t using the value which is m times before. In this study, we take m as 24, 48 and 168 hours.

3.6.2 Seasonal Auto-Regressive Integrated Moving Average (SARIMA)

A widespread and simple time series forecasting method SARIMA is used as another benchmark in this study. According to [72], a full ARIMA model can be written as

$$y_t^{'} = c + \phi_1 y_{t-1}^{'} + ... + \phi_p y_{t-p}^{'} + \theta_1 \epsilon_{t-1} + ... + \theta_1 \epsilon_{t-q} + \epsilon_t$$

where $y_t^{'}$ is the differenced series (it may have been differenced more than once). The "predictors" on the right hand side include both lagged values of y_t and lagged errors. It is called an ARIMA(p,d,q) model, where p is order of the autoregressive part, d is the degree of first differencing involved and q is the order of the moving average part. The backshift notation of this model can be written as,

$$(1 - \phi_1 B - ... - \phi_p B^p)(1 - B)^d y_t = c + (1 + \phi_1 B + ... + \phi_q B^q)\epsilon_t$$

where the firstelement in right side of the equation defines $AR(p)$, second part defines d differences and left side of the equation defines moving averages with q, $MA(q)$. A seasonal ARIMA model is formed by including additional seasonal terms,$(P, D, Q)_m$ in ARIMA models where m is the number of observations in a period. It is taken as 24 in this study. The seasonal part of the model consists of terms that are similar to the non-seasonal components of the model, but involve backshifts of the seasonal period. An $ARIMA(1,1,1)(1,1,1)_{24}$ model without a constant is for a lag 24 data and can be written as

$$(1 - \phi_1 B)(1 - \Phi B^{24})(1 - B)(1 - B^{24})y_t = (1 + \phi_1 B)(1 + \Theta B^{24})\epsilon_t$$

The additional seasonal terms are simply multiplied by the non-seasonal terms. The auto.arima() function in R is used to create model for SARIMA time series. Considering the AIC (Akaike Information Criterion), an iterative procedure called Hyndman-Khandakar algorithm, [73], is used to reach minimum AIC value. The parameters searched in the study are given in Table 3.9

Table 3.9: Parameters Searched in SARIMA model

SARIMA Parameters	**Values Searched**
p	0,1,2,3
d	1
q	0,1,2,3
P	0,1,2,3
D	0
Q	0,1,2,3

The modification for upper and lowest price limits are also conducted for the predictions of Naive and SARIMA models.

CHAPTER 4

NUMERICAL ANALYSIS AND RESULTS

This chapter presents the findings from a variety of models that were generated using varying input parameters in order to assess the models' performance on the time series forecasting problem. Numerical values of the performance metrics are given using the effectiveness of each lag setting used in the ensemble methodology, which follows the procedures previously stated in Chapter 3. The accuracy and robustness of the models are evaluated using the metrics including $MAPE$, MAE, $RMSE$, MSE, ME, $MASE$ and $SMAPE$ given in Chapter 3. Also, benchmark model findings are included to the results to compare the effectiveness of ensemble approach.

Several input configurations are used in different models, including independent variables. When compared to previous research in the literature for Turkish DAM, the suggested ensemble model introduced in Section 3.5 performs competitively, demonstrating its dependability and robustness. In addition, several input configurations are examined to see how they affect the model's performance, emphasizing the significance of careful feature selection and parameter adjustment. The best-performing models are thoroughly discussed among the other models tried in this chapter along with their advantages and disadvantages when applied to the particular time series data.

4.1 Model Results

In this section, the numerical results of the methodology given in Chapter 3 are presented. Different combinations of the features are tried and their results are presented. Besides, since the data is time series, the numerical values of the evaluation metrics are analyzed based on the time periods.

4.1.1 Results of All Models

After optimizing parameters separately using grid search, the ensemble model results are achieved using the best parameter settings. The grid search results belonging to the models are given in Appendix. There are 6 main ensemble models, 6 individual models created with XGBoost and SVR, 3 Naive models with lags 24, 48 and 168 and 1 SARIMA model tried in the study. For the simplicity, the aliases are given to the models and the numbers are given to the features. Since the seasonal features, HOUR_SIN, HOUR_COS, DAY_SIN and DAY_COS are always going to be put in feature combination calculations with electricity price target variable, numbers to these features are not given. In the next subsections the given aliases are going to be used to mention the models and features. Model aliases and feature aliases can be seen in Table 4.1 and Table 4.2 respectively.

Table 4.1: Feature Aliases

Feature	**Number**
WORKDAY	1
DEMAND	2
TAVG	3
RENEW_PERC	4

According to the initial results, without any feature combination and using the best parameters for each individual algorithm, the model results are presented in Table 4.3.

In Table 4.3, the results are presented for the test data set. According to the results, Model B_mae and Model B_SVR are the dominant models which are shown as bolt characters in the table. The optimal weight values found in Model B_mae are 0.631 and 0.369 for XGBoost and SVR, respectively. It can be said that when lag 24 and lag 168 values used together, the overall performance increased. A similar behavior can be observed in [74]. Also, it can be observed that the proposed methodology

Table 4.2: Model Aliases

Model Alias	Model Description
Model A_mae	Ensemble model obtained from problem EMMmae in which lag 168 values of the features are used.
Model A_mape	Ensemble model obtained from problem EMMmape in which lag 168 values of the features are used.
Model A_SVR	Individual SVR model in which lag 168 values of the features are used.
Model A_XGB	Individual XGBoost model in which lag 168 values of the features are used.
Model B_mae	Ensemble model obtained from problem EMMmae in which lag 24 and 168 values of the features are used.
Model B_mape	Ensemble model obtained from problem EMMmape in which lag 24 and 168 values of the features are used.
Model B_SVR	Individual SVR model in which lag 24 and 168 values of the features are used.
Model B_XGB	Individual XGBoost model in which lag 24 and 168 values of the features are used.
Model C_mae	Ensemble model obtained from problem EMMmae in which lag 24 values of the features are used.
Model C_mape	Ensemble model obtained from problem EMMmape in which lag 24 values of the features are used.
Model C_SVR	Individual SVR model in which lag 24 values of the features are used.
Model C_XGB	Individual XGBoost model in which lag 24 values of the features are used.
Naive_24	Naive model using lag 24 value of target variable
Naive_48	Naive model using lag 48 value of target variable
Naive_168	Naive model using lag 168 value of target variable
SARIMA	SARIMA model defined in Chapter 3.

performs better than Naive and SARIMA models in terms of MAE, $MASE$, MSE, $RMSE$ and $SMAPE$. Therefore, the rest of the analysis are conducted for Model B_mae and Model B_SVR.

In Figure 4.1a and Figure 4.1c, the predicted and actual values of the whole period of the test set for the two models are given. An example time period of 01/07/2023 - 07/07/2023 can be seen in Figure 4.1b and Figure 4.1d to see a detailed prediction plot of the two models for the same time period. It can be said that both of the models perform similar for the same example time period. To examine how the residuals behave for both models, error analysis are conducted for test data sets. Figure 4.2

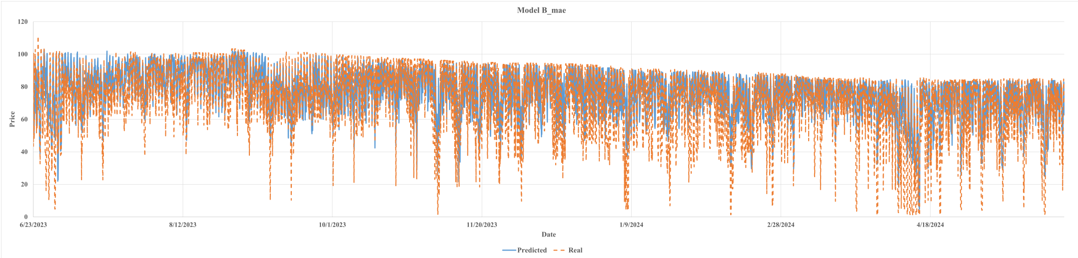
Table 4.3: All Model Results for Test Data Set

Models	MAE	MAPE (%)	MASE	ME	MSE	RMSE	SMAPE (%)
Model A_mae	11.664	55.002	0.525	-5.741	293.553	17.133	20.174
Model A_mape	11.664	55.002	0.525	-5.741	293.552	17.133	20.174
Model A_SVR	11.663	55.000	0.525	-5.740	293.550	17.133	20.174
Model A_XGB	18.004	97.537	0.810	-17.132	648.648	25.469	26.932
Model B_mae	**10.614**	**46.009**	**0.477**	**-5.438**	**223.800**	**14.960**	**18.760**
Model B_mape	12.168	52.121	0.547	-7.665	288.695	16.991	20.579
Model B_SVR	**10.272**	**38.948**	**0.462**	**-1.637**	**209.108**	**14.461**	**18.772**
Model B_XGB	12.168	52.121	0.547	-7.665	288.696	16.991	20.579
Model C_mae	13.880	58.057	0.627	-11.348	388.564	19.712	22.605
Model C_mape	13.879	58.057	0.627	-11.348	388.559	19.712	22.605
Model C_SVR	13.879	58.057	0.627	-11.348	388.559	19.712	22.605
Model C_XGB	13.700	54.669	0.619	-10.642	362.789	19.047	22.354
Naive_24	11.981	40.620	0.541	-0.342	310.167	17.612	22.298
Naive_48	12.567	37.096	0.568	-0.053	353.770	18.809	23.144
Naive_168	14.659	51.748	0.662	-0.094	458.857	21.421	26.161
SARIMA	14.053	45.117	0.635	3.602	365.226	19.111	25.047

shows the residuals for both models.

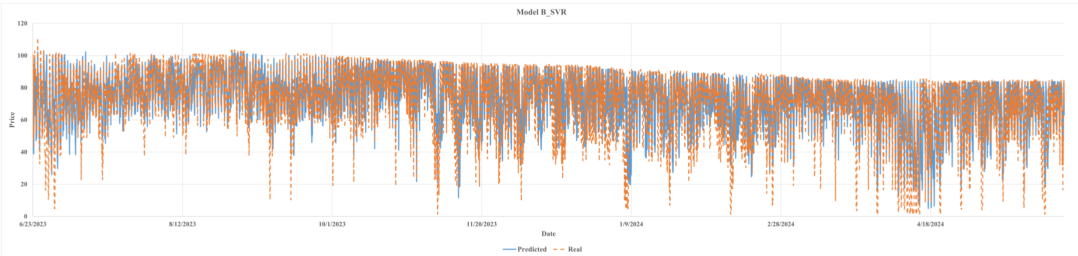
In Figure 4.2a, it is seen that residuals are mostly below zero, which is actually expected from the ME value, -5.438, in Table 4.3 for Model B_mae. Therefore, it can be said that Model B_mae nearly overestimates the real values. In Figure 4.2b, it can be said that residuals are distributed closer to zero, which again can be observed in ME value for Mode B_SVR, -1.637 in Table 4.3. Although, ME value is much closer than -5.438, it is less than zero and a few amount of overestimation can also be observed in Model B_SVR.

Therefore, it can be inferred that, if the limit value for price prediction increases, model performance can increase.

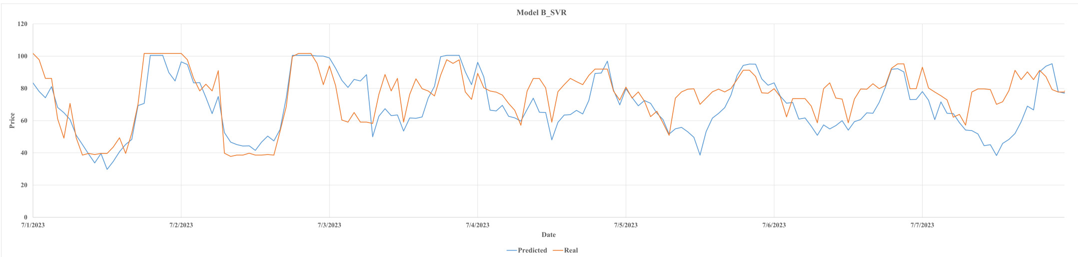


(a) Plot for Model B_mae

(b) Model B_mae Plot Covering 01/07/2023 - 07/07/2023



(c) Plot for Model B_SVR

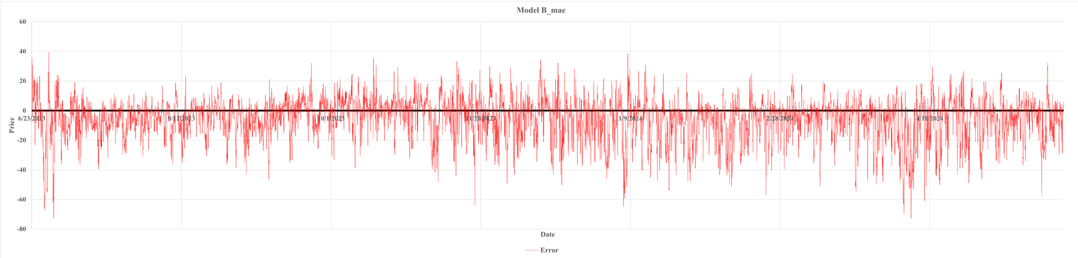


(d) Model B_SVR Plot Covering 01/07/2023 - 07/07/2023

Figure 4.1: Plots for Model B_mae and Model B_SVR

4.1.2 Results with Different Feature Combinations

Initial model results shows that Model B_mae and Model B_SVR outperforms among the other developed models. These results are obtained using all the features. How-



(a) Residuals for Model B_mae

(b) Residuals for Model B_SVR

Figure 4.2: Plots for Residuals in Model B_mae and Model B_SVR

ever, using all the features may cause the training period to slow down because utilizing the lag variables increases the data size. Also, there is a probability of getting same or better results with a smaller sets of features. Therefore, it is logical to search all the feature combinations to find the best model result.

In this section, different combinations of input features are tried and the 5 best results in all of the trials are presented. For the simplicity, the same parameters found in the training period are used in all the results of combinations. The name of the combinations are given according to the feature aliases given in Section 4.1.1.

In Model B_mae and Model B_SVR, there are 8 main features. As it mentioned before, HOUR_SIN, HOUR_COS, DAY_SIN and DAY_COS are always going to be put in feature combination calculations. There will be 15 feature combinations including the Model B_mae and Model B_SVR themselves.

In the results, MAE and $SMAPE$ values are presented. Since $SMAPE$ assigns equal weight to over-estimations and under-estimations, it is used instead of $MAPE$ value.

4.1.2.1 Feature Combination Results for Model B_mae

Table 4.4: Model B_mae Top 5 Results for MAE

Model B_mae Combination	Test Data
1	9.653
1-4	9.872
1-2	9.883
1-2-4	9.896
2	9.897

According to Table 4.4, combination "1", which uses only WORKDAY feature has the smallest MAE value for the test set. Again combination "1" has the smallest $SMAPE$ value for the test set according to Table 4.5.

Table 4.5: Model B_mae Top 5 Results for $SMAPE$ (%)

Model B_mae Combination	Test Data
1	17.530
1-4	17.858
1-2-4	17.875
4	17.905
2	17.985

4.1.2.2 Feature Combination Results for Model B_SVR

According to Table 4.6, combination "1-2-4" has the smallest MAE value for test set. It is seen that Combination "1-2-4" have the smallest $SMAPE$ value as well for test set respectively when the output values in Table 4.7 are examined.

According to the feature combination results, Model B_mae outperforms compared to Model B_SVR in terms of both MAE and $SMAPE$ metrics. Therefore, performance metric analysis are conducted for Model B_mae in the next sections.

Table 4.6: Model B_SVR Top 5 Results for MAE

Model B_SVR Combination	Test Data
1-2-4	10.259
1-2-3-4	10.272
2-4	10.275
2-3-4	10.287
1-4	10.418

Table 4.7: Model B_SVR Top 5 Results for $SMAPE$ (%)

Model B_SVR Combination	Test Data
1-2-4	18.751
2-4	18.760
1-2-3-4	18.772
2-3-4	18.781
1-4	18.998

4.2 Metric Analysis for Model B_mae

After different feature combination results for the Model B_mae and Model B_SVR are investigated, the performance metrics are examined for Model B_mae. Since the test data set is comprised of 12 months, the values found in the previous sections are average values for that time horizon. However, there may be better time periods when the metrics exhibit better performance.

Two of the performance metrics, MAE and $SMAPE$ are examined on hourly, daily, weekly basis. Metric analysis are presented for the top five models of combinations mentioned in Section 4.1.2.1 in the following steps. Tables showing the values in detail can be seen in Appendix B.

To analyze the metrics, confidence intervals are considered. The significance factor (α) is taken as 0.05.

In order to construct hourly confidence intervals, the following equation is used,

$$\bar{x} - t_{\frac{\alpha}{2}}(\frac{s}{\sqrt{n}}) \leq \mu \leq \bar{x} + t_{\frac{\alpha}{2}}(\frac{s}{\sqrt{n}})$$

where $\bar{x}$ is hourly, daily or weekly average value for a metric, α is significance factor, s is sample standard deviation of a metric and n is the number of hours, days or weeks. Since the population standard deviation is not known, t-distribution is used for constructing the confidence intervals.

In Figures 4.3, 4.4, the best five combination results which are derived from Model B_mae for MAE and $SMAPE$ metrics can be seen for the hourly basis.

As it can be seen in Figure 4.3, all five combinations have the similar average $SMAPE$ values for each hour. 12 am has the highest values while 8 pm has the lowest average values for $SMAPE$. Moreover, it can be observed that when the average value for $SMAPE$ is lower, the possible range where the population mean shrinks. However, larger average $SMAPE$ values mean wider range for the true value of the mean for that hour.

MAE metric has more stable outputs for the same five feature combinations. According to Figure 4.4, it can be said that the behavior is the same with the behavior observed in $SMAPE$ values. The smallest average MAE values are again seen in 8 pm for each feature combination. The highest average MAE values are again at 12 am.

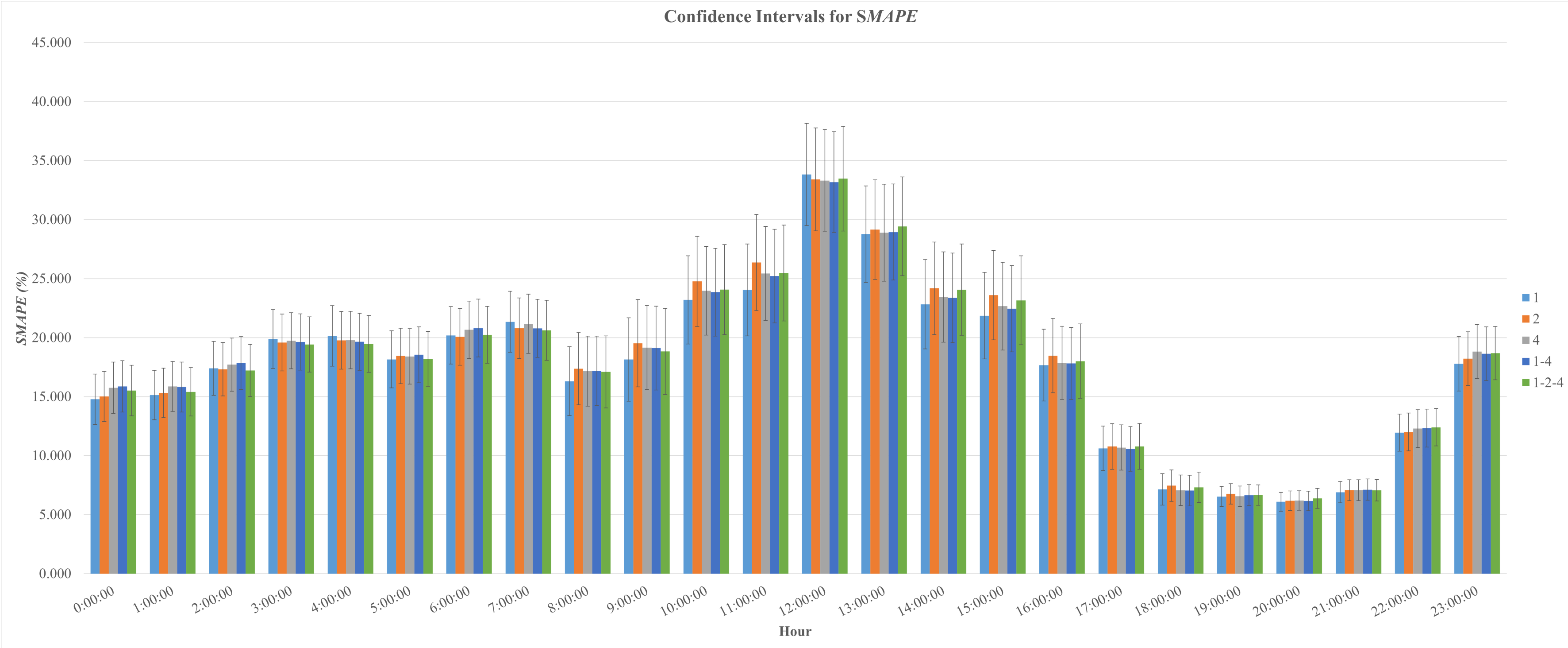


Figure 4.3: Confidence intervals for $SMAPE$ values constructed hourly for the best 5 feature combination results.

