KADIR HAS UNIVERSITY

SCHOOL OF GRADUATE STUDIES

DEPARTMENT OF ENGINEERING AND NATURAL SCIENCES

SHORT-TERM FORECAST FOR TURKEY'S ELECTRICITY DEMAND DNN VS LSTM

BERK DEDE

MASTER'S DEGREE THESIS

ISTANBUL, MARCH, 2023

SHORT-TERM FORECAST FOR TURKEY'S ELECTRICITY DEMAND DNN VS LSTM

BERK DEDE

ADVISOR: ASSOC. PROF. GÖKHAN KİRKİL

MASTER'S DEGREE THESIS

submitted to the school of graduate studies

with the aim to meet the partial requirements required to receive a master's degree in the program of energy and sustainable development

ISTANBUL, MARCH, 2023

ACCEPTANCE AND APPROVAL

This study, SHORT TERM FORECAST OF TURKEY'S ELECTRICITY DEMAND titled, prepared by the BERK DEDE, was deemed successful with the UNANIMOUS/MAJORITY VOTING as a result of the thesis defense examination held on the 10.01.2023 and approved as a MASTER'S DEGREE THESIS by our jury.

JURY: SIGNATURE:

Prof. Dr. Volkan Ş. Ediger, Kadir Has University _____

Assistant Prof. Dr. Emre Çelebi, Yeditepe University _____

Assoc. Prof. Dr. Gokhan Kirkil, (Advisor) Kadir Has University _____

Prof. Dr. Ahmet Yücekaya, (Co-Advisor) Kadir Has University _____

I confirm that the signatures above belong to the aforementioned faculty members.

_____

Prof. Dr. Mehmet Timur AYDEMİR

Director of the School of Graduate Studies

APPROVAL DATE:….\.....\.....

NOTICE ON RESEARCH ETHICS AND PUBLISHING METHODS

I, BERK DEDE;

- hereby acknowledge, agree and undertake that this Master's Degree Thesis that I have prepared is entirely my own work and I have declared the citations from other studies in the bibliography in accordance with the rules;
- that this Master's Degree Thesis does not contain any material from any research submitted or accepted to obtain a degree or diploma at another educational institution;
- and that I commit and undertake to follow the "Kadir Has University Academic Codes of Conduct" prepared in accordance with the "Higher Education Council Codes of Conduct".

In addition, I acknowledge that any claim of irregularity that may arise in relation to this work will result in a disciplinary action in accordance with the university legislation.

BERK DEDE

_____

DATE AND SIGNATURE

Aileme…

ACKNOWLEDGEMENTS

Firstly, I want to acknowledge and give my thanks to my dear professors; Assoc. Prof. Dr. Gökhan Kirkil and Prof. Dr. Volkan Ş. Ediger. Working with them for many years gave me important experience and knowledge. The guidance and advice carried me through all the stages of writing my thesis. Many thanks to my coworkers in CENAL Elektrik Üretim A.Ş in Istanbul, Messrs. Asst. Manager Eray Ural and Manager Melike Kaya for their support.

The last words of thanks go to my family. I thank my mom and dad for their patience and encouragement. Without them, non of this would be possible.

SHORT TERM FORECAST OF TURKEY'S ELECTRICITY DEMAND DNN VS LSTM

ABSTARCT

This study aims to estimate Turkey's short-term electricity demand using artificial intelligence algorithms. Electrical systems are complex structures; therefore, many details must be considered for the prediction. Electricity demand forecasting depends on many conditions such as climate, calendar effect (holidays, day of the week, etc.), demographic data, and economic data. Turkey is a relatively large and crowded country, whose population distribution is concentrated in some regions and climatic conditions, population-weighted meteorological data were used as independent variables. Predicting the future is challenging machine learning can help us understand how systems behave by identifying and analyzing patterns in data. Two advanced artificial neural network models were deployed in this study: a deep neural network (DNN) model, and stacked (deep) long short-term memory (LSTM) model. Their outputs provided estimates of hourly electricity consumption compared with the actual data. For this comparison, mean square error (MSE), mean absolute error (MAE), and mean absolute percentage error (MAPE) metrics were used. It was observed that the DNN model predicted more accurately than the stacked LSTM model.

Key words: Short term electricity demand forecast, LSTM, deep learning, artificial intelligence, neural networks, time series

TÜRKİYE'NİN KISA VADELİ ELEKTRİK TALEP TAHMİNİ DNN VS LSTM

ÖZET

Bu çalışma yapay zeka algoritmaları kullanarak Türkiye'nin kısa vadeli elektrik talebini tahmin etmeyi amaçlamaktadır. Elektrik sistemleri karmaşık yapılardır; bu nedenle, tahmin için birçok ayrıntı dikkate alınmalıdır. Elektrik talep tahmini, iklim, takvim etkisi (tatiller, haftanın günü vb.), demografik veriler ve ekonomik veriler gibi birçok koşula bağlıdır. Türkiye, nispeten büyük ve kalabalık bir ülkedir ve nüfus dağılımı bazı bölgelerde yoğunlaşmış olup, iklim koşulları, nüfus ağırlıklı meteorolojik veriler bağımsız değişken olarak kullanılmıştır. Geleceği tahmin etmek zorlu bir iştir makine öğrenimi, verilerdeki kalıpları belirleyip analiz ederek sistemlerin nasıl davrandığını anlamamıza yardımcı olabilir. Bu çalışmada iki gelişmiş yapay sinir ağı modeli uygulandı: bir derin sinir ağı (DNN) modeli ve yığılmış (derin) uzun kısa süreli bellek (LSTM) modeli. Bu modellerin çıktıları olarak saatlik elektrik tüketim tahminleri elde edildi ve gerçek verilerle karşılaştırıldı. Bu karşılaştırma için ortalama kare hatası (MSE), ortalama mutlak hata (MAE) ve ortalama mutlak hata yüzdesi (MAPE) metrikleri kullanıldı. DNN modelinin yığın LSTM modelinden daha doğru tahmin ettiği gözlemlendi.

Anahtar kelimeler: Kısa vadeli elektrik talep tahmini, LSTM, derin öğrenme, yapay zeka, sinir ağları, zaman serileri

INDEX

ACKNOWLEDGEMENTS v

ABSTARCT vi

ÖZET vii

FIGURE LIST x

TABLE LIST xi

ABBREVIATIONS LIST xii

1. INTRODUCTION 1

2.TURKISH ELECTRICITY MARKET 9

3.METHODOLOGY 12

3.1. What Is Machine Learning 12

3.1.1. Supervised learning 12

3.1.2. Unsupervised learning 13

3.1.3. Semisupervised learning 14

3.1.4. Reinforcement learning 15

3.2. Artificial Neural Networks 16

3.3. Deep Learning 18

3.4. Hyperparameters Of Deep Neural Networks 18

3.5. Recurrent Neural Network 21

3.6. Long-Short Term Memory 22

3.7. Deep/Stack Long-Short Term Memory 24

4. CONSUMPTION FORECAST 24

4.1. Historical Hourly Consumption Data 25

4.2. Feature Selection 25

4.3. Model Implementation 30

4.3.1. Normalization 30

4.3.2. Accuracy validation 30

4.3.3. Train test split 31

4.4. Deep Neural Network Model 32

4.5. LSTM Model 35

5. RESULTS 39

6. CONCLUSION 44

6.1. Future Policy 45

REFERENCES 46

CURRICULUM VITAE 51

FIGURE LIST

Figure 2.1 Historcal Electricity Demand Of Turkey (PwC Turkey & Presidency of the Republic of Turkey, 2021) 10
Figure 2.2 Turkey's Net Energy Demand By Sectors (PwC Turkey & Presidency of the Republic of Turkey, 2021) 10
Figure 2.3 Turkey's Installed Capacity By Energy Source (PwC Turkey & Presidency of the Republic of Turkey, 2021) 11
Figure 2.4 Turkey's Electricity Generation By Source (IEA, n.d.) 11
Figure 3.2.1Illustration of a biological neuron 17
Figure 4.2.1 Data set raw data 26
Figure 4.2.2 Feature correlation heat map 28
Figure 4.2.3 OLS regression statistics of the variables 29
Figure 4.2.4 OLS regression statistics of independent the variables after dropping snowfall 29
Figure 4.3.3.1 Train And Test Split 31
Figure 4.3.3.2 Time Series Array To 3D Matrix 31
Figure 4.4.1 Python Code For DNN 35
Figure 4.5.1 Python Code For LSTM Model 38
Figure 5.1 DNN Model Train And Validation Loss 39
Figure 5.2 The Stacked LSTM Model Train And Validation Loss 40
Figure 5.3 DNN Model's Predicted and Real Consumption Graph 40
Figure 5.4 Stack LSTM Model's Predicted And Real Consumption Graph 41
Figure 5.5 DNN Model's Forecast Results For First Two Week Of May 2019 41
Figure 5.6 DNN Model's Forecast Results For First Two Week Of August 2018 42
Figure 5.7 DNN Model One Week Forecast 43

TABLE LIST

Table 4.4.1 Hidden Layers And Validation Losses 32
Table 4.4.2 Activation Functions And Validation Losses 33
Table 4.4.3 Number Of Neurons In Layers And Validation Losses 34
Table 4.5.1 Hidden Layers And Validation Losses 36
Table 4.5.2 Activation Functions And Validation Losses 36
Table 4.5.3 The Number Of Neurons In Layers And Validation Losses 37
Table 5.1 DNN and LSTM metrics 41
Table 5.2 Cross validation and EXIST's forecast metrics 42

ABBREVIATIONS LIST

2D	Two Dimentional
3D	Three Dimentional
ANN	Artificial Neural Network
API	Application Programming Interface
ARIMA	Autoregressive Integrated Moving Average
DLSTM	Deep Long Short Term Memory
DNA	Deoxyribonucleic Acid
DNN	Deep Neural Network
ELU	Exponential Linear Unit
EPSİM-NN	Expert Systems In Addition To The Artificial Neural Network
EXIST	Energy Exchange Istanbul
GDP	Gross Domestic Product
GNP	Gross National Product
GWh	Gigawatt Hours
IPPs	Independent Power Producers
KWh	Kilowatt Hours
LSTM	Long Short Term Memory
MAE	Mean Absolute Error
MAPE	Mean Absolute Percentage Error
MBE	Mean Bias Error
MENR	Ministry of Energy and Natural Resources
MSE	Mean Squared Error
NaN	Not a Number
NASA	National Aeronautics and Space Administration
OLS	Ordinary Least Squares
RNN	Recurrent Neural Network
ReLU	Rectified Linear Unit

SARIMA	Seasional Autoregressive Integrated Moving Average
SGD	Stochastic Gradient Descent
SSL	Semi Supervised Learning
SVM	Support Vector Machine
SVR	Suppor Vector Regression
TEİAŞ	Turkish Electricity Transmission Corporation
TLU	Threshold Logic Unit
TUIK	Turkish Statistical Institute
TWh	Terawatt hour
Tanh	Hyperbolic Tangent Sigmoid

1. INTRODUCTION

Since the industrial revolution, energy has become a critical position for the economies of developed and developing countries. After the industrial revolution, electrical energy became the basic need of modernized society due to the transition from steam-powered machines to electrical power machines and the increasing use of electricity. Today, the first form of energy that comes to mind when referring to energy is electricity. The fact that electrical power has become the most critical energy source for today's society necessitates that electricity should be continuous, reliable, and accessible. That fact is an obligation for energy companies and has certain challenges. Challenges faced by companies that produce, transmit and distribute energy is the necessity of producing electrical energy from other energy sources, and it should be consumed as soon as it is produced because it cannot store with today's technology.

Planning is therefore highly significant due to generating electricity from limited resources and the need for a production consumption balance. Providing continuous, reliable, and accessible energy is only possible with good planning for the future. High accurate modeling and forecasting for the future mean making investments with a high cost-performance ratio (Bulut & Başoğlu, 2017). The failure to make new investments to create demand may cause bottlenecks in the energy sector. Problems in energy supply for countries mean serious economic losses, such as was witnessed during the oil crisis in the 1970s (Sanlı, 2022).

Our electrical systems are also highly complex. Electricity is produced and transmitted simultaneously for millions of people to use (Veljanovski et al., 2020). Many companies from many fields need forecasts for the future because of this complex structure. The first ones that come to mind are energy-producing companies and energy transmission and distribution companies, which require. The biggest reason for this is the need for precise forecasting and modeling for the correct management of the electrical system. One of the most important forecasts is electricity consumption because of the increasing share of non-dispatchable renewable sources in energy production, the electricity market

structure, the optimal unit commitment, and the planning of conventional power plant operations (Hong, 2014).

Future consumption forecasts can be classified into short-term, medium-term, and long-term. Short-term forecasts usually cover a period of up to six months (Uslu et al., 2013). Long-term forecasts refer to a broad time ranging from one year to decades and are vital for planning to build new generation facilities and expand transmission lines. It also helps that utilities and governments make informed decisions regarding the electricity sector's development (Esteves et al., 2015). Short-term forecasts can be used for many purposes. Fossil-sourced power plants with production costs and hydroelectric power plants with reservoirs need to adjust their production schedules to meet demand in high consumption hours and periods. Resource management is thus crucial for these power plants. Short-term forecasts are also essential for the system operator to keep the system in balance. If the high consumption cannot be predicted by the generation and transmission-distribution companies, there will be an energy deficit; otherwise, an energy surplus occurs. In these cases, the system operator intervenes by giving instructions and ensuring balance between production and consumption. Forecasts also useful for scheduling the time of unit work programs, the production economy, providing effective price offers in the free market, and unit maintenance (Bulut & Başoğlu, 2017).

Predicting the future for a complex system is not easy. Therefore, many different prediction methods and models have been developed, but none is prophetic in practice. Every country has different climatic conditions and energy portfolios; therefore, a general model cannot be created (Bulut & Başoğlu, 2017). Meanwhile, many details influence a prediction. An electricity demand forecast depends on many conditions such as climate, the calendar effect (holidays, day of the week, etc.), demographic data, and economic data. Temperature, day type, humidity, and wind speed are some of the climate and calendar elements that substantially affect electricity consumption and power system load in the short term. The effect of air temperature on system load is particularly essential in terms of power system operational management in the short term (Veljanovski et al., 2020).

