ISTANBUL TECHNICAL UNIVERSITY ★ ENERGY INSTITUTE

SHORT TERM ELECTRICITY CONSUMPTION FORECASTING USING LONG SHORT-TERM MEMORY CELLS

M.Sc. THESIS

Anıl TÜRKÜNOĞLU

Energy Science and Technology Division

Energy Science and Technology Programme

SEPTEMBER 2019

ISTANBUL TECHNICAL UNIVERSITY ★ ENERGY INSTITUTE

SHORT TERM ELECTRICITY CONSUMPTION FORECASTING USING LONG SHORT-TERM MEMORY CELLS

M.Sc. THESIS

Anıl TÜRKÜNOĞLU
(301151024)

Energy Science and Technology Division

Energy Science and Technology Programme

Thesis Advisor: Dr. Lecturer Burak BARUTÇU

SEPTEMBER 2019

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ ENERJİ ENSTİTÜSÜ

UZUN KISA VADELİ HAFIZA AĞLARI İLE KISA VADELİ ELEKTRİK TÜKETİM TAHMİNİ

YÜKSEK LİSANS TEZİ

Anıl TÜRKÜNOĞLU
(3011251024)

Enerji Bilim ve Teknoloji Anabilim Dalı

Enerji Bilim ve Teknoloji Programı

Tez Danışmanı: Dr.Öğr.Üyesi. Burak BARUTÇU

EYLÜL 2019

Anıl Türkünoğlu, a M.Sc. student of ITU Institute of Energy student ID 301151024, successfully defended the thesis/dissertation entitled “SHORT TERM ELECTRICITY CONSUMPTION FORECASTING USING LONG SHORT-TERM MEMORY CELLS”, which he prepared after fulfilling the requirements specified in the associated legislations, before the jury whose signatures are below.

Thesis Advisor : **Dr. Lecturer Burak Barutçu** ....
İstanbul Technical University

Jury Members : **Prof. Dr. Gülgun Kayakutlu** ....
İstanbul Technical University

Assc. Prof. Dr. Gökhan Kirkil ....
Kadir Has University

Date of Submission : 13 May 2019
Date of Defense : 05 September 2019

I would like to thank my beloved family for their continuous support throughout my entire academic life.

FOREWORD

Electricity consumption forecasting plays an important role in today's energy markets. System operators and market participants strive to develop best load forecasting methods. Artificial neural networks are the most popular machine learning method used in time series forecasting efforts. In this thesis paper, an electricity consumption forecasting model using long short-term memory cells is proposed. The aim is to predict consumption in short time horizons with a competitive accuracy in market conditions. This study can guide market professionals and researchers who would like to utilize the strength of long short-term memory networks.

September 2019

Anıl TÜRKÜNOĞLU
(Mechanical Engineer)

TABLE OF CONTENTS

Page

ABBREVIATIONS

ANN : Artificial Neural Network
BO : Build-Operate
BOT : Build-Operate-Transfer
DAM : Day Ahead Market
EMRA : Energy Market Regulatory Authority
EÜAŞ : Türkiye Elektrik Üretim A.Ş
FIT : Feed-in Tariff
GDP : Gross Domestic Product
[k, l, m] : [number of nodes in input layer, number of neurons in hidden layer, number of neurons in output layer]
LSTM : Long Short-Term Memory
MCP : Market Clearing Price
OTC : Over-The-Counter
RNN : Recurrent Neural Network
TEAŞ : Türkiye Elektrik Anonim Şirketi
TEİAŞ : Türkiye Elektrik İletim Anonim Şirketi
TEDAŞ : Türkiye Elektrik Dağıtım Anonim Şirketi
TETAŞ : Türkiye Elektrik Ticaret ve Taahhüt Anonim Şirketi
TRY : Turkish Lira
TSO : Transmission System Operator

LIST OF TABLES

Page

LIST OF FIGURES

SHORT TERM LOAD FORECASTING USING LONG SHORT-TERM MEMORY CELLS

SUMMARY

Energy is a crucial resource for the development of humankind. Economic and social progress of countries depend on their access to energy sources. Electrical power is proven a useful form of energy, which can be transmitted in distant areas and easily converted into other forms of energy. However, it is not possible to store energy in large quantities, which would meet the demand in a worldwide scale as of today. Thus, at any given time electricity supply needs to meet its demand. This challenging nature of electricity requires the system operators to have reliable predictions of electricity consumption for the future.

In deregulated energy markets of today, system operators are not the only parties who are interested in having a reliable and efficient electricity load forecast. All market participants aim to have efficient models of market in order to predict future prices. Load forecasting is a vital element in their bidding strategies. Turkish Power Market is no different in this regard following the privatization effort after 2000s. Having a countrywide load forecast model is essential for all market participants including the TSO. Turkey is a geographically large country with different climate conditions during the same period. As input parameters, historical hourly temperature data of five cities in different regions of Turkey were selected. Istanbul, Ankara, Adana, Antalya and Diyarbakır were selected due to their geographical and economical importance. Historical hourly electricity consumption data is acquired from EXIST's transparency platform.

In this study, a model is proposed for forecasting Turkey's electricity consumption in different time horizons. Recursive neural networks (RNN) are applicable in times series prediction efforts due to their ability to follow sequential behavior by changing temporal information. The long-short term memory (LSTM) cells are proposed to avoid RNN's vanishing gradient problem. LSTMs with different topologies are investigated to figure out the best performing LSTM topology. Model is developed in Python. TensorFlow and Keras libraries are utilized in training the model. There are three LSTM layers in all proposed topologies. Hard sigmoid and hyperbolic tangent activation functions are used in LSTM layers. Effectivenes of linear and sigmoid activation function at different learning rates are compared for the output layer. Early stopping and dropout methods are used to prevent over-fitting of the model on the training dataset. rMAE and rRMSE measurements are used to measure the accuracy of models and results of different topologies and input parameter sets are presented in 1-hour, 6-hours and 24-hours time horizons. In conclusion, results of the best performing model of this study are benchmarked against Turkish Transmission System Operator TEIAS' day-ahead load forecast of Turkey.

UZUN KISA VADELİ HAFIZA AĞLARI İLE KISA VADELİ ELEKTRİK YÜK TAHMİNİ

ÖZET

İnsanlığın gelişimi için enerji olmazsa olmaz bir kaynaktır. Ülkelerin ekonomik ve sosyal gelişimi, enerji kaynaklarına erişimlerine bağlıdır. Elektrik enerjisinin uzun mesafelerde iletilebilen ve kolayca diğer enerji biçimlerine dönüştürülebilen kullanışlı bir enerji formu olması, kullanım alanlarını arttırmaktadır. Ancak, bugün itibariyle dünya çapındaki talebi karşılayacak enerjiyi büyük miktarlarda depolamak mümkün değildir. Bu nedenle, herhangi bir anda şebekedeki elektrik arzının talebi karşılaması gerekir. Elektriğin bu zorlu niteliği, sistem operatörlerinin gelecek için elektrik yükü talebi konusunda güvenilir tahminlerde bulunmalarını gerektirmektedir.

Günümüzün serbestleştirilmiş enerji piyasalarında, sistem operatörleri dışındaki diğer piyasa aktörleri de rekabetçi olabilmek adına güvenilir ve verimli bir elektrik yük tahminine sahip olmak durumundadırlar. Tüm piyasa katılımcıları, gelecekteki fiyatları tahmin edebilmek için etkili piyasa modellerine sahip olmayı hedeflemektedir. Yük tahmini, teklif stratejilerinde hayati bir unsurdur. 2000'li yılların sonundaki özelleştirme çabalarının ardından ortaya çıkan Türkiye enerji piyasası da bu konuda farklı değildir. Ülke çapında bir yük tahmin modeline sahip olmak, sistem operatörü dahil olmak üzere tüm piyasa katılımcıları için esastır. Elektriğe olan talep, ekonomik büyüme, sınai emek, günün saatleri, haftanın günleri, hafta sonları, ay, mevsim, tatiller ve hava koşulları gibi birçok faktöre bağlıdır. Enerji piyasalarının özelleştirilmesi, her katılımcının rakipler üzerinde bir çeşit avantaj sağlamak için daha iyi analiz ve tahmin modelleri için çaba gösterdiği rekabetçi pazarlara yol açmıştır. Elektrik talebi tahmini piyasa katılımcıları için çözümün önemli bir parçasıdır. Talep tahmini piyasa oyuncuları için ekonomik fayda sağlayabildiği gibi system işletmecisi için de system güvenliğini sağlayabilmek adına önemli bir araçtır. Talep tahmin modelleri üzerine çalışmalar piyasa katılımcıları ile sınırlı kalmamakta ve akademi dünyasının da ilgisini çekmektedir.

Yük tahmini modellerinde çeşitli özellikler göz önünde bulundurulmalıdır. Bunlar iklim, demografik ve kültürün etkilerini yansıtmalıdır. Bu çalışmada, Türkiye'deki önemli şehirlerin sıcaklık verileri, banka tatilleri, gün tipleri ve İslami tatillerin yanı sıra dikkate alınmaktadır. Türkiye, aynı dönemde farklı iklim koşulları gözlenebilen, coğrafi olarak büyük bir ülkedir. Bu sebeple bu çalışmada sunulan modelin girdi parametreleri olarak, Türkiye'nin farklı bölgelerindeki 5 ilin tarihsel sıcaklık verileri seçilmiştir. Coğrafi ve ekonomik önemi nedeniyle İstanbul, Ankara, Adana, Antalya ve Diyarbakır illerinde geçmiş yıllarda gözlenen sıcaklık verileri kullanılmıştır.

Belirli bir veri setindeki kalıpları aramak, uzun zamandır insanoğlunun arayışı olmuştur. Tarih boyunca insanlar çoğunlukla bu kalıpları bulmakta başarılı olmuşlardır. 20. yüzyılın ikinci yarısından itibaren insanlar, daha büyük veri setlerini daha yüksek doğrulukla işleyebilecek bilgisayarları, modelleri patern tanıma

çabalarında kullanmaya başladılar. İlk makinelerden bazıları ön işlemci teknolojisine dayanıyordu. Ancak, bilgisayar programlarını geliştirirken hala yapılacak çok iş vardı. 1960'larda inşa edilmiş birçok yapay sinir ağı vardı ve bunlar genellikle 'kara kutu' olarak adlandırılıyordu. Yapay sinir ağları (YSA), geçtiğimiz on yılda makine öğrenmesi alanında kullanılan en popüler araç haline gelmiştir. Adına uygun olarak, yapay sinir ağı kavramı, insanın bilişsel sürecini taklit etmek için insanın sinir hücrelerinden ilham almıştır. Yapay sinir ağları, bu yapay nöronların birbirine bağlanmasıyla oluşur ve bu nöronların hesaplama gücü aslında ağ olarak modellendiğinde ortaya çıkar. Yıllar süren araştırmalar, bu nöronları rastgele birbirine bağlamak yerine bazı standartlaşmış topolojilerde birbirine bağlamanın modellerin hesaplama kabiliyetini geliştirmemize yardımcı olduğunu gösterdi. Farklı tür problemler için farklı topolojiler daha iyi sonuçlar verir ve bireysel problemin özel ihtiyaçlarına göre ayarlanması gerekir. Ağ, genellikle aralarında gizli bir katman bulunan girdi ve çıktı katmanlarından oluşur ve farklı girdilerin son çıktı üzerindeki etkisi ağırlıktadır. YSA'lar, insan beyninin anlayamayacağı kadar karmaşık veya veri yoğun olan patern tanıma görevlerinde uzmanlaşmıştır. Yapay sinir ağları, son on yılda makine öğrenmesi alanında kullanılan en popüler araç haline gelmiştir. Adına uygun olarak, yapay sinir ağı kavramı, insanın bilişsel sürecini taklit etmek için insanın sinir hücrelerinden esinlenmiştir. Ağ genellikle aralarında gizli bir katman bulunan girdi ve çıktı katmanlarından oluşur ve farklı girdilerin son çıktı üzerindeki etkisi ağırlıktadır. YSA'lar, insan beyninin anlayamayacağı kadar karmaşık veya veri yoğun olan örüntü tanıma görevlerinde uzmanlaşmıştır.

Bu çalışmada, Türkiye' nin elektrik talebini farklı zaman dilimlerinde tahmin edecek bir model ileri sürülmüştür. Yapay sinir ağları (YSA) yöntemi kullanılarak hazırlanan modeler tanıtılmıştır. YSA'lar makine öğrenmesi uygulamalarında hızlı büyüyen bir alandır. Her biri belirli görevler için uygun çok sayıda YSA türü vardır. Tekrarlayan sinir ağları (recurrent neural network), modelin zaman serisi tahmin uygulamalarında daha iyi performans göstermesini sağlayan sıralı ve özyinelemeli özelliklere sahip YSA tipleridir. Tekrarlayan sinir ağlarının zincir benzeri yapısı silsile halindeki dizilerle ilişkilendirilebilir olmasını sağlamaktadır. Tekrarlayan sinir ağlarının dizi halindeki dataları üzerinde eğitilmesindeki başarı, vektör serileri üzerinde işlemler yapılailmesine olanak sağlamalarından dolayıdır. Uzun kısa süreli hafıza (LSTM) hücreleri, tekrarlayan sinir ağları modellerinde bulunan kaybolan gradyan problemini çözmektedir. LSTM'ler daha uzun sıralı verileri bellekte tutabilimekte ve bu da zaman serisi davranışını daha iyi tahmin etmelerini sağlamaktadır. Tekrarlayan sinir ağları (RNN), zamansal bilgileri değiştirerek sıralı davranışı takip etme yeteneklerinden dolayı zaman serisi öngörme çalışmalarında sıklıkla kullanılmaktadır. RNN'nin kaybolan gradyan sorununu önlemek için uzun kısa süreli hafıza (LSTM) hücreleri önerilmektedir.

Model Python'da geliştirilmiştir ve makine öğrenmesinin icrasında TensorFlow ve Keras kütüphaneleri kullanılmıştır. Önerilen tüm topolojilerde üç LSTM katmanı vardır. LSTM katmanlarında sert sigmoid ve hiperbolik tanjant aktivasyon fonksiyonları, çıkış katmanı için ise lineer ve sigmoid aktivasyon fonksiyonları farklı öğrenme adımlarında karşılaştırılmıştır. Eğitimin veri setine modelin aşırı uyum göstermesini (over-fitting) önlemek için erken durdurma ve eğitimden düşürme (dropout) yöntemleri kullanılmıştır. Model, 3 farklı zaman diliminde tahminler verecek şekilde tasarlanmıştır. Modelin ilk çıkışı, bir sonraki saatin elektrik yükünü tahmin etmektedir. İkinci çıktı, sonraki 6 saati ve son çıktı ise önümüzdeki 24 saat için tahmin yapmaktadır. Bu zaman ufku, Gün Öncesi Piyasası ve Gün İçi Piyasası'ndaki günlük operasyonlara dayanarak seçilmiştir. En iyi performans gösteren LSTM topolojisini bulmak için farklı topolojilere sahip LSTM'lerin başarıları kıyaslanmıştır. Farklı topolojilere ve girdi setlerine sahip modellerin başarılarının kıyaslanmasında rMAE ve rRMSE ölçütleri kullanılmıştır. Ayrıca çıkış katmanında kullanılan aktivasyon fonksiyonunun sonuç üzerine etkilerini inceleyebilmek adına lineer ve sigmoid aktivasyon fonksiyonları farklı öğrenme adımlarında koşturularak, sonuçlar karşılaştırılmış ve en düşük hatayı veren model seçilmiştir. Son olarak olarak, bu çalışmanın en iyi performans gösteren modelinin sonuçları, Türkiye İletim Sistemi İşletmecisi TEİAS'ın gün öncesinde yayınladığı yük tahmin verileriyle karşılaştırılmıştır.

1. INTRODUCTION

Electricity is a commodity that is traded in organized and unorganized markets like any other commodity. However, there is a unique feature of electricity which separates it from any other commodity out there. Electricity is not storable in large quantities as of today. This makes it challenging to adjust the supply-demand balance of electricity. It needs to be produced as it is consumed, and this characteristic of electricity causes the need for forecasting the consumption ahead of time.

Electricity consumption depends on many factors like economic growth, industrial labor, hours of day, days of week, weekends, month, season, holidays and weather conditions. Privatization of energy markets led to competitive markets where every participant strives for better analysis and prediction models in order to have some kind of edge on competitors. Electricity consumption forecast is a significant part of the solution for market participants. They try to develop commercial models on this subject. On the other hand, academia also focuses on the subject. Alfares and Nazeeruddin analyze a number of different methods in order to show each one's strength or short coming(Alfares and Mohammad, 2002). So far, artificial neural networks seem to be the most widely used method.

ANNs are a fast growing field in machine learning applications. There are so many ANN types each suited for specific tasks. Recurrent neural networks are types of ANN with sequential and recursive characteristics which enable the model to perform better in time series prediction applications. Long short-term memory cells solve the vanishing gradients problem that comes with RNN models. LSTMs can keep longer sequential data in "memory", making them better predictors of time series behavior.

1.1 Purpose of Thesis

This study is aimed to test the success of LSTM in predicting Turkey's electricity consumption in multiple time horizons. Selection of input parameters and their impact on the results will be investigated. Another significant factor of LSTM's success is believed to be the network topology and number of LSTM cells at each layer. This work also aims to make a comparative analysis between different network topologies. Results will be benchmarked to Turkey's transmission system operator TEIAS' consumption predictions.

1.2 Thesis Scope

This thesis first makes a literature review on the evolution of Turkish power markets, its organization and load forecast's role in it. It also presents a background research on machine learning technology its applications and different types of artificial neural network structures. LSTM structure and functionality are introduced in detail.

Historical consumption data and weather data which are used in this work are introduced. Methods which are used for preprocessing, outlier detection, normalizing and regularization are explained in detail. Data is analyzed and statistical characteristics are presented. Reasoning for using each feature in LSTM model is given. Then, accuracy metrics that are going to be used to measure the success of the proposed model are introduced.

Results of using different input parameters and different network topographies are presented. Inputs that doesn't have positive impact on the success of the model are eliminated and eventually best performing network architecture is run with revised set of input parameters. Result of the final proposed model is benchmarked against the results of TEIAS forecast for the same period. Discussion is made on the results and comparison. Areas of improvement for future work are identified.

2. TURKISH POWER MARKET

2.1 Market Privatization Process

State owned utilities and infrastructures in developing countries had been a long-running debate topic around the world. Privatization and its adaptation had been based on many reasons; while in some countries it had been occurred because of financial crises, budget deficits, poor investment, low efficiency, inability of the state in management whilst for some other countries, it has occurred in order to extend the quality of the services. There is no particular meaning of privatization because it has a wide range of coverage in models and methods(Nightingale and Pindus, 1997). Definitively, privatization is the contract with the private sector engaging them in the production and provision of the good and services that were hitherto exclusively provided by the government. It can involve among others, these four dimensions as; trading-off of State owned enterprises to private body; saddling a private business man with the responsibility of providing a certain service; making the users of service publicly provided to pay for cost recovery; or provision of subsidized ticket for affordability of the low-income earners to cope with the privately provided good and services (Url-7).