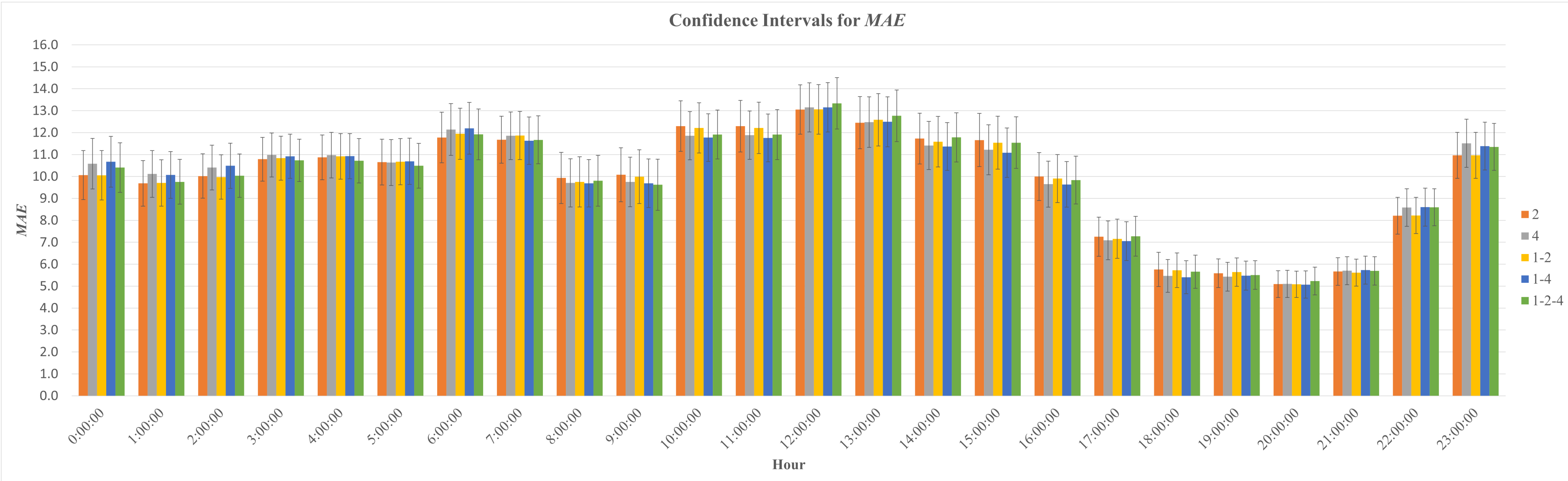


Figure 4.4: Confidence intervals for MAE values constructed hourly for the best 5 feature combination results

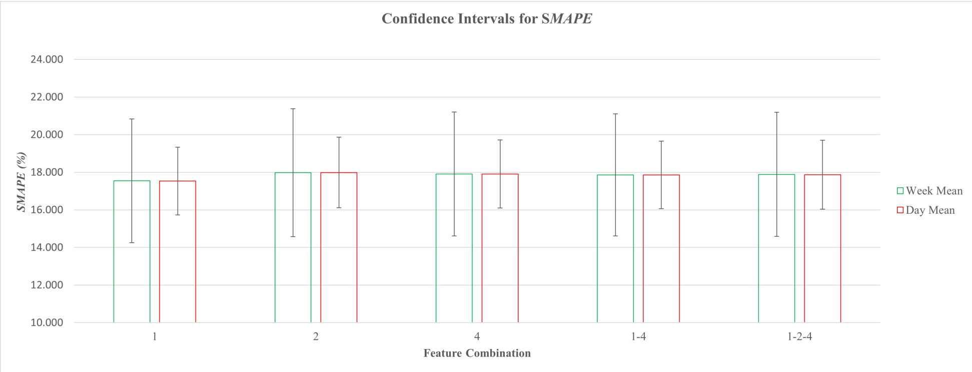


Figure 4.5: Confidence intervals for $SMAPE$ values constructed weekly and daily for the best 5 feature combination results.

When the confidence intervals constructed for daily and weekly are seen in Figure 4.5 and 4.6 a similar behavior is observed. In Figure 4.5 and 4.6, confidence intervals for weekly means are wider than the confidence intervals for daily means. Also, it can be said that MAE values on daily basis are more consistent compared to $SMAPE$ metrics on daily basis.

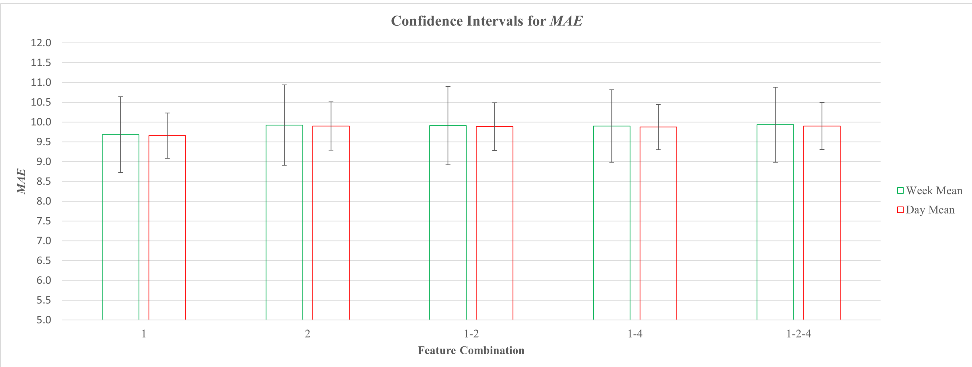


Figure 4.6: Confidence intervals for MAE values constructed weekly and daily for the best 5 feature combination results.

As it can be seen in Figure 4.5 and Figure 4.6, feature combination "1", which is generated using "WORKDAY" feature performs better for daily and weekly analysis than other models generated using other features.

Giving the hourly, daily and weekly performance metric analysis, the sensitivity analysis results for the weight values of Model B_mae are presented in the next section.

4.3 Sensitivity Analysis for Weight Values

In this section, the sensitivity analysis of the weight value are conducted for Model B_mae. Left side of the weight constraint of Problem EMM given in Section 3.5 is changed by adding or subtracting 0.025, 0.050, 0.075 and 0.1. According to those modifications, the performance metric results on the test data set are observed and analyzed. Again, the same best hyperparameters found for both algorithms, XGBoost and SVR, are used in all constraint changes.

In Figure 4.7, all the performance metric results are given for different weight constraints. There is clear fact that subtracting a specific amount from the weight constraint results in a better performance metric. Especially for ME, it can be seen that the least mean deviation from zero is obtained as -0.066 when the weight constraint is equal to 0.925. It is an expected situation because the model overestimates and decreasing weight values reduces the overestimation. Thus, the overestimation of the model decreases by using sum of the weights as 0.925. Decreasing the weight constraint by a specific amount also positively affects the other performance metrics. Some performance metric values; MAE, $MASE$, and $SMAPE$, increase when weight constraint is kept decreasing. For $MAPE$, it can be seen that decreasing weight constraint below 0.9 may positively affect the result. However, when analysis for $MAPE$ are conducted, it is seen that the minimum $MAPE$ value is around 43%. For less values of weight constraint such as 0.75, the $MAPE$ value again increases to 46$\%$. It is also logical that increasing the constraint value increases the overestimation and the performance metric performances decrease.

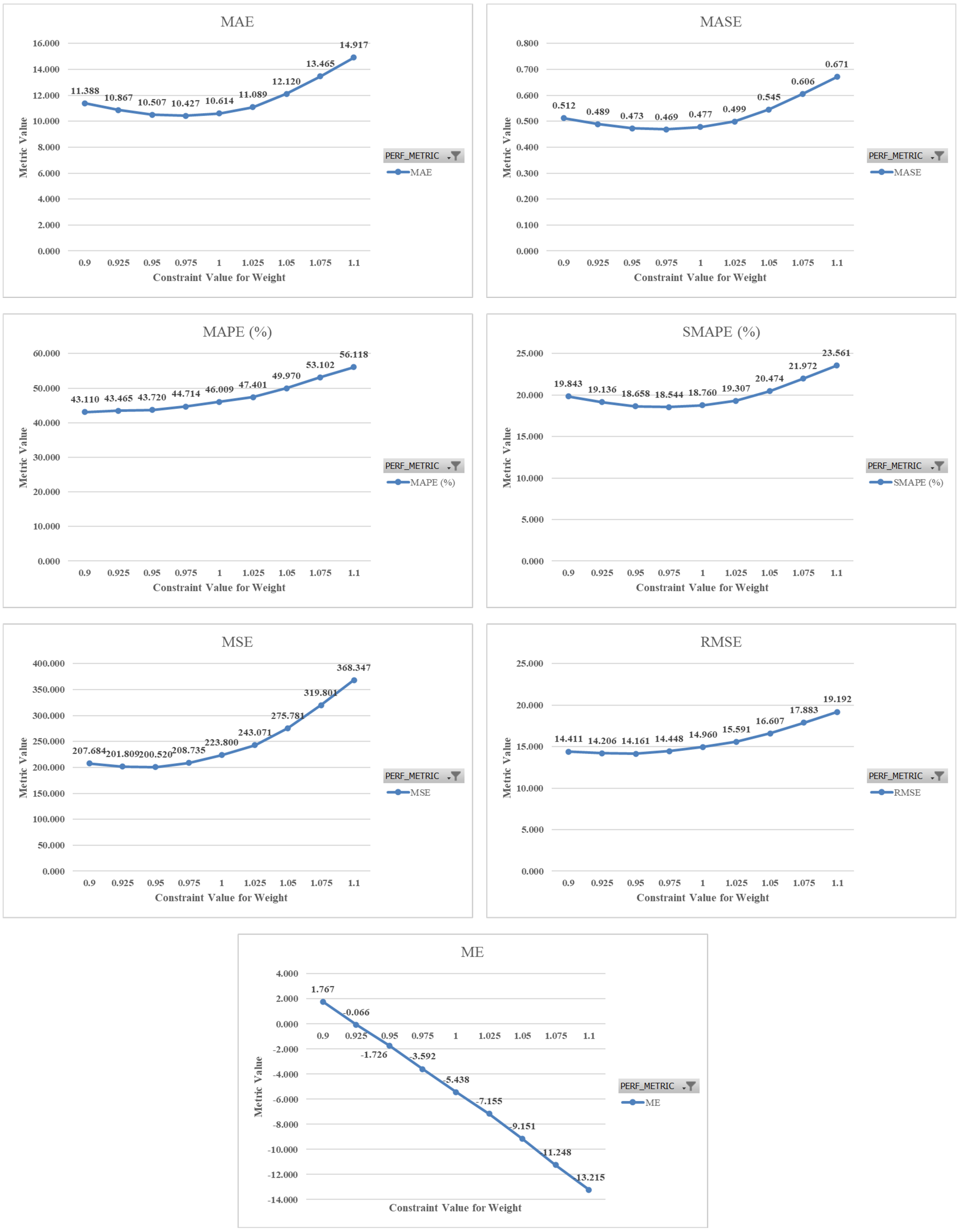


Figure 4.7: Results of the Sensitivity Analysis

4.4 Feature Importance

Lastly, the feature importance can be checked using the SHAP library in python.

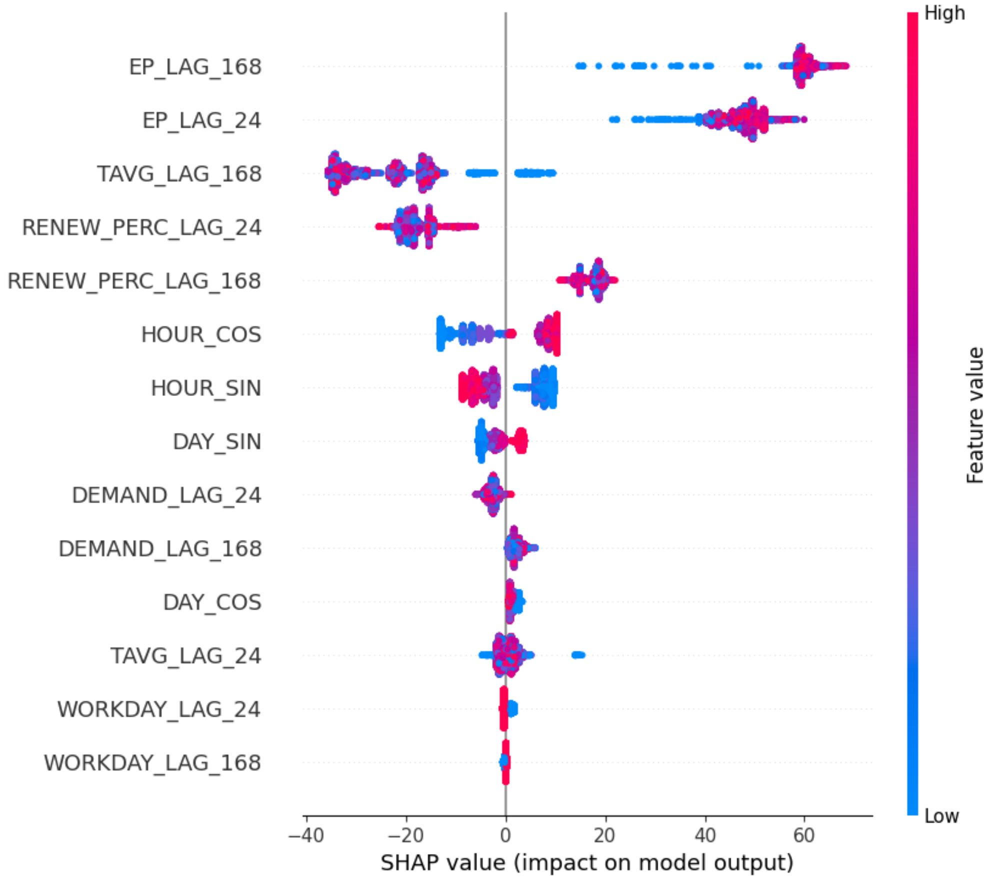


Figure 4.8: Feature Importance Values for Model B_mae

In Figure 4.8, each feature in the model are given vertically on the left side ranked from top to bottom by their mean absolute SHAP values for the entire data set. Each instance value of the feature is given as a point in relative feature row. On the x-axis, the SHAP values are distributed. The color bar on the right shows whether an instance value is high or low. If the value of a variable for a particular instance is relatively high, it appears as a red dot. Relatively low value instances appear as blue dots. In places where there is a high density of SHAP values, the points are stacked vertically. Examining how the SHAP values are distributed reveals how a variable may influence the model's prediction.

According to the SHAP plot in Figure 4.8, it can be said that lag 168 and lag 24 of electricity price are the most important features. Also, it is seen that the high or low values for electricity prices of lag 24 and 168 always cause the prediction values to increase since all high or low instance points have the positive SHAP values. Temperature and renewable resource ratio are other important features. It can be said that higher values of last week's temperature gives lower prediction values for electricity prices. It can be logical because higher temperatures means the season is summer and days are longer than nights which also explains the need for electricity is lower.

CHAPTER 5

CONCLUSIONS

Accurately estimating the electricity prices has been a major problem in the electricity market since the liberalization of the Turkish electricity market in 2001. Accurate forecasting is challenging because of high volatility, many seasonality levels, and nonlinear correlations. Consequently, there are numerous researches to predict the electricity prices in Türkiye.

This thesis examines the Turkish electricity market, starting with its history and presenting a look at how the Turkish DAM works. By understanding how the market has developed and how it operates now, the framework for analyzing and forecasting the electricity prices is given. A review of related studies, both global and specific to Türkiye, on the use of different time series and machine learning methods for the electricity price forecasting, shows the progress and gaps in the existing research.

The methodology reviewed in Chapter 3 for data collection, pre-processing, and modeling, focuses on XGBoost and SVR. These algorithms are chosen for their power and potential to improve predictive accuracy. An ensemble of these two algorithms is considered for predicting the day-ahead electricity prices for Turkish electricity market. The analysis and results in Chapter 4 present the outcomes of the different models produced, including various feature combinations, and evaluate the performance metrics. The results show that two of the models produced, Model B_mae, using Problem EMMmae for determining weights of the algorithms, and Model B_SVR, dominate other alternative models and benchmark models in terms of performance metric values. According to the results, it can be observed that the new methodology compete with the time series and other methodologies given in the literature. Also, it is shown that the model has slightly overestimates the electricity prices. Thus, it can be said that possible increases in price limitations would affect positively the outputs of the model.

Moreover, Model B_mae performs better than Model B_SVR in feature combination processes in terms of both $SMAPE$ and MAE values. Hourly analysis show that model performance is better when the time is 8 or 9 pm. In daily and weekly analysis, it is seen that model generated only using WORKDAY feature and lag of target variable gives better performance. Besides, confidence intervals of daily calculated metric values are narrower compared to weekly calculated values.

When optimal weights are investigated, in Model B_mae, XGBoost algorithm has more weight than SVR. For weight values, also, sensitivity analysis are conducted. As a result of sensitivity analysis, it is seen that decreasing the total weight positively affect the model performance and overestimation decreases.

Lastly, when feature importance analysis conducted, lag values of electricity prices seem as the most effective features with temperature and renewable resource ratio in electricity supply.

In future studies, in order to provide a more comprehensive modeling, future studies might include optimizing the parameters for each feature combination used in the study. In this way, feature combination results would improve with the best parameter settings for each feature combination set. Moreover, exchange rate prediction models can be used in real systems for price limitations which are given as TL by EPIAS. However, the model predicts in USD and by predicting the exchange rates in the future periods, predictions can be modified.

REFERENCES

[1] B. Akkaya, "Comparison of multi-class classification algorithms on early diagnosis of heart diseases," 2019.

[2] J. Somekh, N. Lotan, E. Sussman, and G. A. Yehuda, "Predicting mechanical ventilation effects on six human tissue transcriptomes," *PLoS ONE*, vol. 17, 3 2022.

[3] İlknur Yeşim DİNÇEL, "Turkey's electricity market current situation and alternative policy recommendations," *Ekonomi, Politika & Finans Araştırmaları Dergisi*, pp. 304–321, 8 2021.

[4] S. Singh, S. Patro, and S. K. Parhi, "Evolutionary optimization of machine learning algorithm hyperparameters for strength prediction of high-performance concrete," *Asian Journal of Civil Engineering*, vol. 24, pp. 1–23, 05 2023.

[5] R. H. Assaad and S. Fayek, "Predicting the price of crude oil and its fluctuations using computational econometrics: Deep learning, lstm, and convolutional neural networks," *Econometric Research in Finance*, vol. 6, pp. 119–137, 09 2021.

[6] https://seffaflik.epias.com.tr/electricity/electricity-markets/day-ahead-market-dam/market-clearing-price mcp, "Enerji piyasaları İsletme a.s. (epias)."

[7] J. Lago, G. Marcjasz, B. De Schutter, and R. Weron, "Forecasting day-ahead electricity prices: A review of state-of-the-art algorithms, best practices and an open-access benchmark," *Applied Energy*, vol. 293, p. 116983, 2021.

[8] X. Hao, Y. Zhao, and Y. Wang, "Forecasting the real prices of crude oil using robust regression models with regularization constraints," *Energy Economics*, vol. 86, p. 104683, 2020.

[9] D. J. Swider and C. Weber, "Extended arma models for estimating price developments on day-ahead electricity markets," *Electric Power Systems Research*, vol. 77, pp. 583–593, 4 2007.

[10] C.-I. Kim, I.-K. Yu, and Y. H. Song, "Prediction of system marginal price of electricity using wavelet transform analysis."

[11] R. Steinert and F. Ziel, "Short-to mid-term day-ahead electricity price forecasting using futures," *The Energy Journal*, vol. 40, no. 1, pp. 105–128, 2019.

[12] S. J. Koopman, M. Ooms, and M. A. Carnero, "Periodic seasonal reg-arfima-garch models for daily electricity spot prices," 3 2007.

[13] R. Weron and A. Misiorek, "Short-term electricity price forecasting with time series models: A review and evaluation," *HSC Research Reports*, no. HSC/06/01, 2006.

[14] J. Contreras, R. Espínola, F. J. Nogales, and A. J. Conejo, "Arima models to predict next-day electricity prices," *IEEE Transactions on Power Systems*, vol. 18, pp. 1014–1020, 8 2003.

[15] J. H. Zhao, Z. Y. Dong, Z. Xu, and K. P. Wong, "A statistical approach for interval forecasting of the electricity price," *IEEE Transactions on Power Systems*, vol. 23, pp. 267–276, 5 2008.

[16] A. N. Alkawaz, A. Abdellatif, J. Kanesan, A. S. M. Khairuddin, and H. M. Gheni, "Day-ahead electricity price forecasting based on hybrid regression model," *IEEE Access*, vol. 10, pp. 108021–108033, 2022.

