APPLICATION OF A HYBRID MACHINE LEARNING MODEL ON SHORT TERM ELECTRICITY DEMAND PREDICTION

A THESIS SUBMITTED TO
THE BOARD OF GRADUATE PROGRAMS
OF
MIDDLE EAST TECHNICAL UNIVERSITY, NORTHERN CYPRUS CAMPUS

BY

Ahmed Khaled Ahmed Farouk Assar

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN SUSTAINABLE ENVIRONMENT AND ENERGY SYSTEMS PROGRAM

FEBRUARY 2022

Approval of the Board of Graduate Programs

Prof. Dr. Cumali Sabah
Chairperson

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science

Assoc. Prof. Dr. Ceren İnce Derogar
Program Coordinator

This is to certify that we have read this thesis and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science.

Assoc. Prof. Dr. Murat Fahrioğlu
Supervisor

Examining Committee Members

Prof. Dr. Serkan Abbasoglu	CIU/ESE	
Assoc. Prof. Dr. Murat Fahrioğlu	METU NCC/EEE	
Asst. Prof. Dr. Canras Batunlu	METU NCC/EEE	

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

Name, Last name:

Signature :

ABSTRACT

APPLICATION OF A HYBRID MACHINE LEARNING MODEL ON SHORT TERM ELECTRICITY DEMAND PREDICTION

Assar, Ahmed Khaled Ahmed Farouk
Master of Science, Sustainable Environment and Energy Systems Program
Supervisor: Assoc. Prof. Dr. Murat Fahrioğlu

February 2022, 160 pages

Electricity demand forecasting is an important procedure in the electricity market and plays a great role in assuring a sustainable and efficient operation chain. By accurately forecasting the demand, one can see a considerable reduction in production costs as well as saving energy resources. Therefore, optimizing the demand forecasting techniques became an inseparable goal of power economics, leading to the introduction of machine learning to this sector that proved to be superior to other predefined alternatives. This thesis proposes to apply a Hybrid model that combines two forecasting machine learning algorithms; namely Support Vector Regression (SVR) and Long Short-Term Memory (LSTM). The hourly data from the Spanish electricity market is used to forecast the day-ahead electricity consumption in the last quarter of the year 2018, with weather Variables being fed to the models as the inputs. The performance of the proposed model is compared with the Temperature regression and load projection model (the actual model used in Spain), Autoregressive Integrated Moving Average (ARIMA), Artificial Neural Networks (ANNs), and SVR and LSTM separately. The combined method's forecasting results were shown to be superior to all four suggested independent approaches, and it was able to successfully minimize errors and enhance the accuracy between actual and forecasted values. However, the proposed Hybrid model didn’t outperform the already applied approach and achieved a Mean Absolute Percentage Error (MAPE) of 1.71557 and a Mean Absolute Error (MAE) of 488.269.

Keywords: Power Economics, Electricity Markets, Machine Learning, Demand Forecasting, Prediction Models

ÖZ

HİBRİT MAKİNE ÖĞRENME MODELİNİN KISA DÖNEM ELEKTRİK TALEP TAHMİNİ ÜZERİNE UYGULAMASI

Assar, Ahmed Khaled Ahmed Farouk
Yüksek Lisans, Sürdürülebilir Çevre ve Enerji Sistemleri Programı
Tez Yöneticisi: Doç. Dr. Murat Fahrioğlu

Şubat 2022, 160 Sayfa

Elektrik talep tahmini, elektrik piyasasında önemli bir işlem olmakla birlikte sürdürülebilir ve verimli bir operasyon zincirinin sağlanmasında büyük rol oynamaktadır. Elektrik talebi doğru bir şekilde tahmin edilerek, üretim maliyetlerini önemli ölçüde azaltmanın yanı sıra enerji kaynaklarından da tasarruf sağlanabilir. Bu nedenle, talep tahmin yöntemlerini optimize etmek, güç ekonomisinin vazgeçilmez bir hedefi haline gelmiş olup, sektörde önceden tanımlanmış diğer alternatif yöntemlerden daha üstün olduğu kanıtlanan makine öğrenimi sistemlerinin kullanılmasına yol açmıştır. Bu tez, makine öğrenim algoritmalarını birleştiren bir hibrit model önerir. Bu hibrit modeli oluştururken kullanılan algoritmalar, Destek Vektör Regresyonu (Support Vector Regression - SVR) ve Uzun Kısa Dönemli Bellek (Long Short-Term Memory - LSTM) olup, İspanya elektrik piyasasının saatlik verilerini kullanarak 2018 yılının son çeyreğinde gün öncesi elektrik tüketimini tahmin etmek için kullanıldı. Önerilen modelin performansı, sıcaklık regresyonu ve yük projeksiyon modeli (İspanya'da kullanılan model), Otomatik Regresif Hareketli Ortalama (Autoregressive Integrated Moving Average - ARIMA), Yapay Sinir Ağları (Artificial Neural Networks - ANNs), SVR ve LSTM ile ayrı ayrı karşılaştırıldı. Yöntemin hataları başarılı bir şekilde en aza indirildi ve bu sayede gerçek ile tahmin edilen değerler arasındaki doğruluk arttırılmış oldu. Ancak önerilen Hibrit model, hali hazırda uygulanan yöntemden daha iyi performans gösteremedi ve 1.71557'lik bir Ortalama Mutlak Yüzdelik Hata (MAPE) ve 488.269'luk bir Ortalama Mutlak Hata (MAE) değerine ulaştı.

Anahtar Kelimeler: Güç Ekonomisi, Elektrik Piyasaları, Makine Öğrenimi, Talep Tahmini, Tahmin Modelleri

To my beloved Parents

ACKNOWLEDGMENTS

First and foremost, My utmost gratitude is for ALLAH.

Secondly, I would like to thank Assoc.Prof.Dr.Murat Fahrioğlu for his constant guidance throughout my degree.

I would also like to show my gratitude to my colleague Zekican Buden who always offered his expertise and help with programming in python.

I would also like to show my gratitude to my colleague Khaled el Doa who always offered his expertise and help with conceptual discussions on the thesis's implementation.

I would also like to thank my friend Kaan Büyüktaş for providing his personal computer as means for parallel programming to save time.

Last but not least, I would thank my supporting mother Rania Abdelwahab, and Loving father Khaled Assar for being there for me throughout my life and guiding me on the right path to reach where I am now. If you are reading this, I want you both to know how much I am thankful that you are my parents.

I also want to point out my appreciation to my brother Abdelrahman Assar for always being by my side as a fundamental support system.

TABLE OF CONTENTS

CHAPTERS

APPENDICES

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

LIST OF ABBREVIATIONS

ABBREVIATIONS	
ADF	: Augmented Dickey-Fuller
AI	: Artificial Intelligence
ANN	: Artificial Neural Networks
AR	: AutoRegression
ARIMA	: Autoregressive Integrated Moving Average
ARMA	: Autoregressive Moving Average
CSA	: Cuckoo Search Algorithm
EfuNN	: Evolving Fuzzy Neural Network
ENTSO-E	: European Network of Transmission System Operators for Electricity
FL	: Fuzzy Logic
FNL	: Fuzzy Neural Network
GENCO	: Generation Company
ISO	: Independent System Operator
LRM	: Linear Regression Models
LSTM	: Long Short-Term Memory
LTLF	: Long-Term Load Forecasting
MAE	: Mean Absolute Error
MAPE	: Mean Absolute Percentage Error

MLP	: Multilayer Perception Network
MLR	: Multiple Linear Regression Models
NARX	: Nonlinear Autoregressive Models With Exogenous İnputs
NN	: Neural Networks
RBF	: Radial Basis Function
Relu	: Rectifier Function
RNNs	: Recurrent Neural Networks
SSE	: Sum of Squared Errors
STLF	: Short-Term Load Forecasting
SVM	: Support Vector Machines
SVR	: Support Vector Regression
TMP	: Temporary Market Price
WI	: Willmott’s Index

CHAPTER 1

INTRODUCTION

Nowadays, with all the technological advancements that the globe is experiencing, everything is turning to be automated and the main source that this whole transition is depending on is electricity. Electricity, transforming into an essential commodity that has to be available instantaneously led to the development of electricity markets which can be considered as the most vital market in a country’s economy.

Natural energy consumption continues to rise as people's living standards and socioeconomic status have been rising dramatically in the past century. As the problem of energy scarcity becomes more serious, more and more governments are placing a premium on finding solutions to this topic. Electricity, one of the most essential energy resources, plays a critical function in the stability of any power system, which resulted in it being a primary incentive for improving societal development.

Electricity is a difficult-to-store resource; also, electricity consumption is influenced by a variety of unpredictable variables such as weather, population, holidays, emergencies, and so on. Having absolutely no control over these variables’ behavior, the electricity market finds it challenging to estimate the demand due to all of these massive challenges. As a result, in the power market, an exact and precise forecasting system is required since incorrect electricity demand projections, on the other hand, will be unproductive. A procedure that is overestimated will raise the workload of electricity production and waste energy resources whereas an underestimate will fail to meet the demand and may cause fatal systems’ breakdowns. As a result, whether for developed or developing countries, precise electrical demand forecasting is a requirement to pre-meet the demand. Developing a creative approach that is not only effective but also improves forecasting accuracy is necessary for any type of electricity demand projections, whether short-term, mid-term, or long-term. [1]

Also, the electricity market is unique in a way that there is no direct communication between the customer and the supplier. Moreover, the demand in this market is not absolute, which means that the demand is always fluctuating, which is also one of the main incentives that led the market to adopt forecasting techniques to predict the demand and supply accordingly. All the previous points show the complete volatility of the market. Even though the electricity market can be arguably considered the most critical market in a country's economy, it is prone to failure the most. That is why all the countries adopt strict regulations to ensure that the electricity market operates efficiently. Having these unique characteristics, the electricity market operates on the principles of forecasting.

As the essence of finding reliable prediction models increases, many novelties are being presented by specialists in an attempt to narrow the gap between the forecasted data and actual reality. These prediction models work using different principles and utilize different techniques, however, they can fall under two main categories; parametric and nonparametric approaches.

Parametric approaches usually use time-based and regression methods to deal with data. Goia et al. published their research which used specifically linear regression to forecast the peak electricity demand [2]. In the case of time-based methods, Autoregressive integrated moving average (ARIMA) models are extensively used in the electricity load prediction, one of the multiple research that used such an approach was for the demand prediction in turkey by Erdogdu et al. [3].

On the other hand, nonparametric approaches employ artificial intelligence (AI) and machine learning algorithms to deal with the available data. Some of the mainstream blocks that have been used since the introduction of these methods to electricity load forecast are Neural Networks (NN) [4], Artificial Neural Networks (ANN) [5], and Support Vector Machines (SVM) [6].

Even though these approaches proved to be of great accuracy, the reliability of their results depends heavily on the nature of the data that is being fed to them. The variables that are being input into the system have different trends and behaviors, some data may vary linearly, others may vary nonlinearly whereas some other data can vary both linearly and nonlinearly but in a seasonal manner (variation repetition every specific time interval). The models that these approaches incorporate, may be reliable to one specific type of variation and faulty to others. That is why Hybrid models were introduced recently in the electricity markets.

Coming from its name, Hybrid models are a combination of two or several models that can be both parametric and nonparametric based, which in return can deal with different variations within the same data set and has proved to be of higher superiority to the conventional models already available in the market. [7]

This thesis will combine parametric and nonparametric approaches to form a hybrid model that will be tested on a specific data set (to be justified in upcoming sections) obtained from the Spanish electricity market. The results of this model will be compared with the already used technique in the literature and conclusions will be drawn.

This thesis will be sectioned into 7 chapters, chapter 2 will be familiarizing the reader with what power economics is and how does electricity markets operate, chapter 3 will be a literature review covering what are the different principles that forecasting models use. The case study and data justification will be presented in chapter 4, later on, chapter 5 will thoroughly explain the theory behind the models used and illustrate the methodology implemented on the datasets., and before concluding and stating final remarks in chapter 7, chapter 6 will represent the results.

CHAPTER 2

POWER ECONOMICS

Power economics is a term that engulfs the general electricity market including its policies, entities, and regulations. This type of economics emerged as a result of the availability of multiple energy scarcities which are used by numerous suppliers to produce electricity. As urbanization is continuously spreading through the human lands, the demand for electricity is exponentially increasing and becoming chaotic, therefore an organized system had to be established to efficiently connect the demand side and the suppliers [8]. The designed system is governed by general policies and regulations that are divided into two categories; one which is common to all places around the globe and this category aids in facilitating electricity trading and sharing between countries and the second category contains other “local” policies that are specifically tailored to suit the special cases in a country, the mentioned system is what is known as electricity market. [9]

2.1 General Analogy

To aid the reader in understanding the full concept of electricity transactions that occur in the market, some of the complicated terms will be simplified using the following figure.

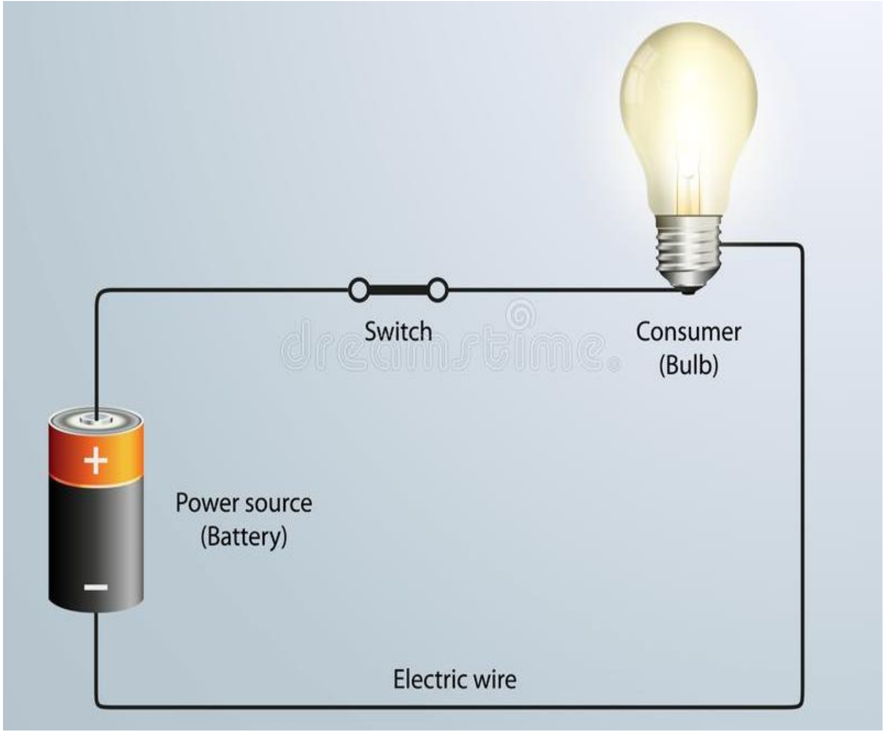


Figure 2.1. Simple Electric Circuit

Figure 2.1 above illustrates a simple electric circuit with basic components. The simple schematic shown above can be related to the real-time application in power economics. The battery shown above resembles the "**Suppliers**" who are responsible for providing electricity, whereas the bulb represents the "**Demand**" that consumes electricity. The electric wire connecting them is known as "**Transmission lines**". Finally, the whole circuit can be related to what is known as the "**Grid**". The grid is a huge circuit in practice that connects all the demand (bulb) in a country to the total number of suppliers (battery). The grid is defined by its total capacity which depends on the number of suppliers with the amount of electricity that they can provide, as well as its reach, i.e. the connection of transmission lines between the number of consumers and the suppliers. However, in actual operation, the circuit is meticulous as shown below in figure 2.2. [10]

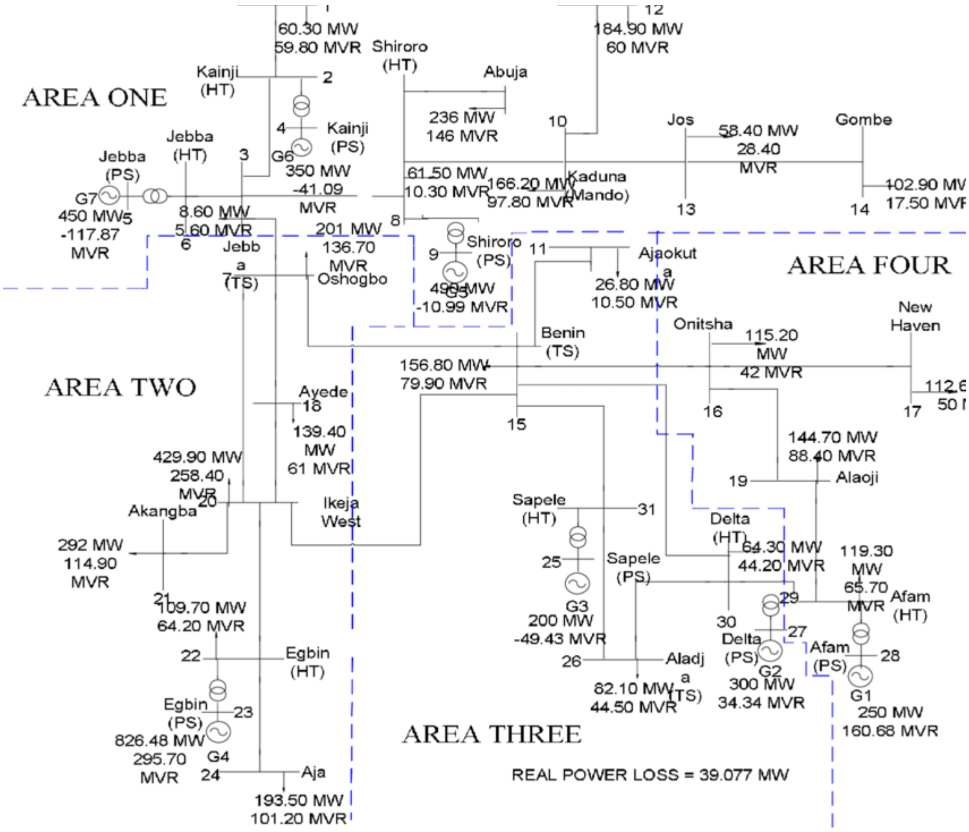


Figure 2.2. Random Power Network

2.2 The Main Entities in the market

The electricity markets are composed of four main parties that are; Demand, Supply, transmission &distribution, and Operator.