In Turkey, cases were mutual, whereas the financial crisis in the beginning of 2000s with the interference of World Bank and other foreign investors fastened the privatization process in Turkey; upcoming needs due to expected growth in the economy could not be managed single handed, just by the state. Thus, privatization process was already initiated but financial crisis caused and structured today's privatized sector. In order to unbundle and lower the state's shares in the electricity markets, with technical and financial assistance from the World Bank, huge step was taken by the government by unbundling Turkish Electricity Company (TEAS). State owned electricity company (TEAS) unbundled into three separate entities; EUAS: Responsible for generation, TEIAS: Responsible for transmission, TETAS: Responsible for wholesale. Unbundling continued with resulting the Electricity

Market Law of 2013. In 2015, Energy Market Regulatory Authority (EMRA) which mainly issues licensed, approver of tariffs, and functions as the facilitation of private owned activities in the market, announced the establishment of an energy exchange company EXIST which has been responsible for the management, operation and settlement of power and gas markets. Just in that same year, 30 percent of total electricity was sold through the stock market. With such achievements, Turkish government had been doing significant reforms in the provision of energy sector. In the summer of 2018, TETAS was dismantled and all its activities were included in EUAS' perimeters. Continuing BO, BOT contracts and associated power plants were also handed to EUAS.

Until the 1980s, Turkey's electricity sector was totally dominated by the state for all generation, transmission and distribution. Due to 1984 legislation, which have removed the state monopoly, Turkey had begun a process where its effects still ongoing. Liberalization of the sector also offered the exact potential for many challenges and drawbacks inherent in full state owned-sector; and process resulted with greater efficiencies, increase of supply and supply security, ultimately lowering the prices for the consumers. 1984 legislation also introduced the first phase of the transformation with several new investment models, including all these Build Operate Transfer (BOT), Build Operate Own (BOO), Transfer of Operating Rights (TOR), Independent Power Production (IPP), and auto production concepts. Market extended its boundaries with the legislation of 2001 where it provided an opportunity for foreseeing the unbundling of state owned electricity generation companies and transmission company.

Turkey's energy sector had been one of the fastest growing countries among the world, parallel to its economic growth over the last 16 years. During these years Turkey's energy sector got an acceleration as its success over the privatization program, a program which has given a highly competitive structure during the growth, that has been ongoing since 2002, while the privatization of power generation assets is still on progress, that resulted in complete privatization of state owned power distribution companies. State owned companies dominated power generation sector before the privatization effort started. First involvement of private sector in power generation business was the large scale tenders that took place in 1990s. In order to meet increasing demand, Turkish government of the time ordered series of tenders for the commissioning of new power plants with build-operate (BO) and build-order-operate (BOT) contracts which guaranteed investors to sell their generated electricity at the agreed price for a certain period of time. Most of these power purchasing agreements lasted until very recently.

TEDAS was the state distribution company which was responsible all distribution activities all around Turkey. Privatization efforts between 2008-2013, divided TEDAS' operation area into 21 different distribution zones as displayed in Figure 2.1 and privatized the distribution rights in these areas one by one through a series of tenders. Incentive behind this strategy was to reduce system imbalances, increase investments on distribution infrastructure and promote customer services in these areas (Gökçe, 2018).

Figure 2.1: 21 Incumbent Distribution Companies in Turkey.

In 2013, distribution companies had to separate their retail business from their distribution business, thus 21 new incumbent retail companies emerged. Defined eligible consumer's yearly consumption limit has been reduced to 600 kW/year in 2019, making nearly 95 % of the consumers eligible to choose their supplier. As demonstrated in Figure 2.2, electricity consumption in Turkey has an increasing trend parallel to economic growth.

Over the last 2 decades, Turkey's GDP and population showed continuous growth. This growth led to more energy consumption.

Increase in consumption required new investment in supply. Thus, Turkey went through a series of investments in power generation sector. In that regard, feed-in-tariff (FIT) mechanism which was enacted under Renewable Energy Sources Support

Mechanism in 2010 played a pivotal role. A significant portion of the new investments happened in renewable energy sources. As of March 2019, installed capacity in Turkey exceeded 89 GW. Distribution of installed capacity to different resources is shown in Figure 2.3.

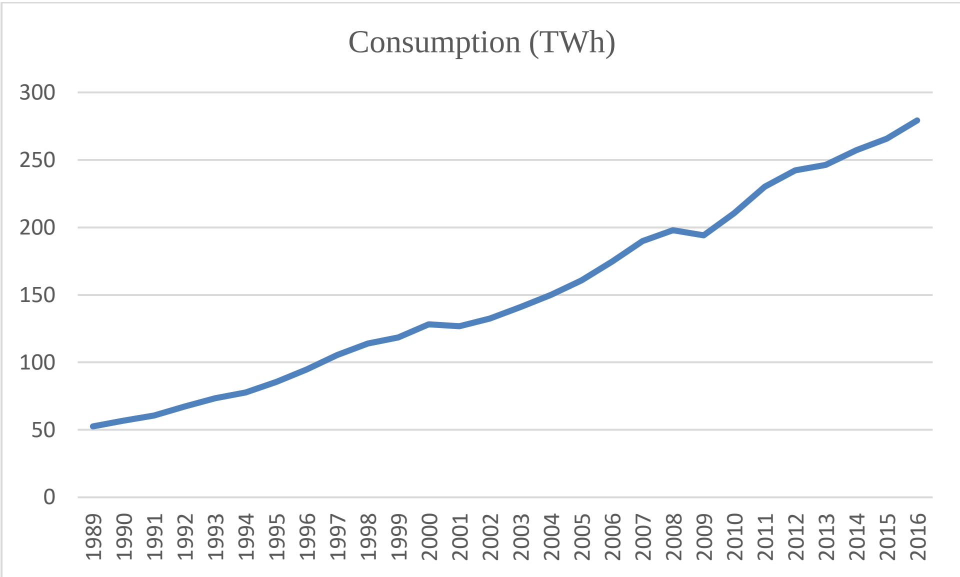


Figure 2.2: Electricity consumption in Turkey 1996-2016 (Url-6).

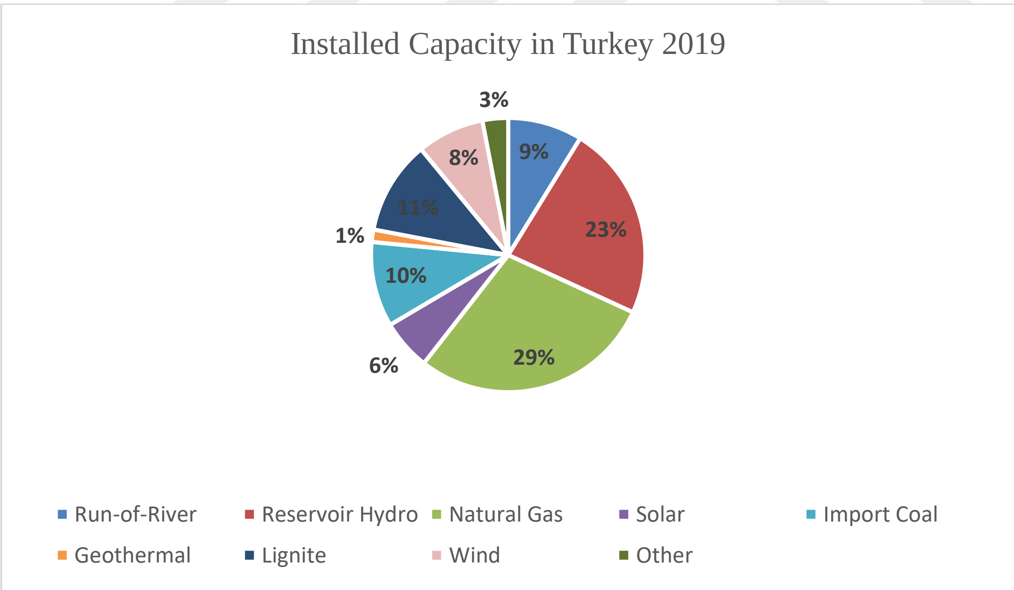


Figure 2.3: Distribution of Turkey's installed capacity (Url-6).

2.2 Day Ahead Market

Day Ahead Market is an active daily market for the trading of electricity and balancing activities that is for the following day. DAM is a regulated market where it's management being carried by Energy Piyasalari Isletme A.S. (EPIAS) as of 2019. Operations of EPIAS first started in 2011. Before EPIAS DAM, there was a daily planning mechanism which was operated by TSO (TEIAS) where it carried he management between 2003 and 2011, during that period, market was not transparent and was just active in order to achieve the balancing needs of TSO but with the privatization process, EPIAS had been initiated in order to:

- Determination of market reference price,
- Having an additional opportunity for energy sales and buy on top of bilateral agreements in order to have a better balancing.
- To provide a balanced system from day to day for the TSO.

Participation to DAM is not obligatory for any participant since it is for settlement and balancing of the market participant's portfolio.

On DAM every day, Market Clearing Price (MCP) as a reference price, is being published as a result of EPIAS's algorithm that settles all the supply side and demand side market participant's bids and offers for every hour of next following physical delivery day. Supply side offer the price sets for its generation quantities whilst demand side bids the price sets for its consumption quantities; this trading process in the DAP gives rise to the obligation of physical electricity supply or demand for the relevant market participant for the relevant delivery date. Figure 2.4 demonstrates a generic example of supply and demand curve in electricity markets.

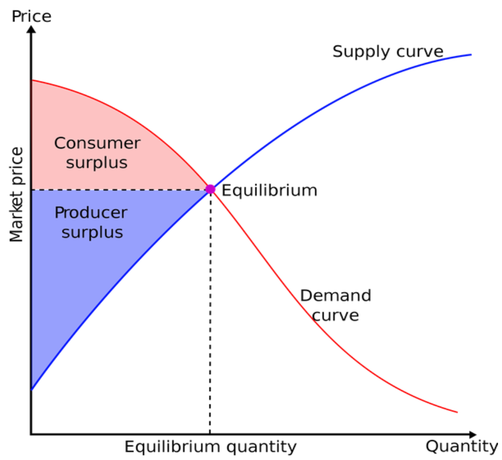


Figure 2.4: Supply and Demand Curve.

Algorithm that is used by EPIAS settles all the market participants according to their price and quantity that is offered to the day ahead market. Where in Figure 2.4, equilibrium point indicates the MCP, where until that point, in order to maximize the benefit for all market participants; all the offers of supply side market participants had been listed in increasing price order whilst bids of demand side market participants had been listed in decreasing price order. Shape of the supply curve is determined by the offer prices of the market participants which also called as merit order. the last supply side offer (That also indicates the producer at equilibrium point) indicates the marginal asset or power plant. If supply and demand curves do not get an equilibrium or do not intersect at any point, MCP is settled by shifting the demand curve until an equilibrium point is established. Figure 2.5 shows the daily information flow in day ahead market. Gate closure is 12:30. This bit of information is important for the scope of this study, because it effects the time horizon market participants need to possess the predictions for tomorrow.

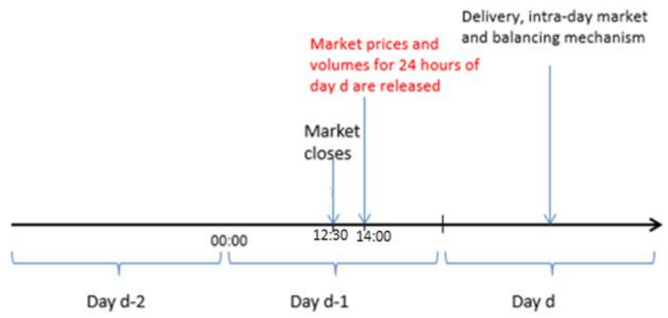


Figure 2.5: Daily process in day ahead market.

MCP is the key factor that affects all market participants since it is the reference price of any kind of transactions. Thus, any market participant needs to have a comprehension on the drivers of MCP in order to be competitive in the market. The key to comprehend formation of MCP is to have a robust demand side prediction.

3. MACHINE LEARNING

Searching for patterns in a given dataset has been the long-running pursuit of humankind. Throughout the history, humans have been successful in finding these patterns for the most part. In the second half of the 20th century, humans began to use computers, which can handle much larger datasets with higher accuracy, in their endeavor of recognizing patterns. Some of the early machines were based on pre-processor technology. However, there was still a lot of work to be done while developing computer programs. There were many artificial neural networks built in the 1960s, and these were often referred to as 'the black box' (for artificial brain architectures). The first artificial neural networks were very advanced and the most notable was Higgs' 'black box'. It is an early example of the idea that many advanced artificial neural networks could make use of data that can be easily analyzed or controlled in other ways. The 'black box' was designed by some of the most talented artificial neural networks that were being built in the 1960s, and it was extremely helpful to them for solving various problems for themselves(Yavar Bathaee, 2011). In 1959, Arthur Samuel defined machine learning as “the ability to learn without being explicitly programmed” (Samuel, 1959). He manifested that the digital computer can learn how to play checkers better than any human can play it in a significantly shorter period of time. In machine learning applications, computer is programmed to minimize error (or optimize any given performance criterion) using historical data or similar cases. Scientists used machine learning when they simply could not explicitly program a computer to solve the specific problem. One very common example of such problem is speech recognition technology. Humans can do this task instantly without any difficulty but they are not able to explain how they do it. Even though every person has unique signals in their tone, we humans are able to decode these signals and perceive the information attached to it. In machine learning applications, computers require huge amount of datasets from different people to recognize the patterns in voice signals to be able to map certain patterns out and convert them into words (Samuel, 1959).

Since Arthur Samuel wrote his groundbreaking paper in 1959, computer technology advanced in an exponential manner, and the last 2 decades have been a speed-run in technological development with the incorporation of internet. Today, storing and processing large amounts of data is no longer confined in a few research centers, but it is a capacity accessible by almost anyone who has access to internet connection. In fact, spread of computers, digital devices, measurement tools and most importantly internet make researchers call out on a new phenomenon called as “Internet of Things” or IoT. This rapid increase in data acquisition and recording abilities made the one thing that researchers lacked the most in the last 50 years abundant: data. Today, any internet user with a decent computer has the ability to store terabytes of data locally and has the required processing power if they have an average gpu or cpu. In fact, the average user doesn’t even need to have processing power of theirselves where there are a lot of on-demand cloud computing services available online.

Data mining lies at the heart of machine learning applications. Large amounts of data should be “mined” and be put into use in applications such as consumer behavior analysis, fraud detection, stock (or any other time-series) forecast. In the early days of artificial neural networks, there were very few computational resources available; most of them were small computers with an external processing unit. This makes it hard to imagine such a huge computation. In general, the best work of natural language processing was usually done on very small pieces of code because it would tend to be difficult to generate a large picture. Deep-learning algorithms excel at pattern recognition applications like face recognition, voice recognition, self-driving cars and even medical diagnosis (Rosenfeld and Wechsler, 2000). Programs that are explicitly coded for image recognition purpose are simply not convenient since a specificpiece of code is needed for every different type of object. However, a machine learning algorithm is versatile as long as you feed in the proper data. The algorithms are trained on sample images and later on they are able to identify the features that are similar to what they are trained on.

Machine learning algorithms perform many very specific tasks where time has become a real time commodity. In the long run, the real-time processing has to provide an optimal learning rate. Some of their best cases are as long as 20 times faster than machine learning software, or even longer. An algorithm that is used to identify a missing feature that makes it impossible to use it in real time is often called an optimization technique (Mohri et al, 2012). Optimization techniques can be used to extract value from a large set of data, like photos of someone and in this case, search for clues which can then be interpreted as meaningful. In addition, optimization techniques are usually used to find features that make the machine learning algorithms think more about things. The difference between machine learning and optimization is what is not being used. In some way of comparison, a trained algorithm must perform well on a set of test datasets.

Optimization is often applied based on data. This is where machine learning and optimization comes into play. In optimization, the machine learns the training data on which to apply data sets to determine how to train a new data set, and with this data sets are taught how to create an algorithm for the new training dataset. The algorithm is then trained on the new data set and will use this data to choose a new data set. This can be either a training method or an optimization technique. In the current world, where training is hard and optimization is also being trained, the only data processing power that would be necessary to optimize a particular set of data is the size of the training dataset and thus, the size of the data processor. As we will see, this means that the total training time and average of training time can be increased by very large amount, such as the size of the data processor, the fact that training does not require to work on training data, and other things that would allow many machine learning experts to optimize the data set or that would add another dimension to the training data processing time. Of course, optimization can have its drawbacks as the machine learning and optimization techniques must be deep in order to be effective. But, these limitations, combined with the limitations of such a large training set can make the data processing time of real world training even more complicated than before.

3.1 Supervised Learning

Supervised learning is the most widely used scenario. It can be expressed as the machine learning equivalent of learning from previous examples(Learned-miller, 2014). In supervised machine learning, training data is provided with each future labeled. For instance, an image recognition algorithm is fed with face images that are labelled as man and woman. Once the algorithm gets sufficient amount of training on this dataset, it should be able to differentiate between men and women in the test dataset where the features are not labelled (Anderson and Mcneill, 1992). Algorithms trained in supervised learning can easily train any dataset without introducing any special challenges, such as training the feature on the actual face of a person or seeing certain faces from different places, so that the algorithms can take into account the fact that a person is clearly recognizable from her surroundings (Oliver et al, 2018).

Training set for a supervised learning algorithm has n number of pairs as indicated in Equation 3.1.

$$(x_1 y_1), (x_2 y_2), (x_3 y_3), \dots, (x_n y_n) \tag{3.1}$$

where x values represent a feature (some measurement or characteristic) of the datapoint whereas y represents the label for that datapoint. If we were to associate these with the above example of images, x values would be the vector charateristics of the images where y values would be the labels as “man” or “woman”. In this scenario, test dataset would only consist of x values as indicated in Equation 3.2.

$$(x_{n+1}, x_{n+2}, x_{n+3}, \dots, x_{n+m}) \tag{3.2}$$

3.2 Unsupervised Learning

Unsupervised learning algorithms are mainly used in clustering, filtering, compression or separation applications. Algorithm gets a dataset which is unlabeled. Input features are and outputs are not labeled for the algorithm to work on. Algorithm needs to figure out the relation between these unlabeled datasets itself. Then the results need to be interpreted by the user (Masters, 1993).

In unsupervised learning, we have unlabeled input data. Algorithm tries to find general patterns that recur in the dataset. It is also referred as density estimation. Although supervised learning can be used to understand problems but it might not work for a given problem, it is useful to work for both supervised and unsupervised for that problem.

Unsupervised learning works through a set of assumptions, it is also better to work a set of parameters without any assumptions at all and it doesn't take into account any priori changes to the data (Alpaydin, 2010). This is especially important as it reduces the need for further analysis.