[17] J. Che and J. Wang, "Short-term electricity prices forecasting based on support vector regression and auto-regressive integrated moving average modeling," *Energy Conversion and Management*, vol. 51, pp. 1911–1917, 10 2010.

[18] Y. Q. Tan, Y. X. Shen, X. Y. Yu, and X. Lu, "Day-ahead electricity price forecasting employing a novel hybrid frame of deep learning methods: A case study in nsw, australia," *Electric Power Systems Research*, vol. 220, 7 2023.

[19] S. Anbazhagan and N. Kumarappan, "Day-ahead deregulated electricity market price forecasting using recurrent neural network," *IEEE Systems Journal*, vol. 7, pp. 866–872, 2013.

[20] W. Li and D. M. Becker, "Day-ahead electricity price prediction applying hybrid

models of lstm-based deep learning methods and feature selection algorithms under consideration of market coupling," *Energy*, vol. 237, 12 2021.

[21] A. Brusaferri, M. Matteucci, P. Portolani, and A. Vitali, "Bayesian deep learning based method for probabilistic forecast of day-ahead electricity prices," *Applied Energy*, vol. 250, pp. 1158–1175, 9 2019.

[22] M. Kabak and T. Tasdemir, "Electricity day-ahead market price forecasting by using artificial neural networks: An application for turkey," *Arabian Journal for Science and Engineering*, vol. 45, pp. 2317–2326, 3 2020.

[23] T. Ulgen and G. Poyrazoglu, "Predictor analysis for electricity price forecasting by multiple linear regression," in *2020 International Symposium on Power Electronics, Electrical Drives, Automation and Motion (SPEEDAM)*, pp. 618–622, IEEE, 2020.

[24] A. Arifoğlu and T. Kandemir, "Electricity price forecasting in turkish day-ahead market via deep learning techniques," *Mehmet Akif Ersoy Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi*, vol. 9, pp. 1433–1458, 7 2022.

[25] M. H. Yildirim, A. Ozmen, O. T. Bayrak, and G. W. Weber, "Electricity price modelling for turkey," in *Operations Research Proceedings 2011: Selected Papers of the International Conference on Operations Research (OR 2011), August 30-September 2, 2011, Zurich, Switzerland*, pp. 39–44, Springer, 2012.

[26] M. Kaya, M. B. Karan, and E. Telatar, "Electricity price estimation using deep learning approaches: An empirical study on turkish markets in normal and covid-19 periods," *Expert Systems with Applications*, vol. 224, 8 2023.

[27] S. K. Depren, M. T. Kartal, H. M. Ertuğrul, and Özer Depren, "The role of data frequency and method selection in electricity price estimation: Comparative evidence from turkey in pre-pandemic and pandemic periods," *Renewable Energy*, vol. 186, pp. 217–225, 3 2022.

[28] A. B. İpek, "Prediction of market-clearing price using neural networks based methods and boosting algorithms," *International Advanced Researches and Engineering Journal*, vol. 5, pp. 240–246, 8 2021.

[29] B. TO and S. H, "A comparison of various electricity tariff price forecasting techniques in turkey and identifying the impact of time series periods," 2016.

[30] Ömer Özgur Bozkurt, G. Biricik, and Z. C. Taysi, "Artificial neural network and sarima based models for power load forecasting in turkish electricity market Ö," *PLoS ONE*, vol. 12, 4 2017.

[31] G. O. Kaya, O. F. Demirel, S. Zaim, and R. Gorener, "Day-ahead electricity price forecasting using a wavelet-support vector machine conjunction model," in *Conference on Engineering And Technology Management GCETM-2015*, p. 179, 2015.

[32] C. O'Leary, C. Lynch, R. Bain, G. Smith, and D. Grimes, "A comparison of deep learning vs traditional machine learning for electricity price forecasting," in *2021 4th International Conference on Information and Computer Technologies (ICICT)*, pp. 6–12, IEEE, 2021.

[33] B. Bedir and F. Abut, "Forecasting the system marginal price using long short-term memory and support vector machine."

[34] S. Zhong, K. Zhang, M. Bagheri, J. G. Burken, A. Gu, B. Li, X. Ma, B. L. Marrone, Z. J. Ren, J. Schrier, *et al.*, "Machine learning: new ideas and tools in environmental science and engineering," *Environmental science & technology*, vol. 55, no. 19, pp. 12741–12754, 2021.

[35] S. Siami-Namini, N. Tavakoli, and A. S. Namin, "A comparison of arima and lstm in forecasting time series," in *2018 17th IEEE international conference on machine learning and applications (ICMLA)*, pp. 1394–1401, Ieee, 2018.

[36] F. Ng, R. Jiang, and J. C. Chow, "Predicting radiation treatment planning evaluation parameter using artificial intelligence and machine learning," *IOP SciNotes*, vol. 1, no. 1, p. 014003, 2020.

[37] L. Tschora, E. Pierre, M. Plantevit, and C. Robardet, "Electricity price forecasting on the day-ahead market using machine learning," *Applied Energy*, vol. 313, 5 2022.

[38] A. Cruz, A. Muñoz, J. L. Zamora, and R. Espínola, "The effect of wind generation and weekday on spanish electricity spot price forecasting," *Electric Power Systems Research*, vol. 81, no. 10, pp. 1924–1935, 2011.

[39] https://www.ncei.noaa.gov/, "National centers for environmental information."

[40] https://evds2.tcmb.gov.tr/index.php?/evds/serieMarket/collapse_2 ~5862/DataGroup/turkish/bie_dkdovizgn/, "Tcmb evds."

[41] D. Chakraborty and H. Elzarka, "Advanced machine learning techniques for building performance simulation: a comparative analysis," *Journal of Building Performance Simulation*, vol. 12, no. 2, pp. 193–207, 2019.

[42] J. Leites, V. Cerqueira, and C. Soares, "Lag selection for univariate time series forecasting using deep learning: An empirical study," *arXiv preprint arXiv:2405.11237*, 2024.

[43] T. Fischer, C. Krauss, and A. Treichel, "Machine learning for time series forecasting-a simulation study," tech. rep., FAU Discussion Papers in Economics, 2018.

[44] O. Surakhi, M. A. Zaidan, P. L. Fung, N. H. Motlagh, S. Serhan, M. Alkhanafseh, R. M. Ghoniem, and T. Hussein, "Time-lag selection for time-series forecasting using neural network and heuristic algorithm," *Electronics (Switzerland)*, vol. 10, 10 2021.

[45] V. R. Joseph, "Optimal ratio for data splitting," *Statistical Analysis and Data Mining*, vol. 15, pp. 531–538, 8 2022.

[46] V. Sharma, "A study on data scaling methods for machine learning," *International Journal for Global Academic & Scientific Research*, vol. 1, no. 1, pp. 31–42, 2022.

[47] P. J. M. Ali, R. H. Faraj, E. Koya, P. J. M. Ali, and R. H. Faraj, "Data normalization and standardization: a technical report," *Mach Learn Tech Rep*, vol. 1, no. 1, pp. 1–6, 2014.

[48] T. Yu and H. Zhu, "Hyper-parameter optimization: A review of algorithms and applications," *arXiv preprint arXiv:2003.05689*, 2020.

[49] L. Yang and A. Shami, "On hyperparameter optimization of machine learning algorithms: Theory and practice," *Neurocomputing*, vol. 415, pp. 295–316, 2020.

[50] M. Schnaubelt, "A comparison of machine learning model validation schemes for non-stationary time series data," tech. rep., FAU Discussion Papers in Economics, 2019.

[51] D. Berrar *et al.*, "Cross-validation.," 2019.

[52] F. Budiman, "Svm-rbf parameters testing optimization using cross validation and grid search to improve multiclass classification," *Scientific Visualization*, vol. 11, pp. 80–90, 2019.

[53] M. Adnan, A. A. S. Alarood, M. I. Uddin, and I. ur Rehman, "Utilizing grid search cross-validation with adaptive boosting for augmenting performance of machine learning models," *PeerJ Computer Science*, vol. 8, 2022.

[54] I. K. Nti, O. Nyarko-Boateng, J. Aning, *et al.*, "Performance of machine learning algorithms with different k values in k-fold crossvalidation," *International Journal of Information Technology and Computer Science*, vol. 13, no. 6, pp. 61–71, 2021.

[55] K. Roy, S. Kar, and R. N. Das, *Understanding the basics of QSAR for applications in pharmaceutical sciences and risk assessment*. Academic press, 2015.

[56] V. Vapnik and A. Y. Lerner, "Recognition of patterns with help of generalized portraits," *Avtomat. i Telemekh*, vol. 24, no. 6, pp. 774–780, 1963.

[57] V. Vapnik and A. Y. Chervonenkis, "A class of algorithms for pattern recognition learning," *Avtomat. i Telemekh*, vol. 25, no. 6, pp. 937–945, 1964.

[58] B. E. Boser, I. M. Guyon, and V. N. Vapnik, "A training algorithm for optimal margin classifiers," in *Proceedings of the fifth annual workshop on Computational learning theory*, pp. 144–152, 1992.

[59] H. Drucker, C. J. Burges, L. Kaufman, A. Smola, and V. Vapnik, "Support vector regression machines," *Advances in neural information processing systems*, vol. 9, 1996.

[60] V. Vapnik, *The nature of statistical learning theory*. Springer science & business media, 2013.

[61] D. Basak, S. Pal, and D. C. Patranabis, "Support vector regression," 2007.

[62] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pp. 785–794, 2016.

[63] J. Friedman, T. Hastie, and R. Tibshirani, "Additive logistic regression: A statistical view of boosting," 2000.

[64] M. Zounemat-Kermani, O. Batelaan, M. Fadaee, and R. Hinkelmann, "Ensemble machine learning paradigms in hydrology: A review," *Journal of Hydrology*, vol. 598, p. 126266, 2021.

[65] M. Shahhosseini, G. Hu, and H. Pham, "Optimizing ensemble weights and hyperparameters of machine learning models for regression problems," *Machine Learning with Applications*, vol. 7, p. 100251, 2022.

[66] A. Botchkarev, "A new typology design of performance metrics to measure errors in machine learning regression algorithms," *Interdisciplinary Journal of Information, Knowledge, and Management*, vol. 14, pp. 045–076, 2019.

[67] B. E. Flores, "A pragmatic view of accuracy measurement in forecasting," *Omega*, vol. 14, no. 2, pp. 93–98, 1986.

[68] C. Tofallis, "A better measure of relative prediction accuracy for model selection and model estimation," *Journal of the Operational Research Society*, vol. 66, no. 8, pp. 1352–1362, 2015.

[69] S. Kim and H. Kim, "A new metric of absolute percentage error for intermittent demand forecasts," *International Journal of Forecasting*, vol. 32, no. 3, pp. 669–679, 2016.

[70] R. J. Hyndman and A. B. Koehler, "Another look at measures of forecast accuracy," *International journal of forecasting*, vol. 22, no. 4, pp. 679–688, 2006.

[71] F. X. Diebold and R. S. Mariano, "Comparing predictive accuracy," *Journal of Business & economic statistics*, vol. 20, no. 1, pp. 134–144, 2002.

[72] G. E. Box, G. M. Jenkins, G. C. Reinsel, and G. M. Ljung, *Time series analysis: forecasting and control*. John Wiley & Sons, 2015.

[73] R. J. Hyndman and Y. Khandakar, "Automatic time series forecasting: the forecast package for r," *Journal of statistical software*, vol. 27, pp. 1–22, 2008.

[74] U. Ugurlu, I. Oksuz, and O. Tas, "Electricity price forecasting using recurrent neural networks," *Energies*, vol. 11, no. 5, p. 1255, 2018.

Appendix A

A.1 Grid Search Results

A.1.1 Grid Search Results For Xgboost Algorithm

Table A.1: Grid Search Results for Model A_XGB

param_XGB__learning_rate	param_XGB__max_depth	param_XGB__n_estimators	split0_test_score	split1_test_score	split2_test_score	mean_test_score	std_test_score	rank_test_score
0.01	4	200	-27.7687	-20.8493	-30.8061	-26.4747	4.166559	1
0.01	6	200	-28.765	-20.8787	-30.0646	-26.5694	4.058779	2
0.1	4	100	-30.1796	-18.8744	-31.1333	-26.7291	5.567729	3
0.01	8	200	-28.9872	-21.8681	-32.3576	-27.7377	4.372529	4
0.1	4	150	-31.0168	-19.068	-33.9812	-28.022	6.446076	5
0.01	4	150	-28.1027	-22.0566	-34.11	-28.0898	4.92077	6
0.01	6	150	-29.4556	-22.3213	-32.8399	-28.2056	4.384241	7
0.1	6	100	-30.9979	-20.1909	-34.5762	-28.5883	6.114928	8
0.01	8	150	-29.1045	-22.691	-34.4725	-28.756	4.816049	9
0.1	4	200	-31.1133	-19.3366	-36.2208	-28.8902	7.069929	10
0.1	6	150	-32.1989	-20.4321	-36.6211	-29.7507	6.832057	11
0.1	8	100	-30.8404	-22.1389	-36.7035	-29.8943	5.983491	12
0.1	6	200	-32.8718	-20.5181	-38.0435	-30.4778	7.352249	13
0.01	4	100	-29.3385	-24.2944	-38.4904	-30.7078	5.875781	14
0.01	8	100	-29.7291	-24.9256	-38.1652	-30.9399	5.472432	15
0.01	6	100	-30.383	-25.047	-37.5363	-30.9888	5.116703	16
0.1	8	150	-31.7444	-22.6771	-38.7589	-31.0601	6.58315	17
0.1	8	200	-31.7872	-22.9959	-39.3443	-31.3758	6.680538	18
0.001	8	200	-33.8791	-33.0575	-51.2165	-39.3844	8.37332	19
0.001	6	200	-33.9592	-33.1026	-51.5356	-39.5325	8.494721	20
0.001	4	200	-33.7525	-32.8313	-52.2176	-39.6005	8.929586	21
0.001	8	150	-34.2489	-33.8326	-52.5283	-40.2032	8.716775	22
0.001	6	150	-34.2922	-33.8869	-52.76	-40.313	8.80288	23
0.001	4	150	-34.223	-33.6723	-53.3866	-40.4273	9.166387	24
0.001	8	100	-34.6327	-34.6608	-53.9321	-41.0752	9.091205	25
0.001	6	100	-34.7374	-34.7298	-54.0284	-41.1652	9.095637	26
0.001	4	100	-34.73	-34.5661	-54.6285	-41.3082	9.419111	27

Table A.2: Grid Search Results for Model B_XGB

param_XGB__learning_rate	param_XGB__max_depth	param_XGB__n_estimators	split0_test_score	split1_test_score	split2_test_score	mean_test_score	std_test_score	rank_test_score
0.1	4	100	-19.0092	-13.912	-13.4051	-15.4421	2.530781	1
0.1	4	150	-20.0667	-14.1846	-13.3108	-15.854	3.000103	2
0.1	4	200	-20.2802	-14.0373	-13.3554	-15.891	3.116139	3
0.1	6	100	-21.0914	-14.3872	-16.926	-17.4682	2.763688	4
0.1	8	100	-19.5779	-15.3877	-18.3386	-17.768	1.757579	5
0.1	6	150	-21.6201	-14.6297	-17.3178	-17.8559	2.879075	6
0.01	4	200	-20.5202	-15.2429	-18.6046	-18.1226	2.181224	7
0.1	8	150	-19.9754	-15.4765	-19.0457	-18.1659	1.939169	8
0.1	6	200	-21.9517	-14.7382	-18.3944	-18.3615	2.944992	9
0.1	8	200	-20.0987	-15.6902	-19.6696	-18.4861	1.984795	10
0.01	6	200	-20.4727	-15.7188	-20.0997	-18.7637	2.158461	11
0.01	8	200	-21.0414	-16.3505	-21.9086	-19.7668	2.44152	12
0.01	4	150	-21.5204	-16.6115	-21.8386	-19.9902	2.392618	13
0.01	6	150	-21.6486	-17.347	-23.1915	-20.7291	2.473012	14
0.01	8	150	-22.1887	-17.9319	-25.1351	-21.7519	2.956871	15
0.01	4	100	-23.5144	-19.3882	-27.2842	-23.3956	3.224613	16
0.01	6	100	-23.884	-20.3297	-29.0615	-24.4251	3.585199	17
0.01	8	100	-24.2842	-20.8554	-30.7838	-25.3078	4.117375	18
0.001	4	200	-31.7704	-31.1514	-48.4946	-37.1388	8.033749	19
0.001	6	200	-31.8462	-31.3136	-49.21	-37.4566	8.313718	20
0.001	8	200	-31.712	-31.5686	-49.6936	-37.6581	8.510584	21
0.001	4	150	-32.6488	-32.3494	-50.5034	-38.5005	8.488168	22
0.001	6	150	-32.7157	-32.5242	-51.1042	-38.7814	8.713917	23
0.001	8	150	-32.5431	-32.6837	-51.4178	-38.8815	8.864655	24
0.001	4	100	-33.5792	-33.6361	-52.6337	-39.9497	8.969003	25
0.001	6	100	-33.6934	-33.7907	-53.072	-40.1853	9.112347	26
0.001	8	100	-33.565	-33.8749	-53.2821	-40.2407	9.222553	27

Table A.3: Grid Search Results for Model C_XGB

param_XGB__learning_rate	param_XGB__max_depth	param_XGB__n_estimators	split0_test_score	split1_test_score	split2_test_score	mean_test_score	std_test_score	rank_test_score
0.1	4	100	-26.4708	-16.5541	-22.3273	-21.784	4.066664	1
0.1	6	100	-26.6236	-16.55	-22.6964	-21.9567	4.145651	2
0.1	4	150	-27.0907	-16.6746	-22.6835	-22.1496	4.269078	3
0.1	4	200	-27.424	-17.2094	-22.9945	-22.5426	4.182303	4
0.01	4	200	-26.7964	-16.6724	-24.2841	-22.5843	4.304301	5
0.1	6	150	-27.4932	-17.1038	-23.9925	-22.8632	4.315996	6
0.01	6	200	-27.265	-17.0929	-24.7662	-23.0414	4.328113	7
0.1	6	200	-28.0868	-17.5871	-24.7109	-23.4616	4.376583	8
0.01	4	150	-27.6775	-17.8033	-26.3971	-23.9593	4.384247	9
0.01	8	200	-28.6208	-16.7866	-26.5095	-23.9723	5.15366	10
0.1	8	100	-28.8441	-17.1122	-26.233	-24.0631	5.029297	11
0.1	8	150	-29.1892	-17.5216	-27.1826	-24.6311	5.093499	12
0.01	6	150	-28.3171	-18.3273	-27.5077	-24.7174	4.530518	13
0.1	8	200	-29.6373	-17.7819	-27.5715	-24.9969	5.17101	14
0.01	8	150	-29.3429	-17.8007	-28.7215	-25.2884	5.300649	15
0.01	4	100	-29.3824	-19.7953	-30.1495	-26.4424	4.710628	16
0.01	6	100	-30.5789	-20.5804	-32.0828	-27.7474	5.104877	17
0.01	8	100	-31.1081	-20.0818	-32.9936	-28.0612	5.694516	18
0.001	4	200	-36.9679	-27.98	-45.3515	-36.7665	7.093338	19
0.001	6	200	-37.1648	-28.2551	-46.2097	-37.2099	7.329973	20
0.001	8	200	-37.1832	-28.0789	-46.6052	-37.2891	7.5637	21
0.001	4	150	-37.6493	-28.8367	-46.878	-37.788	7.365975	22
0.001	6	150	-37.8668	-29.0218	-47.533	-38.1405	7.559638	23
0.001	8	150	-37.8353	-28.9059	-47.8615	-38.2009	7.742881	24
0.001	4	100	-38.464	-29.7443	-48.5052	-38.9045	7.665448	25
0.001	6	100	-38.5797	-29.8654	-48.958	-39.1344	7.804362	26
0.001	8	100	-38.5971	-29.7997	-49.1312	-39.176	7.902682	27