2.2.1 Supply:

The supply chain consists of either one or multiple electricity generation companies (GENCO) that utilize different types of energy sources to produce electricity which vary in magnitude and the availability time i.e. the time in which the GENCO can provide electricity.

2.2.2 Demand

The demand side has mainly two types of customers which can be categorized as “small” customers and “large” customers. Small customers are defined according to their usage, if it falls under a predefined threshold then they are considered as small customers otherwise they are considered as large customers [11]. The difference between them can be explained clearly using the previously mentioned analogy, large costumers’ consumption is large enough to be a bulb, whereas a group of small customers combined together can reach enough consumption to be considered as one. Small customers are like the normal conventional households and on the other hand, large customers are like large production factories. A number of small customers are usually handled by agencies that represent them in the market and they are known to be called “***Retailers***” or “***Demand Forecasting Agencies”***

2.2.3 Distribution and Transmission

These entities are either grouped together or operate separately depending on the conditions in a country. Coming from their names, these entities are responsible for the electricity transmission from one point to the other and also the distribution of electricity to the end customers.

2.2.4 Independent System Operator (ISO)

This entity exists to control the operating environment, it is usually a non-profit organization or a governmental institute that makes sure that no laws are being broken and the transactions are unbiased. [12]

2.3 Types of Market

The following information presented in this section is obtained from [13]

In general, two types of markets exist, the nature of the market depends mainly on a country's condition. These conditions may vary from the availability of natural resources, the advancement of technologies, and the laws and governing policies for the market.

The two types of markets are known as "**vertical company**" market and "**competition**" market. A simplified diagram illustrating the differences between them is shown below.

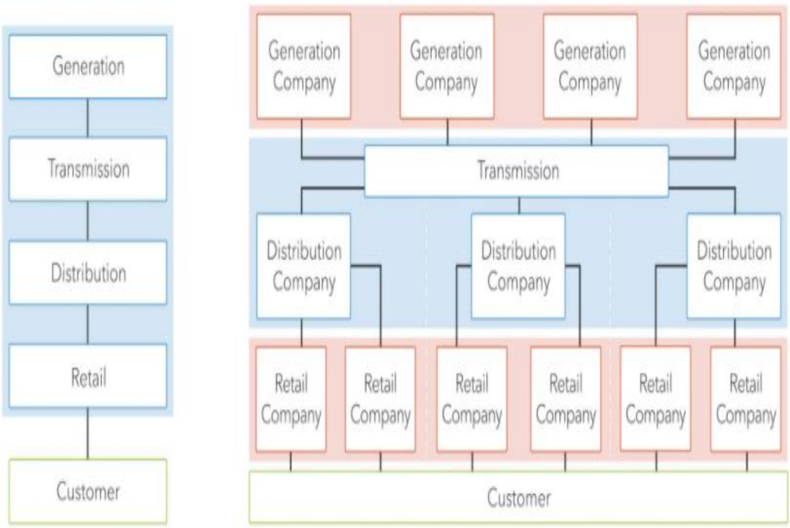


Figure 2.3. Vertical Market v.s Competition Market

The vertical market is represented by the left diagram in figure 2.3, it is important to note that the blue shade in the background defines the entities that are under the government's authority whereas the red shade resembles the private entities. As shown in the figure, vertical company markets usually let their governments operate most of the entities mentioned in the previous section from generation to distribution and even sometimes the retailing stage as well. This type of market can be usually found in developing countries or dictatorship regimes. The competition market on the other hand has a large variety of generation companies that provide electricity using different resources and have different retail prices. In addition, several retail companies exist that are responsible for a distinct number of small customers at different places. It is important to note that no matter what type of market exists, the transmission and distribution entities are always owned or operated by the country's governments or nonprofit organizations due to their high sensitivity, hence; the blue shade.

Important Remark: This Thesis will focus on the competitive market, so the upcoming sections will only be valid for that case.

2.4 The Operation Chain

In this section, the way that the electricity market operates will be explained. Electricity being treated as a commodity with the inability to be stored in large quantities due to cost constraints [14], along with the idea that demand and supply have to meet instantaneously to prevent any system failures which would cost huge losses led the market to operate on the basis of forecasting.

The general procedure is as follows, the demand forecasting agencies (retailers) and larger customers predict their electricity consumption beforehand and submit their predictions to the ISO, the ISO then reviews their predictions and presents the demand capacities to the suppliers. The suppliers, later on, submit their bids which are in the form of energy or power depending on the units used in the market and the time in which they can provide the electricity in. The market stays open until all the demanded electricity is met [15]. The above transactions are always done anonymously while being monitored by the ISO to prevent any manipulations which can cause enormous profits and losses to different parties.

2.4.1 Mechanisms utilized

Different markets exist in different countries or even different cities within a country, and every market tries to optimize its operations using different techniques and procedures depending on the availability of resources. The mechanisms utilized depend on the time frame in which the prediction works; long-term load forecasting (LTLF) and short-term load forecasting (STLF).

2.4.1.1 Long term Load Forecasting

Predictions carried out by forecasting specialists start to be considered as "long-term" when the predictions vary from 1 year to 20 years. Such predictions are known to be inaccurate due to the high probability of externalities occurring that will disrupt the prediction and deem it as invalid [16]. However, these predictions are carried out for the purpose of having a general scheme of how much electricity would be needed to be prepared in terms of resources [17]. One of the commonly used mechanisms that operate on the basis of long-term forecasting is forward contracts.

2.4.1.1.1 Forward Contracts

Indicated earlier at the end of the operation chain section that the transactions are done anonymously, however it is not the case in this type of mechanism. These contracts are usually tailored agreements done specifically between two parties which makes both sides know who are they in business with. These contracts are usually carried out between large customers who require a considerable capacity of electricity regularly and reliable suppliers who proved to supply this amount constantly [18]. Also, it can be done between ISOs and reliable suppliers to serve as a backup in case of any externalities to prevent the market from collapsing.

2.4.1.2 Short term Load Forecasting

This type of forecasting is more accurate and can vary from day-ahead predictions (24 hour period) to “X” min windows. Again, the same ideology applies, the smaller the time frame, the more reliable the prediction is, due to the decrease in the chance of having any externalities. The Day-ahead markets and Real-time markets (interday markets), and X-hour markets are the short-term prediction markets that are considered as the main pillars on which the electricity markets depend. [19]

2.4.1.2.1 Day-Ahead Markets

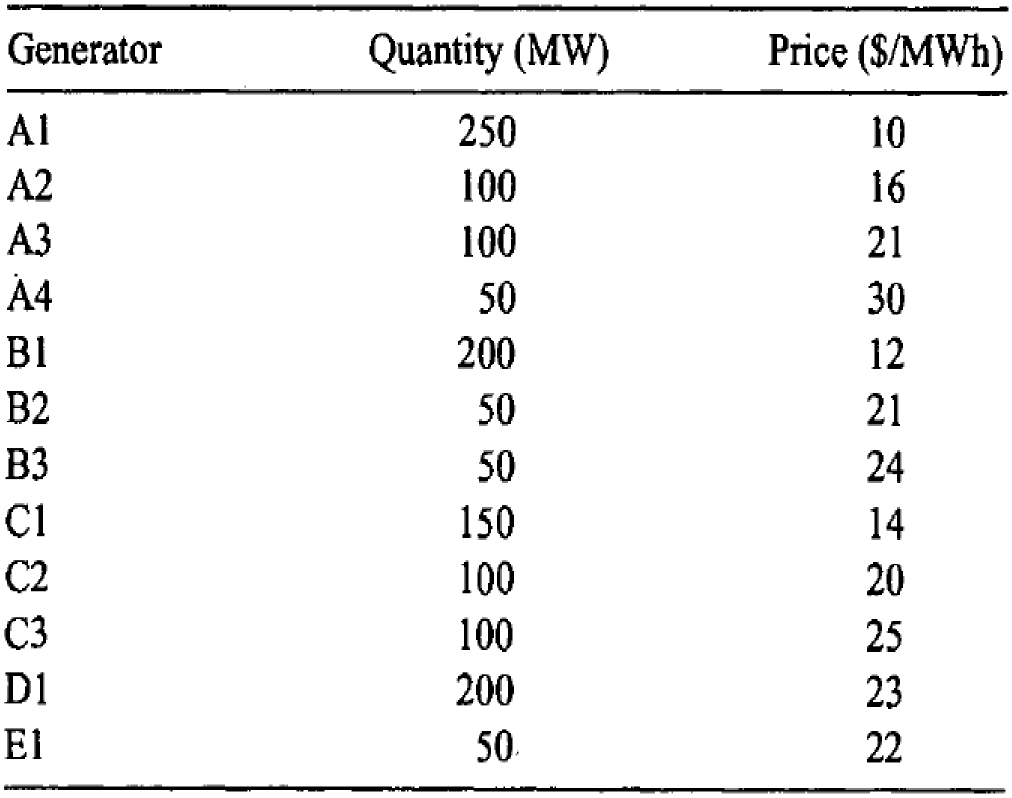
As the name indicates, the day-ahead markets are electricity transactions that are being made 24 hours prior to the actual time in which the electricity will be needed in an online platform. In this market, the suppliers are committing to a certain settlement of electricity which will be provided at a specific time the next day (24-hours later). The end customers to such settlements are kept anonymous to prevent any manipulation in the market, and the bids are being met via the ISO [20]. The day-ahead market usually operates on an hourly basis, i.e. the day-ahead market will try to allocate 50 MW to a customer today to be provided from 1 pm to 2 pm the next day.

2.4.1.2.2 Real-Time Markets

Since the day-ahead market is based on predictions, divergence from the precalculated demands may occur, that is why real-time markets existed. Intraday markets are backup markets that are operating to ensure that any miscalculations are accounted for [20]. After the day-ahead market is concluded, the real-time market starts and continues to operate until a specific time window known as the “Imbalance settlement period”. The imbalance settlement period is a time frame that the ISO decides on in which no more bids can be presented after it [21]. This window is different from one market to the other depending on the technology and resources available. The real-time market is continuously monitoring the fluctuations of the demand, and the frequency in which the generated electricity is being produced, in case of any unwanted or risky fluctuation or disturbance, the ISO which is also governing the real-time market sends dispatch information to the generation companies to be ready to account for any failures and ensure that the market continues to operate.

2.5 The Electronic Trading Process

In this subsection, a detailed example will be explained to connect all the previously mentioned terms and entities and provide a better understanding of the general scheme.



Generator	Quantity (MW)	Price ($/MWh)
A1	250	10
A2	100	16
A3	100	21
A4	50	30
B1	200	12
B2	50	21
B3	50	24
C1	150	14
C2	100	20
C3	100	25
D1	200	23
E1	50	22

Figure 2.4. Electricity Bids Example [19]

For instance, in a random electricity market, the large customers together with the demand forecasting agencies predicted that they would require a total of **500 MW** of electricity from **1 pm till 2 pm on the 23rd of October 2021**. The calculated demand was provided to the ISO on the **22nd of October 2021** in the day-ahead market, the ISO then reviewed this demand and reached out to the suppliers in an attempt to find a match for that specific time frame. Figure 2.4 above shows a table consisting of 3 columns, the left column contains a list of different GENCOS varying from A1 to E1, whereas the middle and right columns show the corresponding capacity of electricity that can be provided and its prices respectively. The variation seen in the table indicated that in that competitive market example; there is a variety of generation companies who use different energy resources and are capable of producing different amounts of electricity at different efficiencies which led to different prices [22].

Since the matching operation is anonymous, only the ISO is aware of all entities i.e. the suppliers don't know their competition, and also the customers are not informed on who is providing electricity. The ISO then starts to allocate the bids according to the increasing price of the supplied electricity which means that the electricity will be bought from the cheapest supplier to the next until the capacity is met [19]. In the provided example, **500 MW** is needed so according to figure 2.4, the cheapest supplier (**A1**) is worth **10 $/MWh** and is providing **250 MW** so the ISO will buy from him first and then move to the second cheapest one which in this case is **B1** worth **12 $/MWh** and providing **200 MW**. After buying from A1 and B1 a total of **450 MW** was obtained and what is left is just 50 MW, following the same procedure, the next and final supplier in this day-ahead market is **C1** with **14 $/MWh** and **150 MW** capacity. As a result, the ISO has fulfilled the needed capacity and will then assign the market price [Temporary Market Price] (TMP) for this period and that will be the corresponding price of the next MW that could be bought, figure 2.5 below is a graphical illustration of the temporary price decision process.[23]

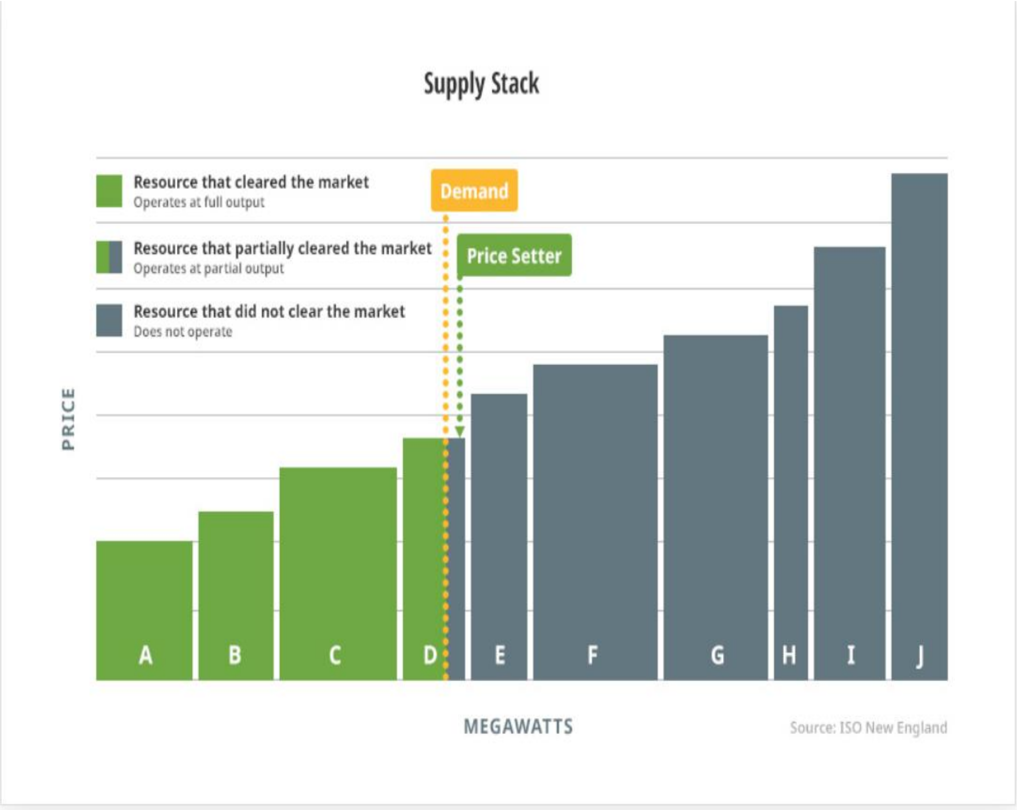


Figure 2.5. Temporary Price Assignment

As indicated in the figure above, the green color resembles the suppliers that have participated in the market and were able to sell their full capacity. D in Figure 2.5 can be considered as C1 in the mentioned example, C1 had a total capacity of 150 MW and after buying the needed electricity, the next MW will still fall under C1's domain (the semi-grey region in D), which consequently leads to the TMP of **14 $/MWh** and this will be the price presented to the demand side. [23]

At this point in the day-ahead market on **22nd of October**, the electricity (500 MW) which is demanded will be sold for the temporary price of **14 $/MWh** for the period of **1 to 2 pm on 23rd of October**. Suppliers A1 and B1 were able to sell all their capacity and profited so far 4$/MWh and 2$/MWh respectively. Supplier C1 only sold 50 MW out of 150 MW and broke even, and the rest of the generation companies weren’t able to take part in the transactions and are awaiting further notice. The day-ahead market concluded and the real-time market starts. The real-time markets then start to operate and continuously monitor any fluctuations or externalities that may alter the predictions.