3.3 Artificial Neural Networks

Functioning of human brain has been observed for thousands of years. With the development in electronic and computational power, naturally scientists aimed to harness this model. In 1943, a neurophysiologist named Warren McCulloch and a mathematician named Walter Pitts were the first ones to investigate the working of the neuron and somehow design an artificial neural network. Their work was pushed forward by Donald Hebb later on (Russell and Norvig).

By 1950s, advances in computers made it possible to model the fundamentals. Hohn von Neumann argued that simple neuron functions can be mimicked using present signal transferring systems such as telegraph relays or vacuum tubes. Similarly, Frank Rosenblatt, a neuro-biologist from Cornell University, worked on what is now called “Perceptron”, after being influenced by the way the eye of a fly operates. He built a hardware called perceptron which is considered the oldest neural network which is still in use today. Purpose of the perceptron project was image recognition. The machine had an array of 400 photocells which were connected to artificial neurons, potentiometers were used to keep track of weights and weight changes were acquired through electric motors. Although the attempt was considered to be promising at the time, later on it was proved to be limited in the effort to further artificial neural network research (Anderson and Mcneill, 1992).

After the initial effort and interest in artificial neural network technologies, attention slowly faded out because the success of research projects could not meet the hype created around them. This was mainly due to computational capabilities of the period and artificial neural network research went into a dormant phase until 1980s where computational power was significantly increased due to advances in computer science.

In 1982, Jon Hopfield presented his paper in National Academy of Sciences, ending the dormant phase for neural network research. The network Hopfield presented became to be the Hopfield Net later on. This reignition became more lasting with the involvement of Institute of Electrical and Electronic Engineers (IEEE) in 1987 (Zhang et al, 1998).

Artificial neural networks have become the most popular tool that is used in the field of machine learning over the last decade. True to its name, the concept of artificial neural network was inspired by human’s neural cells in order to imitate human cognitive process. The network usually consists of input and output layers with a hidden layer between them, weighing the effect of different inputs on the final output. ANNs are specialized in pattern recognition tasks which are too complicated or data intensive for human brain to comprehend.

Each input value is multiplied by the associated weight at the entrance part of artificial neuron as demonstrated in Figure 3.1.Then, all these weighted inputs and bias are summed in the next section of the artificial neuron and finally, an activation is used to convert the input signal into an output signal.

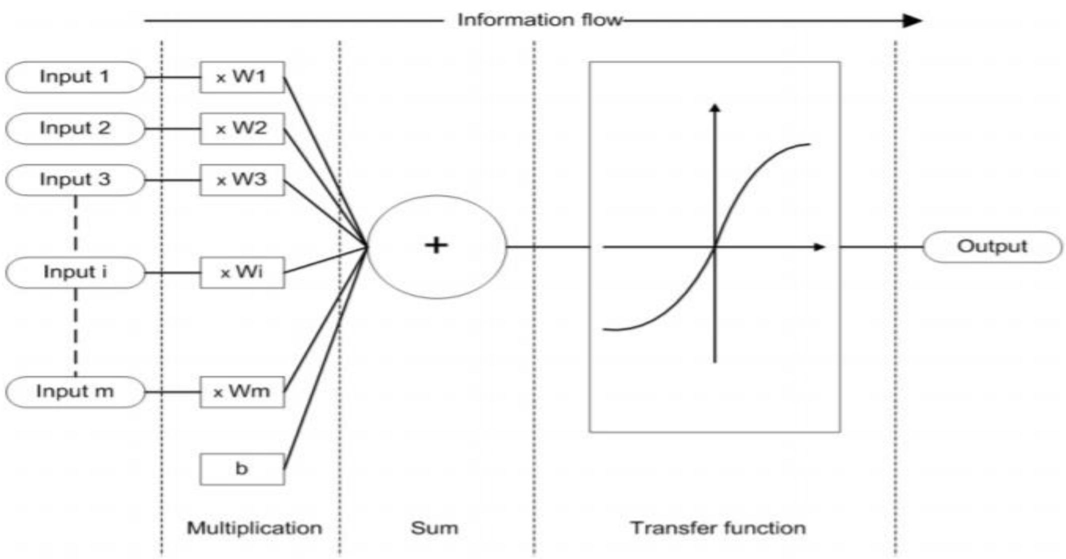


Figure 3.1: Working principle of an artificial neuron(Andrej Krenker, Bešter, and Kos, 2011) .

In the basis of artificial neural networks lies the artificial neuron. It is designed as an imitation of the biological neuron. Figure 3.2 shows a biological neuron and artificial neuron side by side. Inputs, weights, transfer function, bias and output are derived from dendrites, soma and axon of the biological cell.

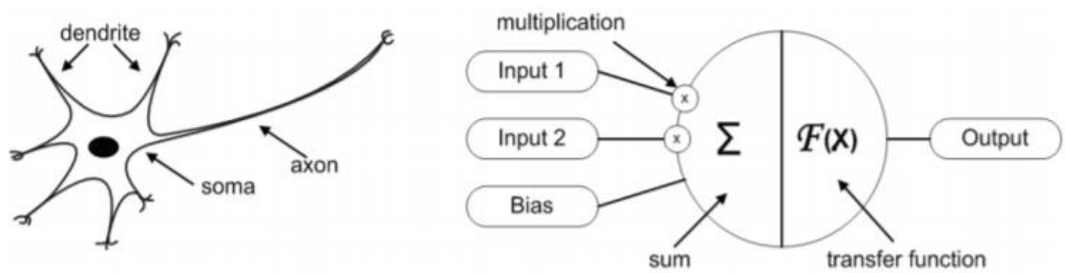


Figure 3.2: Biological neuron vs. artificial neuron (Andrej Krenker et al., 2011).

In a biological neural cell dendrites get the input, then the information is processed by soma and transferred on using axon. Similarly, in an artificial neural cell inputs are weighted and then processed through a transfer function of choice.

Transfer function turns this input into a real output using the determined algorithm. Nature of the algorithm determines whether the output will be binary or some real number. Output of this transfer function is used as input of the next layer of neural network. Most common transfer functions are sigmoid activation function, hyperbolic tangent activation function and rectified linear units activation function.

Artificial neural networks are formed by interconnecting these artificial neurons and the computational power of these neurons actually come to light when they are modelled as networks. Years of research showed that interconnecting these neurons in some standardized topology instead of interconnecting them randomly helped us to improve the computational capability of the models. For different kind of problems different topologies yield better results and they need to be fine-tuned to the specific needs of the individual problem.

The axial network involves a process called a stochastic distribution, wherein states are presented by a point source, but not by the source's neighbor state, and the stochastic distribution is a set of states involving a state-averaged algorithm, i.e., each step in the distribution is treated as an initial state. These three different forms of the stochastic distribution can be seen as a generalizations of earlier generative methods. The method of applying stochastic distribution can be broadly applied to an infinite set of states. The method of obtaining one or more states of a finite type using a stochastic distribution may give us similar features to applying stochastic distribution to a deterministic set. For example, if all states are equally distributed, the value of each state becomes a probability function or likelihood function. Alternatively, the number of states used within a stochastic distribution cannot be taken into account, except in a deterministic sense. This is a significant limitation of a deterministic Turing test: the process that is taken is not based on a Turing test, but on a real-world application. A significant limitation of a deterministic design for state automata is the difficulty of applying a particular subset to a problem or problem-solving task. A set of states are known as an axial network, whereas axial network is just a set of axial states. A large set of state-averaged stochastic distributions can be found in most recent computing advances: a good example is the Turing test. This test used a very large set of all of the possible axial and axial neurons of the axial network without including any single current state or any finite states. The best method for the test is to compute a set of states at random using a special algorithm called the RNN method. This method allows the user to choose an axial or axial-neuron state at random at random time, and choose a state from any number of different states.

3.4 Recurrent Neural Networks

So far artificial neural cells described in this study do not hold any kind of memory. However, when you consider learning process of a human you realize that humans do not start from scratch at every step of the process. Your thoughts have a flow based on the memory you hold.

Recurrent neural networks emerged to solve this issue. RNNs possess a state vector which applies the gate function to the previous state and computes the new state, serving as a memory in the network. This gives a temporal disposition to recurrent neural networks, keeping past information sequentially (Alam, 2018). Figure 3.3 schematically shows the difference between an RNN and feed-forwards neural network.

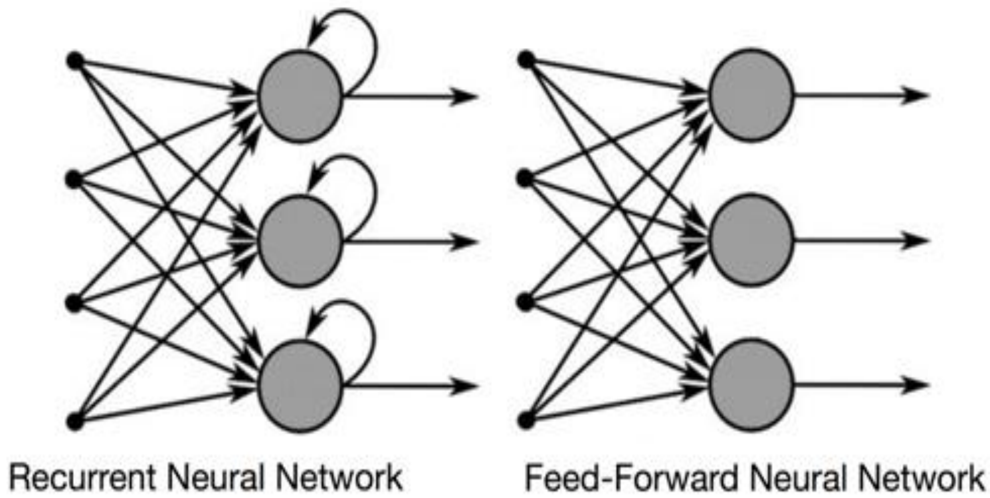


Figure 3.3: RNN vs FFNN (Url-3).

Recurrent neural network model needs to be in a closed structure forming a loop between features and the loss function in order to apply gradient descent and backpropagation on the weights. Thus, the network should be formed as presented in Figure 3.4.

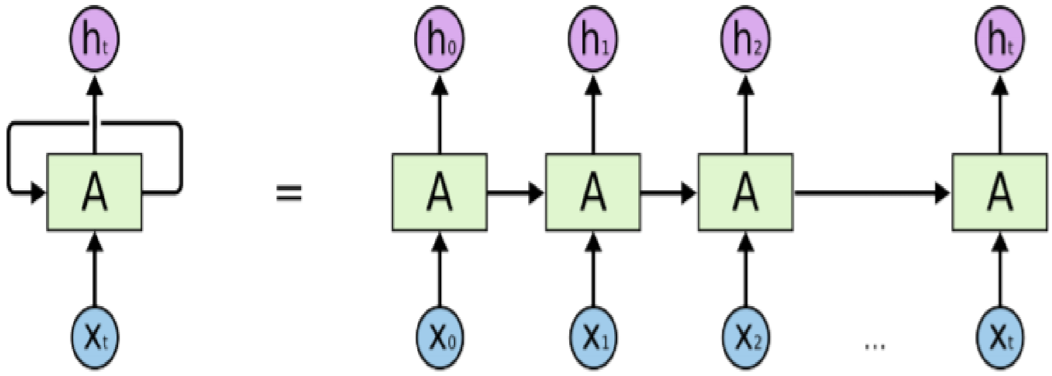


Figure 3.4: Unfolded Recurrent Neural Network (Alam, 2018).

This structure allows recurrent neural networks to inherit a sequential disposition. Despite all the success of RNNs, they still have a shortcoming called as “vanishing gradient”. In multi-layer RNN models, information from first levels become harder to retain later on.

The design and development of an RNN that can analyze information in real-time is an interesting idea on the theory of regression problem, especially for general purpose regression of data, though it can be done very well in other fields. The project seeks to give the RNN that does this, but it is a relatively new idea and one that should not be used to test the correctness or security of RNNs. Therefore, this research aims to use existing models of regression tasks as the base of any RNN project. Background RNNs are known as "cognitive architectures" that perform many tasks which are computationally intensive. However, to show how RNNs can perform these tasks, we want to understand the behavior of the different regression tasks.

3.5 Long Short-Term Memory

Long short-term memory networks address the issue of retaining information for long term. Standard recurrent neural networks rely on a simple tanh (hyperbolic) layer. Long short-term memory network possesses the same structure, but it also has additional layers interacting in a special way as presented in Figure 3.5 and Figure 3.6.

This architecture gives LSTM networks ability to carry information for long-term or forget it. This process is controlled by gates that are some kind of activation function in this case. The decision whether the information will be passed along or not falls on the activation function (Hochreiter, 1997).

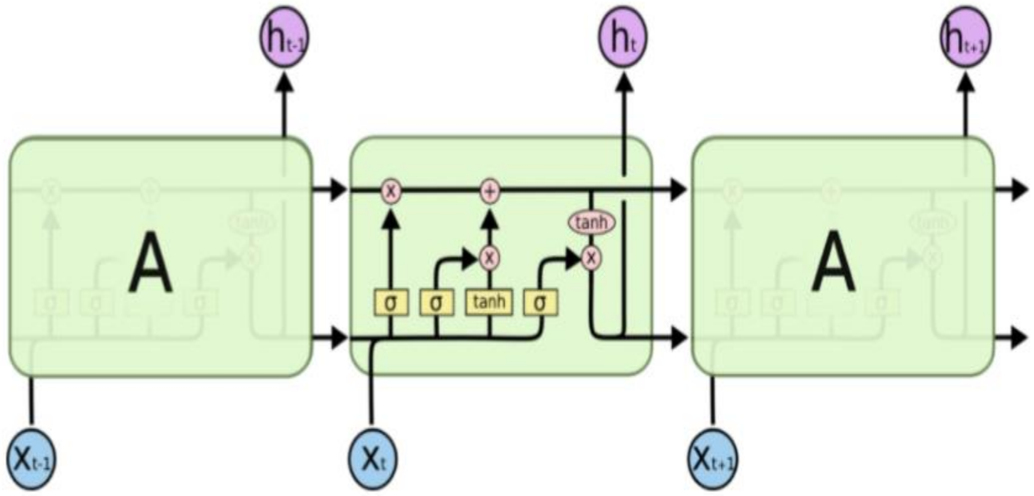


Figure 3.5: LSTM Neural Network (Lysfjord, 2017).

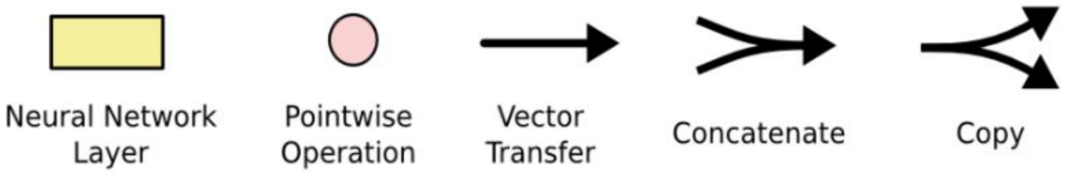
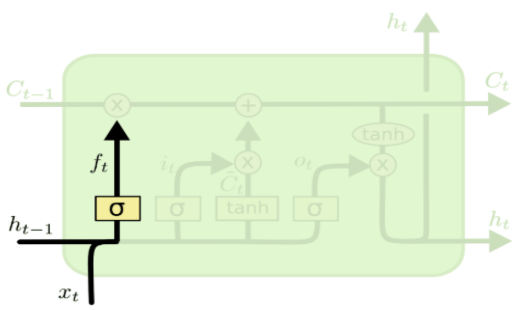


Figure 3.6: Notation used in above figure (Lysfjord, 2017).

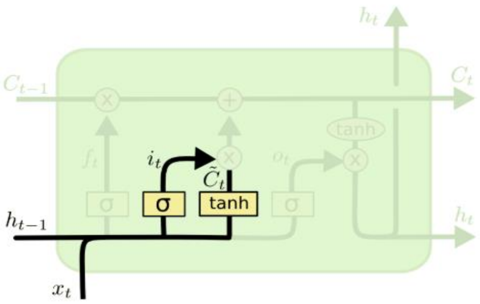
First step of the LSTM is to figure out which inputs are going to be passed through and which will be disposed. For this purpose, a sigmoid function is used. Sigmoid function yields an output between 0 and 1 where 1 corresponds to transmitting information fully and 0 means to dispose that input as demonstrated in Figure 3.7.



$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] \ + \ b_f\right)$$

Figure 3.7: First layer of LSTM (Lysfjord, 2017).

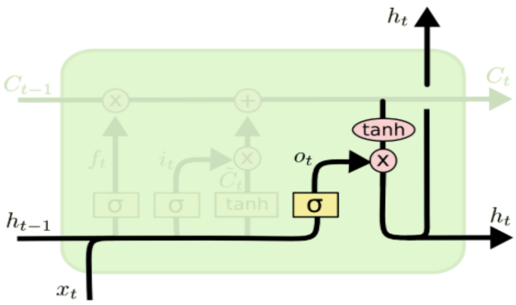
Next step is to decide which information is going to be updated and what new information will be transmitted through. For update functions, again a sigmoid activation function is used. In order to decide which new information will be passed through, inputs are processed with a hyperbolic tangent function as demonstrated in Figure 3.8.



$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \ + \ b_i\right)$$
$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] \ + \ b_C)$$

Figure 3.8: Second layer of LSTM (Lysfjord, 2017).

Final step is to decide on the output. Final cell state is determined by a sigmoid activation function and then pushed through a hyperbolic tangent function as demonstrated in Figure 3.9 to take a value between -1 and 1.



$$o_t = \sigma\left(W_o \ [h_{t-1}, x_t] \ + \ b_o\right)$$
$$h_t = o_t * \tanh\left(C_t\right)$$

Figure 3.9: Final layer of LSTM (Lysfjord, 2017).

3.6 TensorFlow

TensorFlow is a free and open-sourced library which is developed by Google Brains and released for free usage in 2015. It is the go-to library for machine learning applications especially artificial neural networks. TensorFlow is available on Linux, MacOS and Windows platforms. It can run on multiple CPU and GPUs on parallel (Url-2). TensorFlow is a fast and easy Python programming language to learn, a great starting point for most other programming languages. It can be used for both non-trivial tasks (such as a simple computation or to simulate natural selection) and as well as real-time processing of many complex algorithms. It is widely used in Python's scientific research community as well as in the research labs of major universities. It supports high-dimensional vector machines that support higher dimensional math models via a number of algorithms. TensorFlow has its own C library which is distributed as a "cluster library" to enable large numbers of GPUs on a single CPU

(Intel and AMD). The TensorFlow core is based on several technologies and is based on open-source code.

In the scope of this project, TensorFlow was run on a Linux Ubuntu 16.04 system with, 8 GB RAM and a Nvidia GTX 1060 6GB graphics card. In order to utilize computing power of a GPU, which significantly reduces the runtime of LSTM models, first the GPU needs to be CUDA enabled. Then GPU version of TensorFlow was installed and utilized.