A.1.2 Grid Search Results For SVR Algorithm

Table A.4: Grid Search Results for Model A_SVR

param_SVM__C	param_SVM__degree	param_SVM__epsilon	param_SVM__gamma	param_SVM__kernel	split0_test_score	split1_test_score	split2_test_score	mean_test_score	std_test_score	rank_test_score
10	1	1	0.1	rbf	-22.9568	-15.0101	-19.8121	-19.2597	3.26765	1
10	2	1	0.1	rbf	-22.9568	-15.0101	-19.8121	-19.2597	3.26765	1
10	3	1	0.1	rbf	-22.9568	-15.0101	-19.8121	-19.2597	3.26765	1
10	1	1	auto	rbf	-22.9808	-14.9911	-19.8257	-19.2659	3.285735	4
10	2	1	auto	rbf	-22.9808	-14.9911	-19.8257	-19.2659	3.285735	4
10	3	1	auto	rbf	-22.9808	-14.9911	-19.8257	-19.2659	3.285735	4
5	1	1	0.1	rbf	-23.7818	-14.7802	-19.816	-19.4594	3.683526	7
5	2	1	0.1	rbf	-23.7818	-14.7802	-19.816	-19.4594	3.683526	7
5	3	1	0.1	rbf	-23.7818	-14.7802	-19.816	-19.4594	3.683526	7
5	1	1	auto	rbf	-23.712	-14.8499	-19.8553	-19.4724	3.628048	10
5	2	1	auto	rbf	-23.712	-14.8499	-19.8553	-19.4724	3.628048	10
5	3	1	auto	rbf	-23.712	-14.8499	-19.8553	-19.4724	3.628048	10
10	1	5	auto	rbf	-23.069	-15.2253	-20.6888	-19.661	3.283598	13
10	2	5	auto	rbf	-23.069	-15.2253	-20.6888	-19.661	3.283598	13
10	3	5	auto	rbf	-23.069	-15.2253	-20.6888	-19.661	3.283598	13
10	1	5	0.1	rbf	-23.1104	-15.2154	-20.7219	-19.6825	3.305847	16
10	2	5	0.1	rbf	-23.1104	-15.2154	-20.7219	-19.6825	3.305847	16
10	3	5	0.1	rbf	-23.1104	-15.2154	-20.7219	-19.6825	3.305847	16
100	1	1	0.1	rbf	-24.1149	-15.1404	-20.431	-19.8954	3.683357	19
100	2	1	0.1	rbf	-24.1149	-15.1404	-20.431	-19.8954	3.683357	19
100	3	1	0.1	rbf	-24.1149	-15.1404	-20.431	-19.8954	3.683357	19
100	1	1	auto	rbf	-24.2423	-15.1869	-20.3605	-19.9299	3.709392	22
100	2	1	auto	rbf	-24.2423	-15.1869	-20.3605	-19.9299	3.709392	22
100	3	1	auto	rbf	-24.2423	-15.1869	-20.3605	-19.9299	3.709392	22
5	1	5	auto	rbf	-23.9352	-15.1897	-20.9724	-20.0324	3.631702	25
5	2	5	auto	rbf	-23.9352	-15.1897	-20.9724	-20.0324	3.631702	25
5	3	5	auto	rbf	-23.9352	-15.1897	-20.9724	-20.0324	3.631702	25
5	1	5	0.1	rbf	-23.9577	-15.2738	-20.9672	-20.0663	3.601986	28
5	2	5	0.1	rbf	-23.9577	-15.2738	-20.9672	-20.0663	3.601986	28
5	3	5	0.1	rbf	-23.9577	-15.2738	-20.9672	-20.0663	3.601986	28
100	1	1	auto	poly	-24.204	-16.7498	-19.2746	-20.0761	3.095491	31
100	1	1	0.1	poly	-24.205	-16.7504	-19.2735	-20.0763	3.095842	32
10	1	1	auto	poly	-24.2078	-16.7697	-19.2929	-20.0901	3.088448	33
10	1	1	0.1	poly	-24.2109	-16.7691	-19.294	-20.0914	3.089947	34
5	1	1	auto	poly	-24.215	-16.7583	-19.3131	-20.0955	3.094062	35
5	1	1	0.1	poly	-24.2113	-16.7618	-19.3213	-20.0981	3.09049	36
100	1	5	0.1	rbf	-24.4598	-15.4556	-20.6285	-20.1813	3.689525	37
100	2	5	0.1	rbf	-24.4598	-15.4556	-20.6285	-20.1813	3.689525	37
100	3	5	0.1	rbf	-24.4598	-15.4556	-20.6285	-20.1813	3.689525	37
100	1	5	auto	rbf	-24.3259	-15.504	-20.7289	-20.1863	3.621915	40
100	2	5	auto	rbf	-24.3259	-15.504	-20.7289	-20.1863	3.621915	40
100	3	5	auto	rbf	-24.3259	-15.504	-20.7289	-20.1863	3.621915	40
10	1	10	0.1	rbf	-23.2847	-15.5999	-21.7355	-20.2067	3.318314	43
10	2	10	0.1	rbf	-23.2847	-15.5999	-21.7355	-20.2067	3.318314	43
10	3	10	0.1	rbf	-23.2847	-15.5999	-21.7355	-20.2067	3.318314	43
1	1	1	auto	poly	-24.2652	-16.8359	-19.5456	-20.2156	3.069777	46
10	1	10	auto	rbf	-23.3506	-15.6302	-21.7082	-20.2297	3.320705	47
10	2	10	auto	rbf	-23.3506	-15.6302	-21.7082	-20.2297	3.320705	47
10	3	10	auto	rbf	-23.3506	-15.6302	-21.7082	-20.2297	3.320705	47
1	1	1	0.1	poly	-24.2646	-16.8567	-19.5881	-20.2365	3.058832	50
5	3	1	auto	poly	-27.1719	-15.7231	-17.972	-20.289	4.952813	51
10	3	1	0.1	poly	-27.3779	-15.6467	-17.9771	-20.3339	5.070928	52
10	3	1	auto	poly	-27.5409	-15.6282	-17.9776	-20.3822	5.152028	53
5	3	1	0.1	poly	-27.1678	-15.8544	-18.227	-20.4164	4.871227	54
5	1	10	0.1	rbf	-23.9716	-15.6036	-22.0293	-20.5349	3.575938	55
5	2	10	0.1	rbf	-23.9716	-15.6036	-22.0293	-20.5349	3.575938	55
5	3	10	0.1	rbf	-23.9716	-15.6036	-22.0293	-20.5349	3.575938	55
5	1	10	auto	rbf	-24.0567	-15.5457	-22.0193	-20.5405	3.628528	58
5	2	10	auto	rbf	-24.0567	-15.5457	-22.0193	-20.5405	3.628528	58
5	3	10	auto	rbf	-24.0567	-15.5457	-22.0193	-20.5405	3.628528	58
5	3	5	auto	poly	-26.9387	-16.0958	-18.6486	-20.561	4.628526	61
10	3	5	0.1	poly	-27.1692	-15.9872	-18.5775	-20.578	4.779152	62
100	1	5	auto	poly	-24.6095	-17.2198	-20.001	-20.6101	3.047406	63
100	1	5	0.1	poly	-24.6092	-17.2201	-20.0016	-20.6103	3.047123	64
10	1	5	auto	poly	-24.6047	-17.2397	-20.0141	-20.6195	3.037091	65
10	1	5	0.1	poly	-24.6089	-17.2438	-20.0083	-20.6203	3.037776	66
10	3	5	auto	poly	-27.5168	-15.9168	-18.4665	-20.6334	4.977412	67
5	3	5	0.1	poly	-26.8872	-16.2383	-18.7814	-20.6356	4.540832	68
5	1	5	auto	poly	-24.5958	-17.2555	-20.0607	-20.6373	3.024291	69

Table A.4: Grid Search Results for Model A_SVR (Cont'd)

param_SVM__C	param_SVM__degree	param_SVM__epsilon	param_SVM__gamma	param_SVM__kernel	split0_test_score	split1_test_score	split2_test_score	mean_test_score	std_test_score	rank_test_score
5	1	5	0.1	poly	-24.5993	-17.2631	-20.0628	-20.6418	3.022834	70
100	3	1	0.1	poly	-28.1361	-15.653	-18.3573	-20.7154	5.362042	71
1	1	5	auto	poly	-24.6097	-17.3292	-20.2372	-20.7253	2.992225	72
1	1	5	0.1	poly	-24.6206	-17.336	-20.2641	-20.7402	2.99295	73
100	3	1	auto	poly	-28.1993	-15.6931	-18.3996	-20.764	5.372401	74
5	3	10	0.1	poly	-26.4947	-16.5262	-19.5895	-20.8701	4.169152	75
1	1	1	0.1	rbf	-26.2172	-15.2134	-21.2336	-20.888	4.498936	76
1	2	1	0.1	rbf	-26.2172	-15.2134	-21.2336	-20.888	4.498936	76
1	3	1	0.1	rbf	-26.2172	-15.2134	-21.2336	-20.888	4.498936	76
5	3	10	auto	poly	-27.0346	-16.3464	-19.3076	-20.8962	4.505697	79
100	1	10	0.1	rbf	-25.0177	-16.1866	-21.4952	-20.8999	3.62977	80
100	2	10	0.1	rbf	-25.0177	-16.1866	-21.4952	-20.8999	3.62977	80
100	3	10	0.1	rbf	-25.0177	-16.1866	-21.4952	-20.8999	3.62977	80
10	3	10	0.1	poly	-27.2861	-16.1847	-19.2552	-20.9086	4.680514	83
10	3	10	auto	poly	-27.4096	-16.1361	-19.2513	-20.9324	4.753401	84
100	3	5	0.1	poly	-28.155	-15.8354	-18.8327	-20.941	5.245747	85
100	3	5	auto	poly	-28.1482	-15.9039	-18.8832	-20.9784	5.213645	86
100	1	10	auto	rbf	-25.2384	-16.3245	-21.5169	-21.0266	3.655581	87
100	2	10	auto	rbf	-25.2384	-16.3245	-21.5169	-21.0266	3.655581	87
100	3	10	auto	rbf	-25.2384	-16.3245	-21.5169	-21.0266	3.655581	87
1	1	1	auto	rbf	-26.4224	-15.2551	-21.435	-21.0375	4.56768	90
1	2	1	auto	rbf	-26.4224	-15.2551	-21.435	-21.0375	4.56768	90
1	3	1	auto	rbf	-26.4224	-15.2551	-21.435	-21.0375	4.56768	90
1	3	1	auto	poly	-25.8152	-17.1263	-20.4644	-21.1353	3.578807	93
1	3	5	auto	poly	-25.7569	-17.3423	-20.7724	-21.2905	3.454722	94
100	1	10	auto	poly	-25.1314	-17.8335	-20.9873	-21.3174	2.988511	95
100	1	10	0.1	poly	-25.1319	-17.8335	-20.9873	-21.3176	2.988696	96
10	1	10	auto	poly	-25.1293	-17.8574	-21.01	-21.3322	2.977437	97
10	1	10	0.1	poly	-25.1257	-17.8536	-21.022	-21.3338	2.976979	98
5	1	10	0.1	poly	-25.1198	-17.8649	-21.0415	-21.342	2.969409	99
5	1	10	auto	poly	-25.1322	-17.8633	-21.0426	-21.346	2.975282	100
1	3	1	0.1	poly	-25.3065	-17.4609	-21.3536	-21.3737	3.202982	101
1	1	10	auto	poly	-25.063	-17.927	-21.2021	-21.3973	2.916528	102
1	1	10	0.1	poly	-25.0578	-17.9077	-21.2336	-21.3997	2.921381	103
1	1	5	0.1	rbf	-26.4218	-15.6105	-22.2374	-21.4232	4.451069	104
1	2	5	0.1	rbf	-26.4218	-15.6105	-22.2374	-21.4232	4.451069	104
1	3	5	0.1	rbf	-26.4218	-15.6105	-22.2374	-21.4232	4.451069	104
1	3	10	auto	poly	-25.4843	-17.5374	-21.2831	-21.4349	3.24612	107
1	3	5	0.1	poly	-25.1869	-17.6796	-21.6841	-21.5169	3.067118	108
1	1	5	auto	rbf	-26.6283	-15.5953	-22.5057	-21.5765	4.551879	109
1	2	5	auto	rbf	-26.6283	-15.5953	-22.5057	-21.5765	4.551879	109
1	3	5	auto	rbf	-26.6283	-15.5953	-22.5057	-21.5765	4.551879	109
100	3	10	0.1	poly	-28.798	-16.2543	-19.7923	-21.6149	5.280587	112
100	3	10	auto	poly	-28.816	-16.3155	-19.9138	-21.6818	5.254215	113
1	3	10	0.1	poly	-25.2084	-17.8936	-22.075	-21.7257	2.996472	114
0.1	1	1	auto	poly	-24.7793	-18.1737	-22.7469	-21.9	2.762421	115
1	1	10	0.1	rbf	-26.5976	-15.9012	-23.2534	-21.9174	4.467791	116
1	2	10	0.1	rbf	-26.5976	-15.9012	-23.2534	-21.9174	4.467791	116
1	3	10	0.1	rbf	-26.5976	-15.9012	-23.2534	-21.9174	4.467791	116
1	1	10	auto	rbf	-26.7638	-15.9139	-23.5177	-22.0652	4.546983	119
1	2	10	auto	rbf	-26.7638	-15.9139	-23.5177	-22.0652	4.546983	119
1	3	10	auto	rbf	-26.7638	-15.9139	-23.5177	-22.0652	4.546983	119
0.1	1	1	0.1	poly	-24.8335	-18.3487	-23.244	-22.1421	2.75967	122
0.1	1	5	auto	poly	-24.9344	-18.3962	-23.1132	-22.1479	2.755115	123
0.1	1	5	0.1	poly	-25.0297	-18.5953	-23.579	-22.4013	2.755663	124
0.1	1	1	auto	sigmoid	-25.4249	-18.4522	-23.5216	-22.4663	2.942798	125
0.1	2	1	auto	sigmoid	-25.4249	-18.4522	-23.5216	-22.4663	2.942798	125
0.1	3	1	auto	sigmoid	-25.4249	-18.4522	-23.5216	-22.4663	2.942798	125
0.1	1	5	auto	sigmoid	-25.5371	-18.5604	-23.8071	-22.6349	2.96641	128
0.1	2	5	auto	sigmoid	-25.5371	-18.5604	-23.8071	-22.6349	2.96641	128
0.1	3	5	auto	sigmoid	-25.5371	-18.5604	-23.8071	-22.6349	2.96641	128
0.1	1	1	0.1	sigmoid	-25.4128	-18.6061	-23.9372	-22.6521	2.923636	131
0.1	2	1	0.1	sigmoid	-25.4128	-18.6061	-23.9372	-22.6521	2.923636	131
0.1	3	1	0.1	sigmoid	-25.4128	-18.6061	-23.9372	-22.6521	2.923636	131
0.1	1	10	auto	poly	-25.187	-18.8435	-24.0539	-22.6948	2.762321	134
0.1	1	5	0.1	sigmoid	-25.4692	-18.7351	-24.1477	-22.784	2.91341	135
0.1	2	5	0.1	sigmoid	-25.4692	-18.7351	-24.1477	-22.784	2.91341	135
0.1	3	5	0.1	sigmoid	-25.4692	-18.7351	-24.1477	-22.784	2.91341	135
0.1	1	10	0.1	poly	-25.2216	-18.972	-24.4819	-22.8918	2.788134	138
0.1	1	10	auto	sigmoid	-25.808	-18.9312	-24.5197	-23.0863	2.984807	139

Table A.4: Grid Search Results for Model A_SVR (Cont'd)

param_SVM__C	param_SVM__degree	param_SVM__epsilon	param_SVM__gamma	param_SVM__kernel	split0_test_score	split1_test_score	split2_test_score	mean_test_score	std_test_score	rank_test_score
0.1	2	10	auto	sigmoid	-25.808	-18.9312	-24.5197	-23.0863	2.984807	139
0.1	3	10	auto	sigmoid	-25.808	-18.9312	-24.5197	-23.0863	2.984807	139
0.1	1	10	0.1	sigmoid	-25.6984	-19.0312	-24.8697	-23.1998	2.967007	142
0.1	2	10	0.1	sigmoid	-25.6984	-19.0312	-24.8697	-23.1998	2.967007	142
0.1	3	10	0.1	sigmoid	-25.6984	-19.0312	-24.8697	-23.1998	2.967007	142
0.1	3	1	auto	poly	-26.3781	-21.6485	-31.238	-26.4215	3.915021	145
0.1	3	5	auto	poly	-26.4037	-21.8026	-31.3831	-26.5298	3.912227	146
0.1	3	10	auto	poly	-26.5492	-21.9386	-31.4954	-26.661	3.902338	147
0.1	3	5	0.1	poly	-27.0737	-22.4761	-33.5107	-27.6868	4.525666	148
0.1	3	10	0.1	poly	-27.1454	-22.5667	-33.4529	-27.7217	4.462905	149
0.1	3	1	0.1	poly	-27.0584	-22.4772	-33.6296	-27.7217	4.577075	150
0.1	1	10	0.1	rbf	-30.7622	-19.909	-32.5828	-27.7513	5.594944	151
0.1	2	10	0.1	rbf	-30.7622	-19.909	-32.5828	-27.7513	5.594944	151
0.1	3	10	0.1	rbf	-30.7622	-19.909	-32.5828	-27.7513	5.594944	151
0.1	1	5	0.1	rbf	-31.1307	-20.0517	-32.6455	-27.9427	5.613883	154
0.1	2	5	0.1	rbf	-31.1307	-20.0517	-32.6455	-27.9427	5.613883	154
0.1	3	5	0.1	rbf	-31.1307	-20.0517	-32.6455	-27.9427	5.613883	154
0.1	1	1	0.1	rbf	-31.055	-20.1544	-32.8552	-28.0215	5.611217	157
0.1	2	1	0.1	rbf	-31.055	-20.1544	-32.8552	-28.0215	5.611217	157
0.1	3	1	0.1	rbf	-31.055	-20.1544	-32.8552	-28.0215	5.611217	157
0.1	1	10	auto	rbf	-31.14	-20.0825	-33.1778	-28.1334	5.75331	160
0.1	2	10	auto	rbf	-31.14	-20.0825	-33.1778	-28.1334	5.75331	160
0.1	3	10	auto	rbf	-31.14	-20.0825	-33.1778	-28.1334	5.75331	160
0.1	1	5	auto	rbf	-31.5074	-20.3118	-33.2867	-28.3686	5.743198	163
0.1	2	5	auto	rbf	-31.5074	-20.3118	-33.2867	-28.3686	5.743198	163
0.1	3	5	auto	rbf	-31.5074	-20.3118	-33.2867	-28.3686	5.743198	163
0.1	1	1	auto	rbf	-31.4284	-20.3705	-33.4917	-28.4302	5.760972	166
0.1	2	1	auto	rbf	-31.4284	-20.3705	-33.4917	-28.4302	5.760972	166
0.1	3	1	auto	rbf	-31.4284	-20.3705	-33.4917	-28.4302	5.760972	166
100	2	10	0.1	poly	-37.1204	-23.3465	-42.6852	-34.384	8.128617	169
100	2	10	auto	poly	-37.13	-23.3772	-42.7061	-34.4044	8.122931	170
100	2	5	0.1	poly	-37.2066	-23.3084	-42.9974	-34.5041	8.262045	171
100	2	5	auto	poly	-37.2253	-23.3109	-42.9917	-34.5093	8.260984	172
100	2	1	0.1	poly	-37.6522	-23.4966	-42.7348	-34.6278	8.139921	173
100	2	1	auto	poly	-37.6509	-23.5091	-42.7381	-34.6327	8.135134	174
10	2	10	auto	poly	-37.0971	-23.5084	-43.4909	-34.6988	8.332216	175
10	2	10	0.1	poly	-37.0937	-23.4963	-43.7943	-34.7948	8.444581	176
10	2	5	auto	poly	-37.3762	-23.3551	-43.6923	-34.8079	8.498958	177
10	2	5	0.1	poly	-37.4498	-23.4244	-43.8272	-34.9005	8.522244	178
10	2	1	auto	poly	-37.7096	-23.5503	-43.5631	-34.941	8.401456	179
5	2	10	auto	poly	-37.0664	-23.4952	-44.4838	-35.0151	8.690462	180
10	2	1	0.1	poly	-37.6635	-23.5502	-43.8883	-35.034	8.508645	181
5	2	5	auto	poly	-37.5366	-23.5019	-44.3525	-35.1303	8.680583	182
5	2	10	0.1	poly	-37.1514	-23.5686	-44.9812	-35.2337	8.846203	183
5	2	5	0.1	poly	-37.6842	-23.5381	-44.548	-35.2567	8.747299	184
5	2	1	auto	poly	-37.8469	-23.5757	-44.3631	-35.2619	8.681056	185
5	2	1	0.1	poly	-37.9233	-23.6401	-44.7489	-35.4374	8.79506	186
0.1	2	5	0.1	poly	-37.5926	-25.1262	-45.7577	-36.1588	8.483583	187
0.1	2	1	0.1	poly	-37.7754	-25.0733	-45.6469	-36.1652	8.475973	188
0.1	2	5	auto	poly	-37.82	-24.8361	-45.9278	-36.1946	8.68701	189
0.1	2	1	auto	poly	-37.9385	-24.8592	-45.9706	-36.2561	8.70042	190
0.1	2	10	0.1	poly	-37.9479	-25.4644	-46.0696	-36.494	8.474607	191
0.1	2	10	auto	poly	-37.9162	-25.2227	-46.487	-36.542	8.735345	192
1	2	10	auto	poly	-38.4352	-24.1504	-48.1375	-36.9077	9.852106	193
1	2	5	auto	poly	-38.8889	-24.1786	-47.9834	-37.017	9.807986	194
1	2	1	auto	poly	-39.5066	-24.1713	-47.8176	-37.1652	9.794534	195
1	2	10	0.1	poly	-38.789	-24.3136	-48.5855	-37.2294	9.970156	196
1	2	5	0.1	poly	-39.2235	-24.2506	-48.3164	-37.2635	9.922096	197
1	2	1	0.1	poly	-39.7294	-24.2162	-48.1266	-37.3574	9.904435	198
1	1	1	0.1	sigmoid	-61.4807	-28.6859	-38.7715	-42.9794	13.71504	199
1	2	1	0.1	sigmoid	-61.4807	-28.6859	-38.7715	-42.9794	13.71504	199
1	3	1	0.1	sigmoid	-61.4807	-28.6859	-38.7715	-42.9794	13.71504	199
1	1	10	0.1	sigmoid	-60.0259	-31.4906	-45.6464	-45.721	11.64961	202
1	2	10	0.1	sigmoid	-60.0259	-31.4906	-45.6464	-45.721	11.64961	202
1	3	10	0.1	sigmoid	-60.0259	-31.4906	-45.6464	-45.721	11.64961	202
1	1	5	0.1	sigmoid	-60.8878	-32.5561	-46.522	-46.6553	11.56676	205
1	2	5	0.1	sigmoid	-60.8878	-32.5561	-46.522	-46.6553	11.56676	205
1	3	5	0.1	sigmoid	-60.8878	-32.5561	-46.522	-46.6553	11.56676	205
1	1	5	auto	sigmoid	-42.9758	-42.4618	-61.1976	-48.8784	8.713497	208
1	2	5	auto	sigmoid	-42.9758	-42.4618	-61.1976	-48.8784	8.713497	208