Continuing the example, throughout the day (22nd of October), there were some externalities, which led to the actual demand being a total of **700 MW**; ***200 MW excess***. Unfortunately in that case, since the day-ahead market auction was concluded and time is getting closer to the actual time that electricity is needed (1 pm to 2 pm on 23rd of October), the electricity prices increased and the prices presented in figure 2.4 are no longer available.

The difference in prices is due to the following reasons; different GENCOS work using different resources that require different machinery, these machines or generators have different operational costs as well as starting times (Ramp Time) [24]. Ramp time is “the amount of time it takes from the moment a generator is turned on to the moment it can start providing energy to the grid at its lower operating limit “[24]. In the case of the “operational cost” variable, when the electricity suppliers enter the day-ahead market and they don’t succeed in selling their electricity whether partially or fully, they always face a vital dilemma of whether or not to keep their machines operating in case of an extra need due to some externality as mentioned before. If they choose to keep on operating, that means that they worked their machines for a longer time frame, and subsequently, their operation costs increase leading to a proportional increase in price.

As for the “Ramp time”, nearly all machines that produce electricity whether from fossil fuels such as diesel or renewables such as solar, need an amount of starting time to heat and reach their maximum efficiency [24].

Different power plants with their equivalent ramp time and operational costs are shown in table 2.1 and table 2.2 respectively below [24]. The difference tabulated below depends mainly on the operational principles of the machinery used in the power plants.

Table 2.1. Typical ramp and run times for power plants.

Technology	Ramp Time	Min. Run Time
Simple-cycle combustion turbine	Minutes to Hours	Minutes
Combined-cycle combustion turbine	Hours	Hours to Days
Nuclear	Days	Weeks to Months
Wind Turbine (includes offshore wind)	Minutes	None
Hydroelectric (includes pumped storage)	Minutes	None

The minimum run time shown in the right column stands for the shortest amount of time that the plant needs to operate once it is turned on. Both of these variables (ramp time and minimum run time) determine whether or not the power plant is **flexible**. These variables are highly dependent on many constraints and can be considered as functions of regulations, type of fuel, technology, etc... The flexibility of a power plant will then lead to the type of load that they could serve, either baseload or filling the peak demand. [24]

- Baseload: Is the initially predicted demand (day-ahead market) or the constant electricity needed to maintain the minimum electricity consumption. [25]

- Peak demand: This is the electricity demand at its highest and can also be considered as the amount of electricity that exceeded the initial predictions. [26]

In other words, as the flexibility of the power plant decreases (longer minimum run times and slower ramp times), they become more suitable for serving the baseload energy, while more flexible plants (shorter minimum run times and quicker ramp times) are better-suited to filling peak demand.

As for the variation in operational costs, a list of different operational costs is tabulated below.

Table 2.2. Typical operating costs for power plants.

Technology	Operating Cost ($/kWh)
Coal-fired combustion turbine	0.02 — 0.04
Natural gas combustion turbine	0.04 — 0.10
Coal gasification combined-cycle (IGCC)	0.04 — 0.08
Natural gas combined-cycle	0.04 — 0.10
Wind turbine (includes offshore wind)	Less than 0.01
Nuclear	0.02 — 0.05
Photovoltaic Solar	Less than 0.01
Hydroelectric	Less than 0.01

To summarize the two tables above, a simple graph can be drawn having operational flexibility on the x-axis and operation costs on the Y-axis. Plotted on the graph are different powerplants' technologies. [24]

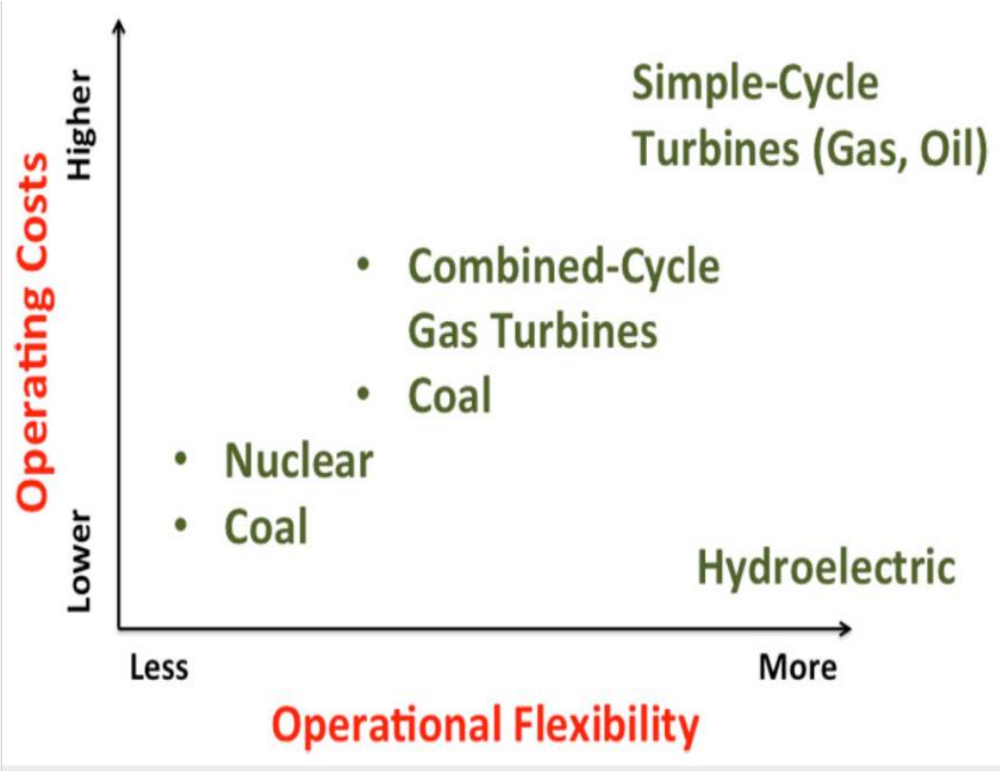


Figure 2.6. Relative comparison of operating cost and operational flexibility for different power plant technologies

Re-visiting the example, that is why, as the time gets closer to the demand meeting time (1 pm to 2 pm), generation companies who own technologies with a low ramp time are needed. Such types of machinery are known as “Spinning Reserves”, these reserves are basically any form of technology that doesn’t involve steam production in them such as simple-cycle turbines. These turbines are best suited for quick responses as indicated in table 2.1 however they have the largest costs as table 2.2 proves.

Spinning reserves usually work using natural gases and oils that not only prove to be expensive but also produce large amounts of greenhouse gases and heat energy to the environment leading to global warming [27]. A graphical illustration below in figure 2.7 shows the difference in carbon dioxide emissions per energy source throughout the years starting from 1750 till 2019. [28]

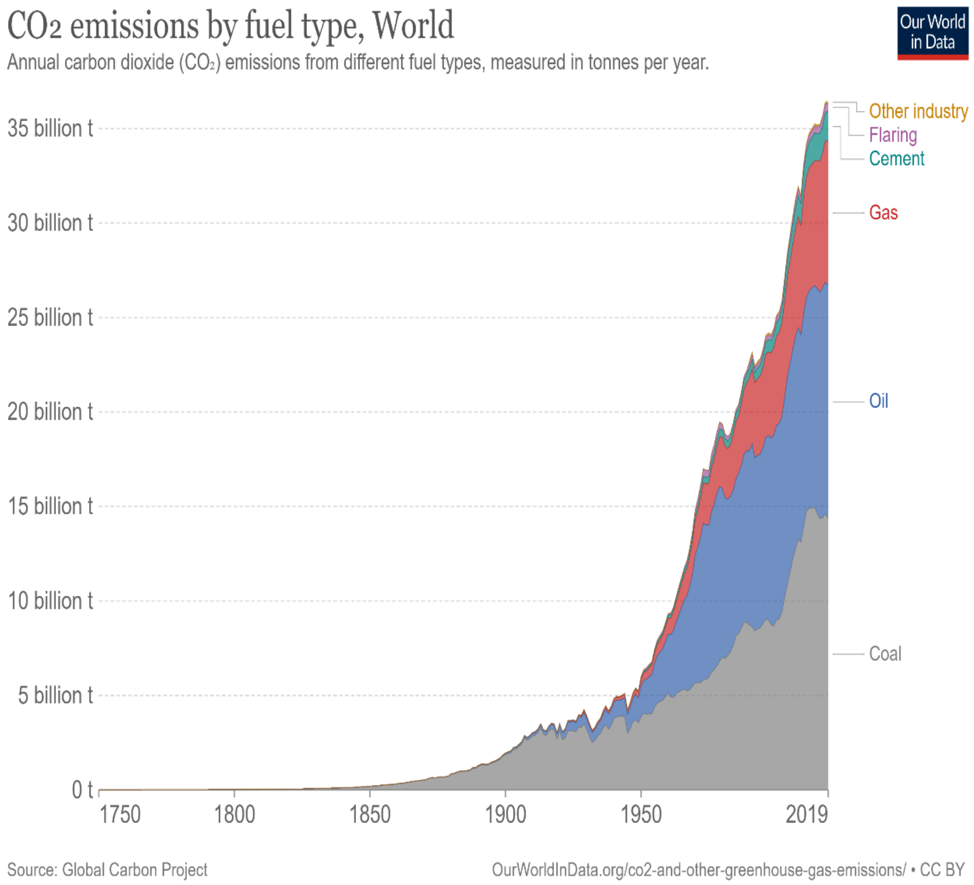


Figure 2.7. World's carbon dioxide emissions by fuel type from 1750 till 2019

As figure 2.7 demonstrates, the carbon dioxide emissions of technologies that operate on gas and oil combined are around 19.98 billion tonnes. [28]

In the example provided, to match the excess demand of 200MW that was newly predicted in the real-time market, the ISO would have to match the bids at a higher price under the condition that the suppliers chose to keep their machines operating. Not only will some suppliers now provide electricity at higher rates due to their increase in operational costs but also there will be another criterion of suppliers who own technologies with low ramp times (spinning reserves) and will also provide electricity at a high price using non-sustainable resources.

In this imaginary scenario, only one intraday market was given as an example (excess 200 MW), in real life there is a high probability or even absolute certainty that more than one intraday market will be entered, and as time gets closer and the marginal error in the day-ahead market increases (a considerable amount of deviation from the day-ahead market prediction) the need for suppliers who own spinning reserves will proportionally increase.

It can be concluded that the forecasting accuracy of the STLF is of great importance and can neither be an overestimate nor an underestimate. Also, as the day-ahead market's demand prediction accuracy increases, the number of the intraday markets will decrease, and subsequently, fewer spinning reserves will be used leading to an overall decrease in electricity prices and a more sustainable economy.

The importance of accurate demand forecasts was emphasized by Bunn and Farmer back in 1985 in the United Kingdom. An error of a 1 percent surplus in the electricity demand led to 10 million pounds of extra expenses [29]. The magnitude of these expenses was 36 years ago, if it is to be propagated to today's inflation rates, the operational costs will be much larger.

That is why constant innovation is needed to improve the accuracy of existing forecasting models and regular optimization procedures are needed to come up with superior novelties in an attempt for reaching a more sustainable and stable energy economics.

CHAPTER 3

LITERATURE REVIEW

As concluded in the previous chapter, achieving an accurate demand forecast is the main goal in the electricity market to ensure maximum efficiency in resource allocations. That is why all previously mentioned entities are continuously hiring highly-skilled individuals to work on optimizing electricity demand prediction models.

Since the electricity market has been established, numerous models have been put into use, even though they all serve the same purpose, the working principles and theory behind their operation differ drastically [30]. The prediction models' principals will be explained together with the general variables that they are a function of.

3.1 Energy Models

To begin with, the demand forecasting models can be classified into several criteria. They can be considered as univariate versus multivariate, static versus dynamic. Also, the techniques that are being utilized can vary from times series to hybrid models [30].

3.1.1 Univariate vs. Multivariate

These two terms refer to the dependency of the models on their variables. In practice, some of the models depend on a single variable, and in this case, they are identified as univariate, whereas on the other hand, multivariate models base their predictions on multiple variables (these variables can be dependent or not). [31]

The choice of creating models that are uni or multivariate depends on many factors, such as the geographic place, living conditions, and most importantly the intuition of the model designer [30]. The general differences between the two models can be tabulated in table 3.1 below [32].

Table 3.1. Differences between Univariate and Multivariate models

Univariate	Multivariate
Involves a single variable	Involves multiple variables
Does not deal with cause and relationships	Deal with cause and relationships
The major purpose is to describe	The major purpose is to explain
It uses dispersion methods like range and variance	It uses correlations
It can be illustrated as frequency distribution	It can be illustrated as relationships and tables
The results can be shown as bar graphs, histograms, pie charts, etc…	The results can be shown as tables where one variable is contingent on other variables
Analysis acquire a short time	Analysis acquire a longer time

3.1.2 Static vs. Dynamic

Static or dynamic models are usually implied on energy models that involve a learning stage then a testing stage. A learning stage is a period in which the machine or the model is fed with previously gathered data to create functions that describe the data's variation.

The data that is input into the system can be either in a form of a single variable (univariate) and in this case, the model will try to create one function that suits the variable's changes, or several variables (multivariate) and in that case, the model would be more complicated and will require more time to create several functions to come up with reliable trends. In this training phase, there is no difference between static and dynamic models.

After passing through the learning stage, the testing stage starts, where the accuracy and reliability of the model's created algorithm would be tested. The difference between static and dynamic models starts to appear in the testing period. The differences can be illustrated in Figures 3.1 and 3.2 below [33].

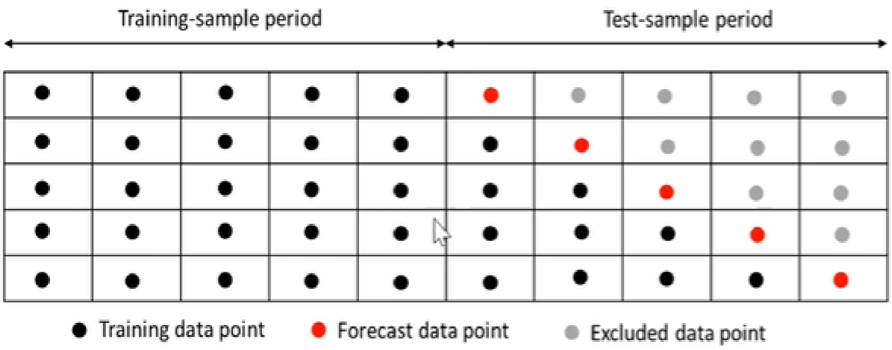


Figure 3.1. Static Forecast

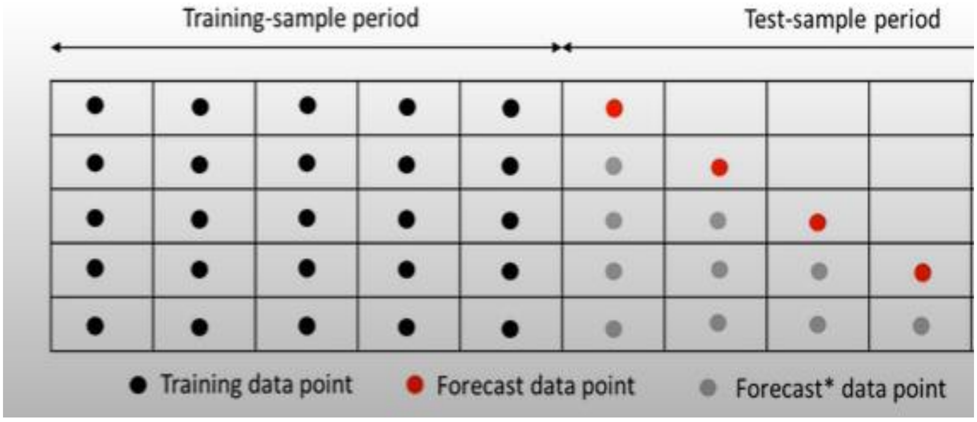


Figure 3.2. Dynamic Forecast

As shown above, in figures 3.1 and 3.2’s legends, the red dots represent the forecasted data, black dots refer to the training data, and the grey dots account for the data that is to be predicted. As the points move horizontally to the right, the time increments which are represented by cells in the table increase. The main difference lies in the data usage for predicting every next time increment. In figure 3.1 for static prediction, the first row presents the training data used to predict the first-time increment which is represented by red. When it comes to the second prediction in row 2, the previously predicted time increment is colored as black which means that the predicted point in row 1 is now considered by the model as part of the training data which concludes that as time progresses, the training set is continuously increasing since the previous predictions are always being added to the training data set. Dynamic models on the other hand stick to the pre-set training data and only uses it for predicting all-time increments as shown in figure 3.2 [34].

3.2 Inputs

Different electricity markets are operating all around the globe, each of these markets may contain sub-markets that co-exist within. Due to the absolute diversity that is available, there is no set list of inputs that can be defined to achieve an efficient energy model.

It is important to note that within one distinct electricity market, demand forecasting agencies may use different prediction models that operate on different principles as discussed in the previous section as well as their choice of what inputs would be utilized, and this is the same scenario for suppliers and ISOs. That leads to the usage of numerous models that work with different inputs by different entities all to serve one purpose; to predict electricity demand as accurately as possible.