Since electricity companies aim to increase their operational profits, forecasting studies have grown in recent years. Scientific publications on load forecasting have risen from

around 100 to over 1,000 in 20 years (Veljanovski et al., 2020). Many forecasting strategies were tested by forecasters, with artificial neural networks (ANN) being one of the most popular. Many scientific studies have been carried out in Turkey. Ediger and Tatlıdil (2002) presented one of the first examples of these studies, estimating the primary energy need of Turkey using a cycle analysis, a semi-statistical method. In this study, many forecasting models were examined, and the most successful among the examined ones was hold-winters method. Then, they analyzed the total amounts of energy demand and GDP, rates of change of energy demand and GDP, energy coefficient, and the annual increase of primary energy demand data between 1950 and 1999 with the cycle analysis method. They estimated the energy need in 2010 to be around 130 million toe.

Akan and Tak (2003) did scientific research to reveal the short- and long-term relationship between Turkey's total energy consumption and consumer groups and estimate Turkey's total consumption, and consumption of different energy groups from 2001-2005. They established a multiple least squares model using electricity prices, GNP and population data, and annual electricity consumption from 1970-2000. According to the three scenarios, low, medium, and high growth, established for 2005, the estimated total consumption wass 127032 GWh, 138169 GWh, and 150102 GWh, respectively (Akan & Tak, 2010).

Hamzaçebi and Kutay (2004) set up an ANN model using electricity consumption and population information. They trained their model with the data from 1970-1990 and used the data from 1991-1998 for the validation. Their test set was the data from 1999-2002. The ANN's prediction had better results when compared with box jenkins and various other regression methods.

Turkey's net electricity consumption on a sectoral basis to 2020 was predicted by Hamzaçebi (2006) using annual electricity consumption data from 1970-2004 that was organized according to sectors and modeled with the ANN method. The total consumption for 2020 was estimated at around 500 TWh, with 49.90% residential consumption, 45.67% industrial consumption, 3.65% agricultural irrigation, and 0.76% transportation.

In their study, Ediger and Akar (2007) estimated the primary energy demand of Turkey from 2005-2020 using Autoregressive Integrated Moving Average (ARIMA) and

seasonal ARIMA (SARIMA) methods. In order to make this estimation, they used the data of consumption and various energy sources from 1950-2004. When the results of the ARIMA and SARIMA methods estimations were compared, they concluded that the ARIMA method is more consistent than SARIMA. They demonstrated the usability of these methods for energy consumption forecasts.

Bilgili (2009) tried to estimate Turkey's net electricity production using linear regression, nonlinear regression, and artificial neural network methods. In this estimation, Turkey's installed power, gross electricity generation, population, and the total number of subscribers between 1990 and 2007 were used as variables and created two different scenarios as low and high. In this estimation, Turkey's installed power, gross electricity generation, population, and the total number of subscribers between 1990 and 2007 were used as variables and created two different scenarios as low and high. As a result of the study, he estimated that consumption in 2012 was 221.07 TWh for the low scenario and 251.1 TWh for the high scenario. In this study, it has been revealed that ANNs give better results than the other two methods.

Altınay (2010) analyzed the seasonal effects of electricity demand in his study and revealed that the trend changes over time with seasonal fluctuations. Therefore, the seasonal ARIMA model was used for demand forecasting. Various models were analyzed and the SARIMA(0,1,1)x(1,1,1) model was chosen. For comparison, a second deterministic model was set in which seasonality is approached with dummy variables. Fort his mode SARIMA(1,1,0)x(2,0,1) choosed. The estimates of these two models for 2009 were then compared, which determined that the short- and long-term trends are stochastic.

Hotunoğlu and Karakaya (2011) used ANNs for demand forecasting and created three different scenarios: static, sustainable, and periodic change. The first scenario was based on stable economic growth, the second one on energy intensity, and the third one on periodic changes in economic growth. The first two scenarios are arranged according to high, medium, and low growth forecasts, and the second scenario also has sub-scenarios where energy intensity decreases and remains constant. Data from 1970-2008 were used in these scenarios. They concluded that the ANN forecasts were reliable and the estimates were below those of the MENR.

Oğcu et al. (2012), forecasted Turkey's electricity consumption using support vector machine (SVM) and artificial neural network (ANN). They tried to estimate electricity consumption from 2010 to 2011. Monthly electricity consumption was used from January 1970 to December 2011. They removed seasonality and trend on data with first differences and seasonal differences. The train data was from 1 January 1970 to 31 December 2009, and the results were tested from 2010 to 2011. According to their train and forecast results, both methods' MAPE score was lower than 5%, and support vector regression (SVR) was more accurate.

Ishik et al. (2015), used the hour, day of the week, month, Ankara temperature, Istanbul temperature, and Izmir temperature that pertains to 2012 to estimate the short-term electricity load of Turkey. In the study feed-forward neural network method was used. 70% of the data set was used as training data, 15% as validation data, and the remaining 15% as test data. When they compared the results, MAPE was 2.3%, and MAE was 600 MWh. Then they separated the data set according to the seasons and tried to estimate load with besides support vector machines (SVM). According to the seasonal estimation results, while the neural network was more successful in winter and spring, SVM was more successful in summer and fall. The data set was divided into three as a day time, evening, and nighttime MAPE scores, respectively 3.5%, 3.0%, and 3.2%.

Uslu et al. (2013) created a model for Turkey's monthly electricity demand. To create the model, monthly consumption data, industrial production index, monthly average temperature data for Istanbul and Ankara, and monthly work and holiday days were used. Present, high, and low were the three scenarios created for the forecast. The forecast for 2014 was 249,124.3 million KWh for the first scenario, 262,948.5 million KWh for the second scenario, and 242,999.1 million KWh for the third scenario (Uslu et al., 2013).

Hamzaçebi and ES (2014) estimated the annual electricity consumption of Turkey with the optimized gray modeling (1.1) method in their study. They used electricity consumption data from 1945-2010 published by the Turkish Statistical Institute (TUIK). While establishing the model, they divided the data into modeling and testing. Modeling data covers from 1945-2005, and test data covers from 2006-2010. The model was applied in direct and iterative ways, and it was revealed that the direct model gave better results.

According to the estimation, the consumption in 2015, 2020, and 2025 would be 19326.2 ktoe, 25358.6 ktoe, 30470.3 ktoe, respectively.

Günay (2016) modeled Turkey's annual gross electricity demand. The model used population, gross domestic product per capita, percentage of inflation, and average summer temperature as independent variables. A multilayer ANN model was created for prediction, and data from 1975-2006 were used. The model was tested with the data from 2007-2013, and that the results were more accurate than those estimates of the Turkey's Ministry of Energy and Natural Resources. In the estimation for the coming years, 460 TWh of consumption was forecast for 2028.

Yukseltan et al. (2017) created a model that considers the harmonics and the modulation of daily periodic changes with seasonality to predict annual, monthly, and daily electricity consumption. The model is based on sinusoidal changes using data from 2012-2014. It has a 3% MAPE for short-term forecasts, while it has a 9% MAPE for long-term forecasts. This study also tried to estimate industrial and residential consumption over total demand. For this, they took the calendar effect as a basis.

Başoğlu and Bulut (2017) created a hybrid model called EPSİM-NN by using expert systems in addition to ANN. In this study, the daily average and 24-hour demands were forecasted with two different ANNs, and then the errors were minimized with the expert system, which used recent demand trends for error minimization. In the example study, it was observed that while ANNs have an error rate of 5% or 6%, this decreases to around 2% with the expert system module. A better result was obtained when compared tothe estimates of Energy Exchange Istanbul (EXIST), which is responsible for managing and operating Turkey's power and commodities markets.

Cömert and Yıldız (2018), used time series in their study and made a parameter optimization. The data set consists of monthly energy demand data from January 2000 to December 2017. In the model established with artificial neural networks, data from 2000 to 2015 were used for training, and validation was made with five months of data. The energy demand for 2016 and 2017 has been estimated and compared with the actual values. The mean percentage error of the estimation was found to be below 1%.

In his study, Melikoğlu (2018) set up an energy consumption forecasting model for Turkey's 2023 energy targets. This model is semi-empirical, and four different annual growth scenarios are used. These are very low, low, medium, and high. While creating these scenarios, per capita electricity consumption and population data are taken as the basis. Attention was drawn to the relationship between per capita electricity consumption and economic growth, and the annual electricity demand increase percentage per capita was used to create models to establish this relationship. The author took into account the crisis years and the last data point, 2016, while calculating the annual electricity demand increase percentage per capita. The results are estimated at 327,000 GWh for the very low scenario, 362,000 GWh for the low scenario, 386,000 GWh for the medium scenario, and 429,000 GWh for the high demand scenario. It is predicted that the wind installed power may remain below the target, but the total installed power may meet the target.

Haliloğlu and Tutu (2018) tried to estimate the daily electricity demand in their study. They emphasized the importance of meteorological factors and calendar effects in their short-term forecasts, using this data as independent variables in their models. Their models defined the temperature threshold value as a variable for seasonality and the relationship between temperature and consumption. The model also included the calendar effect with the dummy variables that they defined. They used the least-squares method to set up the estimation model using 2012-2017 data and then tested data from January 2018 to April 2018. They observed that the estimation deviated from the actual value between 1.3% and 1.6%.

Türkünoğlu (2019), in his study, estimated Turkey's electricity demand for one, six and 24 hours with the LSTM method. He used temperature data and calendar effect as independent variables for his estimation, based on the temperature values of five important provinces since Turkey covers a wide area geographically and experiences different climates. Hourly consumption data from 2010-2016 was used as the dependent variable. Bank holidays, day types, and Islamic holidays are included in the model for calendar effect. As a result, the 24-hour estimation of the model gave close results compared to the TEİAŞ estimation.

Bişkin and Çiftçi (2021), long short term memory (LSTM) and gated recurrent unit (GRU) was used in their study. Root mean square error (RMSE), normalized root mean

square error (NRMSE), root mean squared logarithmic error (RMSLE), mean absolute percentage error (MAPE) used as an performance metrics. Electricity consumption from December, 31 2015 to March 2, 2021 was used. They splitted their data set as a train and test. Train data was October, 30, 2019 to October, 30 2020 and October, 31, 2020 to December, 31, 2020 was used for validation. Test data set was electricity consumption data from January, 1 2021 to March, 2 2021. According to the study results GRU model was slightly better than LSTM model.

Bodur et al. (2021), in their study, used electricity data from January 1, 2013 to December 31, 2017. They separated their data as 80% for train and 20% for test. Their input values are dates, and they have been tried to forecast power demand from input values. They used recurrent neural network (RNN) and long short-term memory (LSTM) methods. They compared the two methods' 24 hours, 48 hours, and one-year forecasts, mean absolute percentage (MAPE) values, and r2 scores. According to the study results, LSTM was more accurate and suitable for short-term estimates.

In this study, the consumption estimation was made with LSTM and deep neural network models, which are popular, and the results of these two models were compared. The study searched which approach to forecasting Turkey's short-term electricity demand would be better. Meteorological data were chosen as an independent variable like some mentioned studies. The data is population-weighted, the most crucial difference between the data in other studies. The study includes the calendar effect, but public holidays were not labeled separately. Chapters are follows; Turkish electricity market, methodology, consumption forecast, results, and conclusion.

2.TURKISH ELECTRICITY MARKET

The Turkish electricity market can be analyzed in three stages. The first stage is the early stage and covers the 1920s and 1960s. In this period, planning was weak and regulatory authorities had not been formed yet. The other stage was the structuring stage; at this stage, regulatory institutions emerged. The Turkish Electricity Administration (TEK) and the Ministry of Energy and Natural Resources (MENR) were established. The first liberalization efforts began in this period, and independent power producers (IPPs) became market shareholders. The period covering the 2000s and today is called the growth stage. The crucial developments in this stage are establishing the organized electricity market, the Energy market regulatory authority (EMRA), and promoting renewable energy (PwC Turkey & Presidency of the Republic of Turkey, 2021).

Turkey is a developing country; at the same time, the energy sector is developing rapidly. In the growth stage, energy consumption has increased, but the increase has stopped due to the economic crisis in the world and the country in the last few years (figure 2.1). When examining the share of sectors in electricity consumption, it is seen that households and services had the largest share in 2019 (figure 2.2). Turkey's total installed power in 2021 is seen in figure 2.3. Regarding production, the biggest shareholders are coal with 31.4%, and natural gas with 32.7%. Considering renewable energy separately, hydro renewables have 16.8% and solar and wind 13.4% (figure 2.4).