Table A.4: Grid Search Results for Model A_SVR (Cont'd)

param_SVM__C	param_SVM__degree	param_SVM__epsilon	param_SVM__gamma	param_SVM__kernel	split0_test_score	split1_test_score	split2_test_score	mean_test_score	std_test_score	rank_test_score
1	3	5	auto	sigmoid	-42.9758	-42.4618	-61.1976	-48.8784	8.713497	208
1	1	1	auto	sigmoid	-43.8641	-42.4638	-61.0501	-49.126	8.450969	211
1	2	1	auto	sigmoid	-43.8641	-42.4638	-61.0501	-49.126	8.450969	211
1	3	1	auto	sigmoid	-43.8641	-42.4638	-61.0501	-49.126	8.450969	211
1	1	10	auto	sigmoid	-44.8216	-42.1011	-61.0439	-49.3222	8.362554	214
1	2	10	auto	sigmoid	-44.8216	-42.1011	-61.0439	-49.3222	8.362554	214
1	3	10	auto	sigmoid	-44.8216	-42.1011	-61.0439	-49.3222	8.362554	214
5	1	5	0.1	sigmoid	-322.89	-153.36	-224.708	-233.653	69.49898	217
5	2	5	0.1	sigmoid	-322.89	-153.36	-224.708	-233.653	69.49898	217
5	3	5	0.1	sigmoid	-322.89	-153.36	-224.708	-233.653	69.49898	217
5	1	1	0.1	sigmoid	-323.1	-153.588	-225.338	-234.009	69.47432	220
5	2	1	0.1	sigmoid	-323.1	-153.588	-225.338	-234.009	69.47432	220
5	3	1	0.1	sigmoid	-323.1	-153.588	-225.338	-234.009	69.47432	220
5	1	10	0.1	sigmoid	-325.381	-153.241	-224.427	-234.35	70.62518	223
5	2	10	0.1	sigmoid	-325.381	-153.241	-224.427	-234.35	70.62518	223
5	3	10	0.1	sigmoid	-325.381	-153.241	-224.427	-234.35	70.62518	223
5	1	5	auto	sigmoid	-404.81	-195.002	-281.068	-293.627	86.11278	226
5	2	5	auto	sigmoid	-404.81	-195.002	-281.068	-293.627	86.11278	226
5	3	5	auto	sigmoid	-404.81	-195.002	-281.068	-293.627	86.11278	226
5	1	1	auto	sigmoid	-405.235	-195.374	-280.42	-293.676	86.18649	229
5	2	1	auto	sigmoid	-405.235	-195.374	-280.42	-293.676	86.18649	229
5	3	1	auto	sigmoid	-405.235	-195.374	-280.42	-293.676	86.18649	229
5	1	10	auto	sigmoid	-407.414	-195.023	-281.515	-294.651	87.20426	232
5	2	10	auto	sigmoid	-407.414	-195.023	-281.515	-294.651	87.20426	232
5	3	10	auto	sigmoid	-407.414	-195.023	-281.515	-294.651	87.20426	232
10	1	10	0.1	sigmoid	-668.173	-316.587	-454.489	-479.75	144.6416	235
10	2	10	0.1	sigmoid	-668.173	-316.587	-454.489	-479.75	144.6416	235
10	3	10	0.1	sigmoid	-668.173	-316.587	-454.489	-479.75	144.6416	235
10	1	5	0.1	sigmoid	-670.155	-316.466	-454.827	-480.483	145.5281	238
10	2	5	0.1	sigmoid	-670.155	-316.466	-454.827	-480.483	145.5281	238
10	3	5	0.1	sigmoid	-670.155	-316.466	-454.827	-480.483	145.5281	238
10	1	1	0.1	sigmoid	-669.919	-316.485	-457.131	-481.178	145.2871	241
10	2	1	0.1	sigmoid	-669.919	-316.485	-457.131	-481.178	145.2871	241
10	3	1	0.1	sigmoid	-669.919	-316.485	-457.131	-481.178	145.2871	241
10	1	1	auto	sigmoid	-801.502	-402.698	-573.088	-592.429	163.3847	244
10	2	1	auto	sigmoid	-801.502	-402.698	-573.088	-592.429	163.3847	244
10	3	1	auto	sigmoid	-801.502	-402.698	-573.088	-592.429	163.3847	244
10	1	10	auto	sigmoid	-827.379	-401.808	-572.746	-600.644	174.855	247
10	2	10	auto	sigmoid	-827.379	-401.808	-572.746	-600.644	174.855	247
10	3	10	auto	sigmoid	-827.379	-401.808	-572.746	-600.644	174.855	247
10	1	5	auto	sigmoid	-827.278	-402.428	-572.363	-600.69	174.5971	250
10	2	5	auto	sigmoid	-827.278	-402.428	-572.363	-600.69	174.5971	250
10	3	5	auto	sigmoid	-827.278	-402.428	-572.363	-600.69	174.5971	250
100	1	10	0.1	sigmoid	-6121.93	-3272.58	-4605.94	-4666.82	1164.037	253
100	2	10	0.1	sigmoid	-6121.93	-3272.58	-4605.94	-4666.82	1164.037	253
100	3	10	0.1	sigmoid	-6121.93	-3272.58	-4605.94	-4666.82	1164.037	253
100	1	1	0.1	sigmoid	-6303.96	-3265.54	-4610.35	-4726.62	1243.151	256
100	2	1	0.1	sigmoid	-6303.96	-3265.54	-4610.35	-4726.62	1243.151	256
100	3	1	0.1	sigmoid	-6303.96	-3265.54	-4610.35	-4726.62	1243.151	256
100	1	5	0.1	sigmoid	-6161.03	-4014.64	-5897.94	-5357.87	955.8641	259
100	2	5	0.1	sigmoid	-6161.03	-4014.64	-5897.94	-5357.87	955.8641	259
100	3	5	0.1	sigmoid	-6161.03	-4014.64	-5897.94	-5357.87	955.8641	259
100	1	1	auto	sigmoid	-7737.26	-4142.39	-5798.87	-5892.84	1469.105	262
100	2	1	auto	sigmoid	-7737.26	-4142.39	-5798.87	-5892.84	1469.105	262
100	3	1	auto	sigmoid	-7737.26	-4142.39	-5798.87	-5892.84	1469.105	262
100	1	10	auto	sigmoid	-7753.86	-4138.74	-5806.18	-5899.6	1477.344	265
100	2	10	auto	sigmoid	-7753.86	-4138.74	-5806.18	-5899.6	1477.344	265
100	3	10	auto	sigmoid	-7753.86	-4138.74	-5806.18	-5899.6	1477.344	265
100	1	5	auto	sigmoid	-7755.65	-4140.57	-5802.68	-5899.64	1477.441	268
100	2	5	auto	sigmoid	-7755.65	-4140.57	-5802.68	-5899.64	1477.441	268
100	3	5	auto	sigmoid	-7755.65	-4140.57	-5802.68	-5899.64	1477.441	268

Table A.5: Grid Search Results for Model B_SVR

param_SVM__C	param_SVM__degree	param_SVM__epsilon	param_SVM__gamma	param_SVM__kernel	split0_test_score	split1_test_score	split2_test_score	mean_test_score	std_test_score	rank_test_score
100	1	1	0.1	poly	-21.0582	-15.0931	-13.1061	-16.4192	3.379144	1
100	1	1	auto	poly	-21.0585	-15.0938	-13.1079	-16.4201	3.378569	2
10	1	1	0.1	poly	-21.0632	-15.0891	-13.1296	-16.4273	3.374281	3
10	1	1	auto	poly	-21.0669	-15.0844	-13.1378	-16.4297	3.373924	4
5	1	1	0.1	poly	-21.0705	-15.0779	-13.1499	-16.4327	3.372522	5
5	1	1	auto	poly	-21.0783	-15.07	-13.1706	-16.4396	3.37047	6
1	1	1	0.1	poly	-21.1108	-15.0719	-13.2853	-16.4893	3.348266	7
1	1	1	auto	poly	-21.1371	-15.0771	-13.3767	-16.5303	3.330669	8
100	1	5	0.1	poly	-21.342	-15.4523	-13.6709	-16.8217	3.277983	9
100	1	5	auto	poly	-21.3433	-15.4522	-13.6757	-16.8238	3.27709	10
10	1	5	0.1	poly	-21.3462	-15.4543	-13.7048	-16.8351	3.268827	11
10	1	5	auto	poly	-21.3387	-15.4575	-13.712	-16.8361	3.26263	12
5	1	5	0.1	poly	-21.3416	-15.4565	-13.7245	-16.8409	3.260091	13
5	1	5	auto	poly	-21.3427	-15.4599	-13.7441	-16.8489	3.253901	14
5	1	1	auto	rbf	-20.6498	-14.2595	-15.8138	-16.9077	2.721088	15
5	2	1	auto	rbf	-20.6498	-14.2595	-15.8138	-16.9077	2.721088	15
5	3	1	auto	rbf	-20.6498	-14.2595	-15.8138	-16.9077	2.721088	15
1	1	5	0.1	poly	-21.3795	-15.4936	-13.9365	-16.9365	3.205304	18
1	1	5	auto	poly	-21.3949	-15.5048	-14.001	-16.9669	3.190702	19
5	1	5	auto	rbf	-20.9992	-14.4048	-15.9997	-17.1346	2.809173	20
5	2	5	auto	rbf	-20.9992	-14.4048	-15.9997	-17.1346	2.809173	20
5	3	5	auto	rbf	-20.9992	-14.4048	-15.9997	-17.1346	2.809173	20
10	1	1	auto	rbf	-21.3387	-14.4573	-15.9225	-17.2395	2.959631	23
10	2	1	auto	rbf	-21.3387	-14.4573	-15.9225	-17.2395	2.959631	23
10	3	1	auto	rbf	-21.3387	-14.4573	-15.9225	-17.2395	2.959631	23
100	1	10	auto	poly	-21.8023	-16.0187	-14.262	-17.361	3.221338	26
100	1	10	0.1	poly	-21.8061	-16.0195	-14.2613	-17.3623	3.223181	27
10	1	10	0.1	poly	-21.8077	-16.0101	-14.2828	-17.3669	3.218343	28
10	1	10	auto	poly	-21.805	-16.0147	-14.2928	-17.3708	3.213274	29
10	3	1	auto	poly	-22.9573	-16.5589	-12.6119	-17.376	4.262858	30
5	1	10	0.1	poly	-21.8035	-16.0167	-14.3118	-17.3773	3.206246	31
5	1	10	auto	poly	-21.8043	-16.0348	-14.3541	-17.3977	3.190538	32
10	1	5	auto	rbf	-21.6173	-14.5132	-16.208	-17.4462	3.029486	33
10	2	5	auto	rbf	-21.6173	-14.5132	-16.208	-17.4462	3.029486	33
10	3	5	auto	rbf	-21.6173	-14.5132	-16.208	-17.4462	3.029486	33
5	3	1	0.1	poly	-23.4266	-16.4875	-12.4787	-17.4643	4.522489	36
5	1	10	auto	rbf	-21.1891	-14.6964	-16.5479	-17.4778	2.730963	37
5	2	10	auto	rbf	-21.1891	-14.6964	-16.5479	-17.4778	2.730963	37
5	3	10	auto	rbf	-21.1891	-14.6964	-16.5479	-17.4778	2.730963	37
1	1	10	0.1	poly	-21.8086	-16.0825	-14.6914	-17.5275	3.079991	40
5	3	1	auto	poly	-22.0196	-16.8167	-13.7689	-17.5351	3.406436	41
5	1	1	0.1	rbf	-21.3465	-14.387	-16.9535	-17.5623	2.873634	42
5	2	1	0.1	rbf	-21.3465	-14.387	-16.9535	-17.5623	2.873634	42
5	3	1	0.1	rbf	-21.3465	-14.387	-16.9535	-17.5623	2.873634	42
0.1	1	1	0.1	poly	-21.4126	-15.6472	-15.6454	-17.5684	2.71826	45
1	1	10	auto	poly	-21.7992	-16.0938	-14.825	-17.5727	3.033162	46
10	3	1	0.1	poly	-24.2219	-16.5036	-12.2719	-17.6658	4.947305	47
10	1	10	auto	rbf	-21.5886	-14.9168	-16.8794	-17.7949	2.79965	48
10	2	10	auto	rbf	-21.5886	-14.9168	-16.8794	-17.7949	2.79965	48
10	3	10	auto	rbf	-21.5886	-14.9168	-16.8794	-17.7949	2.79965	48
5	3	5	auto	poly	-22.5464	-17.0467	-13.8863	-17.8265	3.578198	51
5	3	5	0.1	poly	-24.0717	-16.6893	-12.7971	-17.8527	4.675774	52
10	3	5	auto	poly	-23.8086	-16.7428	-13.0567	-17.8694	4.461133	53
5	1	5	0.1	rbf	-21.7249	-14.573	-17.3206	-17.8728	2.945745	54
5	2	5	0.1	rbf	-21.7249	-14.573	-17.3206	-17.8728	2.945745	54
5	3	5	0.1	rbf	-21.7249	-14.573	-17.3206	-17.8728	2.945745	54
0.1	1	5	0.1	poly	-21.5493	-16.0203	-16.4113	-17.9936	2.519312	57
1	3	1	0.1	poly	-21.3888	-17.5141	-15.2067	-18.0365	2.550709	58
100	3	1	auto	poly	-25.4243	-16.5446	-12.3074	-18.0921	5.465623	59
10	1	1	0.1	rbf	-21.9998	-14.8422	-17.4997	-18.1139	2.954209	60
10	2	1	0.1	rbf	-21.9998	-14.8422	-17.4997	-18.1139	2.954209	60
10	3	1	0.1	rbf	-21.9998	-14.8422	-17.4997	-18.1139	2.954209	60
10	3	5	0.1	poly	-25.0682	-16.8688	-12.4451	-18.1274	5.229648	63
1	3	5	0.1	poly	-21.559	-17.5553	-15.2925	-18.1356	2.590977	64
10	1	5	0.1	rbf	-21.952	-14.8511	-17.6895	-18.1642	2.918284	65
10	2	5	0.1	rbf	-21.952	-14.8511	-17.6895	-18.1642	2.918284	65
10	3	5	0.1	rbf	-21.952	-14.8511	-17.6895	-18.1642	2.918284	65
0.1	1	1	auto	poly	-21.5548	-16.1401	-16.9781	-18.2243	2.379692	68
5	1	10	0.1	rbf	-21.6775	-14.821	-18.1831	-18.2272	2.799328	69

Table A.5: Grid Search Results for Model B_SVR (Cont'd)

param_SVM__C	param_SVM__degree	param_SVM__epsilon	param_SVM__gamma	param_SVM__kernel	split0_test_score	split1_test_score	split2_test_score	mean_test_score	std_test_score	rank_test_score
5	2	10	0.1	rbf	-21.6775	-14.821	-18.1831	-18.2272	2.799328	69
5	3	10	0.1	rbf	-21.6775	-14.821	-18.1831	-18.2272	2.799328	69
1	3	10	0.1	poly	-22.195	-17.5763	-15.5504	-18.4406	2.780669	72
5	3	10	auto	poly	-23.4752	-17.2715	-14.7142	-18.487	3.678466	73
100	3	1	0.1	poly	-26.3235	-16.7345	-12.4593	-18.5058	5.796965	74
10	3	10	auto	poly	-24.644	-17.2299	-13.7536	-18.5425	4.541852	75
10	1	10	0.1	rbf	-21.8989	-15.2107	-18.6458	-18.5851	2.730764	76
10	2	10	0.1	rbf	-21.8989	-15.2107	-18.6458	-18.5851	2.730764	76
10	3	10	0.1	rbf	-21.8989	-15.2107	-18.6458	-18.5851	2.730764	76
5	3	10	0.1	poly	-25.2702	-17.1765	-13.5435	-18.6634	4.901483	79
0.1	1	5	auto	poly	-21.6929	-16.5247	-17.7839	-18.6672	2.200409	80
100	3	5	auto	poly	-26.4651	-17.047	-12.719	-18.7437	5.738642	81
1	1	1	auto	rbf	-22.7088	-15.0763	-18.481	-18.7554	3.122014	82
1	2	1	auto	rbf	-22.7088	-15.0763	-18.481	-18.7554	3.122014	82
1	3	1	auto	rbf	-22.7088	-15.0763	-18.481	-18.7554	3.122014	82
0.1	1	10	0.1	poly	-21.8396	-16.7708	-17.7139	-18.7748	2.201099	85
1	3	1	auto	poly	-20.8735	-18.105	-17.3983	-18.7923	1.499665	86
10	3	10	0.1	poly	-26.4336	-17.1172	-12.8667	-18.8058	5.665888	87
1	3	5	auto	poly	-20.878	-18.2502	-17.8474	-18.9918	1.343806	88
1	3	10	auto	poly	-20.7436	-18.4229	-17.9453	-19.0373	1.222247	89
1	1	5	auto	rbf	-22.6471	-15.328	-19.1681	-19.0477	2.989223	90
1	2	5	auto	rbf	-22.6471	-15.328	-19.1681	-19.0477	2.989223	90
1	3	5	auto	rbf	-22.6471	-15.328	-19.1681	-19.0477	2.989223	90
100	3	10	auto	poly	-27.2161	-17.161	-12.9073	-19.0948	5.999468	93
100	1	1	auto	rbf	-22.5307	-16.067	-18.7977	-19.1318	2.649331	94
100	2	1	auto	rbf	-22.5307	-16.067	-18.7977	-19.1318	2.649331	94
100	3	1	auto	rbf	-22.5307	-16.067	-18.7977	-19.1318	2.649331	94
100	3	5	0.1	poly	-27.3497	-17.2387	-12.8784	-19.1556	6.061364	97
0.1	1	1	auto	sigmoid	-22.6045	-16.4518	-18.5765	-19.2109	2.551592	98
0.1	2	1	auto	sigmoid	-22.6045	-16.4518	-18.5765	-19.2109	2.551592	98
0.1	3	1	auto	sigmoid	-22.6045	-16.4518	-18.5765	-19.2109	2.551592	98
1	1	10	auto	rbf	-22.5102	-15.371	-19.9081	-19.2631	2.950021	101
1	2	10	auto	rbf	-22.5102	-15.371	-19.9081	-19.2631	2.950021	101
1	3	10	auto	rbf	-22.5102	-15.371	-19.9081	-19.2631	2.950021	101
100	1	5	auto	rbf	-22.7497	-16.2478	-19.0974	-19.3649	2.661115	104
100	2	5	auto	rbf	-22.7497	-16.2478	-19.0974	-19.3649	2.661115	104
100	3	5	auto	rbf	-22.7497	-16.2478	-19.0974	-19.3649	2.661115	104
100	3	10	0.1	poly	-27.8904	-17.1942	-13.1706	-19.4184	6.211757	107
100	1	10	auto	rbf	-22.4663	-16.4791	-19.3568	-19.4341	2.444878	108
100	2	10	auto	rbf	-22.4663	-16.4791	-19.3568	-19.4341	2.444878	108
100	3	10	auto	rbf	-22.4663	-16.4791	-19.3568	-19.4341	2.444878	108
0.1	1	10	auto	poly	-21.9538	-17.1707	-19.1918	-19.4388	1.960469	111
0.1	1	5	auto	sigmoid	-22.7465	-16.8604	-19.1902	-19.5991	2.420311	112
0.1	2	5	auto	sigmoid	-22.7465	-16.8604	-19.1902	-19.5991	2.420311	112
0.1	3	5	auto	sigmoid	-22.7465	-16.8604	-19.1902	-19.5991	2.420311	112
0.1	1	1	0.1	sigmoid	-23.5662	-16.636	-18.9497	-19.7173	2.880829	115
0.1	2	1	0.1	sigmoid	-23.5662	-16.636	-18.9497	-19.7173	2.880829	115
0.1	3	1	0.1	sigmoid	-23.5662	-16.636	-18.9497	-19.7173	2.880829	115
1	1	1	0.1	rbf	-23.5792	-15.5075	-20.1396	-19.7421	3.307237	118
1	2	1	0.1	rbf	-23.5792	-15.5075	-20.1396	-19.7421	3.307237	118
1	3	1	0.1	rbf	-23.5792	-15.5075	-20.1396	-19.7421	3.307237	118
0.1	1	5	0.1	sigmoid	-23.6654	-16.8635	-19.1649	-19.8979	2.824813	121
0.1	2	5	0.1	sigmoid	-23.6654	-16.8635	-19.1649	-19.8979	2.824813	121
0.1	3	5	0.1	sigmoid	-23.6654	-16.8635	-19.1649	-19.8979	2.824813	121
1	1	5	0.1	rbf	-23.4892	-15.6399	-20.8893	-20.0061	3.264735	124
1	2	5	0.1	rbf	-23.4892	-15.6399	-20.8893	-20.0061	3.264735	124
1	3	5	0.1	rbf	-23.4892	-15.6399	-20.8893	-20.0061	3.264735	124
0.1	1	10	auto	sigmoid	-22.9713	-17.3961	-20.441	-20.2695	2.279279	127
0.1	2	10	auto	sigmoid	-22.9713	-17.3961	-20.441	-20.2695	2.279279	127
0.1	3	10	auto	sigmoid	-22.9713	-17.3961	-20.441	-20.2695	2.279279	127
0.1	1	10	0.1	sigmoid	-23.7414	-17.2486	-19.8921	-20.294	2.665841	130
0.1	2	10	0.1	sigmoid	-23.7414	-17.2486	-19.8921	-20.294	2.665841	130
0.1	3	10	0.1	sigmoid	-23.7414	-17.2486	-19.8921	-20.294	2.665841	130
1	1	10	0.1	rbf	-23.5635	-15.7059	-21.7475	-20.339	3.358922	133
1	2	10	0.1	rbf	-23.5635	-15.7059	-21.7475	-20.339	3.358922	133
1	3	10	0.1	rbf	-23.5635	-15.7059	-21.7475	-20.339	3.358922	133
100	1	5	0.1	rbf	-22.8264	-17.1191	-21.4398	-20.4618	2.430461	136
100	2	5	0.1	rbf	-22.8264	-17.1191	-21.4398	-20.4618	2.430461	136
100	3	5	0.1	rbf	-22.8264	-17.1191	-21.4398	-20.4618	2.430461	136
100	1	1	0.1	rbf	-23.1314	-17.0166	-21.2596	-20.4692	2.558144	139