All types of data are available to all entities at all times, however since the electricity consumption is chaotic, the probability of experiencing different types of externalities is infinite, which leads to the fluctuations in the reliability of each model being utilized depending on the nature of the externality. The types of input can fall mainly under two categories qualitative and quantitative. [34]

3.2.1 Quantitative input

Quantitative inputs are data or variables that are tangible and can be represented by numbers [35]. This leads to the ease of feeding such type of data into an artificial model. Some of the commonly used quantitative data for electricity demand predictions are [29]:

- Temperature
- Humidity
- Past Consumption
- Income and price elasticity
- Number of customers
- Population
- Climate factors:
    - Dry bulb temperature
    - Wet-bulb temperature
    - Global solar radiation
    - Clearness index
    - Wind speeds
- Energy price
- Technology and advancements
- Previous years' energy's demand
- Agricultural production output, industrial production output
- Population Growth, economic growth, urbanization level

3.2.2 Qualitative Input

The nature of this input is non-numerical, this form of data is gathered through an interactive procedure, where one on one interviews, open-ended surveys, methods of observations, etc... are conducted and the results of such analysis are presented in the forms of categories or clusters of similar individuals. [36]

An appropriate example of such data in the electricity market can be as follows; due to the increase in greenhouse gases emissions, the temperature variations are unstable, and several countries are experiencing unusual heat or cold waves throughout the year. In country “X”, new regulations were announced stating an increase in the wages for army officers, and there are general neighborhoods in this country where the majority of the population are army officers. After the general increase in wages, these people could have an increased tendency to increase their electricity consumption in summer or winter in the form of increasing the operation hours of their heaters or chillers.

Since no previous data is reporting this new trend of consumption, this behavioral change needs to be incorporated through the prediction model, therefore open-ended questionnaires or surveys could be carried out to divide the people into groups and somehow quantitate this nontangible data to increase the accuracy of the prediction model.

To sum up the dissimilarities between qualitative and quantitative Inputs, table 3.2 can be drawn below. [37]

Table 3.2. Qualitative vs. Quantitative data

	Qualitative (Categorical)	Quantitative (Numerical)
Focus	Quality	Quanttity
Method	Ethnography/ observations	Experiments/ correlations
Sample	Small, purposeful	Large and random
Data collection	Interviews and observations	Scales and tests
Analysis	Inductive (by a researcher)	Deductive (statistical methods)
Findings	Comprehensive, descriptive	Precise, numerical

3.3 Trend Variation

Regardless of what type of input is fed into the system or what the system's working principles are, any form of data will vary with respect to a pre-set datum (time in electricity demand's case). These variations are of several types; linear variation, non-linear variation, and seasonal variation. [38]

3.3.1 Linear variation

This variation occurs when one variable is directly proportional to another by a constant factor. This variation is identified graphically when a straight line is seen on a graph as shown below in figure 3.3. or by an equation like equation 3.1.

$$Y = mX + c \qquad \text{(Eq. 3.1)}$$

Where Y is the variable of interest, X is the datum, m is the proportionality constant (slope) and C (Y-axis intercept) is just a constant that is algebraically added to the variable.

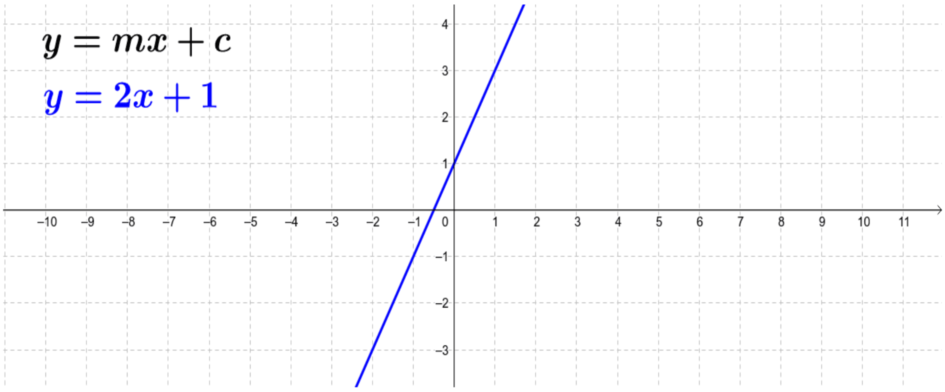


Figure 3.3. Direct linear variation

Figure 3.3 above demonstrates a graphical representation of a simple linear equation with a slope equal to 2 and a y-axis intersection of 1.

3.3.2 Non-linear variation

As the name implies, non-linear variation is any form of relationship that connects two or more variables and is represented by anything other than a straight line. There is no general equation for such variation and there are infinitely many illustrations of this trend [39]. Some of them are represented below in figure 3.4.

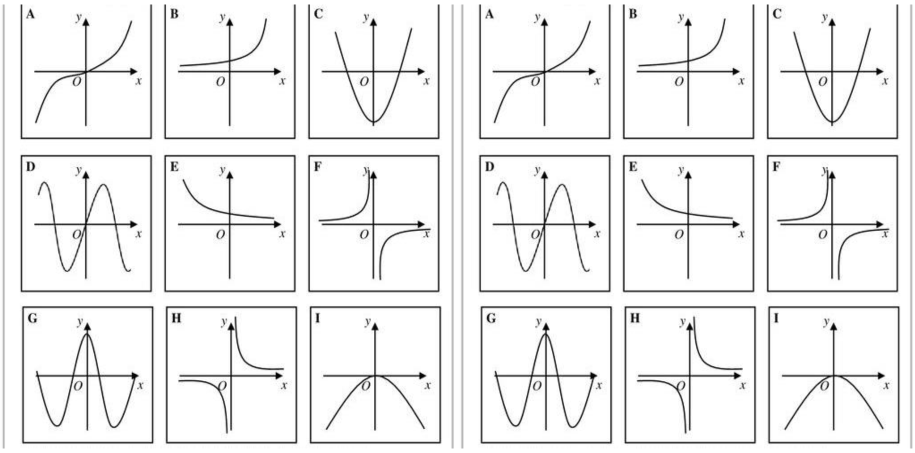


Figure 3.4. Various Non-linear variations

3.3.3 Seasonal Variation

This type of variation is seen in time-dependent relationships. Seasonal variation is unique in the sense that it can involve both linear or non-linear variations, however, the variance is not constant. Seasonal variations can include multiple trends that keep on re-occurring every specific time interval [40]. A good example of seasonal variation is what is shown on a heart rate beep on a monitor.

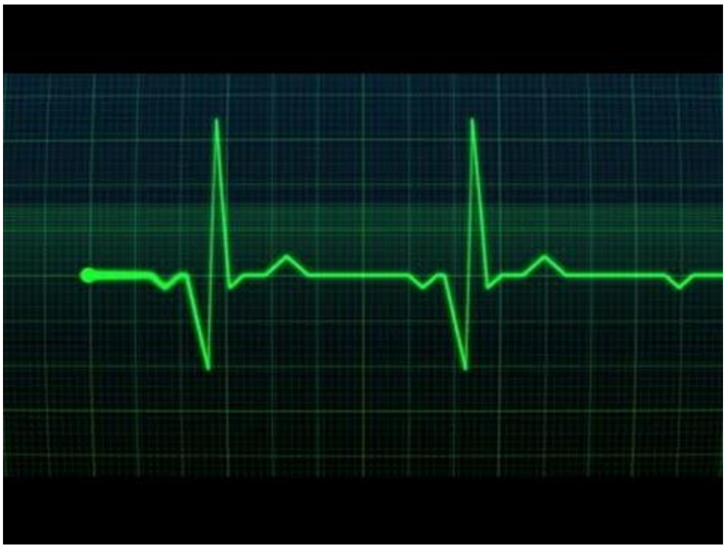


Figure 3.5. Heart Rate Monitor

As shown above, in figure 3.5, the heart's activity is a simple non-linear variation that keeps on repeating itself every specific time interval.

Generally, there are two types of seasonality; additional and multiplicative, the additional seasonality is just numbers being added and this can be seen in linear trends, whereas multiplicative is shown as a result of a compounding effect of percentage growth [41]. Additive and multiplicative seasonality can be shown below in figures 3.6 and 3.7 respectively.

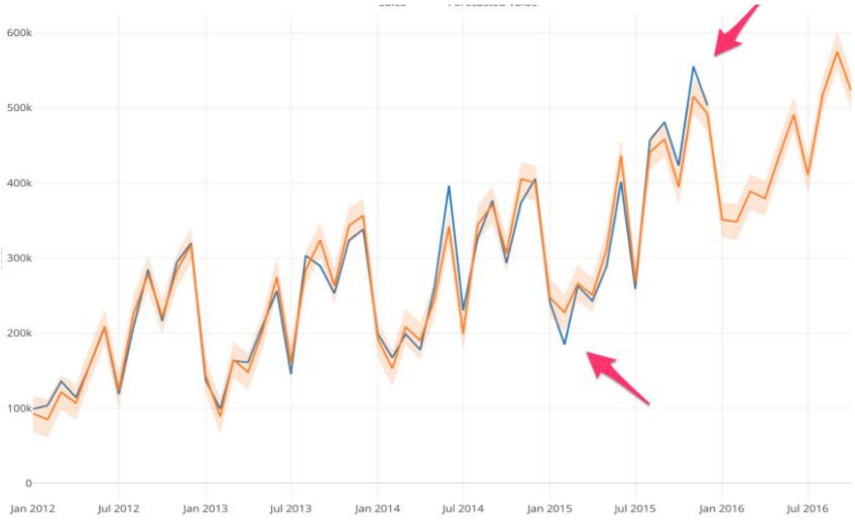


Figure 3.6. Additive Seasonality

As pointed by the red lines in figure 3.6, the difference between the actual values (blue) and the forecasted values (orange) are becoming wider as time goes by. Whereas in figure 3.7 the difference between the actual values (blue) and the forecasted values (orange) are equally spread across the years. [41]

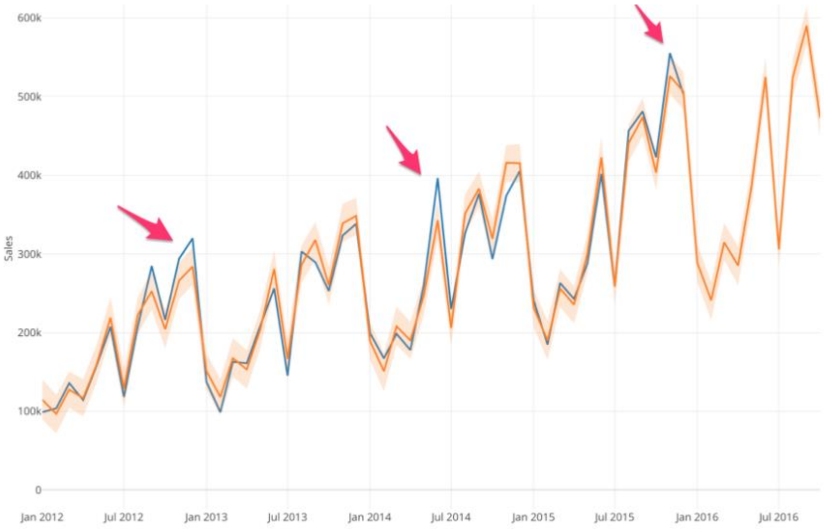


Figure 3.7. Multiplicative Seasonality

3.4 Reliability of the Results

The studied models and techniques are utilized for the sake of predicting the future, and since numerous externalities can occur considering that the predictions are done in an uncontrolled environment, the accuracy and precision of the models are crucial in the decision process of whether or not the model will be used.

Also, it is important to note that a direct comparison between models is not possible, in view of the fact that different models use different prediction techniques as well as different inputs. However, there are general accuracy and precision tests that can be applied to all models to have a singular comparison criterion. These tests can be in a form of a percentage or a scale where the maximum value is identified, it can be set as the smaller the result the better, or vice versa [42]. Some of the commonly used assessment methods are:

- Willmott's Index (WI) (*The Higher The Value The Better*)
- Mean Absolute Error (MAE). (*The Lower The Value The Better*)
- Root-Mean Square Error (RMSE). (*The Smaller The Percentage The Better*)
- Mean Absolute Percentage Error (MAPE). (*The Lower The Value The Better*)

The working principles of these methods will not be discussed in this thesis because it is outside the scope of the topic.

After understanding the discrete types of models and the variety of inputs that can be utilized for electricity demand predictions together with their different nature, in addition to the methods, that validate the models' results. Different forecasting models in the literature will be mentioned in the upcoming sections.

3.5 Forecasting Models

There is a wide variety of forecasting models that are being utilized around the globe. As previously noted, this variety is due to the fact that these models take economic, environmental, and social factors into account which as a consequence leads to seasonal, monthly, daily, and hourly fluctuations in the electricity demand patterns [30]. These models are mainly categorized into three main approaches: ***parametric*** which are Regression and Time-series models, nonparametric that mainly work using artificial intelligence, and the last category is known as hybrid models that mix both of them. Some of the commonly used parametric models are:

- Time-series models
- Regression models

3.5.1 Time-series Models

Time series models work with any variable that is sequentially changing with respect to time. This type of model must require historical data for it to be used and it was proved that it works best with seasonal data [43]. In time series analysis, as in any other study, the data is considered to contain a systematic pattern (typically a set of identifiable components) and random noise (error), which makes the behavior difficult to recognize [44]. Figure 3.8 demonstrates a time series graph with noise included. [45]

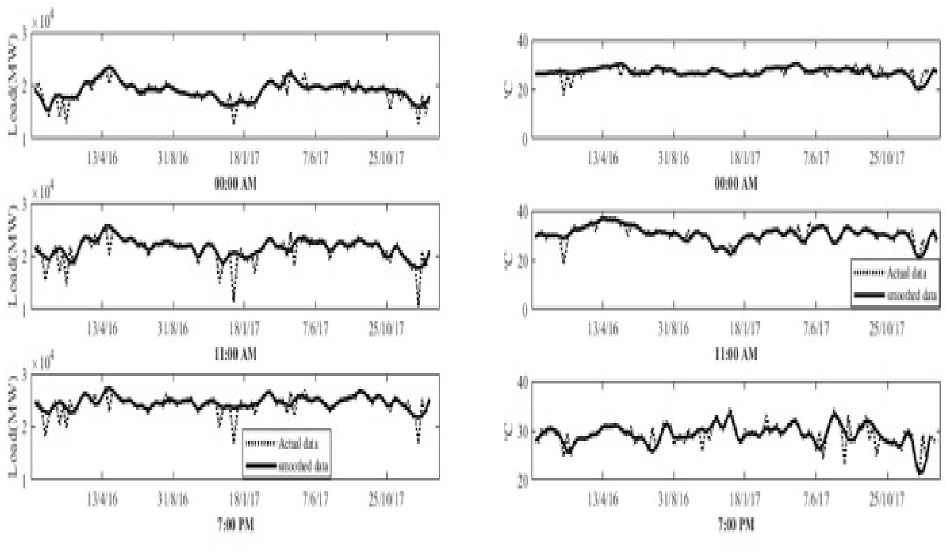


Figure 3.8. Before and After noise reduction

Filtering away noise is a common approach used in time series analysis to make the pattern more smooth. Most time series analysis approaches use some type of noise filtering in order to smooth out the pattern. Time series analysis is used to identify and analyze the effects of time-related elements on time series values [46].

This type of modeling was used for forecasting energy production and consumption in Asturias-Northern Spain with a mean percentage error of 1.6 percent [47], which is considered as a relatively low error proving to be an accurate estimate.

3.5.2 Regression models

The following information is obtained from [48]

In electricity markets, several variables contribute to the demand predictions and these variables are dependent on many other factors (multivariate models), so indirectly demand prediction is dependent on these factors as well. Regression models aim to identify which of these numerous factors have a considerable contribution in changing the behavior of demand.

Regression analysis is a mathematical method of determining which of those factors has an effect. It provides answers to the following questions: Which factors are most important? Which of them can be ignored? What is the relationship between those variables, if there are any?

In regression analysis, there are two main types of variables, the ***dependant variable***; the variable of interest, and there are the ***independent variables***; and these are the factors that are thought to alter the variable of interest. For a regression analysis to be carried out, data has to be gathered for both the dependant and the independent variables, then this data will be plotted against each other with the dependant variable on the y-axis. For instance, it was seen that temperature variation contributes to changing the electricity consumption (demand), data were gathered for both variables and were plotted as shown below.