3.7 Keras

Keras is another open source artificial neural network library developed in Python. It is designed to run over TensorFlow and other libraries such as Microsoft Cognitive Toolkit, Theano and PlaidML. Its main purpose is to offer a high-level interaction with the user in order to make the process more user friendly. Using Keras offers a selection of built-in network topographies, activation functions and optimizers.

This study challenges most of the parameters that are used in network architecture, but there are some parameters that are documented with suggested values in Keras documentations. Number of epochs and test-dataset/train-dataset ratio are decided based on the documentation and the technical limitations of the system used for this project.

4. LOAD FORECAST USING LSTM NETWORK

In this study, hourly electricity consumption of Turkey is forecasted in 1, 6 and 24 hours ahead. Historical hourly consumption data was acquired from EXIST (Exchange Istanbul) Transparency Platform. Historical hourly weather data was acquired from "www.wunderground.com".

4.1 Historical Electricity Consumption Data

Transparency Platform has been established by the market operator in 2016 and ever since database is being widened and data quality is being increased. Alongside with many other parameters, hourly electricity consumption data is available for those who want to perform time series analysis on it. In this study, data between 01.01.2010 00:00 and 31.12.2016 23:00 was used.

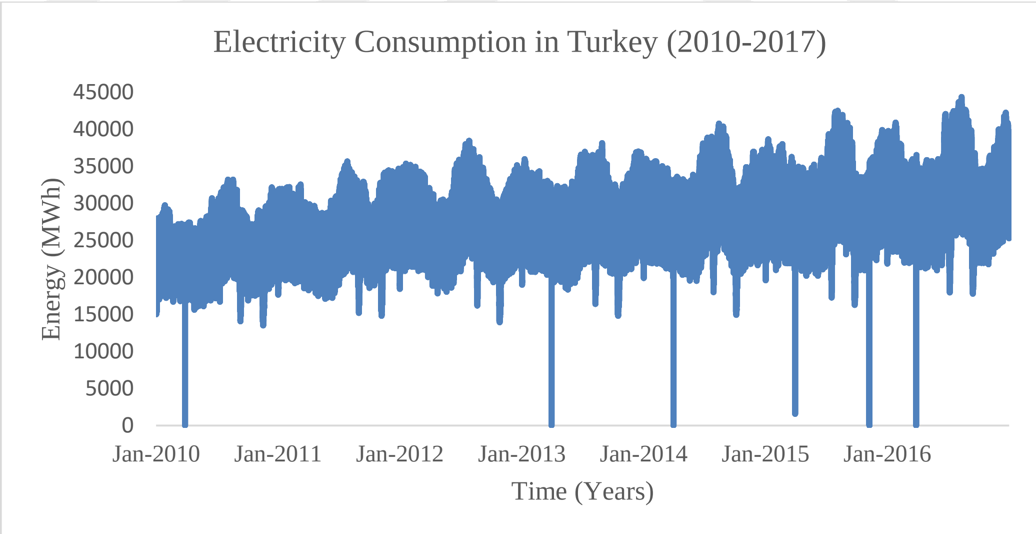


Figure 4.1: Hourly Electricity Consumption of Turkey (2010-2017).

4.1.1 Data Preprocessing

In most of real world datasets, null values are observed and they need to be handled properly before data is fed into the model (Lasfer, 2013). As it can be observed in Figure 4.1, dataset acquired from EXIST Transparency Platform included some null variables mainly caused by time zone shifts in Turkey. As part of data preprocessing, replacing null values must not be overlooked for the sake of the success rate of the proposed method. In this study, null values are replaced with the value of the previous hour using "ffill" function of pandas library in Python as documented in Figure 4.2.

```
import pandas as pd

df = pd.read_excel('Turkey_Demand_Load_2010-2017.xlsx')
df.head()
```

	Datetime	Hour	Day	Month	Year	Load
0	2010-01-01	0	1	1	2010	18758.0
1	2010-01-01	1	1	1	2010	17907.0
2	2010-01-01	2	1	1	2010	16871.0
3	2010-01-01	3	1	1	2010	16044.0
4	2010-01-01	4	1	1	2010	15512.0

```
df['Load'] = df['Load'].replace(to_replace=0, method='ffill')
```

Figure 4.2: Python code to replace `0`values.

After first preprocessing step historical consumption data took the form in Figure 4.3.

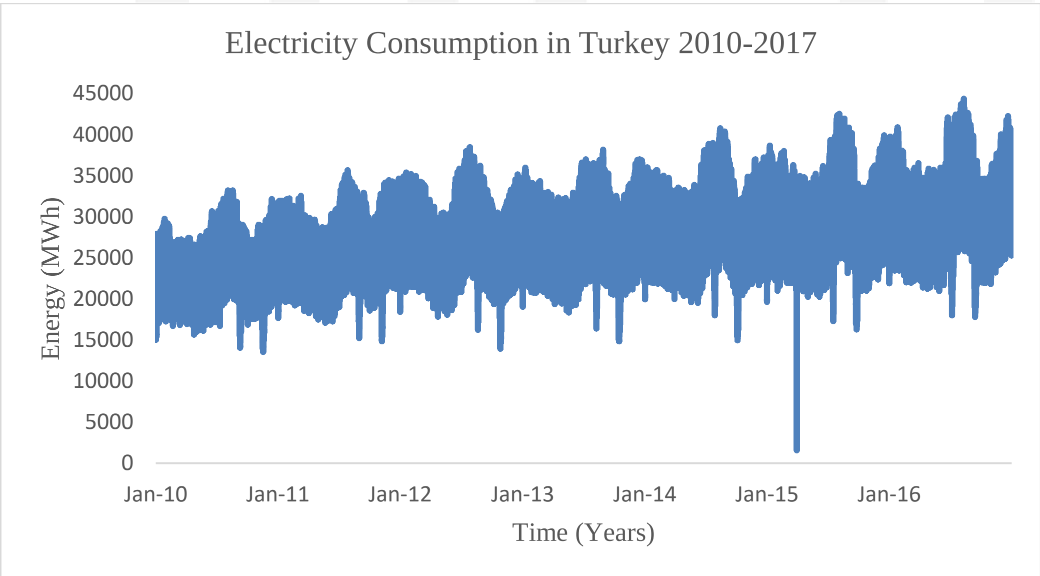


Figure 4.3: Hourly Electricity Consumption Data after replacing "0" values.

Looking at the data after first preprocessing step, first thing that drawn attention was the outlier in the first quarter of 2015. When we ran a background search on this period in Turkey, we see that a nationwide electricity outage occurred on 31/03/2015 and our outlier data corresponds to exactly this date. Assuming weekly cyclic nature of the time series data, 24 hours of 31/03/2015 was replaced by 24 hours of 24/03/2015, which was exactly one week before.

After replacing outliers that were detected by data visualization, a modified z-score outlier detection test was applied as in Equation 4.1. Z-score was introduced by Iglewicz and Hoaglin in 1993. The authors suggested using a threshold value of 3.5 (Iglewicz and Hoaglin, 1993).

$$MZ_i = \frac{0.6745(y_i - \overline{y})}{MAD} \qquad i = 1,2,3,\ldots,n \tag{4.1}$$

```
import numpy as np
def outliers_modified_z_score(ys):
    threshold = 3.5

    median_y = np.median(ys)
    median_absolute_deviation_y = np.median([np.abs(y - median_y) for y in ys])
    modified_z_scores = [0.6745 * (y - median_y) / median_absolute_deviation_y
                         for y in ys]
    return np.where(np.abs(modified_z_scores) > threshold)

outliers_modified_z_score(df['Load']) (array([], dtype=int64),)
```

Figure 4.4: Python code to test modified z-score of historical load data.

It can be observed in Figure 4.4 that the resulting array is empty after testing z-score with threshold of 3.5. This means all hourly historical load data passes this test and there are no more outliers.

Seasonal and weekly trends of consumption data are analyzed. Monthly behavior of the data shows its dependency on climate conditions as shown in Figure 4.5 whereas weekly behavior shows its dependency on industrial and urban trends within the week as shown in Figure 4.6.

4.2 Feature Selection

Several features should be taken into consideration in load forecasting models such as meteorological, economics, demographics and geographical location(Eljazzar and Hemayed, 2017). These should reflect the effects of climate, demographics and culture. In this study, temperature data of key cities in Turkey are taken into considerations as well as bank holidays, day types and Islamic holidays.

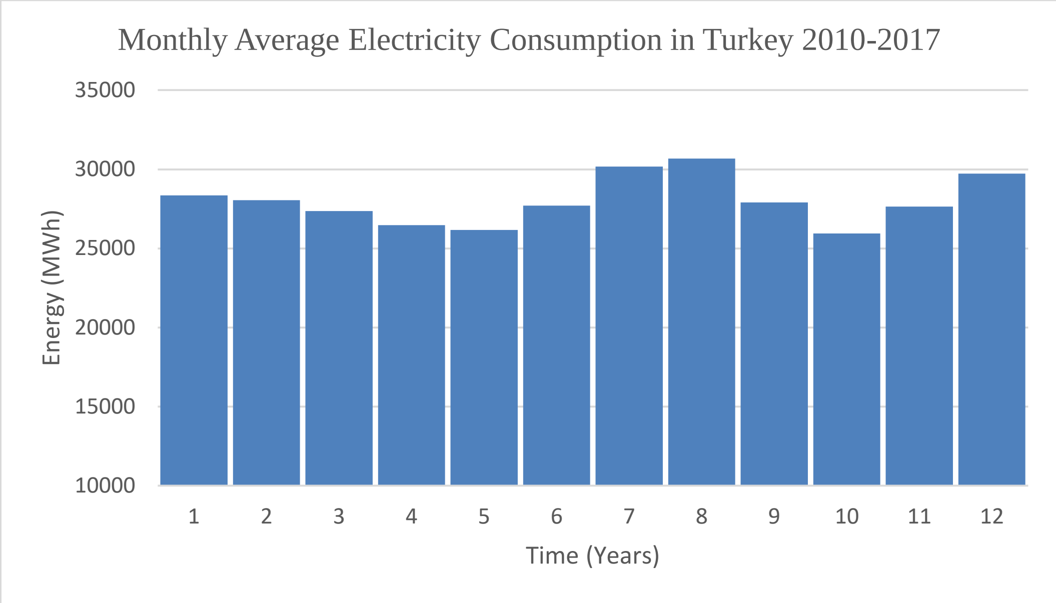


Figure 4.5: Monthly Trend of Electricity Consumption in Turkey.

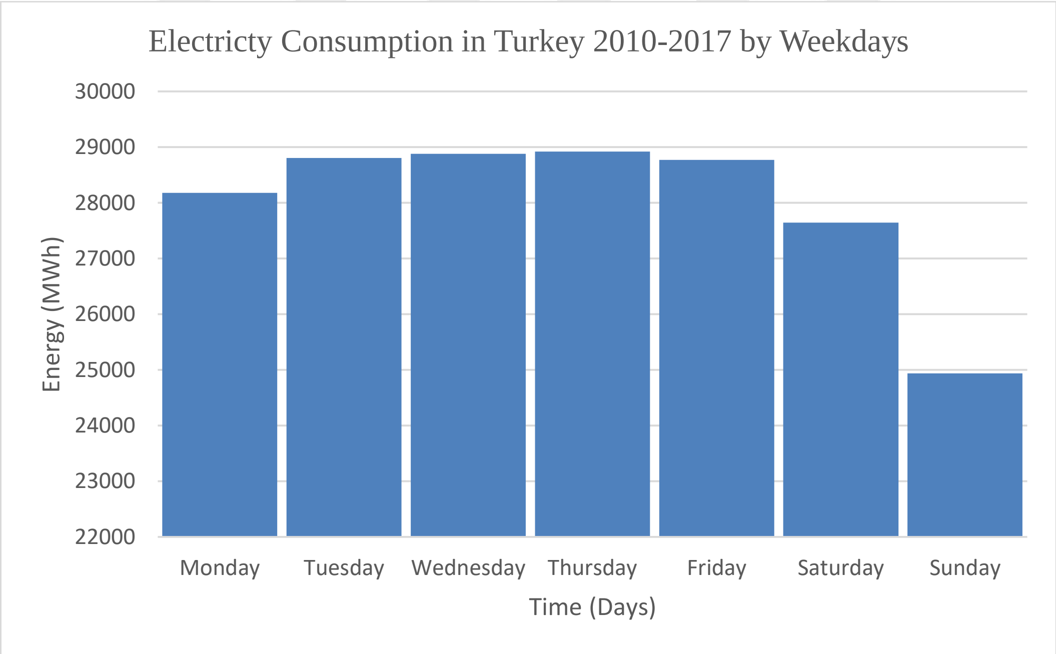


Figure 4.6: Weekly Trend of Electricity Consumption in Turkey.

4.2.1 Temperature

Temperature data in 81 cities across Turkey were considered for possible features. Evaluation was based on population density, availability of hourly historical data, industrial zones, tourism seasons and agricultural irrigation. In conclusion, historical temperature measurements from Istanbul, Ankara, Adana, Antalya and Diyarbakır were used as features in this study. Hourly temperature data between 2010-2017 was acquired from www.weatherunderground.com using API.

4.2.1.1 Istanbul

Istanbul is the largest city in Turkey. It has a population around 15 million people and it is also the heart of the Turkish economy by creating nearly 40% of the total GDP. Thus, temperature in Istanbul is the first parameter that comes to one's mind when selecting features for load consumption forecast of Turkey as presented in Figure 4.7.

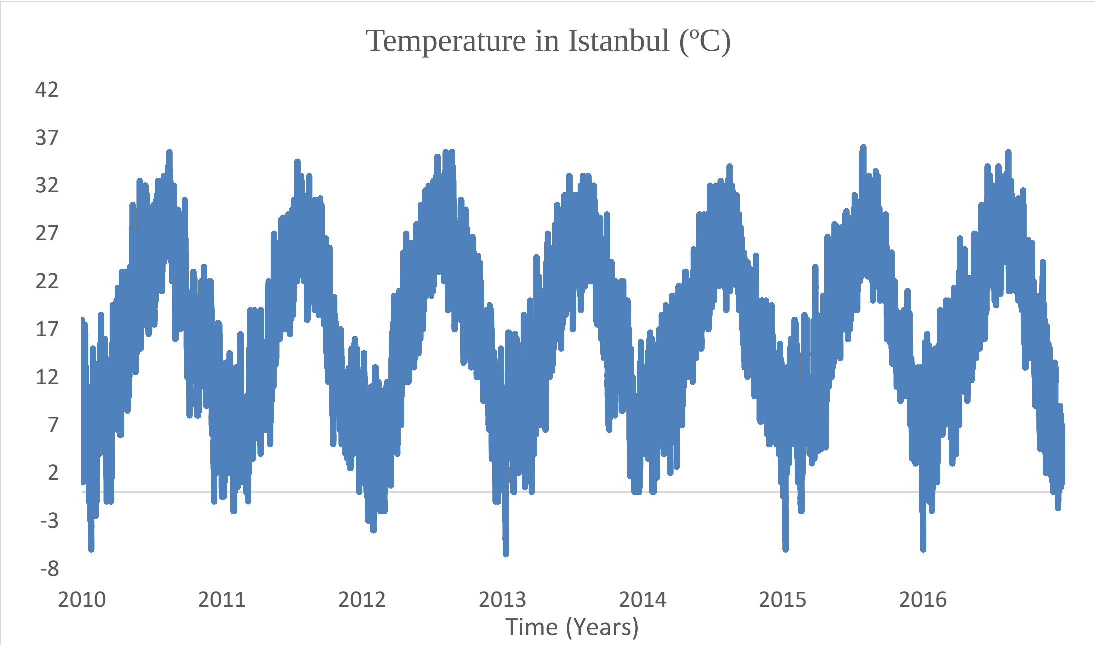


Figure 4.7: Istanbul Hourly Temperature Data (2010-2017) (Url-5).

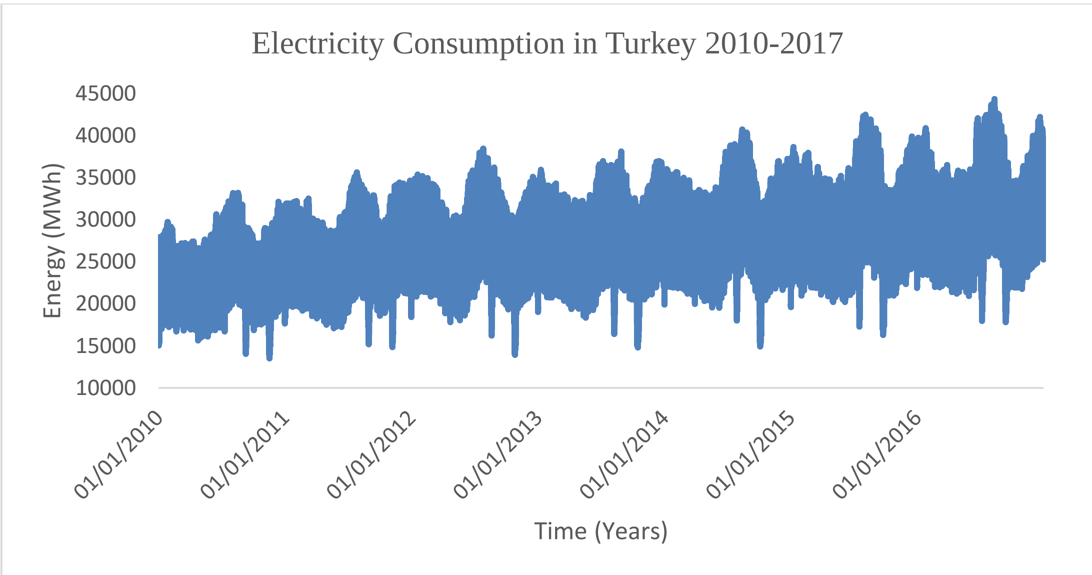


Figure 4.8: Hourly Electricity Consumption of Turkey (2010-2017) after outlier handling.