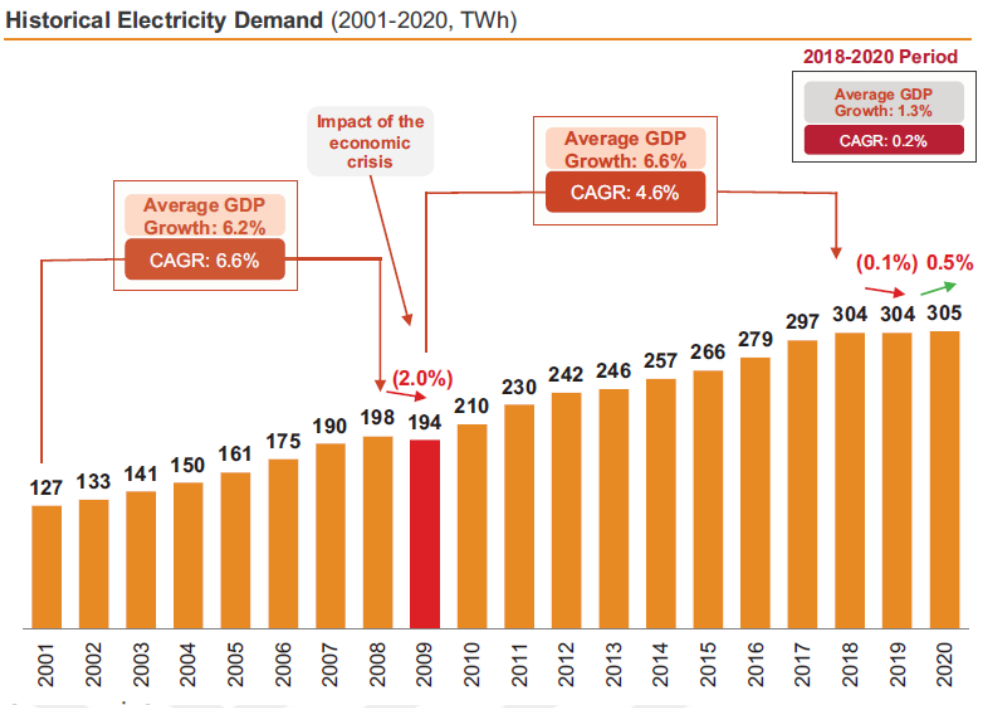


Figure 2.1 Historcal Electricity Demand Of Turkey (PwC Turkey & Presidency of the Republic of Turkey, 2021)

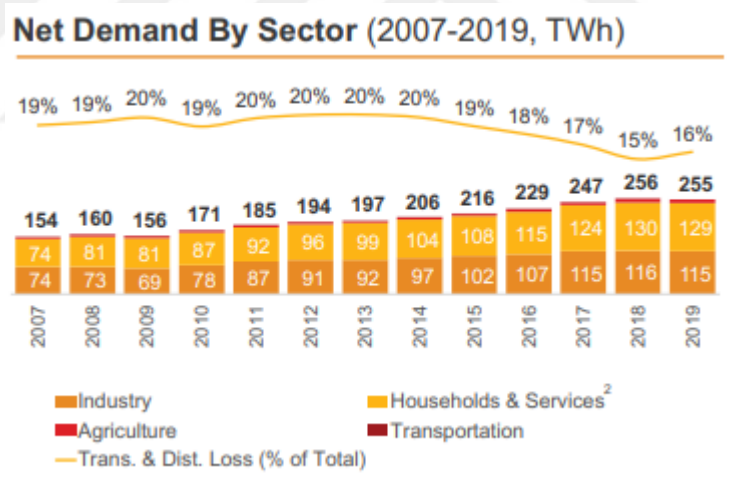


Figure 2.2 Turkey's Net Energy Demand By Sectors (PwC Turkey & Presidency of the Republic of Turkey, 2021)

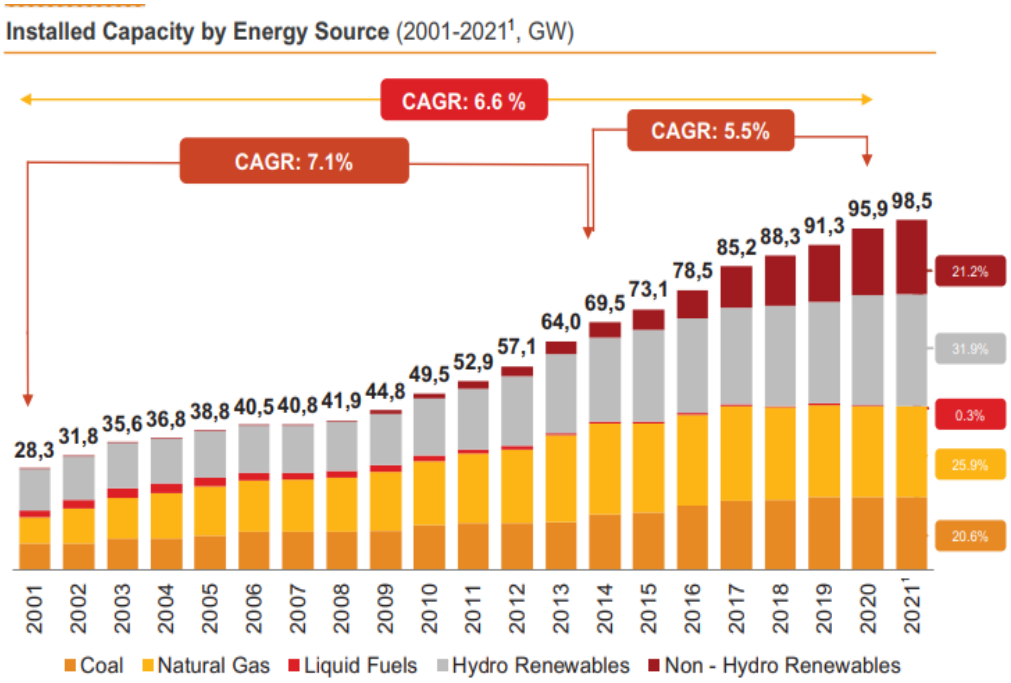


Figure 2.3 Turkey's Installed Capacity By Energy Source (PwC Turkey & Presidency of the Republic of Turkey, 2021)

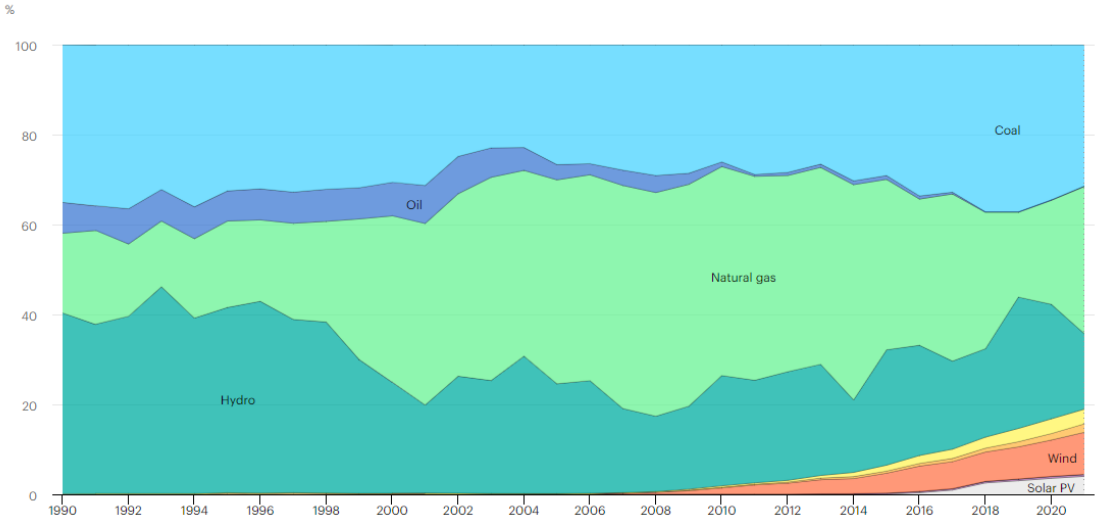


Figure 2.4 Turkey's Electricity Generation By Source (IEA, n.d.)

3.METHODOLOGY

3.1. What Is Machine Learning

Various algorithms are used in computers to understand and solve problems. An algorithm is a set of instructions that converts input to output. For some cases we cannot produce algorithms, but we know what the input is and what the output should be. In these cases, we want the program to learn from the data and create the algorithm automatically, so we appliy machine learning algorithms. Machine learning is a process for discovering necessary models to understand system behaviors. It uses certain patterns and trends that exist in the data to explore these patterns and make meaning of the data or find significant trends within it.

Machine learning can be applied to large databases to deduct meaningful conclusions. We can think of this large amount of data as the raw material before it is processed. Obtaining meaningful and valuable data by processing this raw material is called data mining, which has extensive uses in the field. Artificial intelligence (AI) algorithms form the basis of machine learning. They are dynamic systems and must have the ability to learn. Machine learning aims to optimize the model according to a performance criterion using specified sample data. Models can be predictive or descriptive (Alpaydin, E, 2014, p.2-3). Machine learning is generally categorized in two ways; the learning process and the problem-solving process. The learning stages are also divided into supervised, semi-supervised and unsupervised (Edgar & Manz, 2017, p.154-155).

3.1.1. Supervised learning

Data sets may differ from each other. While the available data can be labeled correctly, sometimes labeling itself may not be possible. The variability in the data set may require the learning styles to be changed for a successful result. When choosing which machine learning algorithms to use, it is necessary to consider their learning styles.

Supervised learning can be briefly defined as machine learning with correct answers. The training set is labeled with the desired solutions (Géron, 2019, p. 30-31). The data

provided as inputs are processed, and its outputs are compared with the required results. After the comparison, the errors are fed back into the system. The weights assigned by the model are re-adjusted, a process that is repeated many times. For this type of learning, the data must contain certain information in order to produce the output, and sufficient data must be provided to the model. Otherwise, the model will not truly learn or converge to the desired result (Anderson, D., and McNeill, G., 1992, p. 26-27).

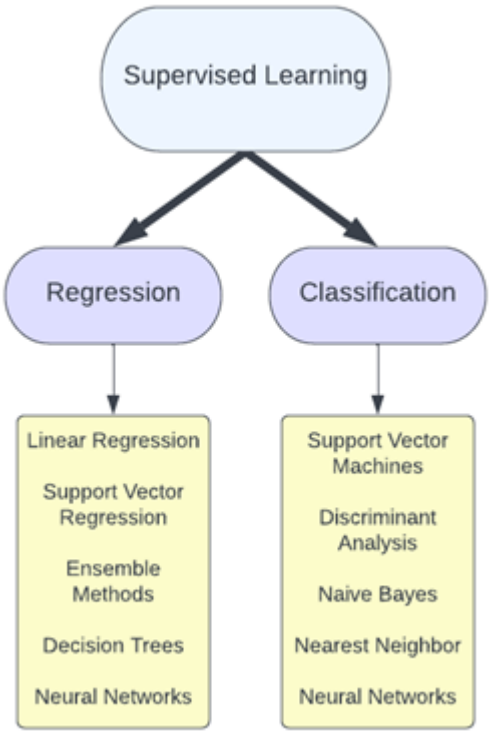


Figure Error! Use the Home tab to apply Başlık 2 to the text that you want to appear here.**.1 Supervised Learning Areas of Usage**

3.1.2. Unsupervised learning

Supervised learning involves learning from output to input with labeled data. In unsupervised learning, there are only inputs that are not marked with outcomes. In this type of learning, the model learns to group the inputs by itself, through a process called self-organization or adaptation (Anderson, D., and McNeill, G., 1992, p. 26-27). To do this, it tries to identify an existing pattern or trend in the input data. In statistics, this is called density estimation. An example of this learning method is clustering, and other commonly used methods are shown in Figure 3.1.2. Customer segmentation, which

companies use for many different purposes, is an example of this learning style and technique. Unsupervised learning is used in many fields, one of it which is the field of bioinformatics. DNA sequences consist of four types of nucleobases: adenine, thymine, cytosine, and guanine. These four bases line up in various ways to form DNA. Proteins are made up of sequences of amino acids, just like DNA, which learns the existing motifs in these sequences with clustering methods (Alpaydin, E, 2014, p.11-13).

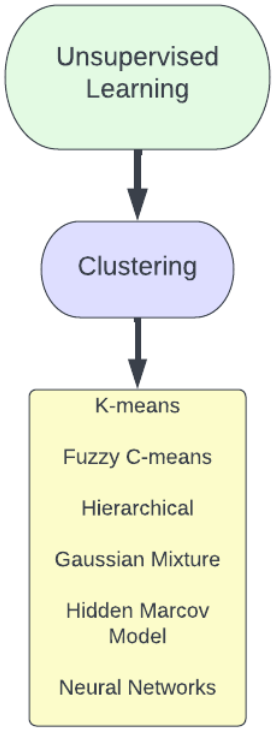


Figure 3.1.2 Unsupervised Learning Areas Of Usage

3.1.3. Semisupervised learning

Supervised learning contains tagged data and works with 2D (x,y) inputs. Here, the goal is to obtain a function f(x) that estimates y from x. Labeling each x with y is sometimes not time and cost-efficient. In these cases, algorithms that can work with small amounts of labeled and large amounts of unlabeled data are needed. Algorithms that work with such data are called semisupervised learning (SSL). For example, photo-hosting services, which detect the same person in many uploaded photos and determines which images are included (unsupervised part), work with this method. By tagging one of these photos, the

user allows that person to be tagged in other photos. As can be seen from the example, semi-supervised learning also works together with supervised and unsupervised learning (Géron, 2019, p.37-37).

3.1.4. Reinforcement learning

Reinforcement learning is a different branch of machine learning. In some cases, the system's output is a sequence of actions. In these cases, a specific series of actions are required to achieve a goal, and individual actions are unimportant (Alpaydin, E, 2014, p.11-13). For example, a humanoid robot sitting on a chair involves a series of actions. Suppose these actions are one step back and one step to the left. Stepping back and to the right does not constitute the sequence of actions necessary to sit on the chair. This situation necessitates not a single action but a whole set of actions and the right policy to choose the right actions. The Markov decision process is used to select these policies.

The Markov property is expressed mathematically as equation 3.1. The probability of occurrence of $S_{(t+1)}$ should depend only on the state of S_t.

$$P\,[St + 1 \mid St\,] \;=\; P\,[St + 1 \mid S1, \ldots., St] \tag{3.1}$$

The Markov process is expressed by equation 3.2. This process is defined by states and state transition probability (S, P). Every situation in this process must show a Markov property. Pss' represents the probability of transition from the current state to a state s'.

$$Pss' \;=\; P\,[St + 1 \;=\; s' \mid St \;=\; s] \tag{3.2}$$

A simple Markov chain is demonstrated in Figure 3.1.4 for a robot example like the one given earlier. In this learning model, the right behavior should be reinforced, and the wrong behavior should be penalized so that the robot can choose the best approach. Penalty points are applied for misconduct. Not every successful approach is the best policy, so a lot of experimentation and a long learning process is crucial to choosing the best approach.

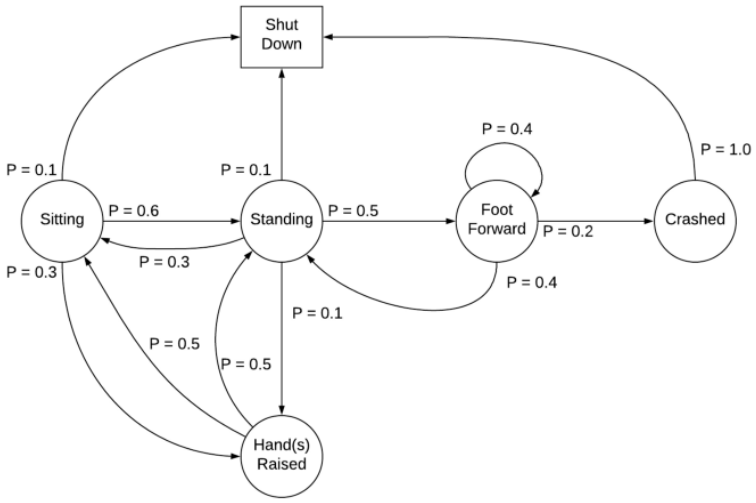


Figure Error! Use the Home tab to apply Başlık 2 to the text that you want to appear here. **Robot Example Marcov Chain (Jagtap, 2022)**

3.2. Artificial Neural Networks

Human have speculated about how the human brain works throughout histor, as humans' ability to process information and make sense of nature is unique, which is what makes them special. People often observe nature and are inspired by it, so it is unsurprising that they try to transfer their abilities to machines. Artificial neural networks (ANNs) result from such aspirational work and are based on the process of trying to imitate the biological neural networks that enable the brain to process information. ANNs first appeared in 1943, when they were introduced by the neurophysiologist Warren McCulloch and the mathematician Walter Pitts (Géron, 2019, p.358).