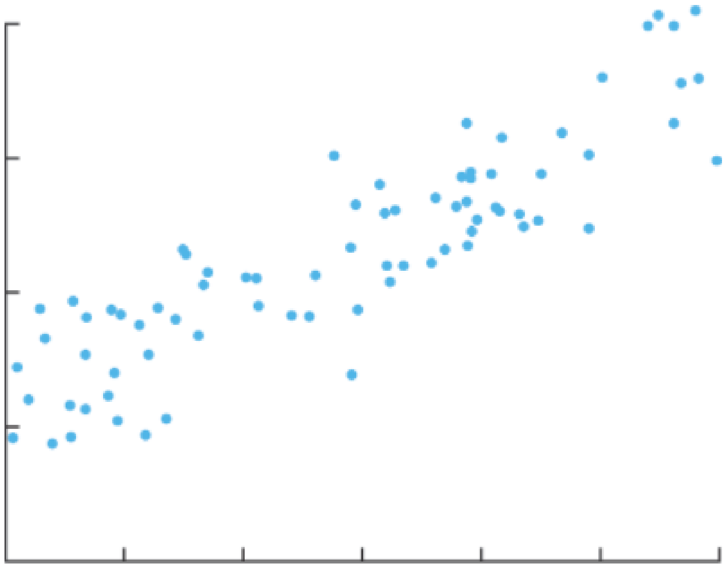


Figure 3.9. Electricity demand vs. Temperature

In figure 3.9, the variable of interest (dependant) which is the demand is plotted on the Y-axis whereas the temperature is plotted on the x-axis. It can be understood from the graph that as the temperature increases, the demand also increases but by what factor is what is important, also the knowledge of how will the demand increase if higher temperatures were seen is of great importance, that is why just observing scattered data is not sufficient for reaching a reliable conclusion. Regression models start first by drawing the best fit line to increase the level of certainty when drawing conclusions.

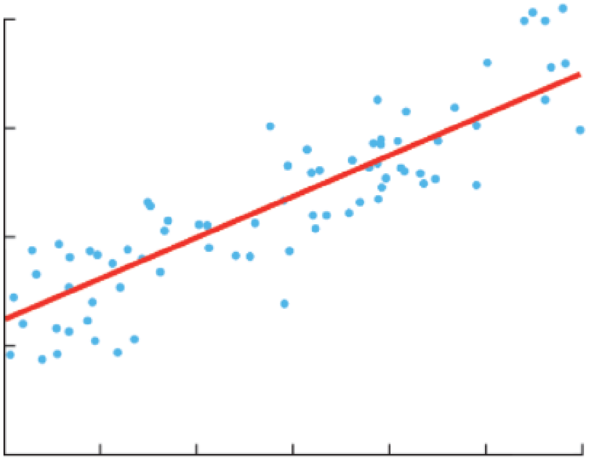


Figure 3.10. Regression line

The best fit line in red which is drawn in figure 3.8 is named as a regression line. Not only does this model draw the regression line, but it also provides an equation for this line.

$$Y = mX + c + error \qquad \text{(Eq. 3.2)}$$

Eq.(3.2) above is a linear regression equation, regression models may use other mathematical operations to explain the best fit line such as multiplication which would result in a completely different equation and also a different line shape (curve for example). However, for the sake of simplicity, a linear example was given. Following this formula, one can know the amount of electricity demand when the designated 0 scaled on the x-axis is reached, the factor by which the demand increases when the temperature increases, and also it provides an error term since the best fit line is not 100 percent accurate. This error term serves as a criterion of whether or not the independent variable can heavily affect the dependant variable's behavior as well as an indication of whether the prediction is accurate or not. The mentioned scenario is an example of a linear trend, however, regression models can deal with non-linear and seasonal data as well. The type of regression model that deals with linearity is named as Linear Regression Models (LRM).

The mentioned example included only one variable but in practice, there are n number of variables that are to be tested, and a more general equation can be listed below [49]:

$$y_i = f(x_i, \beta) + e_i \qquad \text{(Eq. 3.3)}$$

Where y_i is the dependent variable, x_i is the independent variable, β are unknown parameters and e_i are the error terms. The error term is usually represented using the R^2 method.

Regression models were used to study the relationship between several economic factors and annual electricity consumption in northern Cyprus. A correlation was reached connecting energy consumption with the number of customers, the price of electricity, and the number of tourists. The results included R^2 equal to 0.930 which indicated that the model including these factors will have a very strong predictive ability and can be used to forecast future annual electricity consumption. [50].

Also, a multiple linear regression model (MLR) was implemented by Haida & Muto in order to predict the daily demand peak values throughout a year utilizing transformation functions to deal with the non-linear correlation between temperature (independent) and load (dependant). [50]

3.5.3 Machine Learning and Artificial intelligence Approaches

In this section, non-parametric approaches will be discussed and numerous applications where the models were applied will be mentioned. The structure of this section will be a bit different from the previous ones, since the machine learning models are more complicated, covering some of the theories behind them will be difficult, however relevant theory for the proposed models will be explained in detail in the upcoming chapters.

Over the past twenty years, techniques involving computational intelligence have been extensively used in electricity demand forecasting (especially ANN). ANN has been the focus of experts due to its unique ability to model non-linear behaviors of load series. D.C. Park et al. were one of the earliest authors who applied ANN on STLF, in their paper, highly-accurate results were presented for 1-hour and 24-hour (day-ahead) demand predictions as well as peak load values using the data that was presented by Puget Sound Power and Light (an energy utility company based in the US). [51]

Other researches such as [52], used weather variables like temperatures, wind speed, and cloud cover to reduce the forecasting error. They plugged in 51 different scenarios for each weather variable into the system using data from England and Wales and were able to get accurate results for 10-day ahead forecasts, however, the results obtained were underestimates of the actual demand yet it was considered a more reliable model than models that utilize only one weather variable with one scenario. The method they introduced is known as “weather ensemble”.

As time progressed, NN gained higher popularity, and attempts of improving its performance were the main road that the industry was taking. For example, bayesian frameworks were added by the specialist to aid in identifying which type of ANN will be used and assign the number of inputs together with the number of hidden layers that the ANN will consist of. [53]

Nonlinear autoregressive models with exogenous inputs (NARX) are distinct forms of neural networks that are composed of an optimized architecture that reduces the feedback time, they have been recently used in short-term load forecasting however the studies discussing their performance are limited.

Support Vector Machines (SVM) is also a well-known family of Machine Learning approaches, under its umbrella, Support Vector Regression (SVR) is a commonly used approach for demand forecasting that has started to be implemented since the beginning of the current century. Mohamed Ahmed Mohandes wrote one of the first papers that applied SVR for STLF, in his research he also compared SVR with Auto Regression models (AR) for one-hour load forecasting in Saudi Arabia [54]. SVM was also put into comparison with ANN, Guo et al. used both models to forecast monthly electricity in Hebi province in China, and concluded SVM's superiority [55].

R. Weron stated in his book ***" Modeling and forecasting electricity loads"*** that SVM 's speed and accuracy depend heavily on the type of input that is being fed into it, which led to the implementation of attribute reduction models to SVM to counter such a problem [56]. Other methods were suggested in [57], where the usage of the simulated annealing approach was utilized for finding the optimum input parameters for SVR.

SVR and ANN are the widely and commonly used techniques in the market, however, multiple new approaches for STLF are being introduced by the day that also includes artificial intelligence such as fuzzy logic (FL), chaos models, or a combination of them.[58]

Also, it is important to note that there exist regression models that are not considered as parametric approaches and they include heavy computational work. In spite of the fact that they are named as regression, their functions are directly generated using the data itself without any need for parameter estimation. This method was carried out in [59] using kernel estimators rather than assuming a particular distribution.

3.5.4 Hybrid Models

As mentioned earlier in the introduction, each of the models, whether they are parametric or nonparametric have their strengths when dealing with specific data and also weaknesses. In order to deal with a wide variety of variables that have different behaviors (Linear, nonlinear and seasonal), hybrid models were introduced. Hybrid models combine different models to ensure maximum output efficiency. There is no set of mixtures that Hybrid models consist of, the literature is continuously working on finding new combinations that would suit best the type of data used.

Fuzzy logic (FL) and ANN were mixed in a way that FL is used for the training phase of a NN for STLF and resulted in a new model known as fuzzy Neural network (FNL), the results were promising, it was found that FNN was considerably faster than the conventional neural network while still preserving the accuracy.[60]

Another variation of this combination is known as evolving fuzzy neural network (EFuNN) which was applied to half-hourly load data to forecast two days ahead. This proposed hybrid model was then compared to ARIMA and ANN models and proved to be more accurate. [61]

ARIMA and ANN were combined to capture linear and non-linear trends in monthly peak loads for Jeddah, Saudia Arabia. The ARIMA served as the initial forecast and then the resulted data was fed into an ANN which resulted in accurate results [62]. Related work was seen in a combination between ARIMA, SVR, and cuckoo search algorithm (CSA) but in this method, the ARIMA results are then fed into SVR to capture the residual non-linear components [63].

As it can be concluded, there can be infinitely many integrations that result in a hybrid model. These hybrid models have proven to be superior according to the literature. The aim of this thesis is to apply a possible combination of models that would be suitable for the data obtained, to reach better and more accurate forecasts

CHAPTER 4

EMPIRICAL DATA AND CASE STUDY

In this chapter, the decision process that was involved in the procedure of choosing a case study, and obtaining the reliable relevant data for the work-frame will be discussed. In addition, important remarks and comments will be noted for the data used and the real-life factors that contributed to it.

4.1 The Market

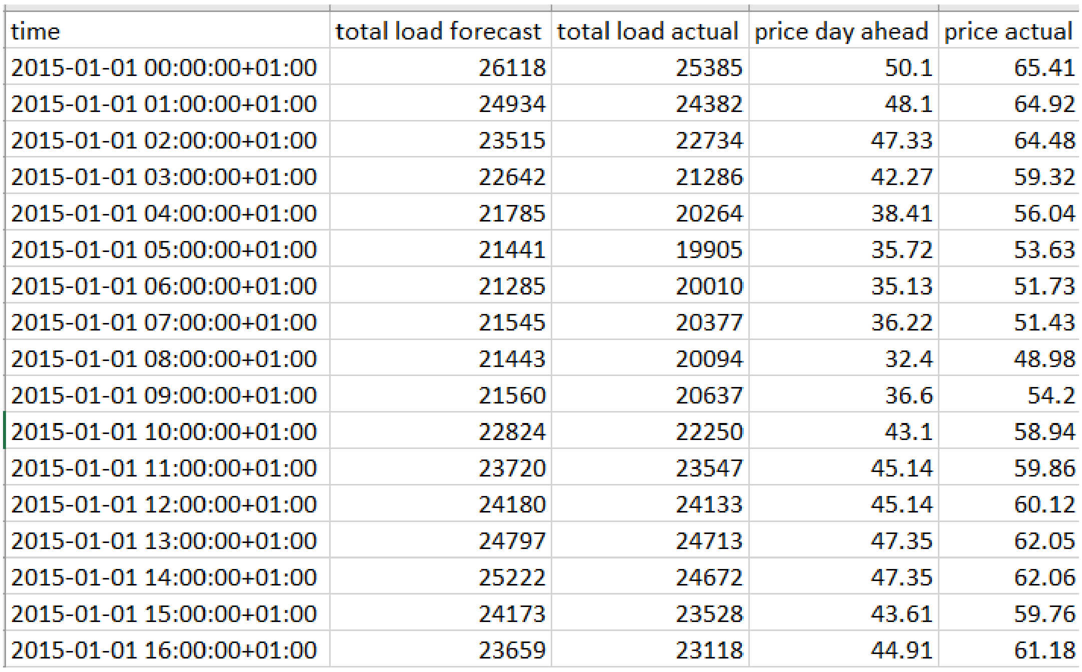
To begin with, the workflow started with the choice of working with either a vertical market or competition market, it was concluded that a competitive market will be best suited for the proposed idea since it contains numerous dynamics that will contribute to superior results if the model operates optimally.

4.2 Type of Forecasting

As discussed earlier, as the time domain for the prediction decreases, it is expected that the obtained forecast would be of greater accuracy since fewer externalities are anticipated to occur, which led to the decision of choosing to apply short-term load forecasting. Under the short-term forecasting, Day-ahead markets were chosen to work with.

4.3 Case Study

In an attempt to find data that suits STLF for a day-ahead market in a country that operates its electricity markets in a competitive nature, reliable historical data was found that contains 4 years worth of electricity consumption and pricing starting from the first of January 2015 till 31st of December 2018. The electricity consumption data was obtained from the European Network of Transmission System Operators for Electricity (ENTSO-E), whereas the settlement prices were gathered from the ISO responsible for managing the Spanish market Red Eléctrica España. A sample of the data found is shown below in figure 4.1 [64].



time	total load forecast	total load actual	price day ahead	price actual
2015-01-01 00:00:00+01:00	26118	25385	50.1	65.41
2015-01-01 01:00:00+01:00	24934	24382	48.1	64.92
2015-01-01 02:00:00+01:00	23515	22734	47.33	64.48
2015-01-01 03:00:00+01:00	22642	21286	42.27	59.32
2015-01-01 04:00:00+01:00	21785	20264	38.41	56.04
2015-01-01 05:00:00+01:00	21441	19905	35.72	53.63
2015-01-01 06:00:00+01:00	21285	20010	35.13	51.73
2015-01-01 07:00:00+01:00	21545	20377	36.22	51.43
2015-01-01 08:00:00+01:00	21443	20094	32.4	48.98
2015-01-01 09:00:00+01:00	21560	20637	36.6	54.2
2015-01-01 10:00:00+01:00	22824	22250	43.1	58.94
2015-01-01 11:00:00+01:00	23720	23547	45.14	59.86
2015-01-01 12:00:00+01:00	24180	24133	45.14	60.12
2015-01-01 13:00:00+01:00	24797	24713	47.35	62.05
2015-01-01 14:00:00+01:00	25222	24672	47.35	62.06
2015-01-01 15:00:00+01:00	24173	23528	43.61	59.76
2015-01-01 16:00:00+01:00	23659	23118	44.91	61.18

Figure 4.1. Sample Data

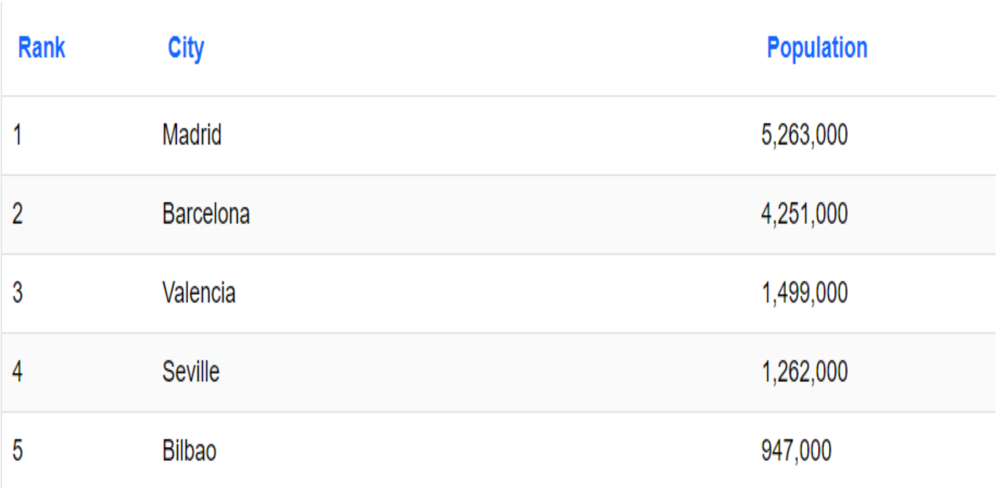
The data represented above in the time column is in an hourly fashion which indicates that the presented data is for the day-ahead markets; as mentioned in the earlier sections, that were entered throughout these 4 years. The other four columns represent the corresponding forecasted load and the actual load together with the forecasted price (TMP) and the actual price. It can be seen that none of the predicted values are equal to the actual ones and as explained in the electronic trading process section, the actual prices are more expensive indicating that intraday markets were seen. This thesis's proposed model aim to decrease the deviation between the actual and forecasted load, to see a corresponding decrease in the actual retail prices for electricity. This data will be used as a reference to the final results derived from the model to compare the performance of the model to the applied methods in the market between the years 2015 and 2018. Also, the actual data presented in figure 4.1 will be fed into the model to create correlations between the inputs chosen and obtain the desired results.

4.3.1 Input Type

At this point, the output or desired data is obtained, the input data that will be fed into the system is now of interest. The decision of whether to choose qualitative or quantitative data was quite easy, carrying out surveys or questionnaires for understanding the behavioral consumption of Spanish consumers in the years 2015 till 2018 now in 2021 is unrealistic, also, extensive research was carried out to find such data however all attempts were in vain, which led the methodology to shift towards quantitative data.

In the literature review carried out, it was seen that one of the most commonly used quantitative types of inputs is weather conditions, also reliable results were obtained from methods containing them as inputs. That is why weather conditions and variables were chosen as the inputs for the proposed model. Another important factor that contributed to this choice is that; the ENTSO-E which is the responsible party for predicting the electricity consumption for most of Europe, and is the source of the used data, is also using weather conditions as their inputs for prediction [65]. Having common inputs with the reference method will flatten out the comparison between models.