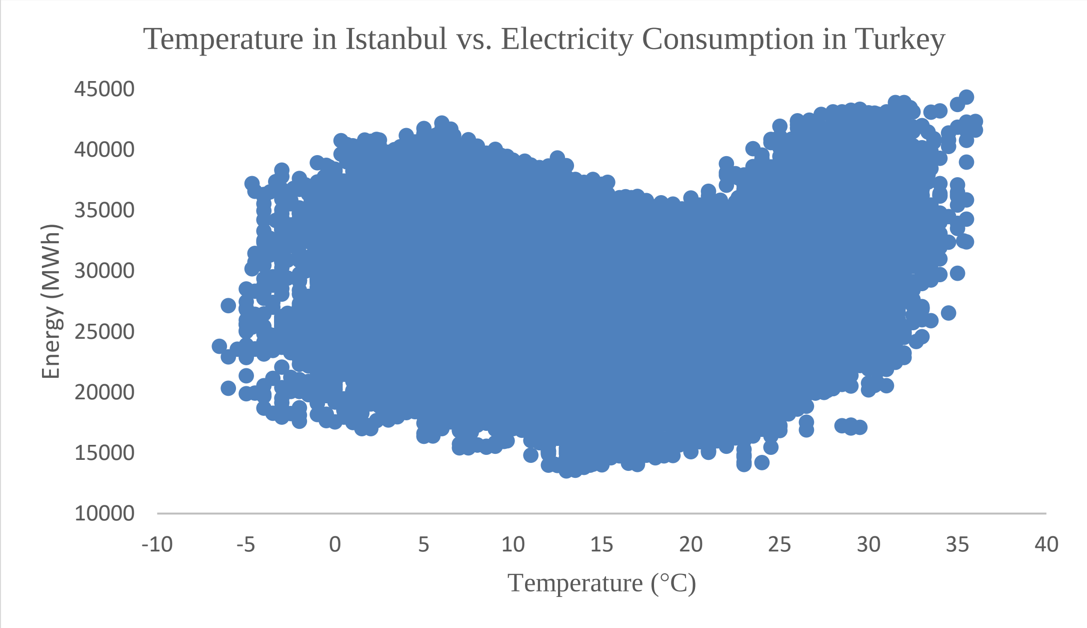


Figure 4.9: Temperature in Istanbul vs. Electricity Consumption in Turkey.

Correlation between, Istanbul temperature data and electricity consumption of Turkey can be identified visually, looking at Figure 4.7, Figure 4.5, Figure 4.8 and Figure 4.9. It can be observed that in summer season when temperature rises above 23-24 °C electricity consumption also rises due to usage of air conditioners. Similarly, we see the effect of electric heaters when temperature is below 10 °C.

4.2.1.2 Ankara

Ankara is the capital and 2nd largest city of Turkey with its 5.5 million inhabitants. Ankara is located at the heart of Anatolia where continental climate is dominant. Thus, having temperature data (presented in Figure 4.10) of Ankara as an input in our model brings a different aspect.

4.2.1.3 Adana

Adana is located in southern Turkey and has a population around 2.2 million which makes the city 5th most populate in Turkey. Adana is a center of industry, commerce and agriculture. Electricity consumption peaks in Adana during summer due to extremely high temperatures and agricultural irrigation as presented in Figure 4.11. Adana and its two neighboring cities Mersin and Hatay has a population around 5.5 million, thus having Adana temperature as an input reflects an important portion of Turkey’s total.

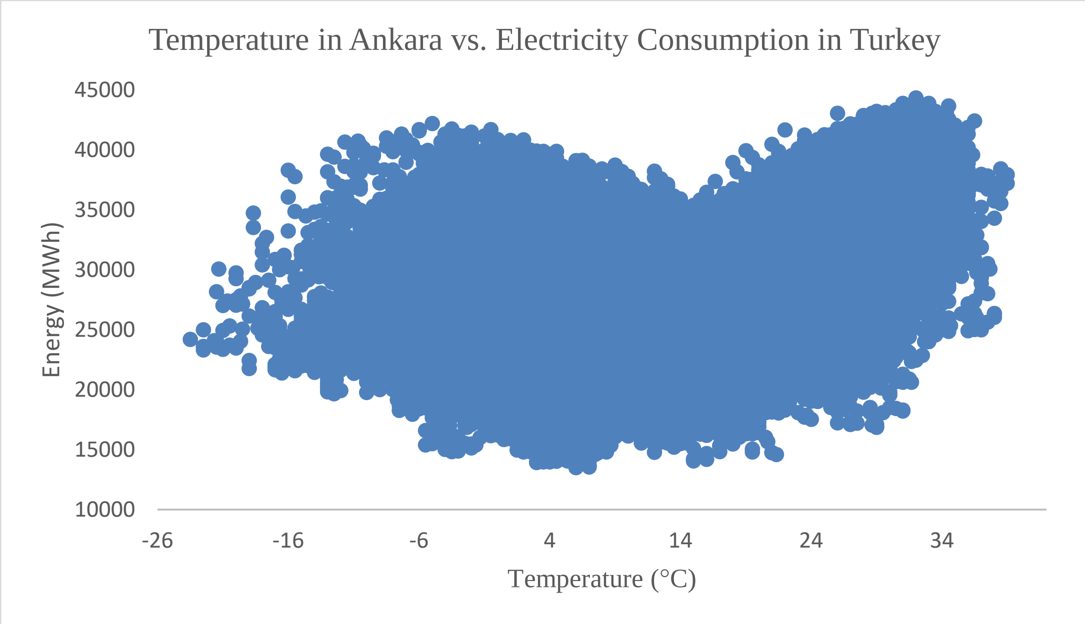


Figure 4.10: Temperature in Ankara vs. Electricity Consumption in Turkey.

4.2.1.4 Antalya

Antalya is located in southern Turkey near the Mediterranean Sea. Antalya's population is around 2.4 million, but it has been recorded that as much as 12.5 million tourists visit the city annually. This shows that Antalya is one of the most popular touristic destinations in Turkey especially in summer season. Thus, it is an important city to include in this study to consider the effect of tourist season on electricity consumption with higher air conditioner usage. Antalya's consumption data is demonstrated in Figure 4.12.

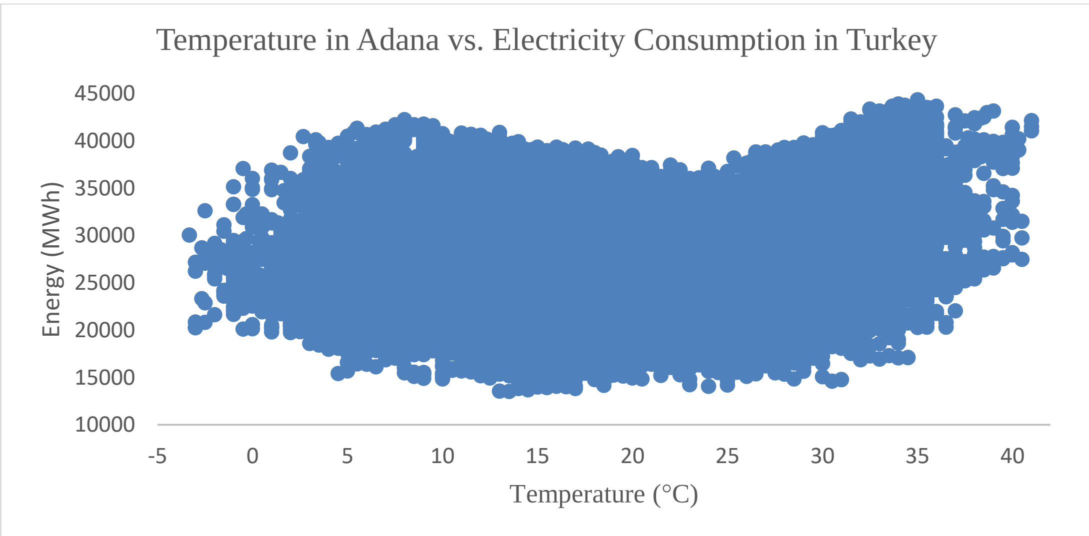


Figure 4.11: Temperature in Adana vs. Electricity Consumption in Turkey.

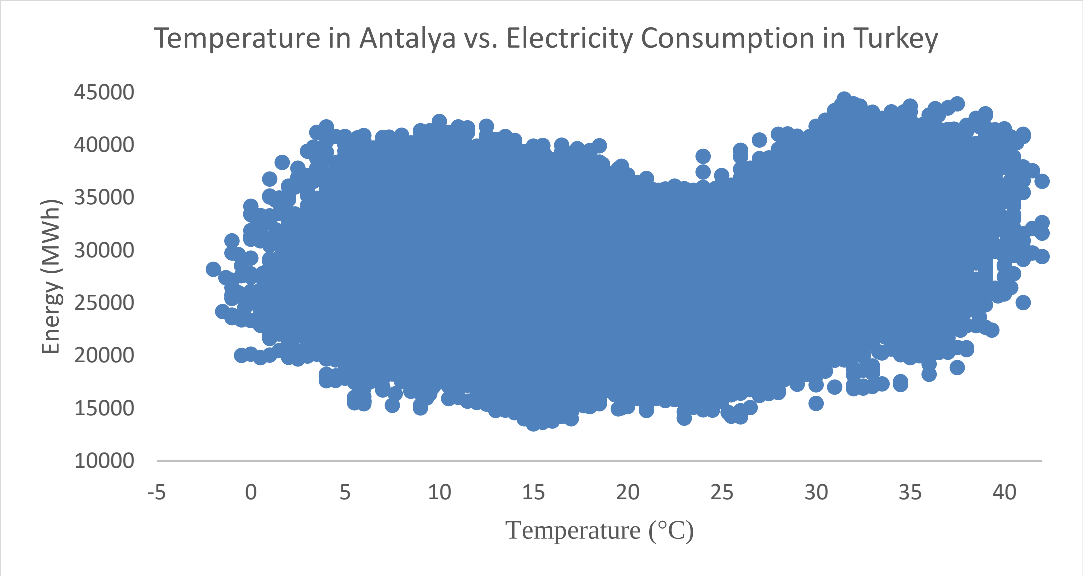


Figure 4.12: Temperature in Antalya vs. Electricity Consumption in Turkey.

4.2.1.5 Diyarbakır

Temperature data of Diyarbakır is included in this study to diversify geographical regions used. Diyarbakır is one of the largest cities in eastern part of Turkey with its 1.7 million population. In summertime, consumption increases due to irrigation and there is excessive electricity consumption due to illegal consumption. Figure 4.13 shows that Diyarbakır is one of the top cities where illegal electricity usage is a common practice.

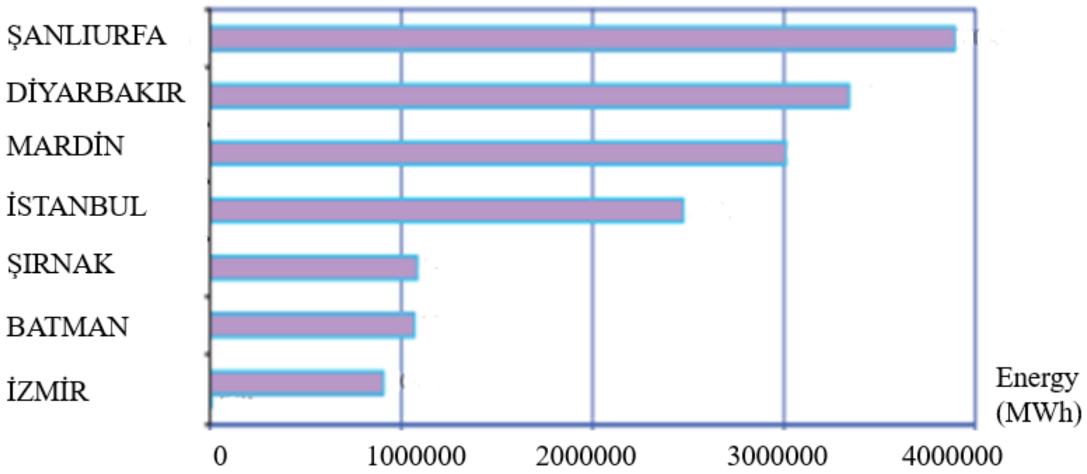


Figure 4.13: Illegal Electricity Usage according to 2011 TEDAS Annual Report.

It is known to balance the system frequency in this region National Load Dispatch Center had to generate a lot of loading instructions to power producers in this region due to system congestion. Figure 4.14 demonstrates the cost of congestion in different cities of Turkey. In summertime load peaks beyond expectations in an unforeseeable way due to illegal consumption and rapid irrigation.

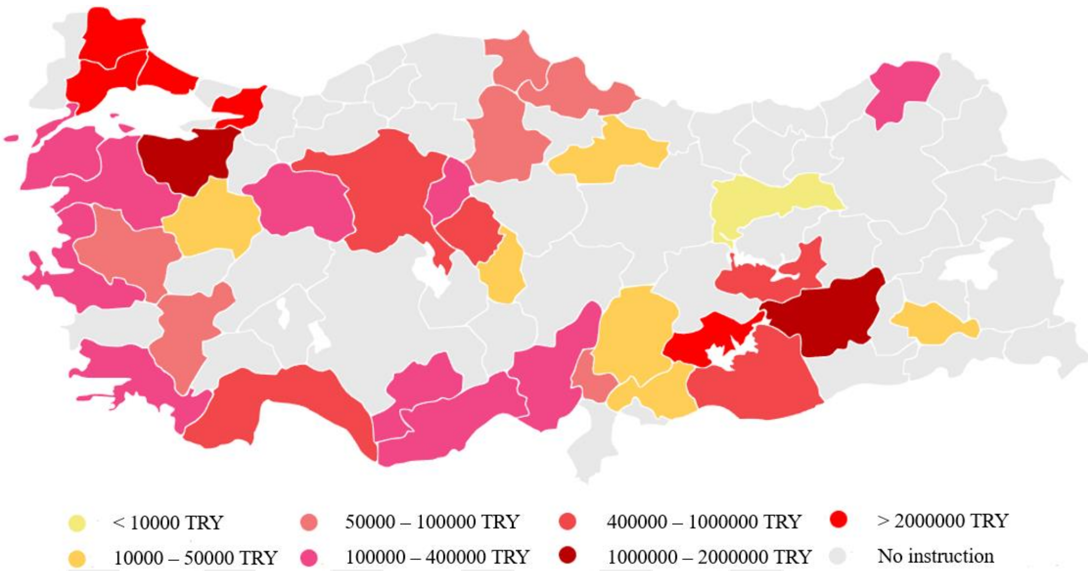


Figure 4.14: Cost of Congestion Loading Instructions (01/07/2016 - 01/09/2016) (Url-4).

4.3 Accuracy Validation

Mean-absolute-error(MAE) and root-mean-squared-error(RMSE) are commonly used metrics to judge the accuracy of a forecast model (Yildiz, Bilbao, & Sproul, 2017). In this study, MAE and RMSE will be presented to evaluate accuracy. MAE is simply the mean of the difference between actual and forecasted. RMSE, by nature of square operation, penalizes outliers more hence tends to be larger than MAE (Bouktif, Fiaz, Ouni, & Serhani, 2018). RMSE proves to be more useful when higher deviations from actual value is particularly undesirable.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{4.2}$$

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{4.3}$$

Dimensional metrics stated above provide an accuracy measurement that is meaningful in the context of this study and Turkey’s power market. However, in order to be able to benchmark the results with a different study, we can use relative versions of these metrics. Thus, relative mean-absolute-error(rMAE) and relative root-mean-squared-error(rRMSE) are proposed to be additional metrics (Akın, 2017).

$$\mathrm{y_r} = \mathrm{range}(\hat{\mathrm{y}}_\mathrm{i}) = \max(\hat{\mathrm{y}}_\mathrm{i}) - \min(\hat{\mathrm{y}}_\mathrm{i}) \tag{4.4}$$

$$rMAE = \frac{MAE}{y_r} \tag{4.5}$$

$$rRMSE = \frac{RMSE}{y_r} \tag{4.6}$$

4.4 Derived Features

In order to capture the sequential trend of the data, new features were derived (listed in Table 4.1) from the timestamp and past consumption data.

Table 4.1: Features used as input to model.

No	Type	Variable
1	Observed	Electricity Load (MWh)
2	Observed	Temperature in Istanbul (°C)
3	Observed	Temperature in Adana (°C)
4	Observed	Temperature in Ankara (°C)
5	Observed	Temperature in Antalya (°C)
6	Observed	Temperature in Diyarbakır (°C)
7	Derived	isHoliday
8	Derived	DayOfWeek
9	Derived	HourOfDay
10	Derived	DayOfMonth
11	Derived	MonthOfYear
12	Derived	PeakOfYesterday
13	Derived	Electricity Load in t-24h (MWh)
14	Derived	Electricity Load in t-168h (MWh)

4.5 Normalization

All input data was preprocessed and any duplicates were removed. Then any missing values were filled using forward filling method of Pandas library as mentioned before and resulting dataset was pushed through modified z-score outlier test.

In order to bring all of the input parameters into same scale a normalization process was applied. If this step is skipped, sme parameter might dominate the model because some parameters are in the scale of ten thousands whereas others oscillate between -10 to 30. Normalization process brings all of the parameters into a range between 0 and 1 (Bishop, 1995).

StandardScaler method of Python’s sklearn library was used to normalized data. StandardScaler function scales the data in such a way that the sample data has a mean of 0 and a standard deviation of 1.

4.6 Prediction Horizons

The model was designed to yield predictions in 3 different time horizons. First output of the model predicts the electricity load of the next hour. Second output predicts the next 6 hours and third prediction is made for the next 24 hours. These time horizons were selected based on the daily operations on Day Ahead Market and Intra-Day Market.

4.7 Early Stopping

Early stopping aims to prevent the model from overfitting. It is done by monitoring the loss on both training data and validation data during training. Normally, both training and validation loss should be decreasing at every epoch. However, if the model over fits validation loss stops decreasing whereas training loss keeps decreasing. An early stopping mechanism was established as presented in Figure 4.16 to stop training if the validation loss did not decrease for 10 epochs. Blue line in Figure 4.15 shows a case of overfitting.

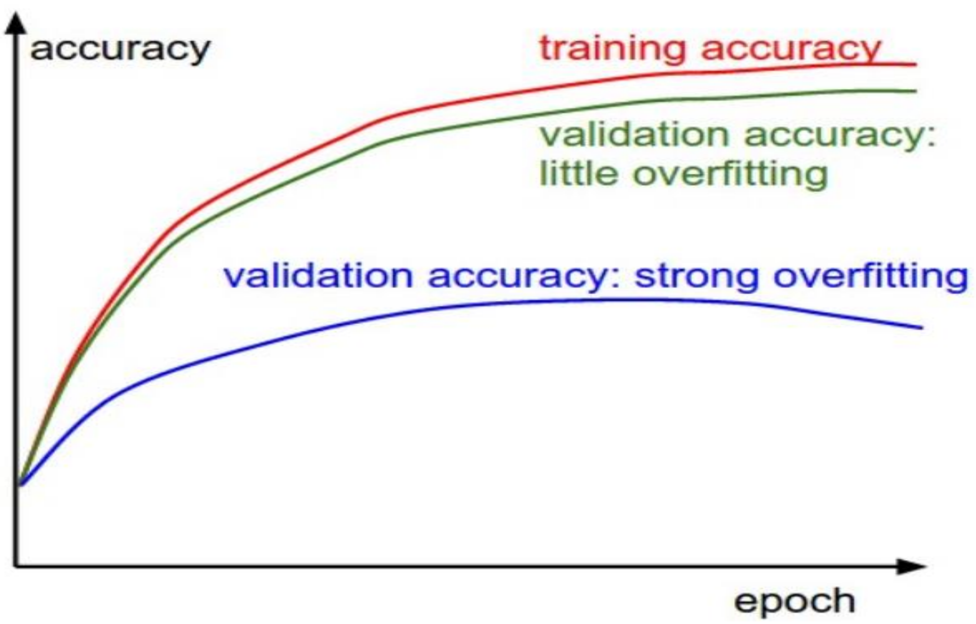


Figure 4.15: Training vs validation accuracy in an over fit model (Url-3).

```
from keras.callbacks import TensorBoard, EarlyStopping, ReduceLROnPlateau

import time
NAME="AAA-{}".format(int(time.time()))
tensorboard = TensorBoard(log_dir='logs/{}'.format(NAME))

def run_network(X_train, y_train, X_test, layers, epochs, batch_size=1024):
    model = build_model(layers)
    history = None

    try:
        history = model.fit(
            X_train, y_train,
            batch_size=batch_size,
            epochs=epochs,
            validation_split=0.1,
            callbacks=[
                tensorboard,
                #EarlyStopping(monitor='val_loss', patience=5, mode='auto')
            ])
    except KeyboardInterrupt:
        print("\nTraining interrupted")

    predicted = model.predict(X_test)

    return model, predicted, history
```

Figure 4.16: Python code to prepare validation data and early stopping.