Table A.5: Grid Search Results for Model B_SVR (Cont'd)

param_SVM__C	param_SVM__degree	param_SVM__epsilon	param_SVM__gamma	param_SVM__kernel	split0_test_score	split1_test_score	split2_test_score	mean_test_score	std_test_score	rank_test_score
100	2	1	0.1	rbf	-23.1314	-17.0166	-21.2596	-20.4692	2.558144	139
100	3	1	0.1	rbf	-23.1314	-17.0166	-21.2596	-20.4692	2.558144	139
100	1	10	0.1	rbf	-22.9794	-17.1629	-21.5272	-20.5565	2.471781	142
100	2	10	0.1	rbf	-22.9794	-17.1629	-21.5272	-20.5565	2.471781	142
100	3	10	0.1	rbf	-22.9794	-17.1629	-21.5272	-20.5565	2.471781	142
0.1	3	1	0.1	poly	-21.2987	-19.5648	-20.9853	-20.6163	0.754398	145
0.1	3	5	0.1	poly	-21.2385	-19.6402	-21.2961	-20.7249	0.767395	146
0.1	3	10	0.1	poly	-21.2018	-19.9342	-21.7245	-20.9535	0.751701	147
0.1	3	1	auto	poly	-21.2447	-22.0711	-27.9225	-23.7461	2.972374	148
0.1	3	5	auto	poly	-21.3113	-22.2437	-28.0873	-23.8808	2.998706	149
0.1	3	10	auto	poly	-21.2962	-22.3074	-28.1924	-23.932	3.0407	150
0.1	1	1	auto	rbf	-27.7839	-21.6641	-33.6425	-27.6968	4.890563	151
0.1	2	1	auto	rbf	-27.7839	-21.6641	-33.6425	-27.6968	4.890563	151
0.1	3	1	auto	rbf	-27.7839	-21.6641	-33.6425	-27.6968	4.890563	151
0.1	1	5	auto	rbf	-27.882	-21.5696	-34.0706	-27.8407	5.103588	154
0.1	2	5	auto	rbf	-27.882	-21.5696	-34.0706	-27.8407	5.103588	154
0.1	3	5	auto	rbf	-27.882	-21.5696	-34.0706	-27.8407	5.103588	154
0.1	1	10	auto	rbf	-27.8283	-21.5815	-34.4893	-27.9664	5.270475	157
0.1	2	10	auto	rbf	-27.8283	-21.5815	-34.4893	-27.9664	5.270475	157
0.1	3	10	auto	rbf	-27.8283	-21.5815	-34.4893	-27.9664	5.270475	157
0.1	1	5	0.1	rbf	-29.1367	-23.4856	-37.5326	-30.0517	5.771042	160
0.1	2	5	0.1	rbf	-29.1367	-23.4856	-37.5326	-30.0517	5.771042	160
0.1	3	5	0.1	rbf	-29.1367	-23.4856	-37.5326	-30.0517	5.771042	160
0.1	1	10	0.1	rbf	-29.1086	-23.2402	-37.8704	-30.0731	6.011542	163
0.1	2	10	0.1	rbf	-29.1086	-23.2402	-37.8704	-30.0731	6.011542	163
0.1	3	10	0.1	rbf	-29.1086	-23.2402	-37.8704	-30.0731	6.011542	163
0.1	1	1	0.1	rbf	-29.0779	-23.5963	-37.6671	-30.1138	5.790902	166
0.1	2	1	0.1	rbf	-29.0779	-23.5963	-37.6671	-30.1138	5.790902	166
0.1	3	1	0.1	rbf	-29.0779	-23.5963	-37.6671	-30.1138	5.790902	166
100	2	1	0.1	poly	-32.6179	-20.4954	-46.6323	-33.2486	10.67965	169
100	2	10	0.1	poly	-32.7212	-20.254	-46.7792	-33.2515	10.83537	170
100	2	5	0.1	poly	-32.7493	-20.5338	-46.6046	-33.2959	10.65038	171
100	2	5	auto	poly	-32.8449	-20.6061	-46.4532	-33.3014	10.55696	172
100	2	10	auto	poly	-32.6927	-20.3369	-46.9075	-33.3124	10.85626	173
100	2	1	auto	poly	-32.7166	-20.7035	-46.6903	-33.3701	10.61915	174
10	2	5	0.1	poly	-32.9779	-21.2767	-47.0764	-33.777	10.54783	175
10	2	1	0.1	poly	-32.8228	-21.3744	-47.3686	-33.8553	10.63717	176
10	2	10	0.1	poly	-32.9169	-21.1818	-47.7577	-33.9521	10.87423	177
10	2	5	auto	poly	-33.0238	-21.9496	-47.8266	-34.2667	10.60073	178
5	2	5	0.1	poly	-33.0207	-21.9793	-47.8695	-34.2898	10.60763	179
10	2	1	auto	poly	-33.0179	-22.0574	-47.8323	-34.3025	10.56166	180
5	2	1	0.1	poly	-33.0201	-22.0853	-47.8621	-34.3225	10.56357	181
10	2	10	auto	poly	-33.1006	-21.7825	-48.4411	-34.4414	10.92455	182
5	2	10	0.1	poly	-33.0994	-21.8145	-48.4313	-34.4484	10.90803	183
5	2	5	auto	poly	-33.1343	-22.7797	-48.8969	-34.937	10.73823	184
5	2	1	auto	poly	-33.0675	-23.1003	-49.0366	-35.0681	10.68251	185
5	2	10	auto	poly	-33.1224	-22.5849	-49.7671	-35.1581	11.19006	186
1	2	10	0.1	poly	-33.4612	-24.5411	-52.1359	-36.7127	11.49775	187
1	2	5	0.1	poly	-33.9707	-24.5444	-52.1777	-36.8976	11.46952	188
1	2	1	0.1	poly	-34.3256	-24.8685	-51.8959	-37.03	11.19838	189
1	2	10	auto	poly	-35.1491	-26.4755	-55.0957	-38.9068	11.98247	190
1	2	5	auto	poly	-35.3768	-26.6764	-55.2554	-39.1029	11.96115	191
1	2	1	auto	poly	-35.6474	-26.9019	-55.2274	-39.2589	11.84245	192
0.1	2	5	auto	poly	-35.1308	-30.2271	-54.8898	-40.0826	10.65998	193
0.1	2	1	auto	poly	-34.9467	-30.2879	-55.3183	-40.1843	10.86904	194
0.1	2	10	auto	poly	-35.3127	-30.4618	-55.0326	-40.269	10.6256	195
0.1	2	1	0.1	poly	-37.2728	-30.3981	-58.4651	-42.0453	11.94493	196
0.1	2	5	0.1	poly	-37.1566	-30.6681	-58.4202	-42.0816	11.85291	197
0.1	2	10	0.1	poly	-37.0958	-30.6059	-58.6021	-42.1013	11.96488	198
1	1	10	auto	sigmoid	-50.6192	-45.5951	-59.1947	-51.803	5.614748	199
1	2	10	auto	sigmoid	-50.6192	-45.5951	-59.1947	-51.803	5.614748	199
1	3	10	auto	sigmoid	-50.6192	-45.5951	-59.1947	-51.803	5.614748	199
1	1	5	auto	sigmoid	-51.4764	-46.0727	-59.427	-52.3254	5.484846	202
1	2	5	auto	sigmoid	-51.4764	-46.0727	-59.427	-52.3254	5.484846	202
1	3	5	auto	sigmoid	-51.4764	-46.0727	-59.427	-52.3254	5.484846	202
1	1	1	auto	sigmoid	-52.161	-46.235	-59.3181	-52.5714	5.349031	205
1	2	1	auto	sigmoid	-52.161	-46.235	-59.3181	-52.5714	5.349031	205
1	3	1	auto	sigmoid	-52.161	-46.235	-59.3181	-52.5714	5.349031	205
1	1	10	0.1	sigmoid	-103.567	-99.508	-127.888	-110.321	12.53187	208
1	2	10	0.1	sigmoid	-103.567	-99.508	-127.888	-110.321	12.53187	208

Table A.5: Grid Search Results for Model B_SVR (Cont'd)

param_SVM__C	param_SVM__degree	param_SVM__epsilon	param_SVM__gamma	param_SVM__kernel	split0_test_score	split1_test_score	split2_test_score	mean_test_score	std_test_score	rank_test_score
1	3	10	0.1	sigmoid	-103.567	-99.508	-127.888	-110.321	12.53187	208
1	1	5	0.1	sigmoid	-103.438	-99.6904	-128.272	-110.467	12.68281	211
1	2	5	0.1	sigmoid	-103.438	-99.6904	-128.272	-110.467	12.68281	211
1	3	5	0.1	sigmoid	-103.438	-99.6904	-128.272	-110.467	12.68281	211
1	1	1	0.1	sigmoid	-103.72	-99.9878	-128.601	-110.769	12.70039	214
1	2	1	0.1	sigmoid	-103.72	-99.9878	-128.601	-110.769	12.70039	214
1	3	1	0.1	sigmoid	-103.72	-99.9878	-128.601	-110.769	12.70039	214
5	1	5	auto	sigmoid	-386.93	-136.704	-358.297	-293.977	111.8213	217
5	2	5	auto	sigmoid	-386.93	-136.704	-358.297	-293.977	111.8213	217
5	3	5	auto	sigmoid	-386.93	-136.704	-358.297	-293.977	111.8213	217
5	1	1	auto	sigmoid	-386.43	-136.617	-360.652	-294.566	112.1817	220
5	2	1	auto	sigmoid	-386.43	-136.617	-360.652	-294.566	112.1817	220
5	3	1	auto	sigmoid	-386.43	-136.617	-360.652	-294.566	112.1817	220
5	1	10	auto	sigmoid	-388.235	-223.325	-359.428	-323.663	71.91747	223
5	2	10	auto	sigmoid	-388.235	-223.325	-359.428	-323.663	71.91747	223
5	3	10	auto	sigmoid	-388.235	-223.325	-359.428	-323.663	71.91747	223
5	1	10	0.1	sigmoid	-732.592	-470.543	-656.762	-619.966	110.0998	226
5	2	10	0.1	sigmoid	-732.592	-470.543	-656.762	-619.966	110.0998	226
5	3	10	0.1	sigmoid	-732.592	-470.543	-656.762	-619.966	110.0998	226
10	1	10	auto	sigmoid	-761.337	-449.889	-724.395	-645.207	138.9318	229
10	2	10	auto	sigmoid	-761.337	-449.889	-724.395	-645.207	138.9318	229
10	3	10	auto	sigmoid	-761.337	-449.889	-724.395	-645.207	138.9318	229
10	1	1	auto	sigmoid	-761.034	-450.073	-724.889	-645.332	138.855	232
10	2	1	auto	sigmoid	-761.034	-450.073	-724.889	-645.332	138.855	232
10	3	1	auto	sigmoid	-761.034	-450.073	-724.889	-645.332	138.855	232
10	1	5	auto	sigmoid	-762.034	-450.345	-725.431	-645.937	139.1091	235
10	2	5	auto	sigmoid	-762.034	-450.345	-725.431	-645.937	139.1091	235
10	3	5	auto	sigmoid	-762.034	-450.345	-725.431	-645.937	139.1091	235
5	1	1	0.1	sigmoid	-731.78	-471.659	-737.986	-647.142	124.1112	238
5	2	1	0.1	sigmoid	-731.78	-471.659	-737.986	-647.142	124.1112	238
5	3	1	0.1	sigmoid	-731.78	-471.659	-737.986	-647.142	124.1112	238
5	1	5	0.1	sigmoid	-732.779	-471.605	-738.44	-647.608	124.4742	241
5	2	5	0.1	sigmoid	-732.779	-471.605	-738.44	-647.608	124.4742	241
5	3	5	0.1	sigmoid	-732.779	-471.605	-738.44	-647.608	124.4742	241
10	1	5	0.1	sigmoid	-1463.58	-555.756	-1474.96	-1164.77	430.6608	244
10	2	5	0.1	sigmoid	-1463.58	-555.756	-1474.96	-1164.77	430.6608	244
10	3	5	0.1	sigmoid	-1463.58	-555.756	-1474.96	-1164.77	430.6608	244
10	1	1	0.1	sigmoid	-1464.36	-911.593	-1477.28	-1284.41	263.6749	247
10	2	1	0.1	sigmoid	-1464.36	-911.593	-1477.28	-1284.41	263.6749	247
10	3	1	0.1	sigmoid	-1464.36	-911.593	-1477.28	-1284.41	263.6749	247
10	1	10	0.1	sigmoid	-1465.79	-932.69	-1474.76	-1291.08	253.4462	250
10	2	10	0.1	sigmoid	-1465.79	-932.69	-1474.76	-1291.08	253.4462	250
10	3	10	0.1	sigmoid	-1465.79	-932.69	-1474.76	-1291.08	253.4462	250
100	1	5	auto	sigmoid	-7571.94	-4495.99	-7309.65	-6459.19	1392.318	253
100	2	5	auto	sigmoid	-7571.94	-4495.99	-7309.65	-6459.19	1392.318	253
100	3	5	auto	sigmoid	-7571.94	-4495.99	-7309.65	-6459.19	1392.318	253
100	1	1	auto	sigmoid	-7571.74	-4500.49	-7320.78	-6464.34	1392.423	256
100	2	1	auto	sigmoid	-7571.74	-4500.49	-7320.78	-6464.34	1392.423	256
100	3	1	auto	sigmoid	-7571.74	-4500.49	-7320.78	-6464.34	1392.423	256
100	1	10	auto	sigmoid	-7576.28	-4502.54	-7318.39	-6465.74	1392.177	259
100	2	10	auto	sigmoid	-7576.28	-4502.54	-7318.39	-6465.74	1392.177	259
100	3	10	auto	sigmoid	-7576.28	-4502.54	-7318.39	-6465.74	1392.177	259
100	1	1	0.1	sigmoid	-14699	-9273.93	-14682.8	-12885.2	2553.58	262
100	2	1	0.1	sigmoid	-14699	-9273.93	-14682.8	-12885.2	2553.58	262
100	3	1	0.1	sigmoid	-14699	-9273.93	-14682.8	-12885.2	2553.58	262
100	1	5	0.1	sigmoid	-14717.5	-9351.29	-14709.6	-12926.1	2527.791	265
100	2	5	0.1	sigmoid	-14717.5	-9351.29	-14709.6	-12926.1	2527.791	265
100	3	5	0.1	sigmoid	-14717.5	-9351.29	-14709.6	-12926.1	2527.791	265
100	1	10	0.1	sigmoid	-14692.9	-9324.18	-14799.2	-12938.8	2556.258	268
100	2	10	0.1	sigmoid	-14692.9	-9324.18	-14799.2	-12938.8	2556.258	268
100	3	10	0.1	sigmoid	-14692.9	-9324.18	-14799.2	-12938.8	2556.258	268