After choosing to work with weather conditions as inputs, an obstacle was encountered which is, the fact that the goal is to predict electricity demand for Spain as a country, but weather conditions vary geographically within the country. To overcome such a problem, it was seen that it is safe that instead of feeding all the weather conditions of all cities and neighborhoods into the model which will result in a more complicated model that will require a long time frame to produce results, the top 5 cities’ (in terms of urbanization and population) weather conditions will be enough or will give accurate estimates. Shown below, is the list of the top 5 cities in Spain in terms of population by the end of 2018. [66]

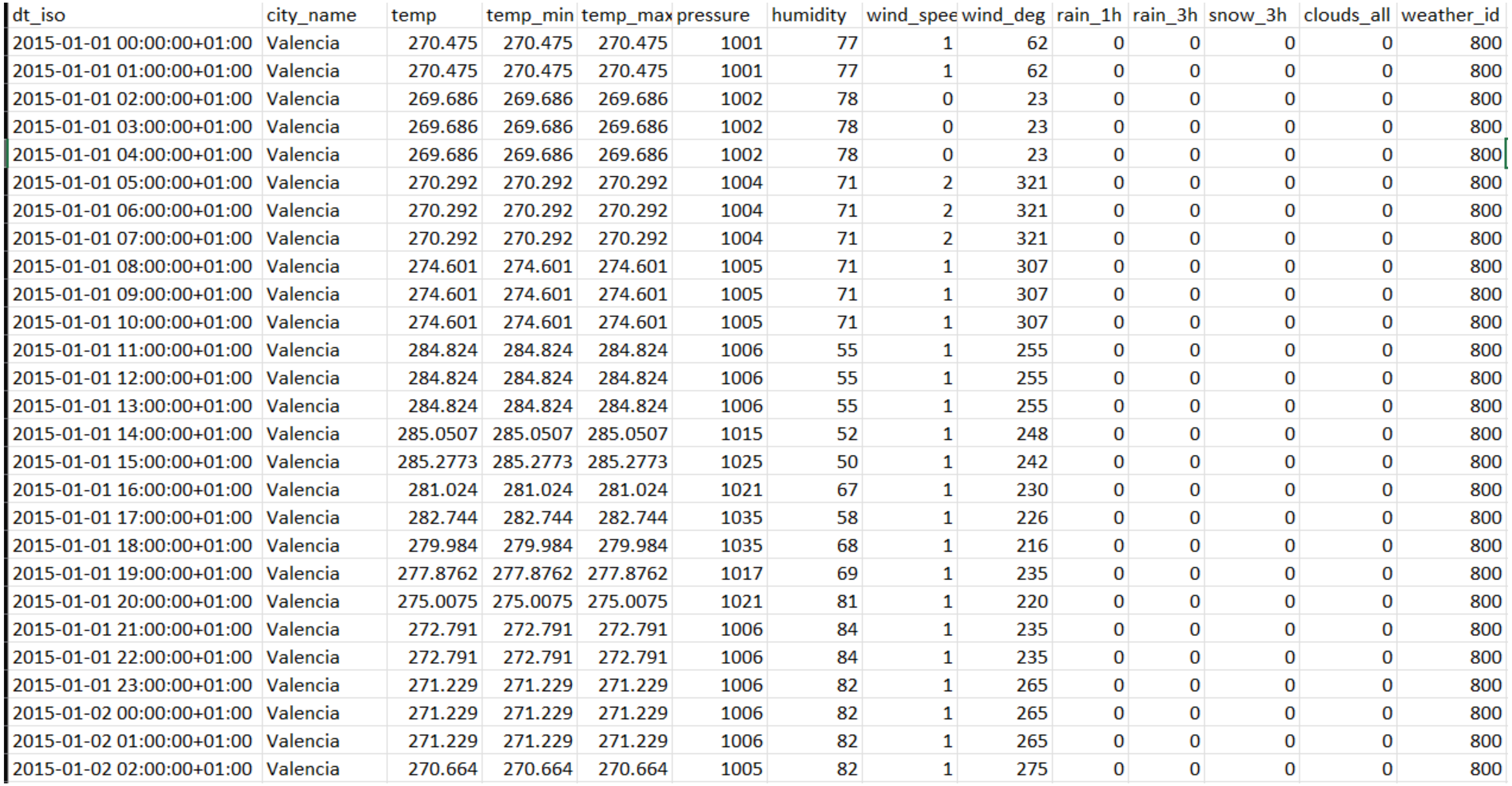


Rank	City	Population
1	Madrid	5,263,000
2	Barcelona	4,251,000
3	Valencia	1,499,000
4	Seville	1,262,000
5	Bilbao	947,000

Figure 4.2. Top 5 cities in terms of population in Spain

As figure 4.2 tabulates, the five largest cities that will be used in the model are Madrid, Barcelona, Valencia, Seville, and Bilbao.

Later on, the search was initiated to find the weather data for these 5 cities and it was found to be available in weather API; an application that provides current and historical weather data on a global scale. Filtering the useful weather data for these cities in Spain and connecting it to the same time domain as the data gathered from the ENTSO-E, a sample of the resulted data presenting Valencia is shown below in figure 4.3. [67]



dt_iso	city_name	temp	temp_min	temp_max	pressure	humidity	wind_spee	wind_deg	rain_1h	rain_3h	snow_3h	clouds_all	weather_id
2015-01-01 00:00:00+01:00	Valencia	270.475	270.475	270.475	1001	77	1	62	0	0	0	0	800
2015-01-01 01:00:00+01:00	Valencia	270.475	270.475	270.475	1001	77	1	62	0	0	0	0	800
2015-01-01 02:00:00+01:00	Valencia	269.686	269.686	269.686	1002	78	0	23	0	0	0	0	800
2015-01-01 03:00:00+01:00	Valencia	269.686	269.686	269.686	1002	78	0	23	0	0	0	0	800
2015-01-01 04:00:00+01:00	Valencia	269.686	269.686	269.686	1002	78	0	23	0	0	0	0	800
2015-01-01 05:00:00+01:00	Valencia	270.292	270.292	270.292	1004	71	2	321	0	0	0	0	800
2015-01-01 06:00:00+01:00	Valencia	270.292	270.292	270.292	1004	71	2	321	0	0	0	0	800
2015-01-01 07:00:00+01:00	Valencia	270.292	270.292	270.292	1004	71	2	321	0	0	0	0	800
2015-01-01 08:00:00+01:00	Valencia	274.601	274.601	274.601	1005	71	1	307	0	0	0	0	800
2015-01-01 09:00:00+01:00	Valencia	274.601	274.601	274.601	1005	71	1	307	0	0	0	0	800
2015-01-01 10:00:00+01:00	Valencia	274.601	274.601	274.601	1005	71	1	307	0	0	0	0	800
2015-01-01 11:00:00+01:00	Valencia	284.824	284.824	284.824	1006	55	1	255	0	0	0	0	800
2015-01-01 12:00:00+01:00	Valencia	284.824	284.824	284.824	1006	55	1	255	0	0	0	0	800
2015-01-01 13:00:00+01:00	Valencia	284.824	284.824	284.824	1006	55	1	255	0	0	0	0	800
2015-01-01 14:00:00+01:00	Valencia	285.0507	285.0507	285.0507	1015	52	1	248	0	0	0	0	800
2015-01-01 15:00:00+01:00	Valencia	285.2773	285.2773	285.2773	1025	50	1	242	0	0	0	0	800
2015-01-01 16:00:00+01:00	Valencia	281.024	281.024	281.024	1021	67	1	230	0	0	0	0	800
2015-01-01 17:00:00+01:00	Valencia	282.744	282.744	282.744	1035	58	1	226	0	0	0	0	800
2015-01-01 18:00:00+01:00	Valencia	279.984	279.984	279.984	1035	68	1	216	0	0	0	0	800
2015-01-01 19:00:00+01:00	Valencia	277.8762	277.8762	277.8762	1017	69	1	235	0	0	0	0	800
2015-01-01 20:00:00+01:00	Valencia	275.0075	275.0075	275.0075	1021	81	1	220	0	0	0	0	800
2015-01-01 21:00:00+01:00	Valencia	272.791	272.791	272.791	1006	84	1	235	0	0	0	0	800
2015-01-01 22:00:00+01:00	Valencia	272.791	272.791	272.791	1006	84	1	235	0	0	0	0	800
2015-01-01 23:00:00+01:00	Valencia	271.229	271.229	271.229	1006	82	1	265	0	0	0	0	800
2015-01-02 00:00:00+01:00	Valencia	271.229	271.229	271.229	1006	82	1	265	0	0	0	0	800
2015-01-02 01:00:00+01:00	Valencia	271.229	271.229	271.229	1006	82	1	265	0	0	0	0	800
2015-01-02 02:00:00+01:00	Valencia	270.664	270.664	270.664	1005	82	1	275	0	0	0	0	800

Figure 4.3. Weather Data

In figure 4.3, the same time distribution is seen and the main weather variables that are included are

- Temperature
- Minimum temperature
- Maximum temperature
- Pressure
- Humidity
- Wind speed
- Wind direction
- Rain in the last hour
- Rain in the last three hours
- Snow in the last three hours
- Clouds
- Weather id

After gathering all the required data for the proposed model to operate with, the next stage of the operation was initiated. As indicated earlier, the focus of this model will be to accurately predict electricity consumption for Day-ahead markets, however with all the data available, there is a total of 1460 days each having 24 hours in them, leading to a total of 35,040 day-ahead markets that were previously entered. At this point of planning, justified filtering processes were applied to the data to reach the specific domain to which the model will be applied.

4.4 Justification

This section will serve the purpose of proofing the reliability of all the data gathered as well as the decisions made to reach the specific day-ahead market that the model will be tested as a justification for the models' results.

4.4.1 Competition in Market

First of all, figure 4.4 shows the different types of generation companies that exist in the Spanish electricity market. [68]

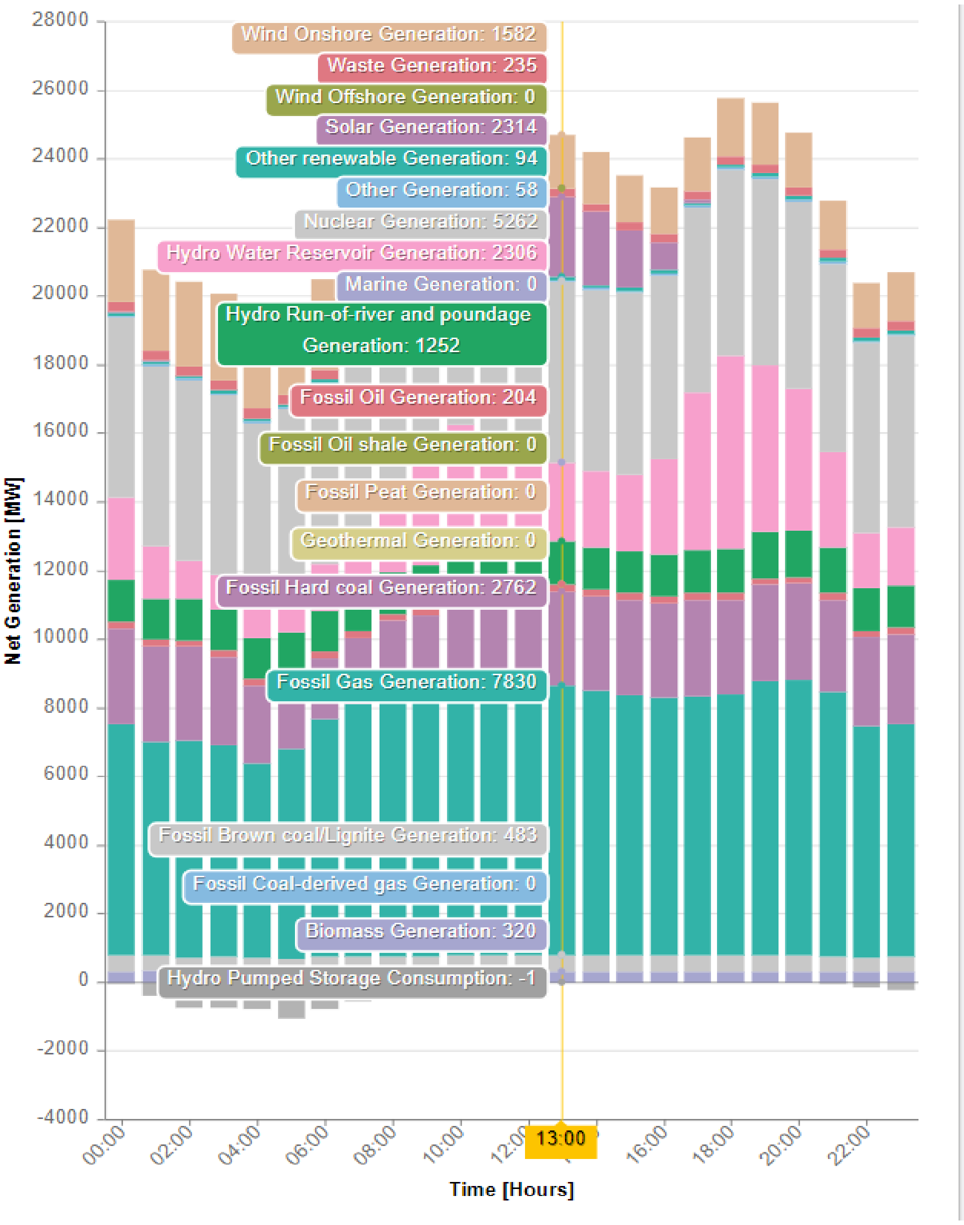


Figure 4.4. Different GENCOS in Spain

In figure 4.4, the different shades of colors present in the bar chart correspond to different generation companies' types as shown in the legend. The Spanish local time is plotted on the x-axis where the energy production is on the Y-axis. These analytics are published by ENTSO-E, showing the competitive nature of the electricity markets in Spain.

4.4.2 Time-domain

The model that will be proposed will include non-parametric approaches such as machine learning algorithms and these methods require as mentioned earlier a training stage and a testing stage.

After gathering the data from January 2015 till December 2018, the aim was to plug in all of this data in the training phase however, the search for more recent data covering the last years and also having the same format wasn't yielding. This led to the division of the obtained data into two parts, one for training and the other for testing.

It was proven by the literature and research, that for AI and Machine learning, the accuracy of the model heavily depends on the amount of data present for the training period [69]. Also, the amount of data is proportional to the purpose of the project, if the model is of high importance and large deviations between forecasts and actual data will cause severe consequences, it is crucial to allocate as much data as possible[70].

Electricity demand prediction is a vital procedure in a country's economy and deviations may cause a large number of materialistic losses as well as an unsustainable operation chain which contributes to global warming. Not only are the effects felt in terms of resources, but also, deviations may cause power cuts, and depending on the industry, human lives may be endangered (Hospitals). Consequently, for the available data, it was decided that the first three years' worth of input will be fed into the system during the training phase.

Studying the data for the fourth year (2018) and attempting to narrow down the scope of interest even more, trends of increase in electricity consumption and increase in forecasted errors during the summer period were noted. H. E. Thornton et al. in their research “The role of temperature in the variability and extremes of electricity and gas demand in Great Britain” concluded that the extreme elevation in temperatures in summer has led to unforeseen increases in electricity demand [71]. As a result, the time domain of interest has shifted specifically to the summer season in 2018 where possible frequent deviations between the actual and forecasted load are to be noted.