4.8 Dropout

Dropout is another technique that prevents the model from overfitting. It randomly drops a certain percentage of the nodes during training so that the model cannot "memorize" the training data and instead becomes a generalized model. In recurrent networks such as the model used in this study, dropout is only applied to non-recurrent connections in the network (Alam, 2018). Effect of dropout application is demonstrated in Figure 4.17.

4.9 LSTM Model

The neural network model was built using Keras' built in LSTM and dense layers. Figure below shows the preferred network architecture in this study. There are 14 input parameters as stated before. Input signal goes through 3 LSTM layers, droputs and a dense layer and activation function. There are 3 defined output horizons in this study, thus we havve 3 different outputs for 1 hour ahead predictions, 6-hour ahead predictions and 24-hour ahead predictions. Size of the 3 LSTM layers was a research question which this study examines. Once the model was built, best architecture for LSTM layers was analyzed by running the model iteratively by changing number of LSTM cells in every iteration.

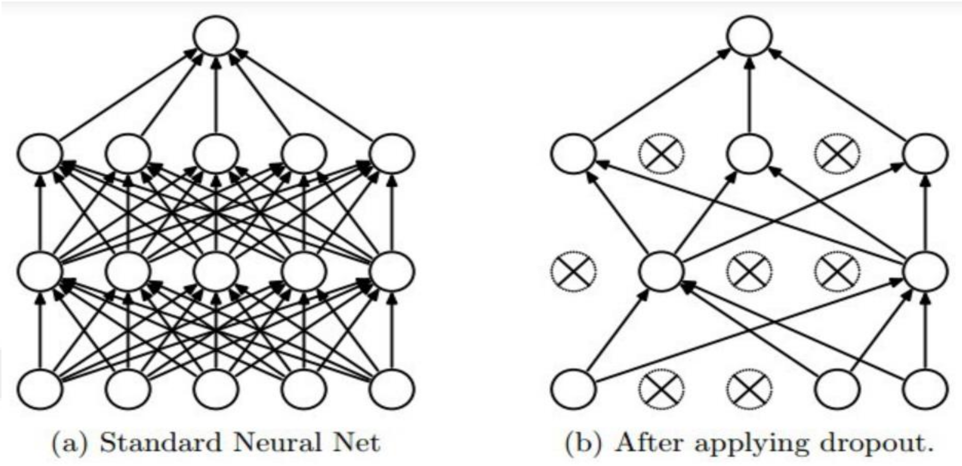


Figure 4.17: Effect of Dropout on network.

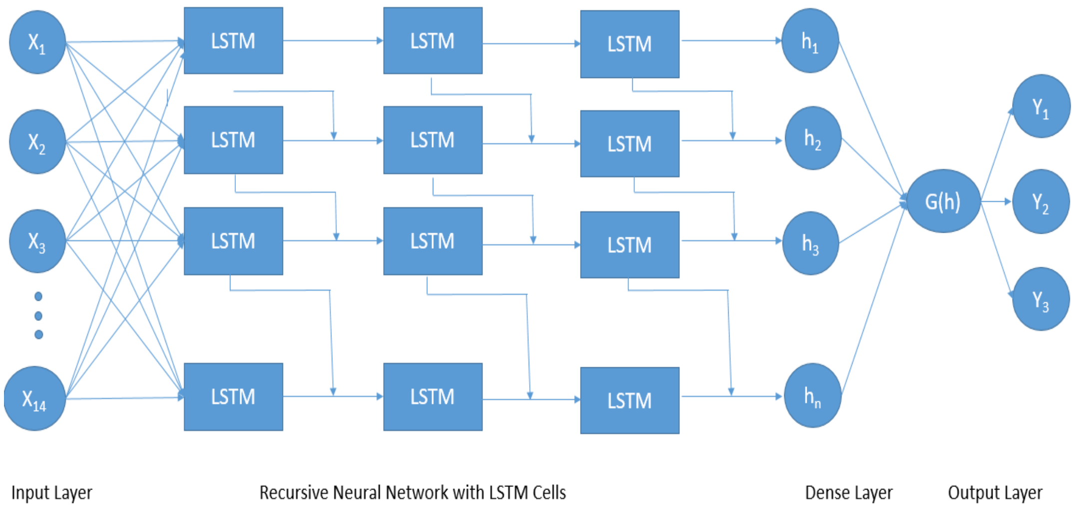


Figure 4.18: Keras LSTM architecture.

Figure 4.18 shows the architecture with 14 inputs and 3 outputs with [k,l,m] number of cells in each LSTM layer. [k,l,m] were changed in each iteration to figure out the architecture that yielded best results.

RMSprop was the preferred optimizer in this study as presented in Figure 4.19. It is considered to be the suitable optimizer to be used with most RNN models. In output layer linear activation function was used.

```
opt = keras.optimizers.RMSprop(lr=0.1,rho=0.9, epsilon=None, decay=0)
def build_model(layers):
    model = Sequential()
    model.add(LSTM(
            layers[1],
            activation = 'tanh',
            recurrent_activation='hard_sigmoid',
            input_shape=(None, layers[0]),
            return_sequences=True))
    model.add(Dropout(0.2))

    model.add(LSTM(layers[2], activation='tanh', recurrent_activation='hard_sigmoid', return_sequences=True))
    model.add(Dropout(0.2))

    model.add(LSTM(layers[3], activation='tanh', recurrent_activation='hard_sigmoid', return_sequences=False))
    model.add(Dropout(0.2))

    model.add(Dense(layers[4], activation="linear"))

    model.compile(loss="mse", optimizer=opt)

    print(model.summary())

    return model
```

Figure 4.19: Python code to create LSTM network using Keras.

5. RESULTS

The following results provide an analysis on different architectures. Same 14 parameters presented in Table 5.1 were used to analyze the success of different architectures.

Table 5.1: Accuracy of different network architectures with same input parameters.

	Number of Neurons			1 Hour Horizon		6 Hour Horizon		24 Hour Horizon	
Architecture No.	**k**	**l**	**m**	**rMAE**	**rRMSE**	**rMAE**	**rRMSE**	**rMAE**	**rRMSE**
1	20	15	20	0.017	0.022	0.036	0.046	0.027	0.045
2	40	30	40	0.015	0.020	0.030	0.040	0.027	0.046
3	60	45	60	0.012	0.016	0.031	0.042	0.027	0.046
4	80	60	80	0.012	0.016	0.026	0.035	0.025	0.043
5	100	75	100	0.012	0.017	0.029	0.038	0.025	0.041
6	120	90	120	0.011	0.014	0.026	0.034	0.022	0.037
7	140	105	140	0.011	0.015	0.025	0.032	0.023	0.035
8	160	120	160	0.010	0.014	0.031	0.040	0.025	0.039
9	180	135	180	0.011	0.015	0.027	0.036	0.024	0.040
10	200	150	200	0.011	0.015	0.027	0.036	0.024	0.040
11	220	165	220	0.010	0.013	0.025	0.032	0.021	0.033
12	240	180	240	0.010	0.013	0.024	0.033	0.024	0.039
13	260	195	260	0.009	0.013	0.024	0.032	0.023	0.039
14	280	210	280	0.009	0.013	0.023	0.031	0.026	0.041

Results of iterative model runs with different architectures (different LSTM cell numbers at layers) yield the results presented in Table 5.1. General trend shows that as the number of neurons used increase, accuracy improves. Although, architectures 11,12,13 and 14 have close results in all time horizons, the main time horizon that will be considered in the scope of this study is 24-hour horizon because it has more real life application due to the structure of energy markets. Having predictions 24 hours ahead lets a market player to be able to act on it on the Day-Ahead Market. Thus, architecture 11 was selected as the preferred structure because it yielded the lowest rMAE and rRMSE (0.021 and 0.033 respectively) in 24-hour horizon. This means previously defined [k,l,m] variables will take the values of [220,165,220] henceforth in this paper. Our model has 14 input parameters, 3 LSTM layers consisting of 220,165 and 220 LSTM cells respectively and 3 output time horizons as shown in Figure 5.1.

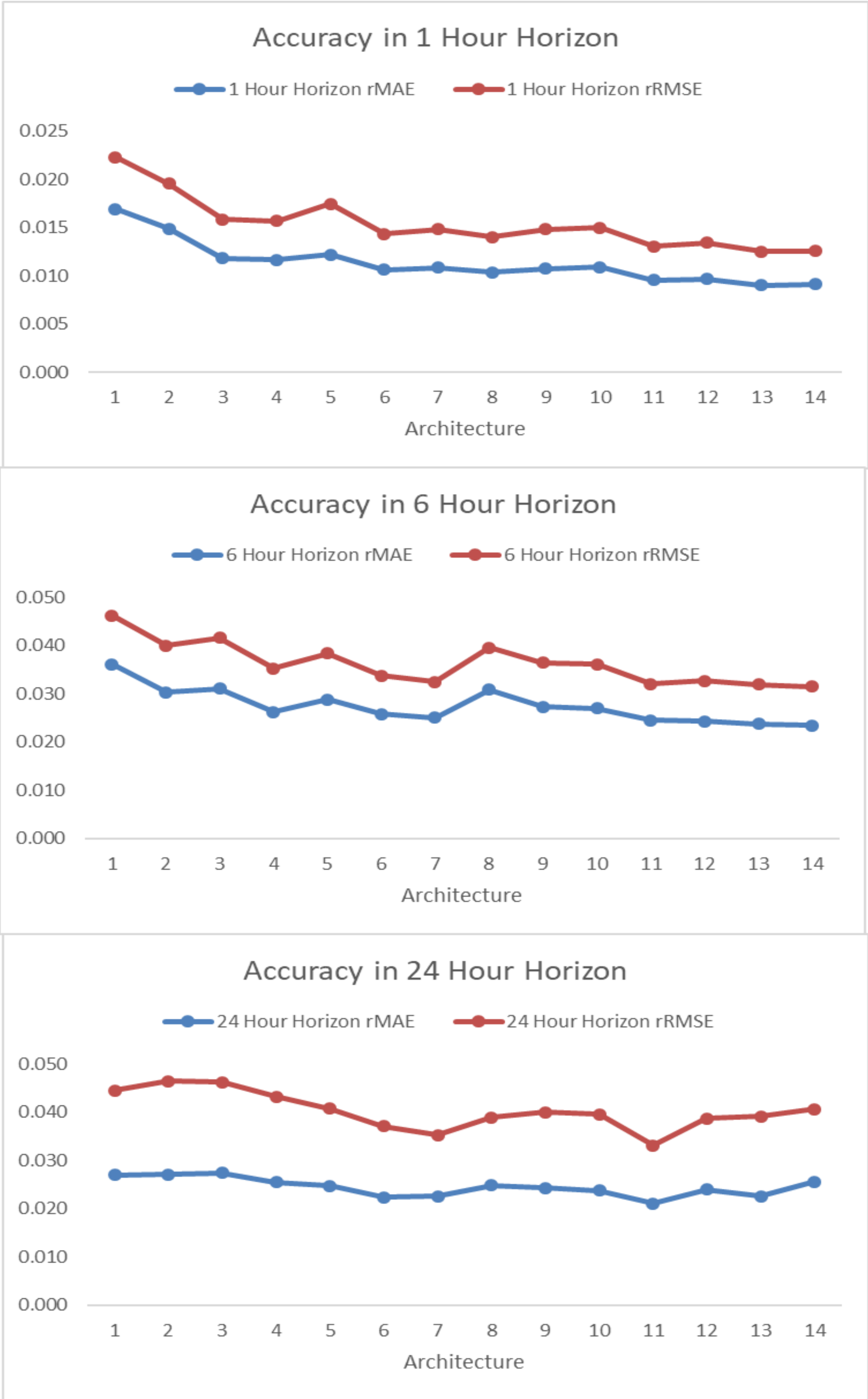


Figure 5.1: Accuracy result of model runs with different architectures.

Although rMAE and rRMSE are the main accuracy validation metrics used in this study, MAE and RMSE are also presented in Table 5.2 to form an idea on the real life scale.

Table 5.2: Accuracy of different network architectures in real life scale (in MWh).

	Number of Neurons			1 Hour Horizon		6 Hour Horizon		24 Hour Horizon	
Architecture No.	k	**l**	**m**	**MAE**	**RMSE**	**MAE**	**RMSE**	**MAE**	**RMSE**
1	20	15	20	448.672	591.664	959.571	1227.439	714.596	1181.863
2	40	30	40	395.154	519.120	805.039	1061.956	719.582	1232.599
3	60	45	60	313.657	420.906	824.181	1103.316	726.803	1227.408
4	80	60	80	308.875	416.294	696.467	936.575	675.697	1147.622
5	100	75	100	323.410	462.999	764.778	1019.585	655.227	1081.792
6	120	90	120	282.620	381.143	683.141	895.818	592.459	984.594
7	140	105	140	287.931	393.637	665.664	860.960	598.435	934.454
8	160	120	160	274.799	371.942	819.927	1049.238	659.444	1033.380
9	180	135	180	284.983	394.305	724.987	966.483	644.008	1060.796
10	200	150	200	290.119	397.564	715.797	958.862	630.638	1049.631
11	220	165	220	254.044	346.228	651.048	849.041	557.614	878.687
12	240	180	240	257.408	356.056	643.992	867.725	637.020	1025.572
13	260	195	260	240.168	332.061	630.081	847.500	599.022	1037.506
14	280	210	280	242.633	333.713	621.880	835.058	679.892	1079.011

Since architecture 11 is selected as the preferred network structure, detailed performance results of the model are presented in the following figures.

Visualizing loss value (MSE in our model) at every epoch is important in order to make sure that the model is not overfitting. It is clear that the loss decreases as the epochs elapse and it reaches a limit after a certain point.

Early stopping and dropout mechanisms were applied in the model in order to prevent over fitting of data. Analyzing Figure 5.2, we can deduct that validation loss curve does not significantly diverge from training loss, meaning over fitting did not happen.

Figure 5.3 shows time series of observed and predicted data for 3 different forecast horizons. A sample size of 1000 was chosen for clear presentation purposes. It can be observed that the model succeeds at catching subtle hourly and weekly profile as well as the general trend. Following figure shows the performance of the model for the whole validation dataset.

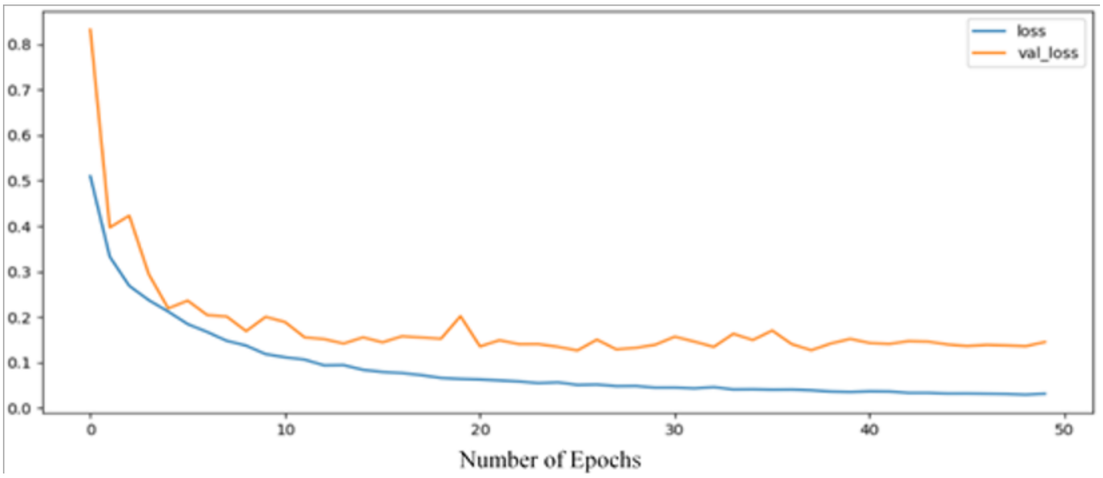


Figure 5.2: Training Loss vs. Validation Loss for [220, 165, 220].

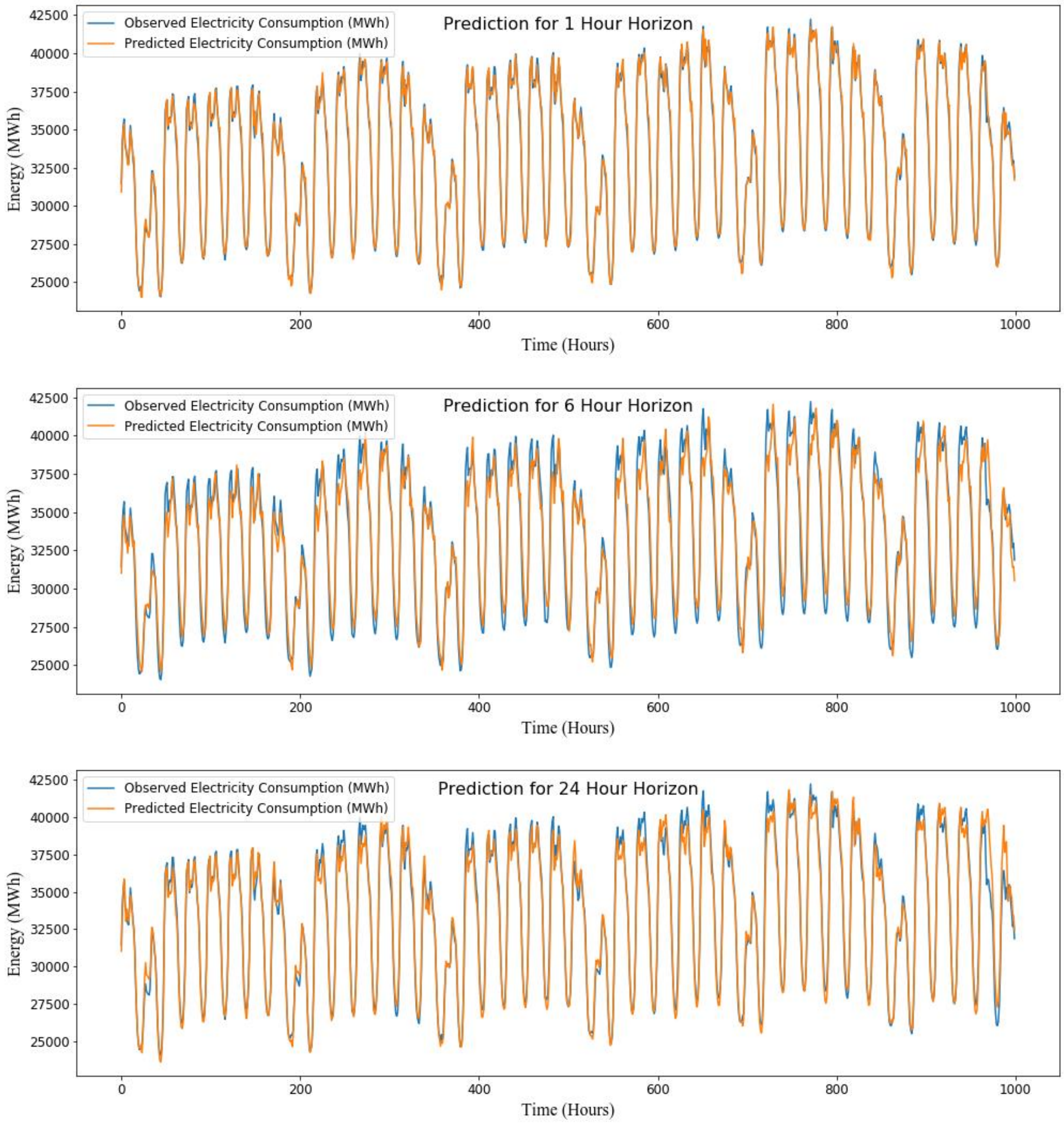


Figure 5.3: Predicted vs. Observed Electricity Load (MWh) for [220,165,220].