Table A.6: Grid Search Results for Model C_SVR

param_SVM__C	param_SVM__degree	param_SVM__epsilon	param_SVM__gamma	param_SVM__kernel	split0_test_score	split1_test_score	split2_test_score	mean_test_score	std_test_score	rank_test_score
100	1	1	auto	poly	-28.1628	-19.627	-22.6952	-23.495	3.530343	1
100	1	1	0.1	poly	-28.1632	-19.6269	-22.6971	-23.4958	3.530382	2
10	1	1	0.1	poly	-28.1521	-19.6359	-22.7238	-23.5039	3.520222	3
10	1	1	auto	poly	-28.1535	-19.6444	-22.7182	-23.5054	3.518165	4
5	1	1	0.1	poly	-28.1307	-19.6248	-22.8329	-23.5295	3.507276	5
5	1	1	auto	poly	-28.1401	-19.6325	-22.8353	-23.536	3.508405	6
1	1	1	0.1	poly	-27.9908	-19.776	-23.2545	-23.6738	3.366753	7
1	1	1	auto	poly	-28.0122	-19.7917	-23.2292	-23.6777	3.370923	8
100	1	5	auto	poly	-28.5437	-20.3132	-23.5472	-24.1347	3.385675	9
100	1	5	0.1	poly	-28.5443	-20.3133	-23.5472	-24.1349	3.385873	10
10	1	5	0.1	poly	-28.5292	-20.3395	-23.6189	-24.1625	3.365423	11
10	1	5	auto	poly	-28.5316	-20.3361	-23.621	-24.1629	3.367658	12
5	1	5	auto	poly	-28.5213	-20.3436	-23.6447	-24.1699	3.359148	13
5	1	5	0.1	poly	-28.5182	-20.3447	-23.6607	-24.1745	3.356545	14
1	1	5	auto	poly	-28.3743	-20.4144	-24.0254	-24.2713	3.254261	15
1	1	5	0.1	poly	-28.363	-20.4338	-24.1202	-24.3057	3.239743	16
0.1	1	1	auto	poly	-26.8746	-21.2103	-27.5074	-25.1974	2.831158	17
100	1	10	auto	poly	-29.1809	-21.3607	-25.1211	-25.2209	3.193349	18
100	1	10	0.1	poly	-29.183	-21.3592	-25.126	-25.2227	3.194792	19
10	1	10	0.1	poly	-29.17	-21.3714	-25.1869	-25.2428	3.184028	20
10	1	10	auto	poly	-29.1793	-21.3789	-25.1752	-25.2445	3.184882	21
5	1	10	auto	poly	-29.172	-21.3788	-25.2438	-25.2649	3.181596	22
5	1	10	0.1	poly	-29.1785	-21.3843	-25.2501	-25.271	3.182011	23
0.1	1	1	auto	sigmoid	-26.779	-21.2094	-28.1319	-25.3734	2.995791	24
0.1	2	1	auto	sigmoid	-26.779	-21.2094	-28.1319	-25.3734	2.995791	24
0.1	3	1	auto	sigmoid	-26.779	-21.2094	-28.1319	-25.3734	2.995791	24
1	1	10	auto	poly	-29.0331	-21.5634	-25.5582	-25.3849	3.051956	27
1	1	10	0.1	poly	-29.021	-21.5865	-25.5953	-25.4009	3.038215	28
0.1	1	1	0.1	poly	-26.8229	-21.3871	-28.2463	-25.4854	2.955645	29
0.1	1	5	auto	poly	-27.1188	-21.5845	-28.2838	-25.6623	2.922426	30
0.1	1	1	0.1	sigmoid	-26.6746	-21.5009	-28.884	-25.6865	3.094048	31
0.1	2	1	0.1	sigmoid	-26.6746	-21.5009	-28.884	-25.6865	3.094048	31
0.1	3	1	0.1	sigmoid	-26.6746	-21.5009	-28.884	-25.6865	3.094048	31
0.1	1	5	auto	sigmoid	-27.0036	-21.5418	-29.0744	-25.8733	3.177325	34
0.1	2	5	auto	sigmoid	-27.0036	-21.5418	-29.0744	-25.8733	3.177325	34
0.1	3	5	auto	sigmoid	-27.0036	-21.5418	-29.0744	-25.8733	3.177325	34
0.1	1	5	0.1	poly	-27.0558	-21.8077	-28.9107	-25.9247	3.008069	37
0.1	1	5	0.1	sigmoid	-26.9282	-21.8144	-29.5315	-26.0914	3.205585	38
0.1	2	5	0.1	sigmoid	-26.9282	-21.8144	-29.5315	-26.0914	3.205585	38
0.1	3	5	0.1	sigmoid	-26.9282	-21.8144	-29.5315	-26.0914	3.205585	38
0.1	1	10	auto	poly	-27.8465	-22.6342	-29.7941	-26.7583	3.022593	41
0.1	1	10	auto	sigmoid	-27.6739	-22.5637	-30.241	-26.8262	3.191019	42
0.1	2	10	auto	sigmoid	-27.6739	-22.5637	-30.241	-26.8262	3.191019	42
0.1	3	10	auto	sigmoid	-27.6739	-22.5637	-30.241	-26.8262	3.191019	42
0.1	1	10	0.1	poly	-27.747	-22.8029	-30.2103	-26.92	3.08006	45
0.1	1	10	0.1	sigmoid	-27.6264	-22.7872	-30.8191	-27.0776	3.301911	46
0.1	2	10	0.1	sigmoid	-27.6264	-22.7872	-30.8191	-27.0776	3.301911	46
0.1	3	10	0.1	sigmoid	-27.6264	-22.7872	-30.8191	-27.0776	3.301911	46
1	1	1	0.1	rbf	-29.718	-20.551	-31.8787	-27.3825	4.910518	49
1	2	1	0.1	rbf	-29.718	-20.551	-31.8787	-27.3825	4.910518	49
1	3	1	0.1	rbf	-29.718	-20.551	-31.8787	-27.3825	4.910518	49
1	3	1	0.1	poly	-31.653	-23.044	-27.7041	-27.4671	3.518591	52
1	3	1	auto	poly	-32.5696	-22.8088	-27.0256	-27.468	3.997122	53
1	1	1	auto	rbf	-29.8918	-20.729	-32.0713	-27.5641	4.914359	54
1	2	1	auto	rbf	-29.8918	-20.729	-32.0713	-27.5641	4.914359	54
1	3	1	auto	rbf	-29.8918	-20.729	-32.0713	-27.5641	4.914359	54
1	1	5	0.1	rbf	-29.9916	-20.6672	-32.0651	-27.5746	4.957122	57
1	2	5	0.1	rbf	-29.9916	-20.6672	-32.0651	-27.5746	4.957122	57
1	3	5	0.1	rbf	-29.9916	-20.6672	-32.0651	-27.5746	4.957122	57
5	1	1	0.1	rbf	-30.8976	-20.9305	-31.0661	-27.6314	4.73875	60
5	2	1	0.1	rbf	-30.8976	-20.9305	-31.0661	-27.6314	4.73875	60
5	3	1	0.1	rbf	-30.8976	-20.9305	-31.0661	-27.6314	4.73875	60
5	1	1	auto	rbf	-30.8884	-21.0185	-31.2265	-27.7111	4.734404	63
5	2	1	auto	rbf	-30.8884	-21.0185	-31.2265	-27.7111	4.734404	63
5	3	1	auto	rbf	-30.8884	-21.0185	-31.2265	-27.7111	4.734404	63
1	3	5	0.1	poly	-31.7058	-23.4829	-27.947	-27.7119	3.361104	66
5	3	1	0.1	poly	-34.8616	-22.6331	-25.6907	-27.7285	5.196079	67
1	1	5	auto	rbf	-30.113	-20.7886	-32.2853	-27.7289	4.987062	68
1	2	5	auto	rbf	-30.113	-20.7886	-32.2853	-27.7289	4.987062	68

Table A.6: Grid Search Results for Model C_SVR (Cont'd)

param_SVM__C	param_SVM__degree	param_SVM__epsilon	param_SVM__gamma	param_SVM__kernel	split0_test_score	split1_test_score	split2_test_score	mean_test_score	std_test_score	rank_test_score
1	3	5	auto	rbf	-30.113	-20.7886	-32.2853	-27.7289	4.987062	68
5	3	1	auto	poly	-35.3449	-22.5645	-25.3669	-27.7588	5.484842	71
1	3	5	auto	poly	-32.7178	-23.3051	-27.3172	-27.78	3.856635	72
10	3	1	0.1	poly	-35.6358	-22.6384	-25.4308	-27.9017	5.586424	73
5	1	5	0.1	rbf	-31.2489	-20.9554	-31.7391	-27.9811	4.971955	74
5	2	5	0.1	rbf	-31.2489	-20.9554	-31.7391	-27.9811	4.971955	74
5	3	5	0.1	rbf	-31.2489	-20.9554	-31.7391	-27.9811	4.971955	74
10	3	1	auto	poly	-35.7443	-22.6795	-25.543	-27.989	5.607085	77
5	3	5	0.1	poly	-35.4803	-22.711	-26.0938	-28.095	5.401715	78
5	1	5	auto	rbf	-31.309	-21.0908	-31.9003	-28.1	4.962154	79
5	2	5	auto	rbf	-31.309	-21.0908	-31.9003	-28.1	4.962154	79
5	3	5	auto	rbf	-31.309	-21.0908	-31.9003	-28.1	4.962154	79
5	3	5	auto	poly	-35.7505	-22.8105	-25.7434	-28.1015	5.539635	82
10	1	1	0.1	rbf	-31.6018	-21.2119	-31.5668	-28.1268	4.889595	83
10	2	1	0.1	rbf	-31.6018	-21.2119	-31.5668	-28.1268	4.889595	83
10	3	1	0.1	rbf	-31.6018	-21.2119	-31.5668	-28.1268	4.889595	83
10	1	1	auto	rbf	-31.5127	-21.3724	-31.5887	-28.158	4.798196	86
10	2	1	auto	rbf	-31.5127	-21.3724	-31.5887	-28.158	4.798196	86
10	3	1	auto	rbf	-31.5127	-21.3724	-31.5887	-28.158	4.798196	86
1	1	10	0.1	rbf	-30.3643	-20.9667	-33.2574	-28.1961	5.24666	89
1	2	10	0.1	rbf	-30.3643	-20.9667	-33.2574	-28.1961	5.24666	89
1	3	10	0.1	rbf	-30.3643	-20.9667	-33.2574	-28.1961	5.24666	89
10	3	5	0.1	poly	-36.2474	-22.8867	-25.6282	-28.2541	5.761828	92
1	3	10	auto	poly	-32.861	-23.6132	-28.3598	-28.278	3.775832	93
10	1	5	0.1	rbf	-31.6969	-21.2243	-31.9501	-28.2904	4.99755	94
10	2	5	0.1	rbf	-31.6969	-21.2243	-31.9501	-28.2904	4.99755	94
10	3	5	0.1	rbf	-31.6969	-21.2243	-31.9501	-28.2904	4.99755	94
1	3	10	0.1	poly	-32.3382	-23.8612	-28.7935	-28.331	3.476143	97
1	1	10	auto	rbf	-30.4354	-21.0772	-33.5423	-28.3516	5.297909	98
1	2	10	auto	rbf	-30.4354	-21.0772	-33.5423	-28.3516	5.297909	98
1	3	10	auto	rbf	-30.4354	-21.0772	-33.5423	-28.3516	5.297909	98
5	3	10	0.1	poly	-35.4598	-23.1505	-26.5675	-28.3926	5.188305	101
10	3	5	auto	poly	-36.8087	-22.9592	-25.5756	-28.4478	6.00775	102
10	1	5	auto	rbf	-31.6714	-21.4958	-32.2232	-28.4635	4.932009	103
10	2	5	auto	rbf	-31.6714	-21.4958	-32.2232	-28.4635	4.932009	103
10	3	5	auto	rbf	-31.6714	-21.4958	-32.2232	-28.4635	4.932009	103
5	3	10	auto	poly	-35.9537	-23.0126	-26.4492	-28.4718	5.473333	106
100	3	1	auto	poly	-37.4605	-22.4882	-25.5369	-28.4952	6.460458	107
100	3	1	0.1	poly	-37.5168	-22.5226	-25.5255	-28.5216	6.477628	108
10	3	10	0.1	poly	-36.766	-22.7414	-26.306	-28.6044	5.951744	109
10	3	10	auto	poly	-36.9542	-22.7672	-26.2693	-28.6636	6.034171	110
5	1	10	0.1	rbf	-31.5108	-21.6879	-32.9142	-28.7043	4.994295	111
5	2	10	0.1	rbf	-31.5108	-21.6879	-32.9142	-28.7043	4.994295	111
5	3	10	0.1	rbf	-31.5108	-21.6879	-32.9142	-28.7043	4.994295	111
5	1	10	auto	rbf	-31.4857	-21.793	-33.0917	-28.7901	4.990966	114
5	2	10	auto	rbf	-31.4857	-21.793	-33.0917	-28.7901	4.990966	114
5	3	10	auto	rbf	-31.4857	-21.793	-33.0917	-28.7901	4.990966	114
100	3	5	auto	poly	-38.0037	-22.7955	-25.9705	-28.9232	6.550354	117
100	3	5	0.1	poly	-37.9553	-22.8748	-25.9597	-28.93	6.504979	118
10	1	10	0.1	rbf	-32.2118	-21.8241	-33.1421	-29.0594	5.130148	119
10	2	10	0.1	rbf	-32.2118	-21.8241	-33.1421	-29.0594	5.130148	119
10	3	10	0.1	rbf	-32.2118	-21.8241	-33.1421	-29.0594	5.130148	119
100	3	10	0.1	poly	-38.1663	-22.7694	-26.4394	-29.1251	6.566328	122
100	3	10	auto	poly	-38.1561	-22.7408	-26.4842	-29.127	6.564884	123
10	1	10	auto	rbf	-32.2434	-21.9294	-33.3875	-29.1868	5.152974	124
10	2	10	auto	rbf	-32.2434	-21.9294	-33.3875	-29.1868	5.152974	124
10	3	10	auto	rbf	-32.2434	-21.9294	-33.3875	-29.1868	5.152974	124
0.1	3	1	auto	poly	-27.6135	-25.7456	-36.4099	-29.923	4.649864	127
0.1	3	5	auto	poly	-27.7444	-25.9215	-36.5677	-30.0779	4.64896	128
100	1	5	0.1	rbf	-32.2279	-22.5573	-35.6476	-30.1443	5.543515	129
100	2	5	0.1	rbf	-32.2279	-22.5573	-35.6476	-30.1443	5.543515	129
100	3	5	0.1	rbf	-32.2279	-22.5573	-35.6476	-30.1443	5.543515	129
100	1	5	auto	rbf	-31.7943	-22.9109	-35.9772	-30.2275	5.448117	132
100	2	5	auto	rbf	-31.7943	-22.9109	-35.9772	-30.2275	5.448117	132
100	3	5	auto	rbf	-31.7943	-22.9109	-35.9772	-30.2275	5.448117	132
100	1	1	0.1	rbf	-32.1126	-22.7828	-35.8745	-30.2566	5.503396	135
100	2	1	0.1	rbf	-32.1126	-22.7828	-35.8745	-30.2566	5.503396	135
100	3	1	0.1	rbf	-32.1126	-22.7828	-35.8745	-30.2566	5.503396	135
0.1	3	10	auto	poly	-28.0327	-26.1639	-36.7163	-30.3043	4.59771	138
100	1	10	0.1	rbf	-32.3147	-22.6522	-36.0676	-30.3448	5.651142	139

Table A.6: Grid Search Results for Model C_SVR (Cont'd)

param_SVM__C	param_SVM__degree	param_SVM__epsilon	param_SVM__gamma	param_SVM__kernel	split0_test_score	split1_test_score	split2_test_score	mean_test_score	std_test_score	rank_test_score
100	2	10	0.1	rbf	-32.3147	-22.6522	-36.0676	-30.3448	5.651142	139
100	3	10	0.1	rbf	-32.3147	-22.6522	-36.0676	-30.3448	5.651142	139
100	1	1	auto	rbf	-31.6285	-23.1632	-36.2707	-30.3541	5.42645	142
100	2	1	auto	rbf	-31.6285	-23.1632	-36.2707	-30.3541	5.42645	142
100	3	1	auto	rbf	-31.6285	-23.1632	-36.2707	-30.3541	5.42645	142
100	1	10	auto	rbf	-32.0323	-23.0865	-36.3428	-30.4872	5.521046	145
100	2	10	auto	rbf	-32.0323	-23.0865	-36.3428	-30.4872	5.521046	145
100	3	10	auto	rbf	-32.0323	-23.0865	-36.3428	-30.4872	5.521046	145
0.1	3	1	0.1	poly	-27.7016	-26.4978	-38.7753	-30.9916	5.525845	148
0.1	3	5	0.1	poly	-27.7858	-26.6576	-39.056	-31.1665	5.597746	149
0.1	3	10	0.1	poly	-27.9704	-26.8167	-38.8876	-31.2249	5.438793	150
0.1	1	1	0.1	rbf	-29.4445	-24.5818	-39.7565	-31.2609	6.326801	151
0.1	2	1	0.1	rbf	-29.4445	-24.5818	-39.7565	-31.2609	6.326801	151
0.1	3	1	0.1	rbf	-29.4445	-24.5818	-39.7565	-31.2609	6.326801	151
0.1	1	5	0.1	rbf	-29.7802	-24.5519	-39.839	-31.3903	6.343948	154
0.1	2	5	0.1	rbf	-29.7802	-24.5519	-39.839	-31.3903	6.343948	154
0.1	3	5	0.1	rbf	-29.7802	-24.5519	-39.839	-31.3903	6.343948	154
0.1	1	1	auto	rbf	-29.6537	-24.8611	-40.4431	-31.6526	6.516482	157
0.1	2	1	auto	rbf	-29.6537	-24.8611	-40.4431	-31.6526	6.516482	157
0.1	3	1	auto	rbf	-29.6537	-24.8611	-40.4431	-31.6526	6.516482	157
0.1	1	10	0.1	rbf	-29.945	-24.9511	-40.2043	-31.7001	6.349573	160
0.1	2	10	0.1	rbf	-29.945	-24.9511	-40.2043	-31.7001	6.349573	160
0.1	3	10	0.1	rbf	-29.945	-24.9511	-40.2043	-31.7001	6.349573	160
0.1	1	5	auto	rbf	-29.9498	-24.8705	-40.4421	-31.7541	6.483852	163
0.1	2	5	auto	rbf	-29.9498	-24.8705	-40.4421	-31.7541	6.483852	163
0.1	3	5	auto	rbf	-29.9498	-24.8705	-40.4421	-31.7541	6.483852	163
0.1	1	10	auto	rbf	-30.1314	-25.0888	-40.7197	-31.98	6.5138	166
0.1	2	10	auto	rbf	-30.1314	-25.0888	-40.7197	-31.98	6.5138	166
0.1	3	10	auto	rbf	-30.1314	-25.0888	-40.7197	-31.98	6.5138	166
100	2	5	auto	poly	-34.7138	-25.63	-47.2501	-35.8647	8.863797	169
100	2	5	0.1	poly	-34.715	-25.6265	-47.2746	-35.872	8.875584	170
100	2	10	auto	poly	-34.337	-25.7943	-47.5233	-35.8849	8.938074	171
100	2	10	0.1	poly	-34.3349	-25.7961	-47.5615	-35.8975	8.954108	172
100	2	1	0.1	poly	-34.9843	-25.7044	-47.2399	-35.9762	8.819771	173
10	2	5	auto	poly	-34.633	-25.7085	-47.5962	-35.9792	8.986168	174
100	2	1	auto	poly	-34.9837	-25.7443	-47.2321	-35.9867	8.800959	175
10	2	10	auto	poly	-34.3577	-25.8317	-47.9438	-36.0444	9.105682	176
10	2	5	0.1	poly	-34.6816	-25.7193	-47.7822	-36.061	9.059778	177
10	2	10	0.1	poly	-34.3805	-25.8215	-48.0297	-36.0772	9.145499	178
5	2	10	auto	poly	-34.324	-25.8964	-48.2091	-36.1432	9.199483	179
10	2	1	auto	poly	-35.0334	-25.8988	-47.6285	-36.1869	8.908565	180
5	2	10	0.1	poly	-34.3633	-25.9259	-48.4396	-36.2429	9.286765	181
5	2	5	auto	poly	-34.7937	-25.8394	-48.1581	-36.2637	9.17066	182
10	2	1	0.1	poly	-35.0578	-25.9537	-47.7821	-36.2645	8.952155	183
5	2	5	0.1	poly	-34.7734	-25.8639	-48.1839	-36.2737	9.173664	184
5	2	1	auto	poly	-35.0788	-26.0134	-47.9974	-36.3632	9.02077	185
5	2	1	0.1	poly	-35.0322	-26.0192	-48.2742	-36.4419	9.140049	186
1	2	10	auto	poly	-34.9371	-26.679	-50.7875	-37.4679	10.00365	187
1	2	5	auto	poly	-35.4753	-26.7369	-50.2567	-37.4896	9.706984	188
0.1	2	1	0.1	poly	-33.804	-28.6468	-50.2745	-37.5751	9.223343	189
1	2	1	auto	poly	-35.573	-26.7453	-50.4884	-37.6022	9.798719	190
0.1	2	1	auto	poly	-33.9907	-28.4609	-50.6555	-37.7024	9.433326	191
0.1	2	5	0.1	poly	-33.9028	-28.851	-50.5278	-37.7605	9.260387	192
1	2	5	0.1	poly	-35.6238	-26.9839	-50.7758	-37.7945	9.833519	193
1	2	10	0.1	poly	-35.362	-26.9052	-51.1242	-37.7971	10.03618	194
1	2	1	0.1	poly	-35.7463	-26.9584	-50.8455	-37.8501	9.864679	195
0.1	2	5	auto	poly	-34.0858	-28.7196	-50.9059	-37.9038	9.451315	196
0.1	2	10	0.1	poly	-34.1016	-29.1927	-51.0319	-38.1087	9.355226	197
0.1	2	10	auto	poly	-34.2771	-28.9583	-51.2071	-38.1475	9.486379	198
1	1	10	0.1	sigmoid	-63.2803	-35.0187	-52.6194	-50.3061	11.65314	199
1	2	10	0.1	sigmoid	-63.2803	-35.0187	-52.6194	-50.3061	11.65314	199
1	3	10	0.1	sigmoid	-63.2803	-35.0187	-52.6194	-50.3061	11.65314	199
1	1	5	0.1	sigmoid	-64.096	-34.8728	-52.2076	-50.3921	11.99916	202
1	2	5	0.1	sigmoid	-64.096	-34.8728	-52.2076	-50.3921	11.99916	202
1	3	5	0.1	sigmoid	-64.096	-34.8728	-52.2076	-50.3921	11.99916	202
1	1	1	0.1	sigmoid	-64.8777	-34.6094	-52.1646	-50.5506	12.4096	205
1	2	1	0.1	sigmoid	-64.8777	-34.6094	-52.1646	-50.5506	12.4096	205
1	3	1	0.1	sigmoid	-64.8777	-34.6094	-52.1646	-50.5506	12.4096	205
1	1	10	auto	sigmoid	-77.4767	-41.7304	-63.652	-60.953	14.71761	208
1	2	10	auto	sigmoid	-77.4767	-41.7304	-63.652	-60.953	14.71761	208