To understand ANNs, we must first understand the structure of biological neurons. A neuron in the human brain can be illustrated simply as in Figure 3.2. These cells consist of a cell body, dendrites, and an axon. The end of the axon divides into many branches, containing synapses that connect with the dendrites of other neurons. Neurons produce electrical impulses that cause synapses to release chemical signals when they are sufficiently stimulated. A single neuron has a simple structure, but billions of neurons can be connected to solve complex problems.

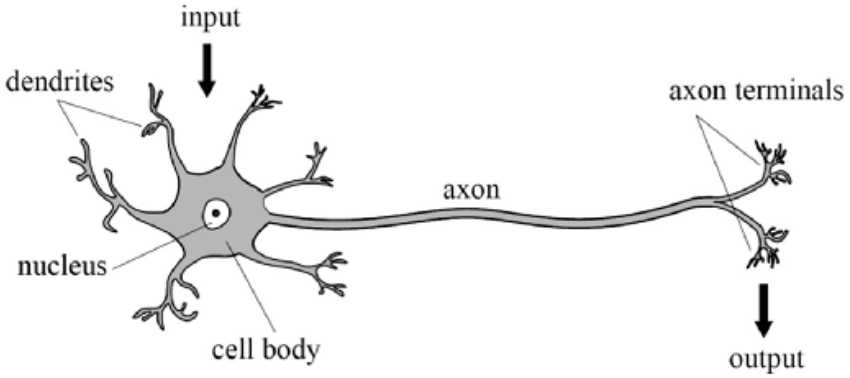


Figure 3.2.1Illustration of a biological neuron (Neves et al., 2017)

ANNS are widely used across an array of fields from image processing to speech recognition and games. Despite their wide use, they were not always so prevalent. The increase in the amount of available data, the increase in processing power as technology developed, and the development of new algorithms all contributed to the growth in the use of ANNs.

The basic ANN is called a perceptron. Perceptrons are based on neurons called the thres hold logic unit (TLU). TLUs are simple neurons with a single input unit and activation function. A perceptron consists of a single fully connected TLU layer. In perceptrons, inputs pass through special structures called input neurons. The layer where the input neurons are located is called the input layer.

Equation 3.3. calculates the outputs of the artificial neuron layers. X represents an input matrix with one row per sample and one column per feature. W defines the weight matrix, contains the weights of all connections, and contains one row for each input neuron and one column for each artificial neuron. Usually, a bias neuron is added to perceptrons. The output of this neuron is always 1. The bias vector W is not included in the weight matrix and is denoted by b. Φ represents the activation function.

$$hw, b\,(x) = \phi\,(XW + b) \tag{3.3}$$

If the biological neurons often stimulate each other, their connections get stronger. This principle underlay Hebb's rule, which says that the weight of the link between neurons increases when they fired together. Perceptrons are trained according to this rule, namely neurons that reduce errors increase the weights of their connections (Géron, 2019, p.360-365).

3.3. Deep Learning

Multilayer perceptrons have a hidden layer, an output layer, and an input layer. The hidden layers and output layer are also the TLU layers. When ANNs contain one or more hidden layers, they are called deep neural networks (DNNs).

The most critical step in developing DNNs was the invention of the backpropagation method in 1985, which is still actively used today. This method, introduced by David Rumelhart, Geoffrey Hinton, and Ronald Williams, calculates the gradient of errors according to each parameter in the model. It also uses the gradient descent technique to make this calculation. In short, it sets the necessary link and bias weights to reduce the error. To do this, it sets the parameters by the derivative of the function and does it iteratively. The aim is to minimize the loss function by reaching the global minimum. As a first step, all weights in the function are randomly assigned, and gradient descent is used until the global minimum is reached (Géron, 2019, p.171-174).

3.4. Hyperparameters Of Deep Neural Networks

With the development of deep learning algorithms, much scientific research has been done regarding multilayer ANNs. Specifically, researchers want to learn how to design deep learning models to solve problems, how many layers these models should consist of, how many neurons will they contain, and which optimization algorithms or activation functions are the most vital for problem-solving.

While creating machine learning models, it is necessary to decide which algorithm and what parameters should be used. While establishing deep learning models, the person who designed the model should decide on many parameters, such as the number of layers, the number of neurons on each layer, and which activation function will be used. These parameters called hyper-parameters. There are no strict rules for the selection of hyperparameters, and in many cases, the choice is entirely determined by the experience of the person designing the model as well as influence from current trends.

The modeler must therefore decide how deep the model will be. Even networks with a single hidden layer can handle complex operations if they have enough neurons, but deep

networks are more efficient for complex problems. Initially, it can be started with a single hidden layer and then increasing the number from there. In some cases, the increased number of layers can cause overfitting, so more layers cannot be added thereafter. The number of neurons in the layers is vital to the model's success. Input and output neurons should be adjusted according to the data set. In hidden layers, the general judgment is that the number of neurons in each layer decreases. Recently, it has been shown that having an equal number of neurons can also increase performance. As with the number of layers, when the number of neurons is more than necessary, overfitting may cause problems (Géron, 2019, p.415-416).

The dataset is the most critical step in building a model. Deep learning models require large datasets and a diverse set of, independent variables to have the best results. The largest disadvantage is that processing large data sets requires a lot of processing power and time. As mentioned before, in deep learning algorithms, the weight of each link is calculated over and over again with the backpropagation process, about which more below, implemented at each step. To overcome the disadvantage of the method, the batch size hyperparameter is used to limit the size of the data that can be processed at one time. Keeping the batch value low increases the value of the loss function while saving time and processing power (Çarkacı, 2020).

Backpropagation uses a chain rule to update weights retrospectively. In this process, it first calculates the partial derivate of the function. This value is then multiplied by the learning rate, and a new weight value is calculated by subtracting the obtained value from the old weight value. The error values used in this process can be calculated using many methods called loss functions. The most used loss funtions for regression are mean squared error (MSE), mean absolute error (MAE), and mean bias error(MBE). The most used loss functions for classification algorithms are support vector machines (SVM) loss and cross-entropy loss (Mulla, 2021).

In deep learning and other ANN algorithms, the learning process of the model is an optimization it needs to find the optimum method for nonlinear values, which requires optimization algorithms. The most known optimization algorithms are stochastic gradient descent (SGD), adagrad, adadelta, adam, and adamax. Each of these has advantages and disadvantages. Although SGD is the default method in deep learning, it works slowly and

is poor at image processing. Adagrad is more suitable to use for sparse parameters. In adagrad, the learning rate gradually decreases, and it can stop over time. Adadelta, on the other hand, does not have the rapid learning decline that adagrad has (Çarkacı, 2020). One of the most used optimization algorithms today is adam, which works similarly to SGD but uses the gradient moving average. The optimization algorithm is one of the parameters for ANNs, which should be selected according to the type of problem and what is expected from the model.

Deep learning models are widely used in solving nonlinear problems. Activation functions are needed so that the model must not be linear. The value obtained from matrix multiplication is linear, and activation functions convert it to a nonlinear form. The standart activation functions are sigmoid, tanh, and rectified linear unit (ReLU) (Çarkacı, 2020). The sigmoid function is defined in equation 3.4. Its outputs are between 0 and 1, depending on the input. The tanh function is similar to sigmoid but assigns values between -1 and 1. In this function, if the input is more positive, it converges to 1; if it is more negative, it converges to -1 (see equation 3.5). Today the most widely used activation function is ReLU. In this activation function, if the value is negative, a 0 output is obtained; if the input is greater than 0, the output is equal to the input (see equation 3.6).

$$\text{Sigmoid}(x) = {}^{1}/_{1 + e^{-x}} \tag{3.4}$$

$$\text{Tanh}(x) = (e^{x} - e^{-x})/(e^{x} + e^{-x}) \tag{3.5}$$

$$\text{ReLU}(x) = \max(0, x) \tag{3.6}$$

The ReLU activation function is also not perfect. In some cases, neurons become unable to produce any value other than 0, which is called the ‘death of neurons’. The phenomenon is observed especially with a high learning rate and when the inputs are mostly negative. Many variants of RELU, however have been developed to solve this problem, for example, LeakyReLU. This function is defined as in equation 3.7, where α is the function's "leakage" coefficient and denotes the slope for z<0. The exponential linear unit (ELU) activation function, which is similar to ReLU but slightly different, has been developed in recent years. It is defined as in equation 3.8 and does not return 0 for z<0. In this function, α denotes the value to which the function converges when z is a large negative number. Although it converges faster than ReLU, the test speed is slower.

In 2017, the Scaled ELU (SELU) method was introduced. This method is self-normalizing and generally outperforms other activation functions for deep neural networks. The model must be sequential and the layers dense to use the SELU activation function [36].

$$LeakyReLU(z) = \max(\alpha z, z) \tag{3.7}$$

$$ELU_{\alpha}(z) = \begin{cases} \alpha(exp(z) - 1), & z < 0 \\ z, & z \geq 0 \end{cases} \tag{3.8}$$

$$SELU(x) = \lambda \begin{cases} x, & x > 0 \\ \alpha e^{x} - \alpha, & x \leq 0 \end{cases} \tag{3.9}$$

In some cases, dropout is applied in fully connected layers. Dropping the nodes below a specific threshold value can increase the performance, and the determined threshold value is called the dropout value, since this weak information is forgotten. In deep learning, the dropout layer is used to prevent overfitting. The threshold value used in the drop-out layer varies according to the problem and the data set, and a different value can be used in each layer (Çarkacı, 2020).

3.5. Recurrent Neural Network

Recurrent neural networks (RNNs) are a type of nonlinear autoregressive moving average model that works with sequencial series or time series. RNN's are quite suitable for time series that must contain moving average components, have a trend, or are in state dependence (Connor et al., 1994). The most significant difference between RNNs from other deep-learning models is their memory. To predict, RNNs establish relationships between inputs and remember all relationships they have found and then use a loop to remember these relationships. This cycle is expressed mathematically as in equation 3.10, where H_t denotes the output value of the neuron, H_{t-1} is the previous output value, and X_t is the input vector.

$$H_t = f(H_{t-1}, X_t) \tag{3.10}$$

The two most important advantages of RNN are that it establishes a relationship between the inputs and has an extensive usage area. They are used in regression and classification problems, language processing, and many other areas. The most critical problem is

gradient vanishing. When gradient vanishing occurs, the weight parameters are updated until 0, and the learning stops. On the contrary, the exploding gradients problem is that the weights get too large for a stable model. A significant drawback is that it is difficult to handle long inputs with RNNs.

3.6. Long-Short Term Memory

RNNs cannot remember for a long time due to the gradient vanishing problem, but LSTM models have been developed to solve this problem. With LSTM, there is a cell state that is responsible for carrying information in the cell. LSTM has units called memory cells in repetitive hidden layers. Each memory block contains units called gates (Sak et al., 2014). The forget gate decides which information should be remembered, the input gate controls the information flow in the memory cell with its activation function, and the output gate controls the data flow out of the cell.

The first gate in the cell is the forget gate. As mentioned earlier, this gate decides which information will be remembered, which is done with the sigmoid function. The result of this function is a number between 0 and 1. When the result of the function approaches 1, it means that the information is important and should be remembered. Otherwise, it approaches 0, which means that the information is not important. This structure is shown in Figure 3.6.1.

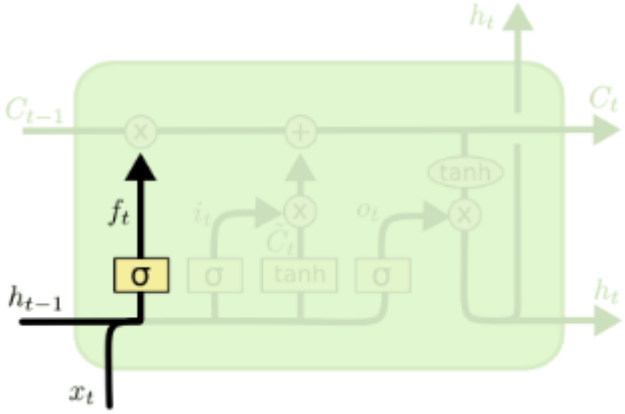


Figure Error! Use the Home tab to apply Başlık 2 to the text that you want to appear here.**6.1 A forget gate illisturation in LSTM cell. Adapted from Olah (Understanding LSTM Networks -- Colah's Blog, 2015)**

The information then moves to the input gate, which chooses what information will be stored to update the cell state. In short, it measures the importance of the new information. The input gate consists of two stages. The first stage is the sigmoid function, which determines the values that will be updated in this part. The vector of the new values is generated with the tanh function, and these two outputs are multiplied. The input gate is visualized in Figure 3.6.2.

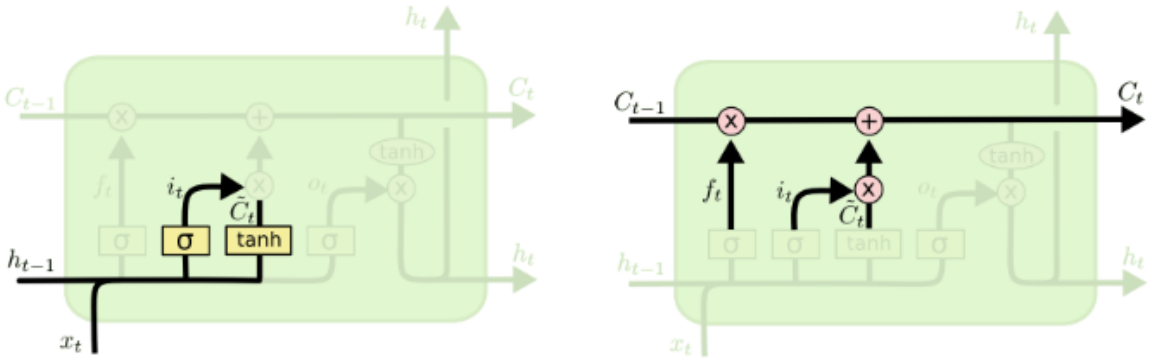


Figure 3.66.2 A input gate illisturation in LSTM cell. Adapted from Olah (Understanding LSTM Networks -- Colah's Blog, 2015)

The last gate is the output gate, which decides the output that goes to the next layer. First, a sigmoid function again determines which values will be included in the output. The cell state values are then arranged from -1 to 1 with the tanh function. The values in range -1 and 1 are multiplied by the output of the sigmoid function. This way, we get the output values [39]. The output gate is visualized in Figure 3.6.3.