Studying the data available, several days in the summer period of 2018 proved to be good candidates for the model’s application. Their eligibility was depending on how bad the forecasts were on that specific day i.e. the bigger the deviations, the better. Some of the days that were chosen are the 16th of June, 21st of July, 1st of August, and 15th of August, their schematics are presented below from figure 4.5 to figure 4.8 respectively. [68]

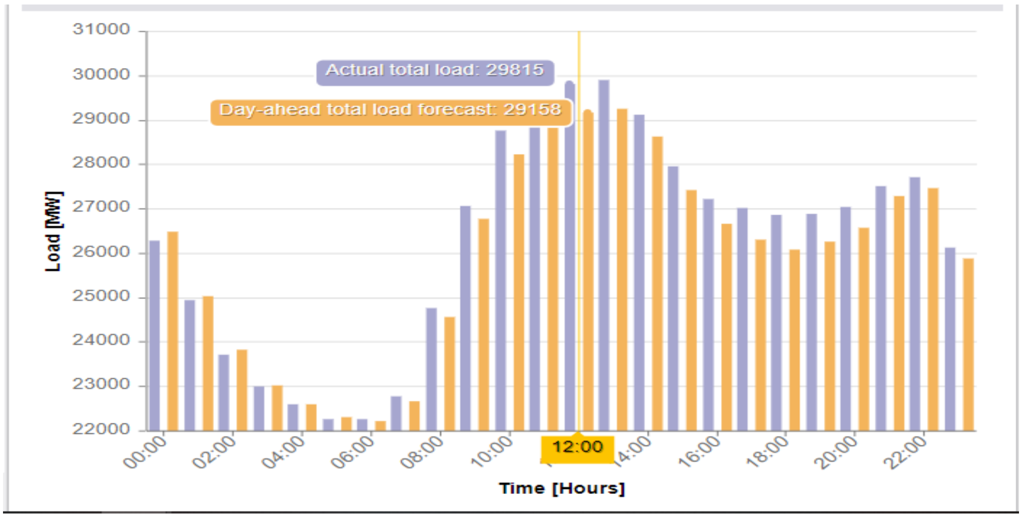


Figure 4.5. Actual vs. forecasted load for 16th of June 2018

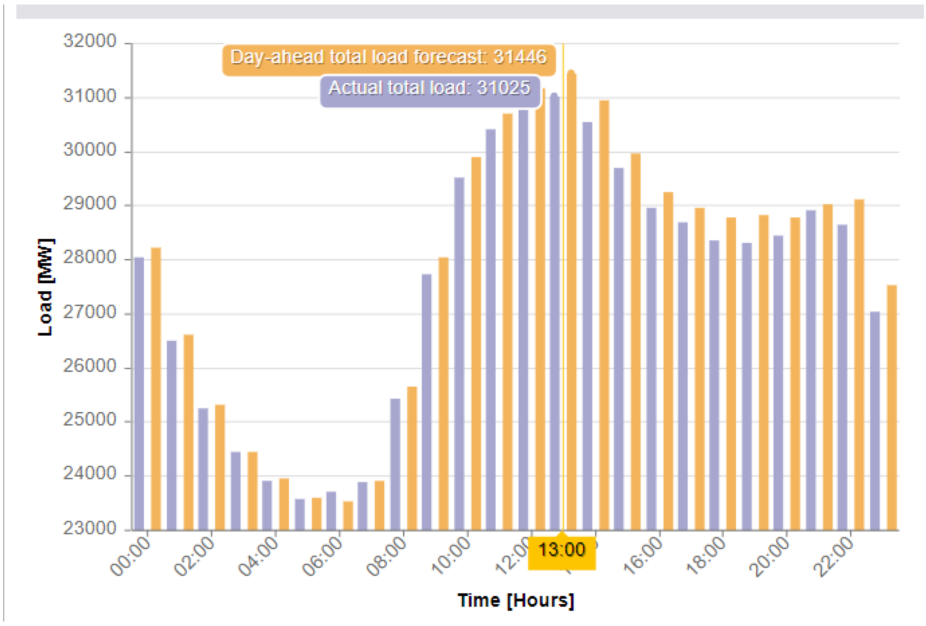


Figure 4.6. Actual vs. forecasted load for 21st of July 2018

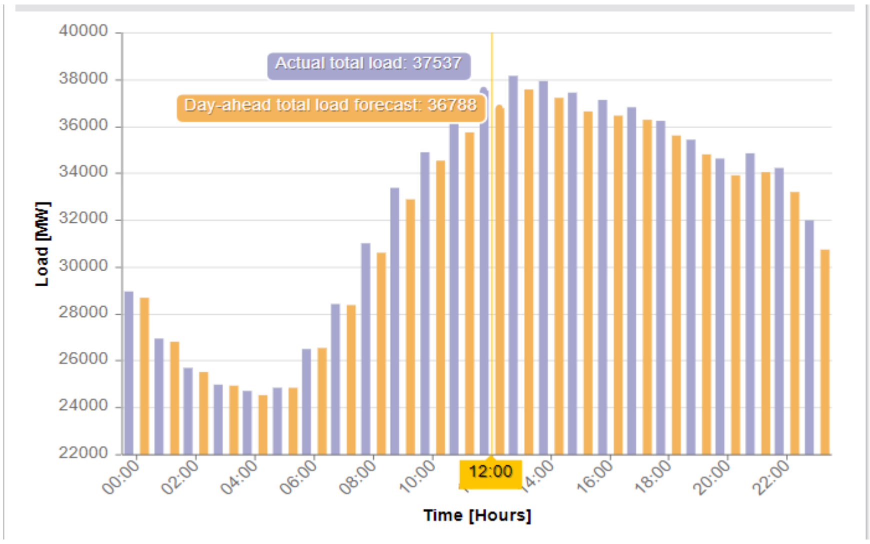


Figure 4.7. Actual vs. forecasted load for 1st of August 2018

Figures 4.5 till 4.7 all show deviations between the forecasted and actual demand, the number presented in each of the pictures were chosen according to the highest peak demand seen on that day, if a closer look is taken considering the peak demand values, it can be seen what H. E. Thornton et al.'s conclusion was right, as the temperature increases (as time flows into the summer season) the deviations increase

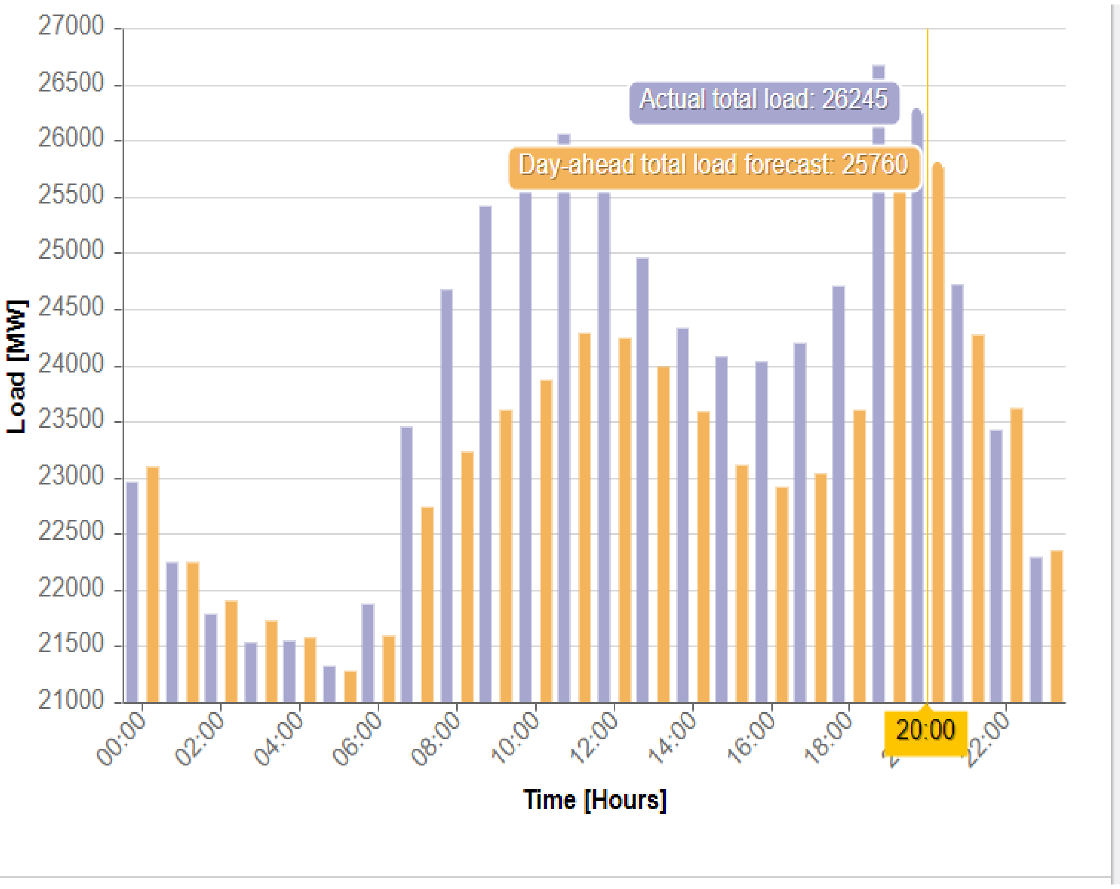


Figure 4.8. Actual vs. forecasted load for 15th of August 2018

In all of the figures presented above, the local time was plotted on the X-axis and the load in MW was on the Y-axis. The blue lines represent the actual demand whereas the orange corresponds to the forecasted values. Figure 4.8 shows extremely high deviations at almost all times during the 15th of August 2018. One of the highest deviations seen was at 8 pm local time as indicated in figure 4.8. Another graph representing the day-ahead prices for electricity sold on that day is shown in figure 4.9 below. Again, on the x-axis time is plotted, and on the other hand, the price per MWh in euros is plotted on the Y-axis. [68]

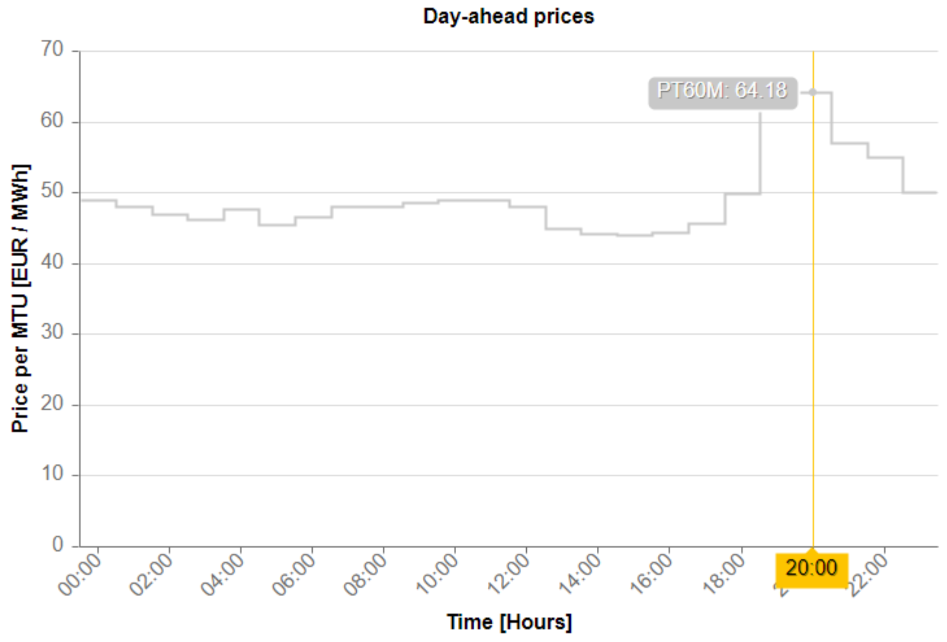
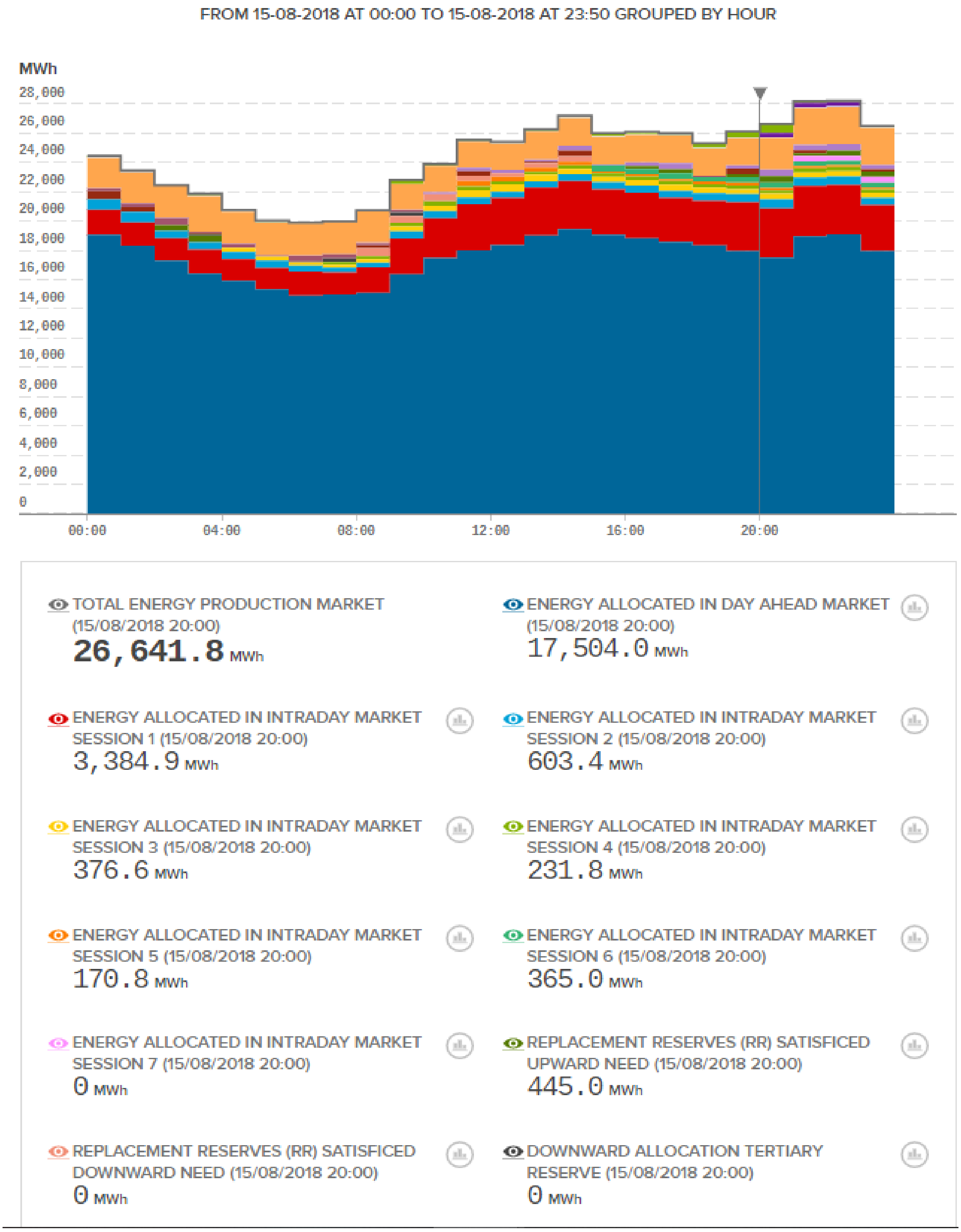


Figure 4.9. Day-ahead prices for 15th of August 2018

As expected since large deviations were seen between the actual and forecasted demand, the market price has also increased indicating that several intraday markets were entered.



TOTAL ENERGY PRODUCTION MARKET (15/08/2018 20:00)
26,641.8 MWh

ENERGY ALLOCATED IN DAY AHEAD MARKET (15/08/2018 20:00)
17,504.0 MWh

ENERGY ALLOCATED IN INTRADAY MARKET SESSION 1 (15/08/2018 20:00)
3,384.9 MWh

ENERGY ALLOCATED IN INTRADAY MARKET SESSION 2 (15/08/2018 20:00)
603.4 MWh

ENERGY ALLOCATED IN INTRADAY MARKET SESSION 3 (15/08/2018 20:00)
376.6 MWh

ENERGY ALLOCATED IN INTRADAY MARKET SESSION 4 (15/08/2018 20:00)
231.8 MWh

ENERGY ALLOCATED IN INTRADAY MARKET SESSION 5 (15/08/2018 20:00)
170.8 MWh

ENERGY ALLOCATED IN INTRADAY MARKET SESSION 6 (15/08/2018 20:00)
365.0 MWh

ENERGY ALLOCATED IN INTRADAY MARKET SESSION 7 (15/08/2018 20:00)
0 MWh

REPLACEMENT RESERVES (RR) SATISFICED UPWARD NEED (15/08/2018 20:00)
445.0 MWh

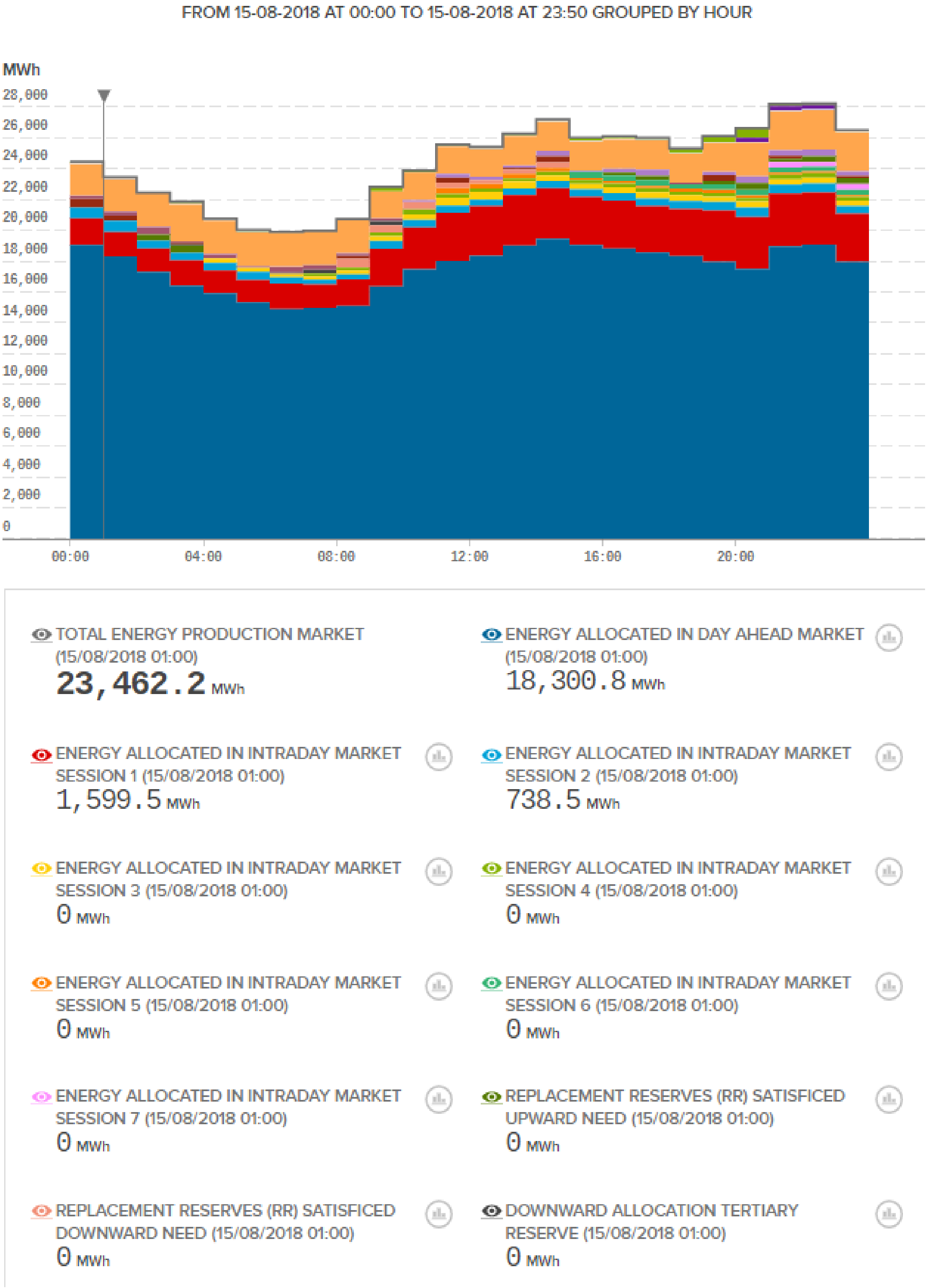
REPLACEMENT RESERVES (RR) SATISFICED DOWNWARD NEED (15/08/2018 20:00)
0 MWh

DOWNWARD ALLOCATION TERTIARY RESERVE (15/08/2018 20:00)
0 MWh

Figure 4.10. Energy Allocation for 15th of August at 8:00 pm

Figure 4.10 shows the electricity capacities there were allocated for balancing the demand at 8:00 pm on the 15th of August 2018 as shown by the cursor [68]. Different colors represent the amount of energy allocated in different markets. It can be seen that the large deviation shown in figure 4.8 led the market to enter 6 intraday markets as indicated in figure 4.10. As derived before in prior sections, as more intraday markets are entered, the electricity prices increase, and the need for using spinning reserves which are not sustainable and more expensive will be inevitable, which was the case in figure 4.10 where reserves were used to satisfy 445 MWh. The entrance in several intraday markets and the usage of reserves all led to the high electricity price for 8:00 pm as illustrated in figure 4.9.