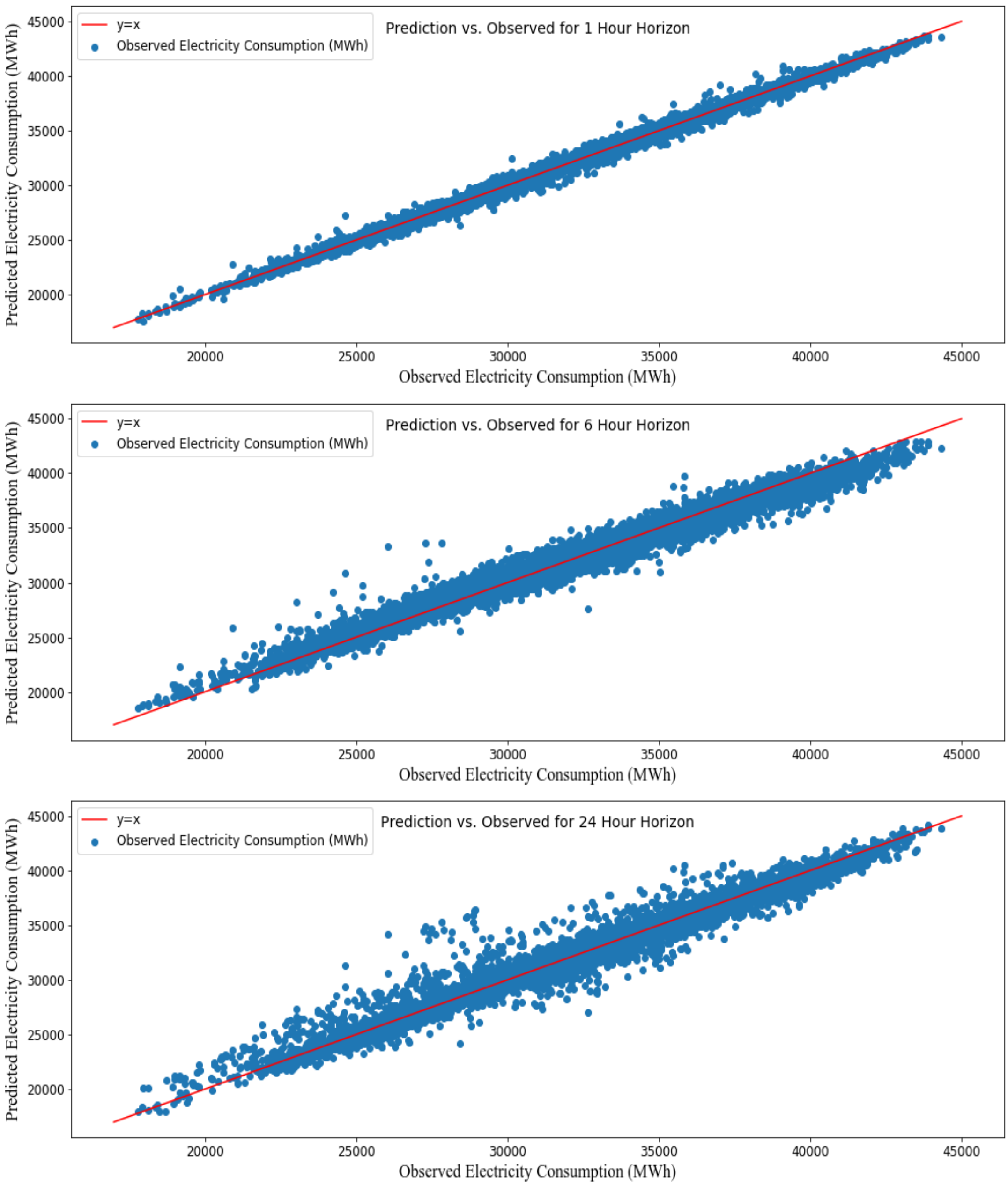


Figure 5.4: Predicted vs. Observed Electricity Load (MWh) for [220,165,220].

In Figure 5.4, predicted vs. observed data is presented alongside a 45° line (y=x). In case of a perfect forecast all data points should lie along y=x diagonal. In 24-hour horizon graph, some outliers are observed although most of the predictions lie near x=y line. This kind of distribution can be interpreted looking at the spread between MAE and RMSE, since RMSE punishes outliers more severely.

After identifying best performing network architecture, it was aimed to figure out which input parameters had more impact on the accuracy of the model. In order to do that, model was run leaving one input parameter listed in Table 5.3 out at every run.

Table 5.3: Effect of input parameters on results.

		1 Hour Horizon		6 Hour Horizon		24 Hour Horizon	
	Parameter_Left_Out	**rMAE**	**rRMSE**	**rMAE**	**rRMSE**	**rMAE**	**rRMSE**
1	None (Reference)	0.010	0.013	0.025	0.032	0.021	0.033
2	without_istanbul	0.010	0.014	0.023	0.031	0.023	0.036
3	without_Adana	0.010	0.014	0.027	0.035	0.025	0.038
4	without_Ankara	0.009	0.013	0.027	0.036	0.024	0.041
5	without_Antalya	0.009	0.013	0.024	0.032	0.023	0.039
6	without_Diyarbakir	0.011	0.015	0.028	0.036	0.023	0.038
7	without_isHoliday	0.009	0.013	0.025	0.035	0.023	0.039
8	without_DayOfWeek	0.010	0.013	0.026	0.034	0.026	0.041
9	without_HourOfDay	0.010	0.014	0.031	0.040	0.024	0.037
10	without_DayOfMonth	0.009	0.012	0.026	0.033	0.022	0.035
11	without_MonthOfYear	0.011	0.015	0.025	0.034	0.023	0.038
12	without_PeakOfYesterday	0.010	0.013	0.025	0.033	0.024	0.038
13	without_Load in t-24h	0.009	0.013	0.025	0.033	0.024	0.040
14	without_Load in t-168h	0.009	0.012	0.022	0.030	0.021	0.038

Results of Table 5.3 and Figure 5.5 show which parameters has no effect or negative effect on the accuracy as well as the parameters that have the most correlation. Cases where feature 9 (HourOfDay) and feature 11 (MonthOfYear) are left out display the highest deterioration in accuracy among all the time horizons.

Thus, it can be concluded that these 2 features are essential for the model. Nearly in all scenarios where an input is left out there is slight deterioration in accuracy, except for input feature 10 (DayOfMonth). This shows most of the features were selected well. Excluding feature 10 yields in slightly better accuracy in all of time horizons compared to our reference scenario. Thus, it can be concluded that having DayOfMonth as a feature has negative effect on the accuracy and it is better to remove it from the model.

Since feature 10 (DayOfMonth) was removed from model, we ended up with a new architecture with 13 input variables. Thus, it was decided to check whether this change would affect the best performing variable.

Model was run again with different number of LSTM cells to figure out the best performing architecture with the new set of input variables.

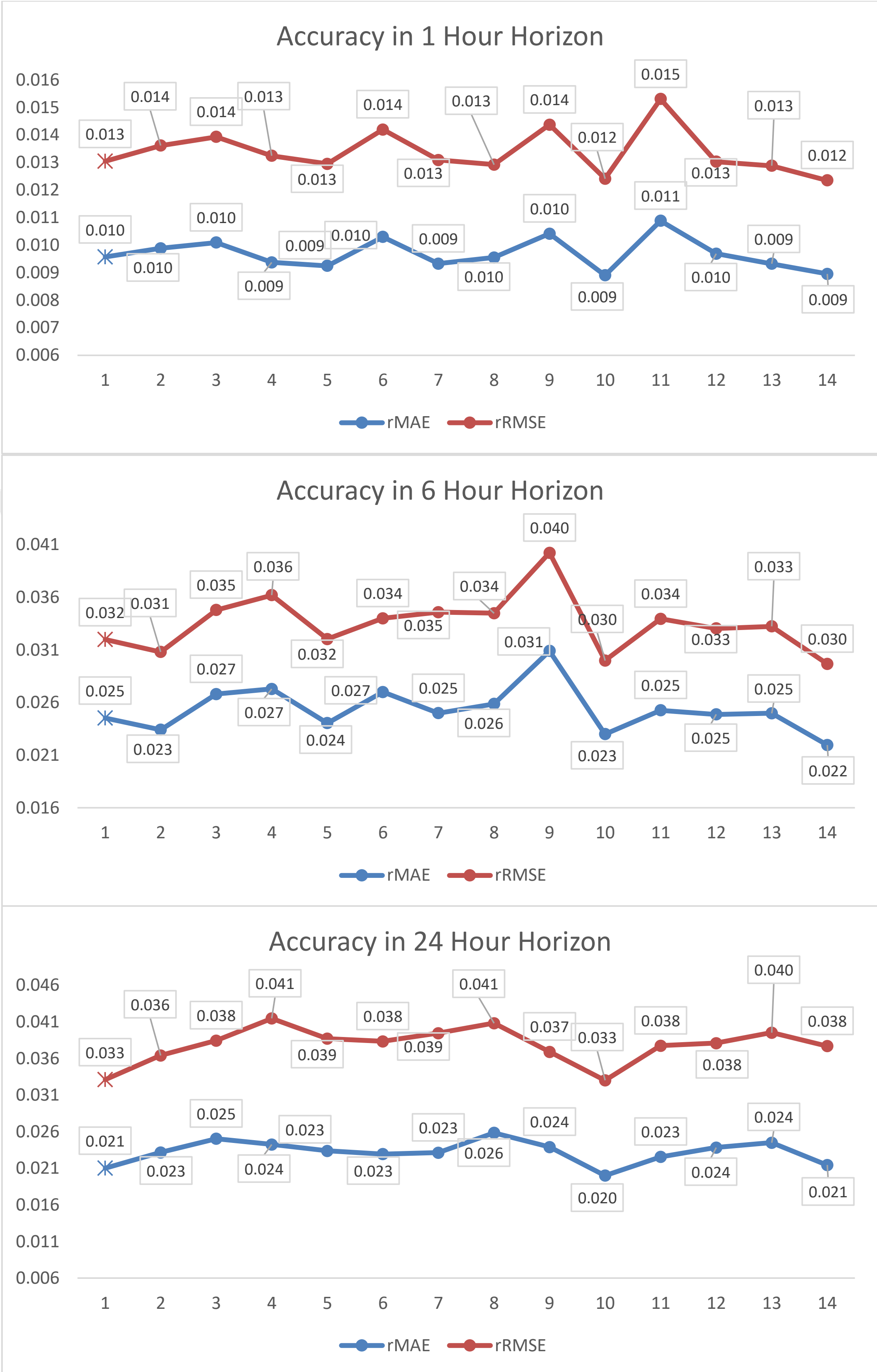


Figure 5.5: Accuracy of model [220,165,220] when input parameters are left out one at a time.

Table 5.4: Accuracy of network architectures with revised input parameters.

	Number of Neurons			1 Hour Horizon		6 Hour Horizon		24 Hour Horizon	
Architecture	**k**	**l**	**m**	**rMAE**	**rRMSE**	**rMAE**	**rRMSE**	**rMAE**	**rRMSE**
1	20	15	20	0.018	0.024	0.035	0.045	0.028	0.044
2	40	30	40	0.014	0.018	0.028	0.037	0.024	0.043
3	60	45	60	0.012	0.016	0.031	0.041	0.026	0.045
4	80	60	80	0.011	0.015	0.026	0.035	0.024	0.044
5	100	75	100	0.012	0.017	0.032	0.042	0.025	0.042
6	120	90	120	0.011	0.015	0.027	0.036	0.025	0.044
7	140	105	140	0.011	0.015	0.025	0.032	0.022	0.034
8	160	120	160	0.010	0.014	0.027	0.035	0.022	0.038
9	180	135	180	0.011	0.015	0.025	0.033	0.027	0.041
10	200	150	200	0.010	0.014	0.024	0.032	0.024	0.039
11	220	165	220	0.010	0.013	0.024	0.032	0.020	0.034
12	240	180	240	0.010	0.013	0.025	0.032	0.022	0.036
13	260	195	260	0.009	0.013	0.027	0.034	0.022	0.036
14	280	210	280	0.011	0.015	0.030	0.039	0.027	0.047

Results in Table 5.4 shows that despite the input parameter set is changed, architecture 11 [220,165,220] still yields the best accuracy. The model is finalized with a rRMAE of 0.02 and rRMSE of 0.034, over the test dataset (2016-2017 consumption data). It can be discussed that the biggest challenge of the model is to predict the peak load after there is a deep in consumption.

More concretely, forecasting the peak hour of Monday is a challenge for the model after the consumption makes a low on Sunday and Monday off-peak hours.

Choice of output layer activation fuction was also investigated in this study. In order to decide whether linear activation function or sigmoid activation function in output layer yields the better results, best performing architecture ([220,165,220]) was run with revised input parameters at varying learning rates with both linear and sigmoid activation functionsand results are presented in Figure 5.7, Table 5.5 and Table 5.6.

The best performing model turned out to be the one with a linear output activation function at a learning rate of 0.001 for 24-hour time horizon prediction. However, it was also observed that stability of the model rapidly dropped for learning rates higher than 0.01.

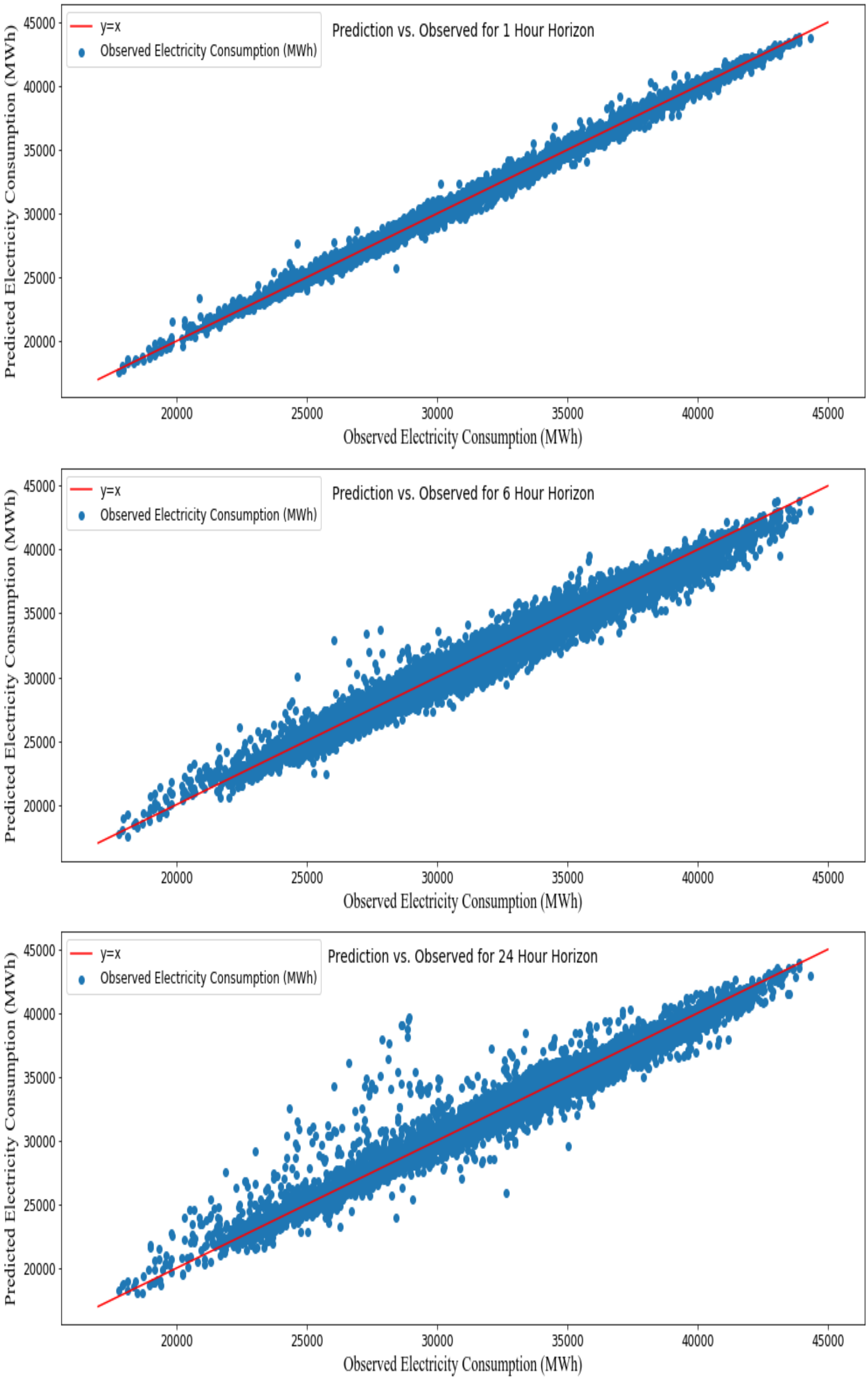


Figure 5.6: Predicted vs. Observed Electricity Load (MWh) for [220,165,220] with revised parameters.