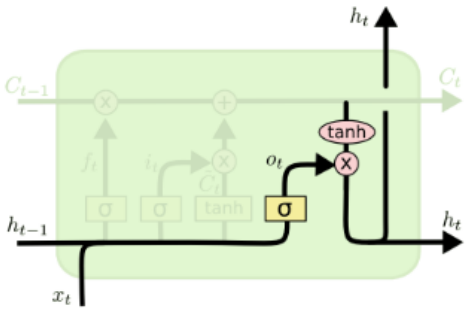


Figure 3.6.3 A output gate illisturation in LSTM cell. Adapted from Olah (Understanding LSTM Networks -- Colah's Blog, 2015)

3.7. Deep/Stack Long-Short Term Memory

Depth is conceptually defined as one or more hidden nonlinear layers between the input and output (Pascanu et al., 2014). Stack LSTMs contain more than one nonlinear layer, as in DNNs. Despite these nonlinear layers, data is processed by a single nonlinear layer before contributing to the output. According to this definition, a stack LSTM is created using more than one LSTM layer, as seen in Figure 3.7. One of the advantages of Stack LSTMs is that the processed data passes through more nonlinear layers without increasing the memory size needed when processing the data (Sak et al., 2014).

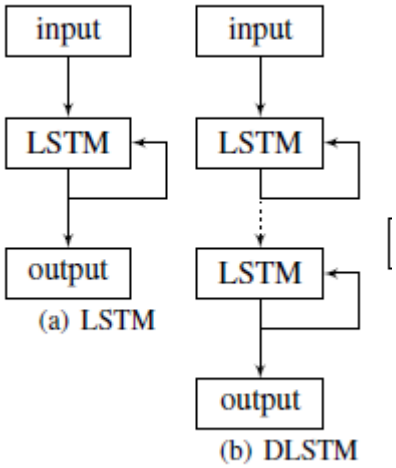


Figure Error! Use the Home tab to apply Başlık 2 to the text that you want to appear here.**7 A deep LSTM structure (Sak et al., 2014)**

4. CONSUMPTION FORECAST

This study attempts to estimate Turkey's hourly electricity consumption using deep neural network and stack LSTM methods. Real historical hourly electricity consumption data was used, which is published on the transparency platform of market operator EXIST. Historical population-weighted meteorological data was gathered from "renewables.ninja," which has a NASA data information policy license.

4.1. Historical Hourly Consumption Data

EXIST created the transparency platform in 2016, since which time considerable data about the electricity market has been made available to the public, including hourly electricity consumption. In this study, data from January 1 ,2016 to December 31, 2019 was used.

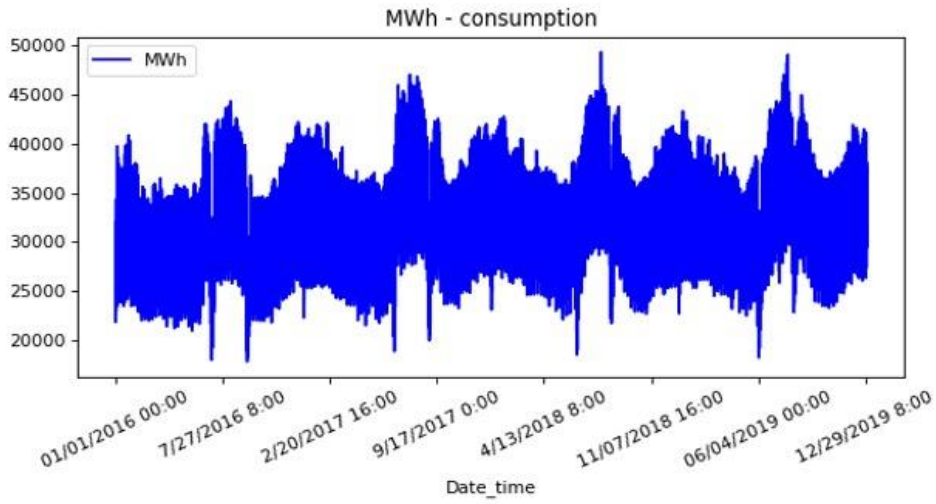


Figure Error! Use the Home tab to apply Başlık 2 to the text that you want to appear here. **Turkey's historical hourly electricity demand**

4.2. Feature Selection

Many factors affect electricity consumption. The main ones relate to climate and seasonality, as previously mentioned. Electricity consumption has a high seasonality, increasing and decreasing at specific times according to the hour, day, and month. The chosen data should explain these increases and decreases. In this study, hourly electricity consumption was assumed to be directly related to weather conditions and the hour of the day. Population-weighted data were used because the effects of weather conditions in areas of high density population are more critical than in areas of low-density population.

The meteorological data included in this study are precipitation (mm/h), temperature (°C), surface irradiance (W/m2), irradiance on top of the atmosphere (W/m2), snowfall (mm/h), snowmass (kg/m2), cloud cover (0,1), and air density (kg/m3), covering the period from January 1, 2016 to December 31, 2019. When the dataset was examined, there was no duplicate or note a number (NaN) data, so missing value handling was not applied.

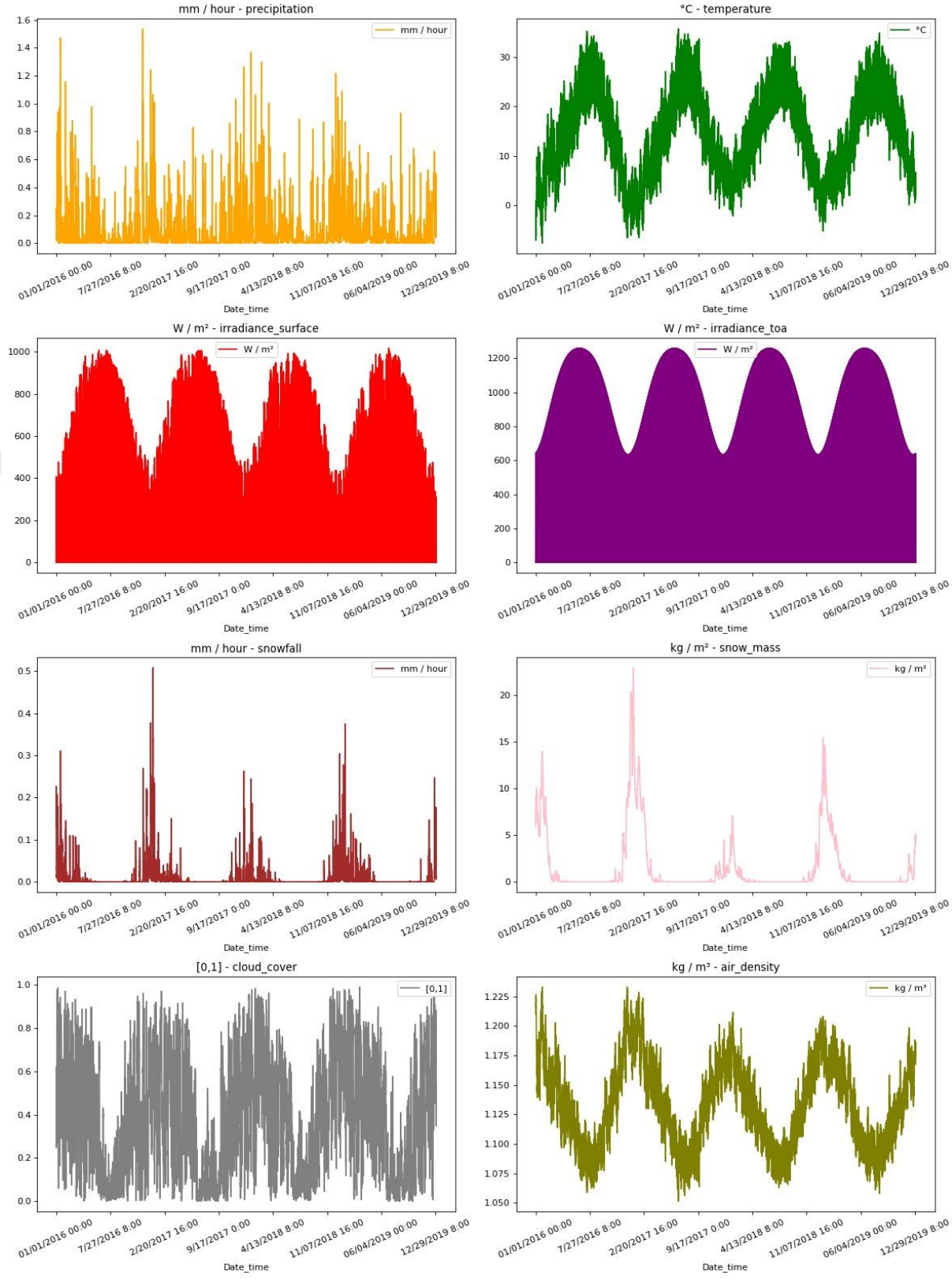


Figure 4.2.1 Data set raw data

To notice the sequential trend in the data, new features were derived using historical data. These derived features are listed in Table 4.2. In the day of the week variable, each day is encoded with a number: 1 for Friday and 7 for Thursday.

Table Error! Use the Home tab to apply Başlık 2 to the text that you want to appear here. **Observed and derived features**

No	Type	Variable
1	Observed	Electricity Consumption (MW)
2	Observed	Day time
3	Observed	Temperature (°C)
4	Observed	Precipitation(mm/h)
5	Observed	Surface Irradiance(W/m2)
6	Observed	Irradiance On Top Of The Atmosphere(W/m2)
7	Observed	Snowfall (mm/h)
8	Observed	Snow Mass(kg/m2)
9	Observed	Cloud Cover(0,1)
10	Observed	Air Density(kg/m3)
11	Derived	Year
12	Derived	Mounth Of Year
13	Derived	Day Of Week
14	Derived	Hour Of Day
15	Derived	Electricity consumption in t-24 (MW)

The relationship between the dependent variable, electricity consumption, and the other independent variables is shown in Figure 4.2.2. Independent variables should be correlated with the dependent variable but not highly correlated with each other. Independent variables that are highly correlated with each other can cause noise during

the learning phase of the model. It can be seen that there is no high correlation between independent variables in Figure 4.2.2.

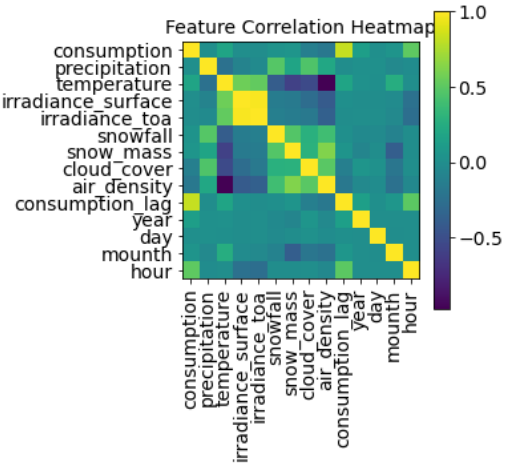
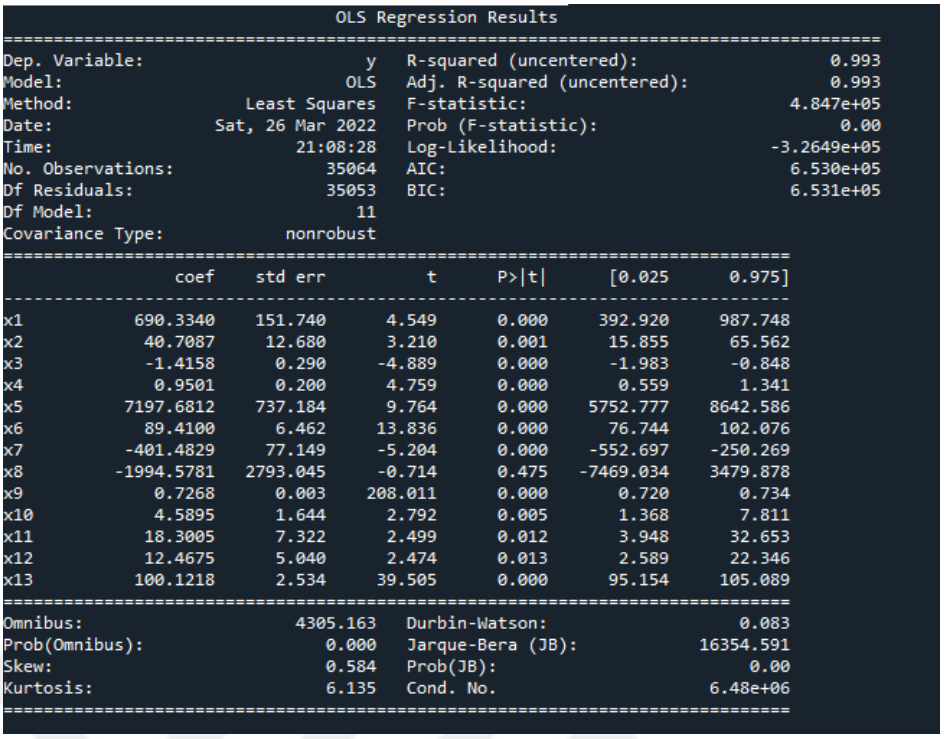


Figure 4.2.2 Feature correlation heat map

The OLS regression results of the 13 independent variables are shown in Figure 4.2.3. The eighth independent variable is snowfall in OLS statistics. Its p-value is higher than the threshold value, which is 0.05. The p-value is a parameter that denotes how statistically significant the variable is. If this value exceeds the threshold value, the variable is statistically insignificant. According to the current situation, the snowfall variable should be removed. At the same time, the multicollinearity problem in regression attracts attention. This problem is due to the independent variables having high coefficients in the regression. This issue is insignificant at this moment. The data will scale in a later stage of the model.