Table A.6: Grid Search Results for Model C_SVR (Cont'd)

param_SVM__C	param_SVM__degree	param_SVM__epsilon	param_SVM__gamma	param_SVM__kernel	split0_test_score	split1_test_score	split2_test_score	mean_test_score	std_test_score	rank_test_score
1	3	10	auto	sigmoid	-77.4767	-41.7304	-63.652	-60.953	14.71761	208
1	1	1	auto	sigmoid	-78.041	-41.6441	-63.3768	-61.0206	14.95208	211
1	2	1	auto	sigmoid	-78.041	-41.6441	-63.3768	-61.0206	14.95208	211
1	3	1	auto	sigmoid	-78.041	-41.6441	-63.3768	-61.0206	14.95208	211
1	1	5	auto	sigmoid	-78.4135	-41.4243	-63.5757	-61.1378	15.19886	214
1	2	5	auto	sigmoid	-78.4135	-41.4243	-63.5757	-61.1378	15.19886	214
1	3	5	auto	sigmoid	-78.4135	-41.4243	-63.5757	-61.1378	15.19886	214
5	1	10	0.1	sigmoid	-321.623	-211.257	-286.281	-273.054	46.01734	217
5	2	10	0.1	sigmoid	-321.623	-211.257	-286.281	-273.054	46.01734	217
5	3	10	0.1	sigmoid	-321.623	-211.257	-286.281	-273.054	46.01734	217
5	1	5	0.1	sigmoid	-319.384	-215.261	-287.285	-273.977	43.53751	220
5	2	5	0.1	sigmoid	-319.384	-215.261	-287.285	-273.977	43.53751	220
5	3	5	0.1	sigmoid	-319.384	-215.261	-287.285	-273.977	43.53751	220
5	1	1	0.1	sigmoid	-321.632	-215.139	-289.127	-275.299	44.56143	223
5	2	1	0.1	sigmoid	-321.632	-215.139	-289.127	-275.299	44.56143	223
5	3	1	0.1	sigmoid	-321.632	-215.139	-289.127	-275.299	44.56143	223
5	1	1	auto	sigmoid	-396.196	-260.527	-364.619	-340.447	57.9637	226
5	2	1	auto	sigmoid	-396.196	-260.527	-364.619	-340.447	57.9637	226
5	3	1	auto	sigmoid	-396.196	-260.527	-364.619	-340.447	57.9637	226
5	1	5	auto	sigmoid	-394.342	-275.656	-364.055	-344.684	50.35233	229
5	2	5	auto	sigmoid	-394.342	-275.656	-364.055	-344.684	50.35233	229
5	3	5	auto	sigmoid	-394.342	-275.656	-364.055	-344.684	50.35233	229
5	1	10	auto	sigmoid	-395.585	-274.245	-364.294	-344.708	51.43647	232
5	2	10	auto	sigmoid	-395.585	-274.245	-364.294	-344.708	51.43647	232
5	3	10	auto	sigmoid	-395.585	-274.245	-364.294	-344.708	51.43647	232
10	1	1	0.1	sigmoid	-629.183	-424.51	-580.745	-544.813	87.33473	235
10	2	1	0.1	sigmoid	-629.183	-424.51	-580.745	-544.813	87.33473	235
10	3	1	0.1	sigmoid	-629.183	-424.51	-580.745	-544.813	87.33473	235
10	1	5	0.1	sigmoid	-637.203	-424.633	-579.985	-547.274	89.81109	238
10	2	5	0.1	sigmoid	-637.203	-424.633	-579.985	-547.274	89.81109	238
10	3	5	0.1	sigmoid	-637.203	-424.633	-579.985	-547.274	89.81109	238
10	1	10	0.1	sigmoid	-644.247	-424.647	-580.155	-549.683	92.20439	241
10	2	10	0.1	sigmoid	-644.247	-424.647	-580.155	-549.683	92.20439	241
10	3	10	0.1	sigmoid	-644.247	-424.647	-580.155	-549.683	92.20439	241
10	1	1	auto	sigmoid	-780.265	-516.682	-733.822	-676.923	114.8826	244
10	2	1	auto	sigmoid	-780.265	-516.682	-733.822	-676.923	114.8826	244
10	3	1	auto	sigmoid	-780.265	-516.682	-733.822	-676.923	114.8826	244
10	1	5	auto	sigmoid	-786.542	-515.641	-732.61	-678.264	117.0808	247
10	2	5	auto	sigmoid	-786.542	-515.641	-732.61	-678.264	117.0808	247
10	3	5	auto	sigmoid	-786.542	-515.641	-732.61	-678.264	117.0808	247
10	1	10	auto	sigmoid	-786.248	-516.753	-732.486	-678.496	116.4563	250
10	2	10	auto	sigmoid	-786.248	-516.753	-732.486	-678.496	116.4563	250
10	3	10	auto	sigmoid	-786.248	-516.753	-732.486	-678.496	116.4563	250
100	1	10	0.1	sigmoid	-6280.34	-4079.74	-5919.25	-5426.45	963.6058	253
100	2	10	0.1	sigmoid	-6280.34	-4079.74	-5919.25	-5426.45	963.6058	253
100	3	10	0.1	sigmoid	-6280.34	-4079.74	-5919.25	-5426.45	963.6058	253
100	1	5	0.1	sigmoid	-6212.6	-4080.24	-6116.73	-5469.86	983.3876	256
100	2	5	0.1	sigmoid	-6212.6	-4080.24	-6116.73	-5469.86	983.3876	256
100	3	5	0.1	sigmoid	-6212.6	-4080.24	-6116.73	-5469.86	983.3876	256
100	1	1	0.1	sigmoid	-6286.74	-4099.34	-6118.14	-5501.4	993.7974	259
100	2	1	0.1	sigmoid	-6286.74	-4099.34	-6118.14	-5501.4	993.7974	259
100	3	1	0.1	sigmoid	-6286.74	-4099.34	-6118.14	-5501.4	993.7974	259
100	1	10	auto	sigmoid	-7739.18	-5144.42	-5801.75	-6228.45	1101.436	262
100	2	10	auto	sigmoid	-7739.18	-5144.42	-5801.75	-6228.45	1101.436	262
100	3	10	auto	sigmoid	-7739.18	-5144.42	-5801.75	-6228.45	1101.436	262
100	1	5	auto	sigmoid	-7638.54	-5155.71	-7432.59	-6742.28	1125.021	265
100	2	5	auto	sigmoid	-7638.54	-5155.71	-7432.59	-6742.28	1125.021	265
100	3	5	auto	sigmoid	-7638.54	-5155.71	-7432.59	-6742.28	1125.021	265
100	1	1	auto	sigmoid	-8331.34	-5048.41	-7430.4	-6936.72	1384.966	268
100	2	1	auto	sigmoid	-8331.34	-5048.41	-7430.4	-6936.72	1384.966	268
100	3	1	auto	sigmoid	-8331.34	-5048.41	-7430.4	-6936.72	1384.966	268

Appendix B

B.1 Tables of Metric Analysis

B.1.1 Daily - Weekly Tables

Table B.1: Daily-Weekly Metric Values of MAE for Model B_mae

Comb	Metric	Week Max	Week Min	Day Max	Day Min	Week Mean	Day Mean	Week Error	Day Error	Week Confidence Interval		Day Confidence Interval	
1	MAE	32.915	9.706	68.184	11.470	9.684	9.656	0.956	0.573	8.728	10.640	9.083	10.229
2	MAE	36.672	13.393	69.925	10.680	9.922	9.900	1.014	0.611	8.908	10.936	9.290	10.511
1-2	MAE	35.156	12.810	69.101	12.730	9.909	9.886	0.988	0.600	8.921	10.898	9.286	10.486
1-4	MAE	32.756	10.222	62.933	10.113	9.901	9.874	0.917	0.571	8.984	10.817	9.302	10.445
1-2-4	MAE	31.834	11.883	63.421	13.442	9.933	9.899	0.949	0.591	8.984	10.882	9.309	10.490

Table B.2: Daily-Weekly Metric Values of SMAPE for Model B_mae

Comb	Metric	Week Max	Week Min	Day Max	Day Min	Week Mean	Day Mean	Week Error	Day Error	Week Confidence Interval		Day Confidence Interval	
1	SMAPE (%)	141.960	48.977	184.398	17.971	17.542	17.532	3.285	1.806	14.257	20.827	15.725	19.338
2	SMAPE (%)	145.423	47.859	189.628	21.038	17.979	17.986	3.398	1.877	14.581	21.377	16.109	19.863
4	SMAPE (%)	138.771	51.463	188.078	10.824	17.912	17.903	3.297	1.809	14.615	21.209	16.094	19.713
1-4	SMAPE (%)	140.856	51.879	188.675	16.976	17.859	17.856	3.248	1.801	14.612	21.107	16.055	19.657
1-2-4	SMAPE (%)	140.913	55.407	189.393	21.260	17.885	17.876	3.302	1.834	14.583	21.187	16.042	19.709

B.1.2 Hourly Table

Table B.3: Table Showing Confidence Values of MAE and SMAPE Values for Model B_mae

	COMBINATION																							
	1				2				4				1-2				1-4				1-2-4			
Hour	**Mean MAE**	**Mean SMAPE**	**MAE Conf**	**SMAPE Conf**	**Mean MAE**	**Mean SMAPE**	**MAE Conf**	**SMAPE Conf**	**Mean MAE**	**Mean SMAPE**	**MAE Conf**	**SMAPE Conf**	**Mean MAE**	**Mean SMAPE**	**MAE Conf**	**SMAPE Conf**	**Mean MAE**	**Mean SMAPE**	**MAE Conf**	**SMAPE Conf**	**Mean MAE**	**Mean SMAPE**	**MAE Conf**	**SMAPE Conf**
0:00:00	9.934	14.777	1.134	2.131	10.064	15.012	1.116	2.113	10.579	15.751	1.153	2.172	10.055	14.993	1.125	2.129	10.668	15.864	1.156	2.173	10.404	15.519	1.128	2.136
1:00:00	9.582	15.134	1.040	2.092	9.684	15.321	1.043	2.098	10.113	15.868	1.064	2.117	9.703	15.354	1.054	2.109	10.067	15.815	1.063	2.118	9.755	15.405	1.023	2.054
2:00:00	10.158	17.399	1.036	2.282	10.020	17.323	1.013	2.262	10.404	17.712	1.024	2.250	9.975	17.254	1.009	2.248	10.487	17.849	1.026	2.260	10.031	17.219	0.988	2.210
3:00:00	11.115	19.885	1.067	2.493	10.786	19.584	0.998	2.409	10.981	19.733	1.003	2.379	10.832	19.714	0.999	2.408	10.922	19.628	1.008	2.383	10.733	19.421	0.967	2.341
4:00:00	11.254	20.152	1.121	2.567	10.871	19.767	1.023	2.438	10.974	19.790	1.039	2.431	10.917	19.887	1.039	2.457	10.924	19.647	1.028	2.411	10.715	19.474	1.011	2.406
5:00:00	10.608	18.158	1.085	2.415	10.653	18.446	1.043	2.355	10.633	18.410	1.052	2.354	10.674	18.443	1.056	2.369	10.689	18.546	1.051	2.367	10.486	18.193	1.024	2.317
6:00:00	11.917	20.194	1.164	2.432	11.776	20.070	1.150	2.405	12.138	20.667	1.181	2.437	11.944	20.329	1.166	2.428	12.198	20.809	1.179	2.446	11.918	20.242	1.159	2.409
7:00:00	12.092	21.344	1.115	2.581	11.676	20.795	1.075	2.571	11.853	21.171	1.085	2.508	11.866	21.156	1.101	2.586	11.631	20.785	1.082	2.463	11.669	20.624	1.094	2.545
8:00:00	9.264	16.305	1.068	2.916	9.929	17.365	1.163	3.056	9.707	17.163	1.100	2.974	9.753	17.068	1.145	3.014	9.690	17.192	1.077	2.938	9.804	17.099	1.156	3.053
9:00:00	9.200	18.147	1.073	3.530	10.080	19.526	1.231	3.704	9.748	19.160	1.126	3.571	9.989	19.497	1.222	3.697	9.687	19.113	1.106	3.547	9.622	18.832	1.166	3.650
10:00:00	11.188	23.199	1.056	3.733	12.297	24.769	1.154	3.817	11.860	23.970	1.098	3.750	12.214	24.736	1.146	3.805	11.774	23.847	1.085	3.718	11.910	24.076	1.116	3.809
11:00:00	11.003	24.041	1.057	3.889	12.291	26.365	1.172	4.075	11.881	25.435	1.102	3.982	12.214	26.312	1.173	4.063	11.755	25.221	1.090	3.970	11.909	25.478	1.143	4.059
12:00:00	13.515	33.831	1.169	4.329	13.054	33.407	1.126	4.362	13.147	33.314	1.123	4.300	13.055	33.553	1.133	4.333	13.153	33.183	1.127	4.271	13.335	33.469	1.174	4.438
13:00:00	12.273	28.768	1.156	4.083	12.449	29.151	1.191	4.219	12.480	28.895	1.153	4.112	12.584	29.496	1.192	4.194	12.494	28.947	1.140	4.070	12.766	29.427	1.173	4.183
14:00:00	10.829	22.828	1.040	3.786	11.729	24.180	1.157	3.925	11.411	23.440	1.097	3.827	11.587	23.977	1.152	3.892	11.367	23.372	1.087	3.799	11.782	24.061	1.122	3.866
15:00:00	10.521	21.860	1.081	3.671	11.658	23.601	1.212	3.789	11.215	22.667	1.138	3.719	11.538	23.412	1.208	3.750	11.084	22.446	1.124	3.658	11.542	23.160	1.177	3.769
16:00:00	9.369	17.671	0.972	3.037	9.996	18.472	1.092	3.153	9.649	17.857	1.047	3.105	9.902	18.347	1.094	3.162	9.636	17.813	1.039	3.056	9.835	18.009	1.090	3.150
17:00:00	7.171	10.623	0.859	1.884	7.251	10.782	0.894	1.935	7.088	10.689	0.884	1.923	7.157	10.615	0.893	1.926	7.051	10.566	0.885	1.891	7.273	10.782	0.910	1.937
18:00:00	5.539	7.143	0.774	1.331	5.759	7.456	0.781	1.337	5.465	7.071	0.751	1.293	5.724	7.400	0.787	1.341	5.406	7.038	0.749	1.294	5.657	7.316	0.753	1.296
19:00:00	5.422	6.536	0.639	0.854	5.587	6.759	0.651	0.867	5.432	6.561	0.658	0.869	5.639	6.818	0.646	0.863	5.477	6.649	0.664	0.889	5.505	6.657	0.654	0.870
20:00:00	5.036	6.091	0.599	0.802	5.091	6.178	0.611	0.821	5.100	6.193	0.619	0.826	5.083	6.170	0.600	0.807	5.069	6.160	0.619	0.828	5.232	6.374	0.630	0.848
21:00:00	5.562	6.902	0.638	0.897	5.666	7.073	0.626	0.890	5.706	7.072	0.638	0.889	5.612	6.994	0.616	0.873	5.729	7.121	0.641	0.899	5.695	7.066	0.648	0.914
22:00:00	8.284	11.953	0.837	1.580	8.212	12.002	0.839	1.603	8.582	12.295	0.861	1.598	8.222	12.004	0.823	1.569	8.601	12.338	0.867	1.609	8.597	12.404	0.845	1.581
23:00:00	10.840	17.782	1.089	2.311	10.961	18.222	1.051	2.280	11.513	18.827	1.099	2.285	10.963	18.202	1.048	2.258	11.385	18.641	1.088	2.273	11.348	18.689	1.068	2.254
	1				2				4				1-2				1-4				1-2-4			
Hour	**Max MAE**	**Min Mae**	**Max SMAPE**	**Min SMAPE**	**Max MAE**	**Min Mae**	**Max SMAPE**	**Min SMAPE**	**Max MAE**	**Min Mae**	**Max SMAPE**	**Min SMAPE**	**Max MAE**	**Min Mae**	**Max SMAPE**	**Min SMAPE**	**Max MAE**	**Min Mae**	**Max SMAPE**	**Min SMAPE**	**Max MAE**	**Min Mae**	**Max SMAPE**	**Min SMAPE**
0:00:00	66.756	0.017	156.271	0.021	69.925	0.020	157.835	0.025	60.230	0.086	152.655	0.099	69.101	0.021	157.439	0.028	62.158	0.019	153.784	0.025	62.596	0.035	154.033	0.048
1:00:00	55.813	0.002	167.725	0.003	55.670	0.015	165.392	0.019	49.775	0.004	164.284	0.005	57.689	0.024	168.610	0.028	48.900	0.025	163.984	0.032	46.163	0.184	160.977	0.235
2:00:00	56.019	0.070	170.428	0.089	51.309	0.001	168.146	0.001	52.902	0.086	166.157	0.111	57.834	0.003	171.223	0.004	50.736	0.023	166.236	0.028	49.872	0.010	164.561	0.012
3:00:00	54.092	0.103	170.283	0.139	54.085	0.005	172.271	0.007	54.366	0.010	163.195	0.021	54.003	0.032	169.383	0.044	54.339	0.099	163.156	0.141	50.513	0.031	161.797	0.040
4:00:00	58.491	0.029	176.004	0.042	51.798	0.022	176.916	0.033	52.187	0.079	169.963	0.159	51.470	0.012	176.098	0.017	52.021	0.021	171.069	0.027	49.033	0.063	169.458	0.086
5:00:00	60.659	0.017	166.336	0.022	59.795	0.049	167.111	0.067	60.639	0.103	157.852	0.129	59.458	0.056	165.873	0.075	60.456	0.067	157.280	0.092	57.161	0.046	156.194	0.064
6:00:00	49.798	0.024	156.720	0.033	51.718	0.073	157.982	0.114	49.545	0.006	139.421	0.009	49.131	0.040	155.717	0.055	49.735	0.024	138.900	0.034	46.301	0.012	145.847	0.014
7:00:00	60.425	0.003	165.131	0.003	57.622	0.031	169.672	0.034	58.885	0.009	155.476	0.015	58.936	0.026	166.370	0.037	60.022	0.015	140.618	0.021	59.422	0.071	164.094	0.104
8:00:00	52.966	0.014	182.371	0.016	57.477	0.014	188.707	0.016	54.129	0.014	179.845	0.016	54.476	0.014	186.903	0.016	50.230	0.014	179.908	0.016	53.382	0.014	186.911	0.016
9:00:00	57.124	0.011	184.398	0.014	60.682	0.003	189.628	0.004	54.234	0.006	184.221	0.007	60.978	0.013	187.519	0.015	51.794	0.052	184.801	0.061	56.328	0.032	189.027	0.035
10:00:00	55.388	0.043	181.443	0.064	54.161	0.062	188.888	0.084	53.773	0.022	184.913	0.028	53.960	0.036	186.585	0.041	53.128	0.074	183.126	0.156	54.375	0.011	186.812	0.013
11:00:00	56.549	0.115	182.889	0.132	56.760	0.113	187.954	0.146	58.972	0.113	185.508	0.137	58.062	0.091	186.996	0.117	59.309	0.026	188.357	0.035	59.457	0.147	189.393	0.199
12:00:00	66.830	0.055	182.285	0.077	53.997	0.042	186.702	0.059	58.331	0.046	184.548	0.065	55.211	0.025	185.331	0.037	56.044	0.056	187.046	0.077	53.396	0.004	187.523	0.008
13:00:00	68.184	0.067	179.369	0.123	57.554	0.018	186.199	0.035	61.674	0.039	187.677	0.048	58.489	0.045	185.748	0.058	59.567	0.061	188.675	0.134	59.174	0.021	188.052	0.030
14:00:00	60.369	0.070	181.460	0.078	60.716	0.167	185.221	0.204	63.594	0.021	188.078	0.029	59.972	0.003	185.016	0.004	69.827	0.059	188.329	0.146	69.542	0.086	188.265	0.106
15:00:00	59.575	0.060	183.418	0.082	57.018	0.055	181.662	0.067	56.172	0.086	186.144	0.114	57.402	0.055	181.619	0.063	61.812	0.004	186.633	0.005	56.971	0.138	189.159	0.157
16:00:00	63.087	0.033	177.502	0.039	62.008	0.244	179.558	0.266	67.943	0.092	179.397	0.124	63.794	0.170	178.436	0.181	56.863	0.050	182.542	0.061	57.557	0.001	183.631	0.001
17:00:00	49.125	0.000	163.245	0.000	49.755	0.044	163.103	0.049	50.269	0.053	157.920	0.054	49.686	0.030	162.748	0.040	50.565	0.009	164.246	0.011	57.221	0.114	162.279	0.136
18:00:00	56.979	0.031	152.761	0.032	54.432	0.007	151.091	0.008	51.200	0.013	148.794	0.014	58.525	0.019	153.720	0.021	49.630	0.035	147.598	0.043	48.912	0.036	147.033	0.041
19:00:00	35.661	0.014	53.815	0.017	35.160	0.029	54.123	0.034	36.673	0.016	53.550	0.017	36.096	0.054	56.121	0.063	35.475	0.029	50.570	0.031	34.662	0.029	52.207	0.033
20:00:00	31.426	0.061	45.065	0.073	30.809	0.014	46.564	0.018	31.900	0.022	43.990	0.023	31.209	0.071	45.303	0.078	32.327	0.031	43.928	0.035	31.090	0.004	45.887	0.004
21:00:00	40.658	0.034	66.752	0.041	40.093	0.010	66.131	0.012	40.005	0.032	66.034	0.040	39.386	0.022	65.346	0.029	39.619	0.006	65.606	0.007	41.261	0.020	67.409	0.030
22:00:00	41.249	0.042	161.542	0.048	43.838	0.055	163.398	0.057	43.200	0.013	162.957	0.016	38.218	0.005	159.115	0.006	43.113	0.011	162.896	0.012	41.147	0.060	161.465	0.078
23:00:00	64.099	0.064	154.606	0.088	61.657	0.005	153.229	0.006	63.672	0.029	154.371	0.037	61.858	0.001	153.346	0.002	62.933	0.028	153.959	0.039	63.421	0.285	154.232	0.397