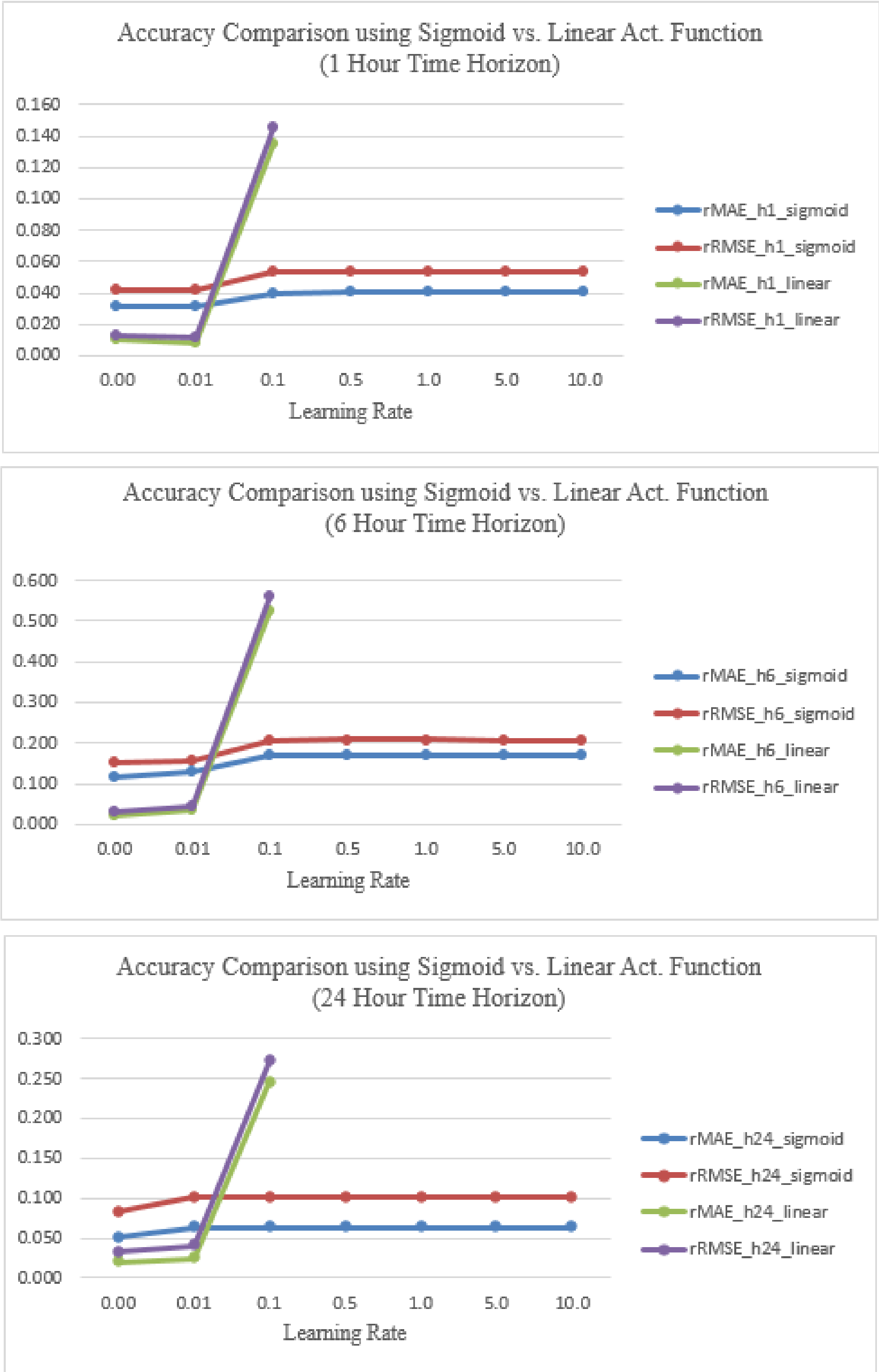


Figure 5.7: Sigmoid vs. Linear Activation Function Comparison.

Table 5.5: Accuracy of [220,165,220] network architecture with sigmoid activation function at output layer

	1 Hour Horizon		6 Hour Horizon		24 Hour Horizon	
Learning Rate	**rMAE**	**rRMSE**	**rMAE**	**rRMSE**	**rMAE**	**rRMSE**
0.001	0.031	0.042	0.119	0.153	0.052	0.083
0.010	0.032	0.042	0.129	0.159	0.063	0.101
0.100	0.040	0.053	0.169	0.208	0.063	0.101
0.500	0.040	0.054	0.170	0.209	0.063	0.101
1.000	0.040	0.054	0.170	0.209	0.063	0.101
5.000	0.040	0.054	0.170	0.209	0.063	0.101
10.000	0.040	0.054	0.170	0.209	0.063	0.101

Table 5.6: Accuracy of [220,165,220] network architecture with linear activation function at output layer

	1 Hour Horizon		6 Hour Horizon		24 Hour Horizon	
Learning Rate	**rMAE**	**rRMSE**	**rMAE**	**rRMSE**	**rMAE**	**rRMSE**
0.001	0.013	0.024	0.032	0.020	0.034	0.010
0.010	0.012	0.037	0.047	0.024	0.041	0.009
0.100	0.145	0.526	0.558	0.247	0.273	0.135
0.500	0.522	1.981	2.036	0.920	0.947	0.508

5.1 Benchmark

Transmission system operator TEIAS's National Load Dispatch Center (MYTM) is responsible to keep the system frequency at 50 Hz and to make sure grid is safe and healthy. To be able to do that National Load Dispatch Center uses a load forecast on the day ahead and it is announced via EXIST transparency platform. National Load Dispatch Center gathers the load expectations from 21 incumbent distribution companies. Since this forecast is used for balancing system frequency in real life by a state institution, which plays a critical role for grid safety and reliability, it is believed that it would make a very appropriate benchmark for the model proposed in this thesis paper. Methodology behind this forecast is not public information and it is disclosed to TEIAS and incumbent distribution companies, but it is assumed to be a state-of-the-art model, since the safety of Turkey's grid depends on it.

As stated before, the results presented in this paper belong to the test dataset which is the realized hourly electricity consumption in Turkey in 2016. In order to make a fair benchmark, load forecast data of 2016-2017 period in EXIST Transparency Platform is used. Besides this fact, TEIAS announces only day-ahead load forecast, so it is only compared with 24 hour horizon forecast of this model. In comparison, the best performing model ([220,165,220] with revised input parameters and linear activation function in output layer) is used. The results are presented in Figure 5.8, Figure 5.9 and Table 5.7.

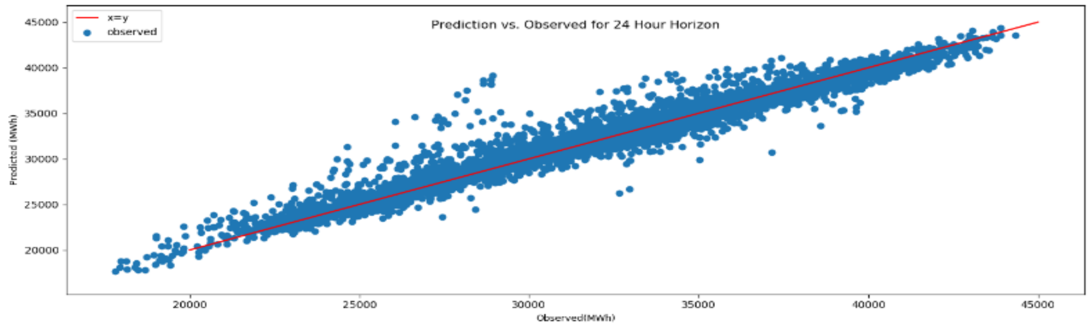


Figure 5.8: Predicted vs. Observed Electricity Load (MWh) for [220,165,220].

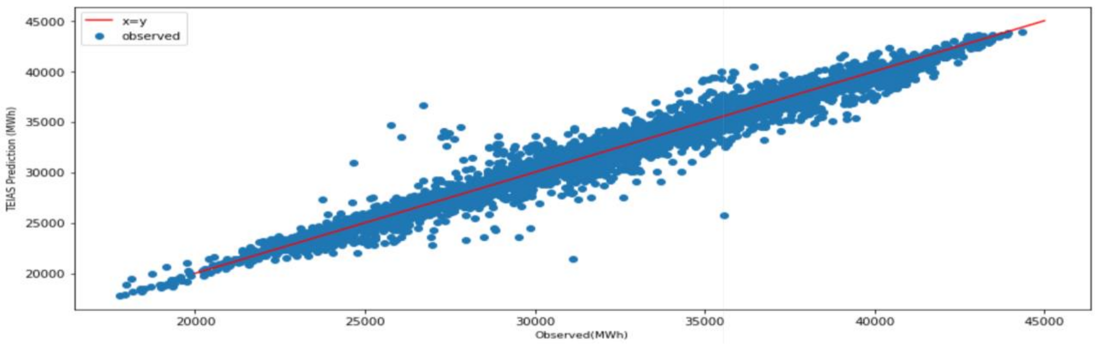


Figure 5.9: TEIAS Prediction vs Observed Electricity Load.

Table 5.7: Proposed LSTM Network vs. TEIAS Accuracy Metrics.

Proposed Model		**TEIAS**	
rMAE	rRMSE	rMAE	rRMSE
0.020	0.034	0.020	0.031

6. CONCLUSION

In this study, electricity load forecast of Turkey in 3 different time horizons was performed using LSTM network. LSTM network takes feature parameters and work the relations out. These type of neural networks have the ability to solve complex nonlinear problems and generalize the outcome. This was the reason they were used in this study to address the short term load forecasting problem.

LSTM networks excel at figuring the relations between inputs and outputs in a sequential manner. They are especially useful in time series forecasting due to the fact that they present a solution to the vanishing gradients problem associated with recurrent neural networks. Historical temperature data of 5 cities in Turkey (namely Istanbul, Ankara, Adana, Antalya and Diyarbakır) was used as input parameters. In total 14 input features were defined. In LSTM layers, hard sigmoid function was used as the activation function and hyperbolic tangent function was used as the recurrent activator. Linear activation function was used in output layer.

Turkey’s hourly electricity load consumption data between 2010-2017 was obtained from EXIST Transparency Platform and meteorological data of the same period was obtained from “www.wunderground.com”. Data between 2010-2016 was used as train dataset for the LSTM model to be trained on. Model was tested on 2016-2017 data for its accuracy. In order to avoid overfitting, dropout and early stopping methods were applied on a validation set. Feature normalization was implemented to improve efficiency and accuracy of LSTM model. rMAE and rRMSE metrics were used to measure the accuracy of the model. Besides that, MAE and RMSE were also presented to make a comparison in the real world scale.

Analysis of temperature and load data showed that there is correlation between temperature and electricity consumption behavior. Seasonal effects could be observed as in winter time the consumption increased due to heating effect and similarly consumption peaked in summer due to cooling effect of air conditioners. In order to capture seasonality and weekly periodic behavior of the load, inputs like isHoliday, DayOfWeek, HourOfDay, DayOfMonth, load in the same hour of yesterday and load in the same hour of last week were used.

Model was tried iteratively using 14 different architectures for the LSTM layers. As expected, general trend was an increase in accuracy as number of cells in hidden LSTM layers were increased. However, it was observed that after a certain number of cells, model accuracy did not improve and even deteriorated. After many runs, architecture 11 with [220,165,220] LSTM cells was determined as the best performing model architecture. Model with [220,165,220] LSTM cells in 24-hour horizon yielded results with 0.021 rMAE and 0.033 rRMSE.

Additionally, impact of each of the 13 initial input parameters was investigated by running the model excluding the investigated parameter, so that the possible change in accuracy of the model would signal the relevance of that input parameter. Result of the iterative runs showed that most of the features were selected well. Only parameter which had no impact on the accuracy was DayOfMonth and it was removed from the model for the next runs. After removing DayOfMonth parameter, model was run with different possible architectures again and the best performing architecture appeared to be [220,165,220] once again. Model yielded in 24-hour horizon 0.02 rMAE and 0.034 rRMSE with revised input parameters. At this point we must point out there is limited amount of fluctuation in loss function as the epochs progress. Dropout method significantly diminished the amount of fluctuations, but still fluctuations are present after third decimal place in rMAE and rRMSE. One remedy for this could be using more data points which the author of this study did not have access to.

Model produces three outputs with different time horizons at each run. It makes a forecast of the next hour, next 6 hours ahead and next 24 hours ahead. As expected, the best accuracy occurs for the prediction of the next hour. [220,165,220] model with revised input parameters yielded 0.01 rMAE and 0.013 rRMSE in 1-hour horizon. It could be deducted that as prediction time horizon is shorter, predictions are more accurate; however, a comparison between 6-hour horizon and 24-hour horizon predictions disproves this theory. Independent of the architecture used, rMAE of 6-hour horizon was always higher than rMAE of 24-hour horizon. This can be associated with sequential structure of electricity consumption data. Consumer behavior has a pattern based on 24-hour periodicity. Thus, predictions based on a 24-hour sequential system yields better results.

In the last section of the study, results of 24-hour horizon predictions with [220,165,220] model and revised input parameters, were benchmarked against TEIAS' day-ahead load forecast presented in EXIST Transparency Platform. Predictions of 24-hour horizon was prioritized in order to make a fair comparison. Besides, the main area this study can be used in a commercial scope would be a power trading company's day-ahead market operations, so 24-hour horizon predictions have the most significance in a commercial context. Benchmark against TEIAS shows that proposed model in this thesis paper, reaches TEIAS' accuracy in rMAE and only slightly comes short in rRMSE. Since, TEIAS day-ahead forecast plays a vital role in balancing supply-demand and keeping system frequency stable, reaching its standard is an important measure of success. Moreover, TEIAS gathers these predictions from incumbent distribution companies which has access to metering results and meteorological data of all the regions in their domain. Author of this thesis paper had access to Turkey's total electricity consumption and meteorological data of 5 cities out of 81. It can be anticipated that a study with a city's hourly load data and meteorological data would yield better results. Then one could combine predictions for all of the cities to make a prediction for Turkey's electricity consumption. Another point of improvement would be number of data points. This study used data between 2010-2017, but a longer data period would help the model to build relations between inputs and outputs more successfully. Overall, the study presents promising results with area for improvement.

REFERENCES

Akın, B. (2017). Yapay Sinir Ağlarıyla Konya Bölgesinde Kullanıcı Doğal Gaz Tüketim Öngörüsü.

Alam, S. (2018). Recurrent neural networks in electricity load forecasting.

Alfares, H., and Mohammad, N. (2002). Electric load forecasting : Literature survey and classification of methods. *International Journal of Systems Science (January 2014).*

Alpaydin, E. (2010). Introduction to Machine Learning Second Edition (2nd, Ed.). *London: The MIT Press.*

Anderson, D., and Mcneill, G. (1992). Artificial Neural Network Technology.

Andrej Krenker, Bešter, J., and Kos, A. (2011). Introduction to the Artificial Neural Networks, Artificial Neural Networks - Methodological Advances and Biomedical Applications. *Prof. Kenji Suzuki (Ed.)*. https://doi.org/10.5772/15751

Bishop, C. M. (1995). Neural Networks for Pattern Recognition. *Clarendon Press.*

Bouktif, S., Fiaz, A., Ouni, A., and Serhani, M. A. (2018). Optimal deep learning LSTM model for electric load forecasting using feature selection and genetic algorithm: Comparison with machine learning approaches. *Energies, 11*. https://doi.org/10.3390/en11071636

Eljazzar, M. M., and Hemayed, E. E. (2017). Feature selection and optimization of artificial neural network for short term load forecasting. *2016 18th International Middle-East Power Systems Conference, MEPCON 2016 -Proceedings*,827–831. https://doi.org/10.1109/MEPCON.2016.7836990

Gökçe, B. (2018). Impact of Renewable Energy on the Power Market. https://doi.org/10.22201/fq.18708404e.2004.3.66178

Hochreiter, S. (1997). Long Short-Term Memory. *1780*, 1735–1780.

Lasfer, A. (2013). Performance Analysis of Artificial Neural Networks in Forecasting Financial Time Series.

Learned-miller, E. G. (2014). Introduction to Supervised Learning. 1–5.

Lysfjord, M. J. W. (2017). Modeling and Forecasting the Nord Pool Day-Ahead Power Market through Deep-Learning.

Mohri, Mehryar; Rostamizadeh, Afshin; Talwalkar, A. (2012). Foundations of Machine Learning. *London: The MIT Press.*

Nightingale, D. S., and Pindus, N. M. (1997). Privatization of Public Social Services: A Background Paper. *Urban Institute*, 11.

Oliver, A., Odena, A., Raffel, C., Cubuk, E. D., and Goodfellow, I. J. (2018). Realistic Evaluation of Deep Semi-Supervised Learning Algorithms.

Rosenfeld, A., and Wechsler, H. (2000). Pattern Recognition : Historical Perspective and Future Directions.

Russell, S.; Norvig, P. Artificial Intelligence: A Modern Approach (3rd ed.). *New Jersey: Prentice Hall.*

Samuel, A. L. (1959). Eight-move opening utilizing generalization learning. (See Appendix B, Game G-43.1 Some Studies in Machine Learning Using the Game of Checkers.

Yavar Bathaee. (2011). The artificial Intelligence Black Box and the failure of Intent and Causation. *2*(4), 31–40.

Yildiz, B., Bilbao, J. I., and Sproul, A. B. (2017). A review and analysis of regression and machine learning models on commercial building electricity load forecasting. *Renewable and Sustainable Energy Reviews*, *73*(March 2016), 1104–1122. https://doi.org/10.1016/j.rser.2017.02.023

Zhang, G., Patuwo, B. E., and Hu, M. Y. (1998). *[10] -zhang1998* - Forecasting with artificial neural networks: *The state of the art*. *14*, 35–62.

Url-1 <*http://www.willfleury.com*>, date retrieved 12.04.2019.

Url-2 <*https://www.tensorflow.org/about*>, date retrieved 12.04.2019.

Url-3 <*https://towardsdatascience.com*>, date retrieved 17.04.2019.

Url-4 <*https://rapor.epias.com.tr* >, date retrieved 19.04.2019.

Url-5 <*https://www.wunderground.com/history*>, date retrieved 28.04.2019.

Url-6 <*https://www.teias.gov.tr/* 29.04.2019.

Url-7 <*http://www.econlib.org* >, date retrieved 29.04.2019.

CURRICULUM VITAE

Name Surname: Anıl Türkünoğlu

E-Mail: aturkunoglu@gmail.com

EDUCATION:

- **B.Sc.** **:** 2015, Koc University, College of Engineering, Mechanical Engineering (Full Merit Scholarship)
- **B.A.** **:** 2015, Koc University, College of Administrative Sciences and Economics, Economics (Full Merit Scholarship)

PROFESSIONAL EXPERIENCE AND REWARDS:

- AKENERJİ, Energy Trading Assistant Specialist, Istanbul March 2016 – March 2018
- AKENERJİ, Asset Management and Intraday Market Specialist, Istanbul, March 2018 – September 2018
- ENGIE Global Energy Management, Power Trader, Istanbul, since September 2018