OLS Regression Results

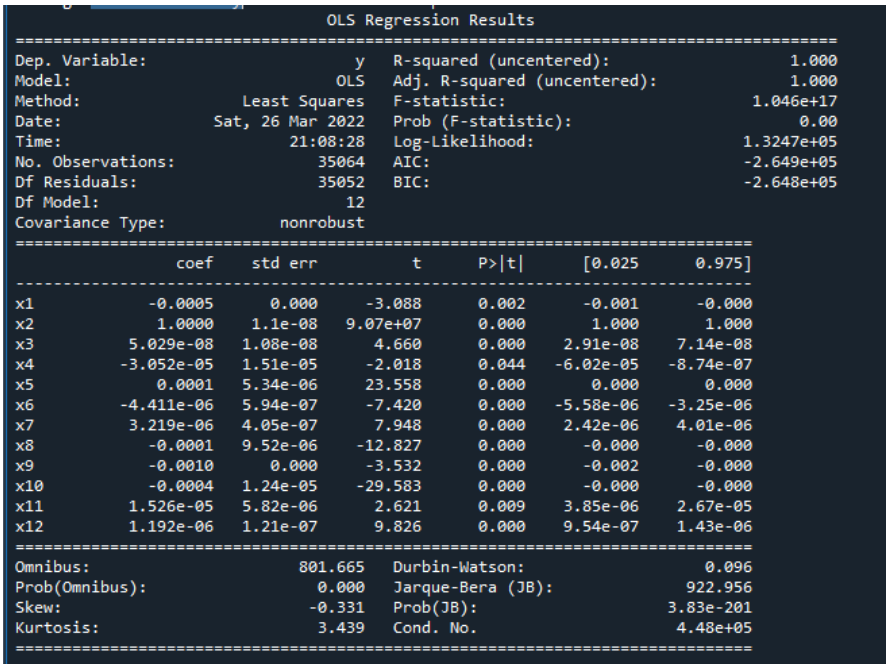
Dep. Variable:	y	R-squared (uncentered):	0.993
Model:	OLS	Adj. R-squared (uncentered):	0.993
Method:	Least Squares	F-statistic:	4.847e+05
Date:	Sat, 26 Mar 2022	Prob (F-statistic):	0.00
Time:	21:08:28	Log-Likelihood:	-3.2649e+05
No. Observations:	35064	AIC:	6.530e+05
Df Residuals:	35053	BIC:	6.531e+05
Df Model:	11		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
x1	690.3340	151.740	4.549	0.000	392.920	987.748
x2	40.7087	12.680	3.210	0.001	15.855	65.562
x3	-1.4158	0.290	-4.889	0.000	-1.983	-0.848
x4	0.9501	0.200	4.759	0.000	0.559	1.341
x5	7197.6812	737.184	9.764	0.000	5752.777	8642.586
x6	89.4100	6.462	13.836	0.000	76.744	102.076
x7	-401.4829	77.149	-5.204	0.000	-552.697	-250.269
x8	-1994.5781	2793.045	-0.714	0.475	-7469.034	3479.878
x9	0.7268	0.003	208.011	0.000	0.720	0.734
x10	4.5895	1.644	2.792	0.005	1.368	7.811
x11	18.3005	7.322	2.499	0.012	3.948	32.653
x12	12.4675	5.040	2.474	0.013	2.589	22.346
x13	100.1218	2.534	39.505	0.000	95.154	105.089

Omnibus:	4305.163	Durbin-Watson:	0.083
Prob(Omnibus):	0.000	Jarque-Bera (JB):	16354.591
Skew:	0.584	Prob(JB):	0.00
Kurtosis:	6.135	Cond. No.	6.48e+06

Figure 4.2.3 OLS regression statistics of the variables

After the snowfall variable is dropped, the OLS regression results are provided in Figure 4.2.4, and all p values are lower than the threshold value.



OLS Regression Results

Dep. Variable:	y	R-squared (uncentered):	1.000
Model:	OLS	Adj. R-squared (uncentered):	1.000
Method:	Least Squares	F-statistic:	1.046e+17
Date:	Sat, 26 Mar 2022	Prob (F-statistic):	0.00
Time:	21:08:28	Log-Likelihood:	1.3247e+05
No. Observations:	35064	AIC:	-2.649e+05
Df Residuals:	35052	BIC:	-2.648e+05
Df Model:	12		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
x1	-0.0005	0.000	-3.088	0.002	-0.001	-0.000
x2	1.0000	1.1e-08	9.07e+07	0.000	1.000	1.000
x3	5.029e-08	1.08e-08	4.660	0.000	2.91e-08	7.14e-08
x4	-3.052e-05	1.51e-05	-2.018	0.044	-6.02e-05	-8.74e-07
x5	0.0001	5.34e-06	23.558	0.000	0.000	0.000
x6	-4.411e-06	5.94e-07	-7.420	0.000	-5.58e-06	-3.25e-06
x7	3.219e-06	4.05e-07	7.948	0.000	2.42e-06	4.01e-06
x8	-0.0001	9.52e-06	-12.827	0.000	-0.000	-0.000
x9	-0.0010	0.000	-3.532	0.000	-0.002	-0.000
x10	-0.0004	1.24e-05	-29.583	0.000	-0.000	-0.000
x11	1.526e-05	5.82e-06	2.621	0.009	3.85e-06	2.67e-05
x12	1.192e-06	1.21e-07	9.826	0.000	9.54e-07	1.43e-06

Omnibus:	801.665	Durbin-Watson:	0.096
Prob(Omnibus):	0.000	Jarque-Bera (JB):	922.956
Skew:	-0.331	Prob(JB):	3.83e-201
Kurtosis:	3.439	Cond. No.	4.48e+05

Figure 4.2.4 OLS regression statistics of independent the variables after dropping snowfall

4.3. Model Implementation

4.3.1. Normalization

Deep learning models and all other ANN models are sensitive to data scaling, and a multicollinearity problem may arise if the data is not scaled. The root of this problem is that some independent variables have large values while others have much smaller values. This situation can be noticed from the OLS statistical results for the independent variables we used in the study.

Different statistical methods are available for normalization. This study used the MinMaxScaler method in Python's sklearn library. In this method, the data is scaled to 0 and 1 intervals.

$$x_{scaled} = \frac{x - x_{min}}{x_{max} - x_{min}} \tag{4.1}$$

4.3.2. Accuracy validation

In the literature, many different performance criteria are used. Mean squared error (MSE), mean absolute error (MAE), and mean absolute percentage error (MAPE), which are widely used and included in Keras API, were used in this study.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|x_i - x| \tag{4.2}$$

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(x_i - \hat{x}_i)^2 \tag{4.3}$$

$$\text{MAPE} = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{x_i - \hat{x}_i}{x_i}\right| \tag{4.4}$$

where :

x_i actual value

$\hat{x}_i$ predicted value

4.3.3. Train test split

Models must not memorize the data. Therefore data must be separated as test and train. In the DNN model, which is the first model, the data from December 18. 2019 to December 31, 2019 was reserved for testing. The model was trained with the rest of the data. During the model training, 20% of the data were used for validation.

```
# normalize features
scaler = MinMaxScaler(feature_range=(0, 1))
scaled_X = scaler.fit_transform(X2)
scaled_Y = scaler.fit_transform(Y)


X_train= scaled_X [:34728,:]
X_test= scaled_X [34728:,:]
y_train = scaled_Y [:34728,:]
y_test = scaled_Y [34728:,:]
```

Figure 4.3.3.1 Train And Test Split

The dataset used for the second model, deep LSTM, had to be made in 3D. The function in Figure 4.3.3.2 is used to convert the 2D time series array to 3D, and a two week portion of the data was reserved for testing.

```
def create_X_Y(ts: np.array, lag=1, n_ahead=1, target_index=0) -> tuple:
    """
    A method to create X and Y matrix from a time series array for
    the training of deep learning models
    """
    # Extracting the number of features that are passed from the array
    n_features = ts.shape[1]

    # Creating placeholder lists
    X, Y = [], []

    if len(ts) - lag <= 0:
        X.append(ts)
    else:
        for i in range(len(ts) - lag - n_ahead):
            Y.append(ts[(i + lag):(i + lag + n_ahead), target_index])
            X.append(ts[i:(i + lag)])

    X, Y = np.array(X), np.array(Y)

    # Reshaping the X array to an RNN input shape
    X = np.reshape(X, (X.shape[0], lag, n_features))

    return X, Y
```

Figure 4.3.3.2 Time Series Array To 3D Matrix

4.4. Deep Neural Network Model

The DNN model was built using the Keras API in Python and Python codes shared on github (Dede, 2022a). As mentioned, 12 independent variables and one dependent variable were used in this model. In the study, one neuron for each independent variable for a total of 12 neurons were used in the input layer. One neuron was used in the output layer because we wanted to forecast hourly consumption. First, to determine the optimal number of hidden layers, a hidden layer was added at each iteration with 50 neurons, the "relu" activation function was used, and the results were recorded. As can be seen in Table 4.4.1, the best results were obtained with four hidden layers.

Table 4.4.1 Hidden Layers And Validation Losses

	VALIDATION LOSS		
HİDDEN LAYER	MAE	MAPE	MSE
1	0,0022	0,7102	0,38
2	0,0020	0,6006	0,10
3	0,0025	0,8202	0,24
4	0,0013	0,3618	0,04
5	0,0019	0,4754	0,08

After the number of hidden layers was determined, the results were recorded with the commonly used “sigmoid”, “tanh”, “relu”, “elu”, “selu”, and “leakyrelu” activation functions to select the best option. It is seen in the literature that activation functions should be used with specific weight initiators. This is because each activation function is different and not compatible with every weight initiator. Therefore, the “sigmoid” and “tanh” functions were used with the “GlorotNormal” weight initializer (Glorot & Bengio, 2010), “relu”, “elu”, and “leaky relu” with the “he_normal” weight initializer (He et al., 2015), and finally, “selu” with the “lecun_normal” weight initializer (Géron, 2019, p.430-434). As seen in Table 4.3.2, the best result was obtained with the “relu” activation function.

Table 4.4.2 Activation Functions And Validation Losses

	VALIDATION LOSS		
Activation Functions	MAE	MAPE	MSE
Sigmoid	0,0026	0,7905	0,17
Tanh	0,002	0,4409	0,07
ReLU	0,0016	0,4613	0,06
ELU	0,0037	1,0186	0,27
SELU	0,0036	1,0477	0,23
LeakyReLU	0,0022	0,6277	0,10

Finally, in each iteration, the number of neurons was changed to determine the optimal number of neurons in the hidden layers.[a, f] represent the input and output layers, respectively, while [b, c, d, e] represent four hidden layers. Neuron numbers were tested as recommended in the literature, which is equal or decreasing. As can be seen in Table 4.4.3, the optimal result was obtained when 50 neurons were in each hidden layer.

Table 4.4.3 Number Of Neurons In Layers And Validation Losses

Architecture No.	Number of Neurons						Validation Loss		
	a	b	c	d	e	f	MAE	MAPE	MSE
1	12	10	10	10	10	1	0,0012	0,389	0,29
2	12	50	10	10	10	1	0,0031	0,8733	0,55
3	12	50	50	10	10	1	0,0014	0,502	0,08
4	12	50	50	50	10	1	0,003	0,9746	0,22
5	12	50	50	50	50	1	0,0012	0,4599	0,06
6	12	100	50	50	50	1	0,0023	0,6957	0,11
7	12	100	100	50	50	1	0,0019	0,4901	0,08
8	12	100	100	100	50	1	0,0018	0,5241	0,07
9	12	100	100	100	100	1	0,0032	0,8541	0,22
10	12	100	100	50	10	1	0,0014	0,3434	0,04
11	12	100	50	50	10	1	0,0018	0,5182	0,07
12	12	100	50	10	10	1	0,002	0,5626	0,08
13	12	100	10	10	10	1	0,0012	0,5353	0,08

```
model= Sequential()
model.add(Dense(12,kernel_initializer='he_normal',activation='relu'))

model.add(Dense(50,kernel_initializer='he_normal',activation='relu'))

model.add(Dense(50,kernel_initializer='he_normal',activation='relu'))

model.add(Dense(50,kernel_initializer='he_normal',activation='relu'))

model.add(Dense(50,kernel_initializer='he_normal',activation='relu'))



model.add(Dense(1, activation='linear'))
model.compile(optimizer = 'adam', loss = 'mae', metrics=["mae","mape","mse"])

path_checkpoint = "model_checkpoint.h5"
es_callback = keras.callbacks.EarlyStopping(monitor="val_loss", min_delta=0,
                                            patience=15)

modelckpt_callback = keras.callbacks.ModelCheckpoint(
    monitor="val_loss",
    filepath=path_checkpoint,
    verbose=1,
    save_weights_only=True,
    save_best_only=True,
)

history = model.fit(X_train,y_train,
                validation_split=0.2,
                batch_size=256,
                shuffle=True,
                verbose=2,
                epochs=500,
                callbacks=[es_callback, modelckpt_callback])

model.summary()
```

Figure 4.4.1 Python Code For DNN

4.5. LSTM Model

The LSTM model was built using Keras API in Python and Python codes shared on github (Dede, 2022b). Like the other model, 12 independent variables and one dependent variable were used, and one neuron for each independent variable, for a total of 12 neurons, was used in the input layer. One neuron was also used in the output layer to forecast hourly consumption. We determined the hidden layer, then the activation function, and finally the number of neurons in the hidden layers in order to establish the hyperparameters as in the previous model. According to Table 4.5.1, two hidden layers gave the best results.

Table 4.5.1 Hidden Layers And Validation Losses

	VALIDATION LOSS		
HIDDEN LAYER	MAE	MAPE	MSE
1	0,0103	2,4226	1,9891
2	0,009	2,1365	1,5402
3	0,0123	2,8783	2,6816
4	0,0106	2,4763	2,1046

Activation functions 'sigmoid', 'tanh', 'relu', 'elu', 'selu', and 'leaky relu' were attempted and defined the best activation function for the model. Among them, “tanh” was slightly better than the others according to the validation loss illustrated in Table 4.5.2.

Table 4.5.2 Activation Functions And Validation Losses

	Validation Loss		
Activation Functions	MAE	MAPE	MSE
Sigmoid	0,0144	3,2522	4,0042
Tanh	0,009	2,1365	1,5402
ReLU	0,0107	2,6442	2,1546
ELU	0,0135	3,1301	3,1882
SELU	0,0124	2,9233	2,6474
LeakyReLU	0,0121	2,8096	2,4991

The number of neurons for the input and output layers is shown by [a,d], and for the hidden layers by [b,c]. According to Table 4.5.3, it was concluded that there should be 150 neurons in each hidden layer.

Table 4.5.3 The Number Of Neurons In Layers And Validation Losses

Architecture No.	Number Of Neurons				Validation Loss		
	a	b	c	d	MAE	MAPE	MSE
1	12	10	10	1	0,0121	2,9016	2,74
2	12	50	10	1	0,0098	2,3045	1,86
3	12	100	10	1	0,0098	2,3388	1,75
4	12	150	10	1	0,0096	2,2679	1,73
5	12	200	10	1	0,0091	2,1582	1,63
6	12	50	50	1	0,0111	2,6979	2,24
7	12	100	50	1	0,0105	2,4285	2,09
8	12	150	50	1	0,011	2,5512	2,21
9	12	200	50	1	0,0094	2,1560	1,70
10	12	100	100	1	0,0099	2,3049	1,85
11	12	150	100	1	0,0096	2,2041	1,70
12	12	200	100	1	0,0098	2,2801	1,80
13	12	150	150	1	0,009	2,1074	1,48
14	12	200	150	1	0,0098	2,2559	1,78
15	12	200	200	1	0,0094	2,1999	1,69

```
model = Sequential()
model.add(LSTM(12, kernel_initializer='GlorotNormal', activation='tanh',
               input_shape=(Xtrain.shape[1], Xtrain.shape[2]), return_sequences=True))
model.add(LSTM(150, kernel_initializer='GlorotNormal', activation='tanh',
               input_shape=(Xtrain.shape[1], Xtrain.shape[2]), return_sequences=True))
model.add(LSTM(150,kernel_initializer='GlorotNormal', activation='tanh' ,
               return_sequences=False))
model.add(Dense(Ytrain.shape[1],activation='linear'))

model.compile(optimizer='adam', loss='mae',metrics=["mae","mape","mse"])
model.summary()

#early stopping
path_checkpoint = "model_checkpoint.h5"
es_callback = keras.callbacks.EarlyStopping(monitor="val_loss",
                                            min_delta=0, patience=15)

modelckpt_callback = keras.callbacks.ModelCheckpoint(
    monitor="val_loss",
    filepath=path_checkpoint,
    verbose=1,
    save_weights_only=True,
    save_best_only=True,
)

# fit the model
history = model.fit(Xtrain,Ytrain,
                validation_split=0.2,
                batch_size=256,
                shuffle=True,
                verbose=2,
                epochs=500,
                callbacks=[es_callback, modelckpt_callback])
```

Figure 4.5.1 Python Code For LSTM Model

5. RESULTS

The DNN model, which is the first of the prediction models used in this study, was run with the selected optimal parameters. As mentioned before, 20% of the data used for the train was separated for validation. Before early stopping was activated, the loss per epoch was calculated with the mean absolute error metric and plotted as in Figure 5.1. This graph gives us information about how successful the model was during training and about the errors such as overfitting or unknown fitting that occurred during the training of the model. Unknown fitting occurs when the validation loss is much lower than the training loss, whereas overfitting occurs if the validation loss and training loss overlap precisely. These two cases reveal that there was a problem during the learning phase. As shown in Figure 5.1, although the learning is a little noisy, these two situations were not observed in the training of the model.

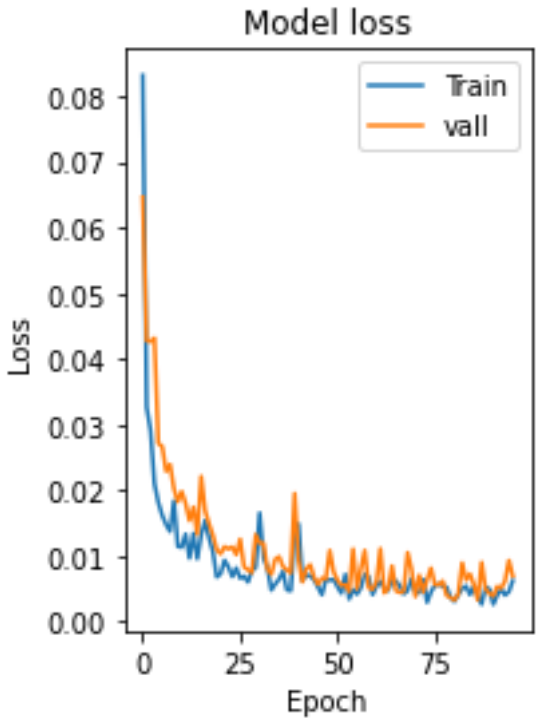


Figure 5.1 DNN Model Train And Validation Loss

The stacked LSTM model was run with the parameters as outlined above. As in the other model, 20% of the data was separated for validation during training, and the mean absolute error metric was used to calculate losses. The training success of the stacked

LSTM model is shown in Figure 5.2. Although the training phase is less noisy than the other model, its accuracy is slightly lower.

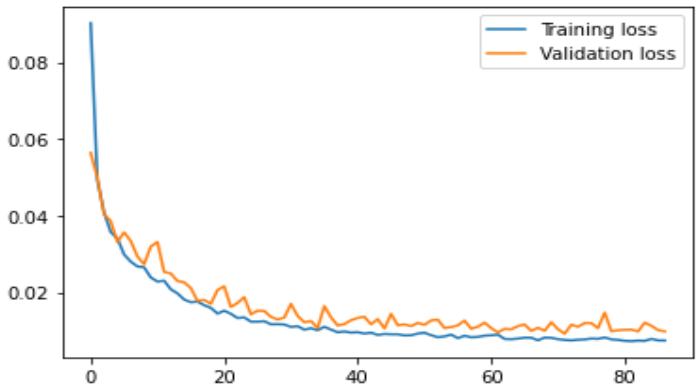


Figure 5.2 The Stacked LSTM Model Train And Validation Loss

The first model ran with the independent variables in the data set separated for the test. Consumption was estimated for two week periods. The estimated and actual consumption graphics are compared in Figure 5.3. Visually, the estimated consumption and actual consumption are quite similar. The accuracy of the prediction was calculated with three metrics; mean squared error(MSE), mean absolute error (MAE), and mean absolute percentage error (MAPE), respectively. As a result of the two week estimation of the first model, MSE was calculated as '265.32876527436383', MAE as '244.703125', and MAPE as '0.007274017701473777'.

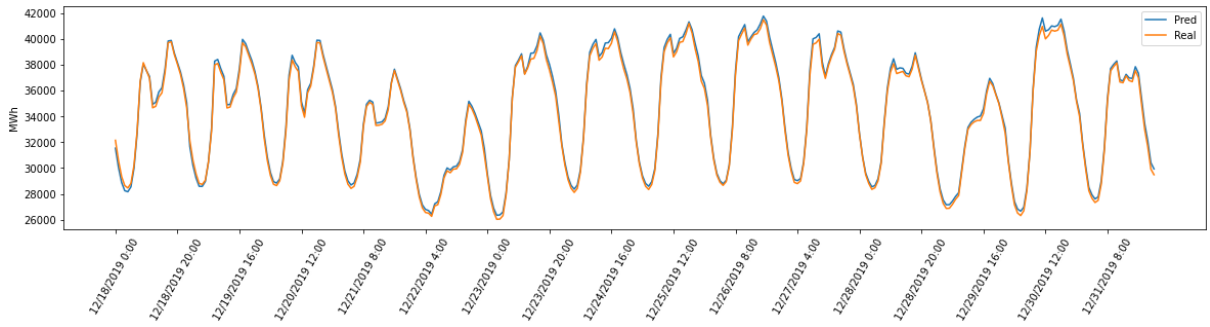


Figure 5.3 DNN Model's Predicted and Real Consumption Graph

In the second model, hourly electricity consumption was estimated by the independent variables in the data set separated for the test, the same as in the first model. The estimated consumption of the model and actual consumption are shown in Figure 5.4. Visually, the prediction results seem more inaccurate than the first model. To measure the accuracy of the estimation, mean squared error (MSE), mean absolute error (MAE), and mean

absolute percentage error (MAPE) metrics were used, respectively, as in the first model. As a result of the accuracy calculation of the estimated consumption, MSE is calculated as '2056.4022046438995', MAE is '1667.5655059614307', MAPE is '0.05251767775430342', and all error metrics can observe from Table 5.1.

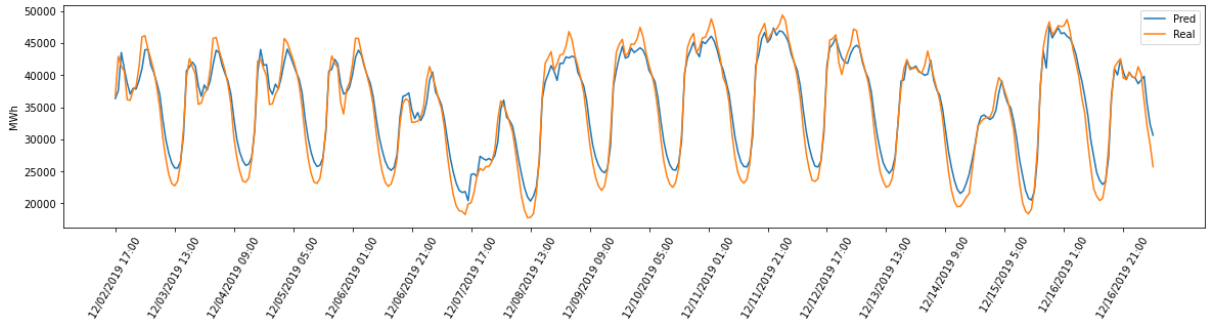


Figure 5.4 Stack LSTM Model's Predicted And Real Consumption Graph

Table 5.1 DNN and LSTM metrics

Model	MAE	MAPE	MSE
DNN	244.70	0.0073	265.33
LSTM	1667.57	0.0526	2056.40

Cross-validation was performed to ensure that the study was robust. Fourteen days from two different years and two different periods were selected for cross-validation. The forecast results were highly accurate, can be seen in Figure 5.5 and Figure 5.6. The first testing was for the first two weeks of May 2019, and the error metrics were recorded as follows, MSE '410.94618336833355', MAE '336.74942', and MAPE '0.011522852'. The consumption was forecasted for the first two weeks of August 2018 secondly, and it was highly accurate. On the second attempt, the error metrics were recorded as follows, MSE '252.22991433908072', MAE '205.47615', and MAPE '0.005456625'.

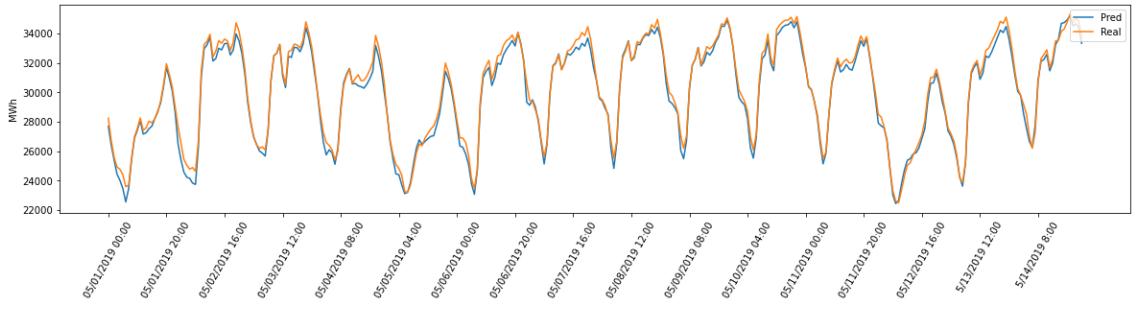


Figure 5.5 DNN Model's Forecast Results For First Two Week Of May 2019

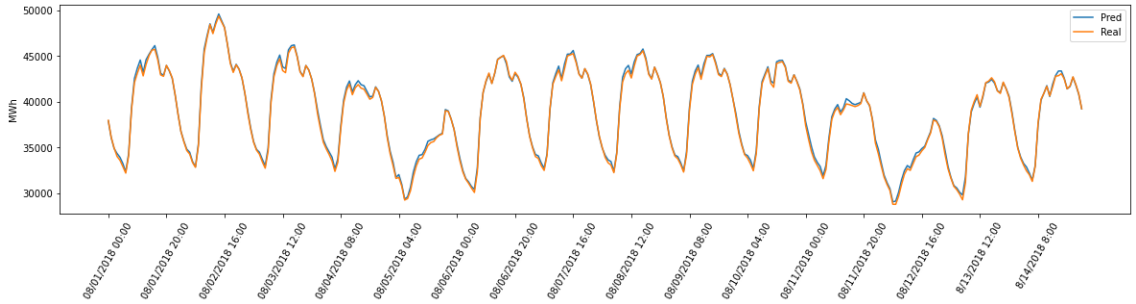


Figure 5.6 DNN Model's Forecast Results For First Two Week Of August 2018

The cross-validation results were compared with the consumption estimations provided by EXIST on the EXIST Transparency Platform. Error measurements for EXIST's 01 May 2019 – 14 May 2019 consumption forecast are MAE '800.5181' and MAPE '0.027285'. Error measurements for EXIST's 1 August 2018 – 14 August 2018 consumption forecast are MAE '864.6486526' and MAPE, '0.022026814', and all error metrics can be observed in Table 5.2.

Table 5.2 Cross validation and EXIST's forecast metrics

Date	MAE	MAPE	MSE
1 – 14 May 2019 (DNN)	336.74942	0.011522852	410.94618336833355
1 -14 August 2018 (DNN)	205.47615	0.005456625	252.22991433908072
1 – 14 May 2019 (EXIST)	800.5181	0.027285	-
1 – 14 August 2018 (EXIST)	864.6486526	0.022026814	-

The consumption estimation of the first model is more accurate. Sometimes, two weeks can be a long timeframe for a short-term forecast. In that case, a one week or even a one day forecast becomes essential. The results of the one week estimation in the first model are seen in Figure 5.7, while the test metrics were calculated as MSE '258.7979540159367', MAE '188.6723400297619', and MAPE '0.00573465100916626'.

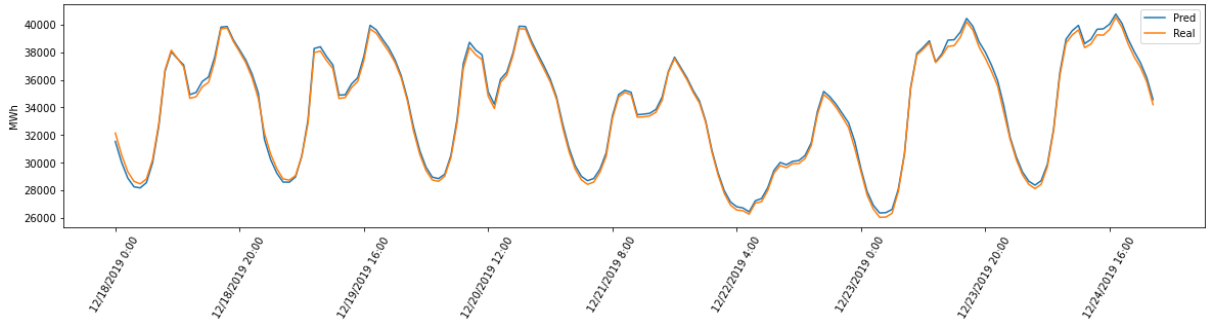


Figure 5.7 DNN Model One Week Forecast

6. CONCLUSION

In a concluson, there are many studies in the literature on short-term electricity consumption forecasts, which discuss the problem in different ways using different methodologies. With the development of technology in recent years, the increase in the data processing capacity of computers and the increasing amount of data, our abilities to predict are improving. In light of these developments, ANN models have been used frequently, especially for problems that are difficult to predict.

This study used stacked LSTM and deep neural network methods to predict Turkey's short-term electricity consumption on an hourly basis. It differs from other studies because it is based entirely on meteorological data and the weighting of these data with population distribution, and also compares the prediction success of the stacked LSTM and DNN methods with a relatively small amount of data.

Nine independent variables were observed, and five independent variables were used in the models, while 12 independent variables were eliminated due to the p values on their OLS regression results. According to the selected error measurement metrics, it was decided that the DNN model should have for hidden layers, a ReLU activation function, and 50 neurons for each hidden layer. It was also concluded that in the stacked LSTM model, there should be two hidden layers, 150 neurons for each hidden layer, and the tanh activation function.

As a result of the estimation made with these selected hyperparameters, the DNN model was found to the most successful with 258.79 MSE and suggest that this model be used when conducting short-term consumption forecasting, as it is superb as dealing predicting short-term actions. This finding has implications for electricity price estimations based on consumption estimations, especially since they must take into account energy facilities that face limited sources. Consumption estimations are also vital in offering ancillary services prices in the electricity market and determining the price for load and deload offers.

The main reason for the lower prediction success of the Stacked LSTM model may be that the model needs more data points than the other model. In addition, the forget gate in the structure of this method may have caused some critical information to be forgotten.

At the same time, this method needs 3D matrices. Slight data loss occurred when 2D matrices were converted to 3D, but it is not at an amount that would affect the estimation.

6.1. Future Policy

The proposed method has a low deviation percentage, as revealed, and can be used for short-term load demand forecasting for companies in the energy sector. It has been observed that it gives a better estimation than the system operator's estimation and can be used by the system operator when needed. Meteorological data should be weighted with population distribution and given to the system with historical consumption data. The proposed method is planned to be developed in future studies using a larger data set and calendar labels for public holidays.

REFERENCES

Akan, Y. and Tak, S. (2010). Türkiye Elektrik Enerjisi Ekonometrik Talep Analizi. Atatürk Üniversitesi İktisadi ve İdari Bilimler Dergisi, 17 (1-2), Retrieved from https://dergipark.org.tr/tr/pub/atauniiibd/issue/2685/35247

Alpaydin, E. (2014). Introduction To Machine Learning 3Rd Edition. PHI.

Altınay, G. (2010). Aylık elektrik talebinin mevsimsel model ile orta dönem öngörüsü. Enerji, Piyasa ve Düzenleme, 1(1), 1-23.

Anderson, D., and McNeill, G. (1992). *Artificial Neural Networks Technology*, 26 -27.

Bilgili, M. (2009). Estimation of net electricity consumption of Turkey. Isi Bilimi Ve Teknigi Dergisi/ Journal of Thermal Science and Technology. 29. 89-98.

Bişkin, O. T., and Çiftçi, A. (2021). Forecasting of Turkey's Electrical Energy Consumption using LSTM and GRU Networks. Bilecik Şeyh Edebali Üniversitesi Fen Bilimleri Dergisi. https://doi.org/10.35193/bseufbd.935824

Bodur, I., Celik, E., and Ozturk, N. (2021). A short-term load demand forecasting based on the METHOD OF LSTM. 2021 10th International Conference on Renewable Energy Research and Application (ICRERA). https://doi.org/10.1109/icrera52334.2021.9598773

Bulut, M., and Başoğlu, B. (2017). Kısa Dönem Elektrik Talep Tahminleri İçin Yapay Sinir Ağları ve Uzman Sistemler Tabanlı Hibrid Tahmin Sistemi Geliştirilmesi. Gazi Üniversitesi Mühendislik-Mimarlık Fakültesi Dergisi, 32(2). https://doi.org/10.17341/gazimmfd.322184

Cömert, M., and Yıldız, A. (2018). Forecasting short-term electricity demand of Turkey by artificial neural networks. 2018 International Conference on Artificial Intelligence and Data Processing (IDAP). https://doi.org/10.1109/idap.2018.8620861

Çarkacı, N. (2020, January 1). Derin Öğrenme Uygulamalarında En Sık kullanılan Hiper-parametreler. *Medium*. https://medium.com/deep-learning-turkiye/derin-ogrenme-uygulamalarinda-en-sik-kullanilan-hiper-parametreler-ece8e9125c4

Dede, B. (2022a, February 20). GitHub - BerkDede/electricity-demand-forecast--dnn-. GitHub. Retrieved February 20, 2023, from https://github.com/BerkDede/electricity-demand-forecast--dnn-

Dede, B. (2022b, February 20). GitHub - BerkDede/electricity-demand-forecast-with-stacked-lstm. GitHub. Retrieved February 20, 2023, from https://github.com/BerkDede/electricity-demand-forecast-with-stacked-lstm

Edgar, T. W., and Manz, D. O. (2017). Machine Learning. *Research Methods for Cyber Security*, 154–155. https://doi.org/10.1016/b978-0-12-805349-2.00006-6

Ediger, V., and Akar, S. (2007). ARIMA forecasting of primary energy demand by fuel in Turkey. *Energy Policy*, *35*(3), 1701–1708. https://doi.org/10.1016/j.enpol.2006.05.009

Ediger, V., and Tatlıdil, H. (2002). Forecasting the primary energy demand in Turkey and analysis of cyclic patterns. Energy Conversion and Management, 43(4), 473–487. https://doi.org/10.1016/s0196-8904(01)00033-4

Esteves, G. R. T., Bastos, B. Q., Cyrino, F. L., Calili, R. F., and Souza, R. C. (2015). Long Term Electricity Forecast: A Systematic Review. Procedia Computer Science, 55, 549–558. https://doi.org/10.1016/j.procs.2015.07.041

Géron, A. (2019). *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems* (2nd ed.), O'Reilly Media.

Glorot, X., and Bengio, Y. (2010). Understanding the difficulty of training deep feedforward neural networks. International Conference on Artificial Intelligence and Statistics, 249–256. https://proceedings.mlr.press/v9/glorot10a/glorot10a.pdf

Günay, M. (2016). Forecasting annual gross electricity demand by artificial neural networks using predicted values of socio-economic indicators and climatic conditions: Case of Turkey. *Energy Policy*, *90*, 92–101. https://doi.org/10.1016/j.enpol.2015.12.019

Haliloğlu, E. Y., and Tutu, B. E. (2018). Türkiye İçin Kısa Vadeli Elektrik Enerjisi Talep Tahmini. Journal of Yaşar University, 13(51), 243–255.

Hamzacebi, C., and Es, H. A. (2014). Forecasting the annual electricity consumption of Turkey using an optimized grey model. Energy, 70, 165–171. https://doi.org/10.1016/j.energy.2014.03.105

Hamzaçebi, C. (2007). Forecasting of Turkey's net electricity energy consumption on sectoral bases. *Energy Policy*, *35*(3), 2009–2016. https://doi.org/10.1016/j.enpol.2006.03.014

Hamzaçebi, C., and Kutay, F. (2004). Yapay Sinir Ağları İle Türkiye Elektrik Enerjisi Tüketiminin 2010 Yılına Kadar Tahmini. Istanbul University - DergiPark. https://dergipark.org.tr/tr/download/article-file/76153

He, K., Zhang, X., Ren, S., and Sun, J. (2015). Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification. International Conference on Computer Vision. https://doi.org/10.1109/iccv.2015.123

Hong, T. (2014). Energy Forecasting: Past, Present, and Future. Foresight: The International Journal of Applied Forecasting, 32, 43–48. https://EconPapers.repec.org/RePEc:for:ijafaa:y:2014:i:32:p:43-48

Hotunluoğlu, H., and Karakaya, E. (2011). Forecasting Turkey's Energy Demand Using Artificial Neural Networks: Three Scenario Applications. *Ege Academic Review*, *11*(5), 87–94.

Ishik, M. Y., Goze, T., Ozcan, I., Gungor, V. C., and Aydin, Z. (2015). Short term electricity load forecasting: A case study of electric utility market in Turkey. 2015 3rd International Istanbul Smart Grid Congress and Fair (ICSG). https://doi.org/10.1109/sgcf.2015.7354928

J. T. Connor, R. D. Martin, and L. E. Atlas, "Recurrent neural networks and robust time series prediction", IEEE transactions on neural networks, vol. 5, no. 2, pp. 240–254, 1994.

Jagtap, R. (2022, November 21). Understanding the Markov Decision Process (MDP). Built In. https://builtin.com/machine-learning/markov-decision-process

Melikoglu, M. (2018). Vision 2023: Scrutinizing achievability of Turkey's electricity capacity targets and generating scenario based nationwide electricity demand forecasts. *Energy Strategy Reviews*, *22*, 188–195. https://doi.org/10.1016/j.esr.2018.09.004

Mulla, M. Z. (2021, December 14). Cost, Activation, Loss Function|| Neural Network|| Deep Learning. What are these? *Medium*. https://medium.com/@zeeshanmulla/cost-activation-loss-function-neural-network-deep-learning-what-are-these-91167825a4de

Neves, A. C., González, I., Leander, J., and Karoumi, R. (2017). A New Approach to Damage Detection in Bridges Using Machine Learning. *Lecture Notes in Civil Engineering*, 73–84. https://doi.org/10.1007/978-3-319-67443-8_5

Oğcu, G., Demirel, O. F., and Zaim, S. (2012). Forecasting Electricity Consumption with Neural Networks and Support Vector Regression. Procedia - Social and Behavioral Sciences, 58, 1576–1585. https://doi.org/10.1016/j.sbspro.2012.09.1144

Pascanu, R., Gulcehre, C., Cho, K., and Bengio, Y. (2014). How to Construct Deep Recurrent Neural Networks. International Conference on Learning Representations. http://lib-arxiv-008.serverfarm.cornell.edu/pdf/1312.6026

PwC Turkey & Presidency of the Republic of Turkey. (2021). Overview of the Turkish Electricity Market. In https://www.pwc.com.tr/. PwC Turkey. Retrieved January 28, 2023, from https://www.pwc.com.tr/overview-of-the-turkish-electricity-market

Sak, H., Senior, A., and Beaufays, F. (2014). Long short-term memory recurrent neural network architectures for large scale acoustic modeling. *Interspeech 2014*. https://doi.org/10.21437/interspeech.2014-80

Sanlı, B. (2022, January 3). Was 2021 Just the Beginning of an Energy Crisis? - Barış Sanlı. Bilkent EPRC. https://www.bilkenteprc.com/post/was-2021-just-the-beginning-of-an-energy-crisis-bar%C4%B1%C5%9F-sanl%C4%B1

Türkiye - Countries & Regions. (n.d.). IEA. Retrieved February 4, 2023, from https://www.iea.org/countries/turkiye

Türkünoğlu, A. (2019, May 13). *Short Term Electricity Consumption Forecasting Using Long Short-term Memory Cells*. https://polen.itu.edu.tr/items/fef15046-a45d-44dd-a9ce-b34a61924a00

Understanding LSTM Networks -- colah's blog. (2015, August 27). Retrieved December 3, 2022, from https://colah.github.io/posts/2015-08-Understanding-LSTMs/

Uslu, F.M., Sanlı, B., and Temur, T. (2013, October). Türkiye Aylık Elektrik Talep Modeli. Http://Www.Barissanli.Com/. Retrieved December 3, 2022, from http://www.barissanli.com/calismalar/2013/mfuslu-bsanli-ttemur-AETM.pdf

Veljanovski, G., Atanasovski, M., Kostov, M., and Popovski, P. (2020). Application of Neural Networks for Short Term Load Forecasting in Power System of North Macedonia. 2020 55th International Scientific Conference on Information, Communication and Energy Systems and Technologies (ICEST). https://doi.org/10.1109/icest49890.2020.9232674

Yukseltan, E., Yucekaya, A., and Bilge, A. H. (2017). Forecasting electricity demand for Turkey: Modeling periodic variations and demand segregation. *Applied Energy*, *193*, 287–296. https://doi.org/10.1016/j.apenergy.2017.02.054

CURRICULUM VITAE

Personal Information

Name and Surname: Berk Dede

Academic Background

- Kadir Has University
- Energy Systems Engineering (2013-2018)

- Kadir Has University
- Sustainable Energy and Development (2020-Cont)

Work Experience

- Sancak Enerji
- Energy Trade Asst. Specialist (05.2019 – 11.2019)

- CENAL Elektrik Üretim A.Ş
- Energy Planning Asst. Specialist (10.2021 – Cont)