The dependency of the electricity prices on the number of intraday markets entered and spinning reserves capacities utilized is huge, and these factors are inversely proportional to the accuracy of the forecasts. It can be seen that using the same forecasting techniques on the same time-domain (15th of August 2018) but with higher accuracy can lead to completely different results. Looking back at figure 4.8 and 4.9, at 1:00 am the deviations between the actual and predicted demand were minimal, and also the day-ahead price was considerably low in comparison to the day ahead price allocated for 8:00 pm. The energy allocations for 1:00 am are given below in figure 4.11 [68].



FROM 15-08-2018 AT 00:00 TO 15-08-2018 AT 23:50 GROUPED BY HOUR

TOTAL ENERGY PRODUCTION MARKET (15/08/2018 01:00)
23,462.2 MWh

ENERGY ALLOCATED IN DAY AHEAD MARKET (15/08/2018 01:00)
18,300.8 MWh

ENERGY ALLOCATED IN INTRADAY MARKET SESSION 1 (15/08/2018 01:00)
1,599.5 MWh

ENERGY ALLOCATED IN INTRADAY MARKET SESSION 2 (15/08/2018 01:00)
738.5 MWh

ENERGY ALLOCATED IN INTRADAY MARKET SESSION 3 (15/08/2018 01:00)
0 MWh

ENERGY ALLOCATED IN INTRADAY MARKET SESSION 4 (15/08/2018 01:00)
0 MWh

ENERGY ALLOCATED IN INTRADAY MARKET SESSION 5 (15/08/2018 01:00)
0 MWh

ENERGY ALLOCATED IN INTRADAY MARKET SESSION 6 (15/08/2018 01:00)
0 MWh

ENERGY ALLOCATED IN INTRADAY MARKET SESSION 7 (15/08/2018 01:00)
0 MWh

REPLACEMENT RESERVES (RR) SATISFICED UPWARD NEED (15/08/2018 01:00)
0 MWh

REPLACEMENT RESERVES (RR) SATISFICED DOWNWARD NEED (15/08/2018 01:00)
0 MWh

DOWNWARD ALLOCATION TERTIARY RESERVE (15/08/2018 01:00)
0 MWh

Figure 4.11. Figure 4.10. Energy Allocation for 15th of August at 1:00 am

Figure 4.11 shows that with better prediction, only two intraday markets were entered and no reserves were used. The difference between 8:00 pm energy allocations and 1:00 am energy allocations is tabulated below in Table 4.1.

Table 4.1. Prediction Accuracy Outcomes for 8:00 pm and 1:00 am

Criterion	**08:00**	**01:00**
Number of intraday Markets	6	2
Reserve capacity	445 MWh	0 MWH
Money Spent	More	Less
Time Spent	More	Less
Sustainability	Not sustainable	Sustainable
Overall Decision	**Worse**	**Better**

It can be concluded that with a better prediction, the overall performance of the market will be more sustainable in terms of energy usage, time consumption, and economy.

This thesis will use the weather variables and the data gathered to create a hybrid model that will focus on achieving a more accurate and superior prediction for STLF in Spain in the last quarter of 2018. The data will be divided into parts, testing and training data, the training data will include all the days until the 1st of August 2018 and will be tested on the day of interest till the end of the year.

In the upcoming chapters, the applied models with their theoretical backgrounds included in the proposed methodology will be explained and the results will be presented and commented on. Also, the proposed model’s performance will be compared to the applied model in real life

CHAPTER 5

APPLIED MODELS

In this chapter, the working principles, theory, and governing equations of the applied models will be discussed in a detailed manner. Four models are applied in total namely; ARIMA, LSTM, ε-SVR, and ANN. According to the way of implementation that will be discussed in chapter 6, ARIMA is considered to be a parametric model whereas, on the other hand, ε-SVR, LSTM, and ANN are both parametric and non-parametric.

5.1 Autoregressive Integrated Moving Average Model

Autoregressive Integrated Moving Average (ARIMA) models have been introduced by Box and Jenkins in 1976 and have been used for forecasting time series models since then [72]. Before going through the theory behind the model it is important to touch upon the family of models that ARIMA belongs to, which is AutoRegression type models (AR) models.

5.1.1 AR Models

AR Models are forecasting models that don’t include any external inputs (features) but rather focus only on the target value (value of interest). These models are time series models that study the trends and behaviors of the target value using its past values and utilize them to predict the future ones. Having this operational principle, the performance of the result is solely dependent on the quality and accuracy of the historical data available (Training Sample). Also, the end performance of these models is graded according to the predicted values (Test Sample) [73].

One of the early versions of AR models is known as Autoregressive Moving Average [ARMA]. The AR process is responsible for dealing with random data and predicting their future values based on historical data, whereas the MA process mitigates the effects of random, short-term variations over a specific time frame by constructing a series of averages of distinct subsets of the whole data set and calculating the moving average [72]. These models are denoted using two important parameters p and q and are represented using the notation ARMA(p,q) and described using the following equation.

$$\dot{y}_t - \phi_1 \dot{y}_{t-1} - \dots - \phi_p \dot{y}_{t-p} = \epsilon_t + \theta_1 \epsilon_{t-1} + \dots + \theta_q \epsilon_{t-q} \qquad \text{(Eq. 5.1)}$$

Presented in the equation above is a series of different parameters, $\dot{y}_t = y_t - \mu$ where μ is the mean of the whole process. $\dot{y}_t$ represents a stationary series of inputs, ϕ and θ are coefficients for the autoregressive and moving average parameters respectively, p and q on the other hand are for denoting the order of the autoregressive and moving average parameters. In any forecast as mentioned before, errors exist, these random errors are included in the ϵ_t .When considering these errors, a general assumption is made which is that these errors are considered to be distributed uniformly as a white noise process with a mean of 0 and a constant standard deviation of σ^2, they are also considered to be independent of each other. Another form of notation for these errors is WN(0,σ^2).

5.1.2 Seasonal AutorRegressive Integrated Moving Average

Having understood the general concept of the AR models, now a dive through the theory behind the ARIMA model would be more constructive. ARIMA models are basically ARMA models that are simpler in nature and include a notion of integration within [72]. ARIMA models can be categorized into two categories; seasonal and non-seasonal, these identifications depend on whether or not the model will be able to deal with seasonal variation in data. The general notation for the Autoregressive Integrated Moving Average model is expressed as ARIMA (p,d,q) where p and q are the same for AR models and d accounts for the number of times that differencing were applied to the data. The differencing term resulted from the concept that AR and MA processes work with stationary data("*Stationary data has the property that the mean, variance, and autocorrelation structure do not change over time*") [74]. And since in reality, some data may be non-stationary, differencing has been introduced. Following this logic, if data is presented for which the ARIMA model is to be applied on and was found to be stationary, the "d" term would be equal to 0. In the case of the seasonal ARIMA, it is denoted as SARIMA(p,d,q)(P, D, Q), where the capitalized parameters are equivalent to the lowercase but for the seasonal data, i.e. P and Q resembles the Autoregressive and Moving Average processes for the seasonal data and D is the differencing for both seasonal and non-seasonal data. Equations 5.2 till 5.7 will exhibit the mathematical interpretation of the model. [72]

$$\phi_p(B)\Phi_P(B^s)(1 - B)^d(1 - B^s)^D\dot{y}_t = \theta_q(B)\Theta_Q(B^s)\epsilon_t \qquad \text{(Eq. 5.2)}$$

Where:

$$\phi_p(B) = 1 - \phi_1 B - \phi_2 B^2 - \ldots - \phi_p B^p \qquad \text{(Eq. 5.3)}$$

$$\Phi_P(B) = 1 - \Phi_1 B - \Phi_2 B^2 - \ldots - \Phi_P B^P \qquad \text{(Eq. 5.4)}$$

$$\theta_q(B) = 1 - \theta_1 B - \theta_2 B^2 - \ldots - \theta_P B^q \qquad \text{(Eq. 5.5)}$$

$$\Theta_Q(B) = 1 - \Theta_1 B - \Theta_2 B^2 - \ldots - \Theta_Q B^Q \qquad \text{(Eq. 5.6)}$$

Φ and Θ are coefficients for the seasonal autoregressive and seasonal moving average parameters respectively, whereas B corresponds to what is known as the backshift operator which is responsible for representing the lagged values in the time series representation of the variable of interest. In other words ;

$$B * y_t = y_{t-1} \qquad \text{(Eq. 5.7)}$$

In all the previous equations, any terms that include lowercase parameters are for the non-seasonal data whereas the opposite is true. For a successful implementation of the ARIMA model on data, the data needs to be processed in a specific way to yield accurate and reliable results. This mentioned preprocessing of the data stage will be explained thoroughly in the next chapter.

5.2 Recurrent Neural Networks

The human brain is considered to be a wonder and their ability to recall information instantly when needed has always been the main feature that the experts are trying to memic in their machine learning or AI algorithms. When humans are required to retain a piece of specific information from their brain, they don’t start from scratch i.e. they don’t have to start thinking from the day they were born till the time the information needed was saved into their brains.

A conventional NN does not have a memory mechanism and thus they consider each input on its own irrespective of what came before. Therefore in an attempt to maintain an internal state which serves as a memory providing context to the current input the creation of what is known as Recurrent Neural Networks (RNNs). [75]

RNNs were created to tackle such a problem where a loop mechanism was added to NNs and a saving feature was obtained. A simple schematic of RNN is shown below.

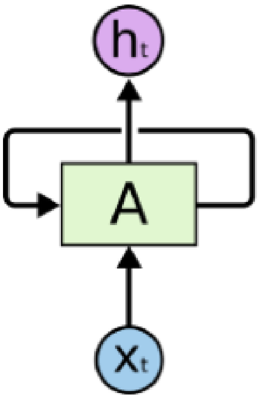


Figure 5.1. Recurrent Neural Network

Figure 5.1 shows one chunk of RNN denoted by A, this block receives an input data of X_t and outputs a value of h_t. The loop serves the purpose of passing information from one part of the network to the other. A complete RNN network (chunk)can be considered as a chain of identical blocks as the one illustrated in the prior figure, each conveying a message to its successor. Figure 5.2 will expand the chunk shown previously and will show the n components that lie within.

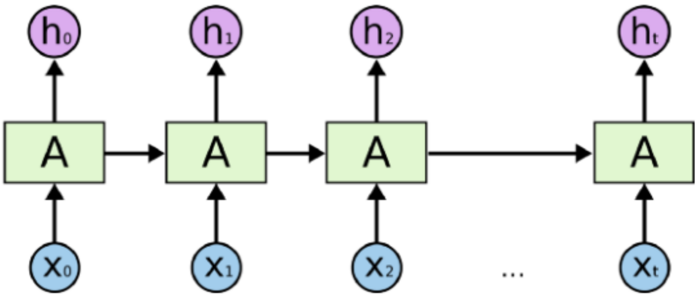


Figure 5.2.An unrolled recurrent neural network.

As it can be interpreted from the figure the RNN consists of a long chain of identical recurrent neural networks that are ordered sequentially where information is being sent from one point to the other after the processing stage, hence the name of Recurrent Neural Networks.

This architecture of Neural Networks has been widely used in the past decades and was seen in a variety of industries such as speech recognition, language modeling, translation, and many more [76]. The Loop feature in the RNNs introduced the idea that such networks could be able to save the information and connect them however unlike the human brain, problems of Long-term Dependencies had risen.

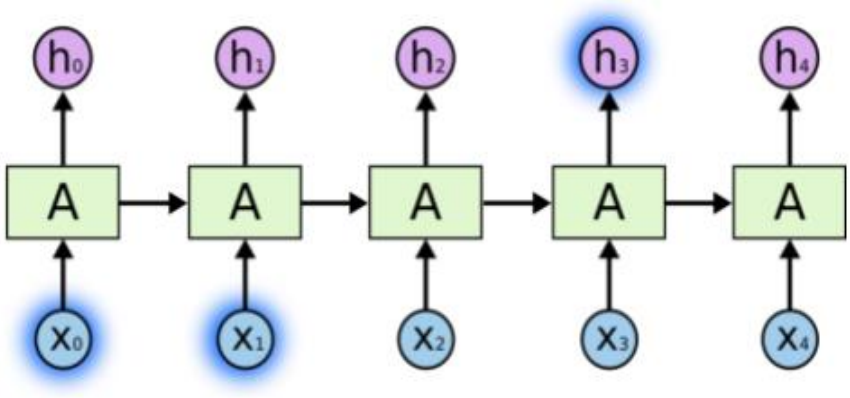


Figure 5.3. Prediction of h_3

The ability to predict information from previously saved data is possible and proved to contain a high level of accuracy, however, this isn't always the case. It was found that the accuracy of the prediction depends highly on where and when the relevant information needed for prediction was stored. In figure 5.3 for example, h_3 was the value of interest and the information relative for its prediction were $X_0$and X_1 and in that case, the model was able to retrieve the information successfully since the gap between them is small. Yet, when a new prediction at a later time required the same relative information the results were completely different.

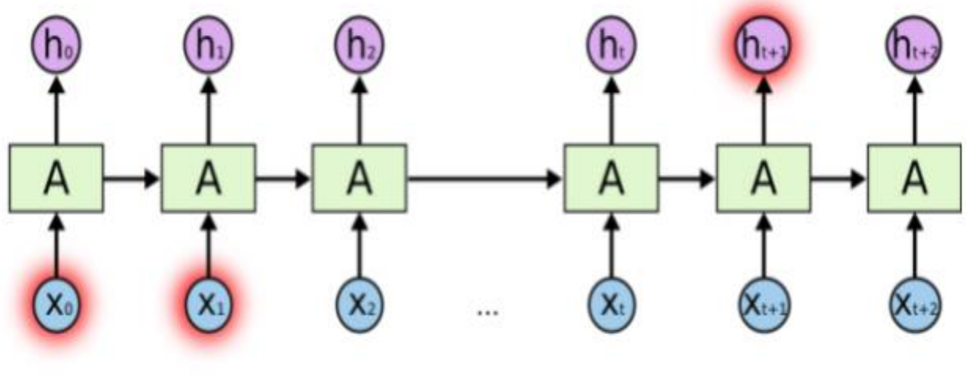


Figure 5.4. Prediction of h_{t+1}

When the gap between the target value (prediction) and its features (relative inputs) increases, the ability of RNN to connect them decreases in other words as indicated in the figure above for h_{t+1}, when a long-term memory is needed for prediction RNN fails to give out results. This phenomenon was studied by Hochreiter in 1991 and it was found that one of the main reasons to why such a problem exists is due to a gradient decay in terms of transfer of information. [77]

Trying to come up with solutions to such problems, Hochrieter and Schmidhuber reached a breakthrough and introduced a new form of RNNs known as Long short term memory in 1997 [78] and this RNN had no problem remembering information for a long period of time.

5.2.1 Long Short Term Memory (LSTM)

This section's information was gathered from [79]and [80]

Normal conventional RNNs consist of a series of modules as mentioned earlier with one neural network layer that applies the tanh function as shown in figure 5.5, LSTM being an upgrade consists of 4 layers of neural networks within one of its modules as figure 5.6 indicates.

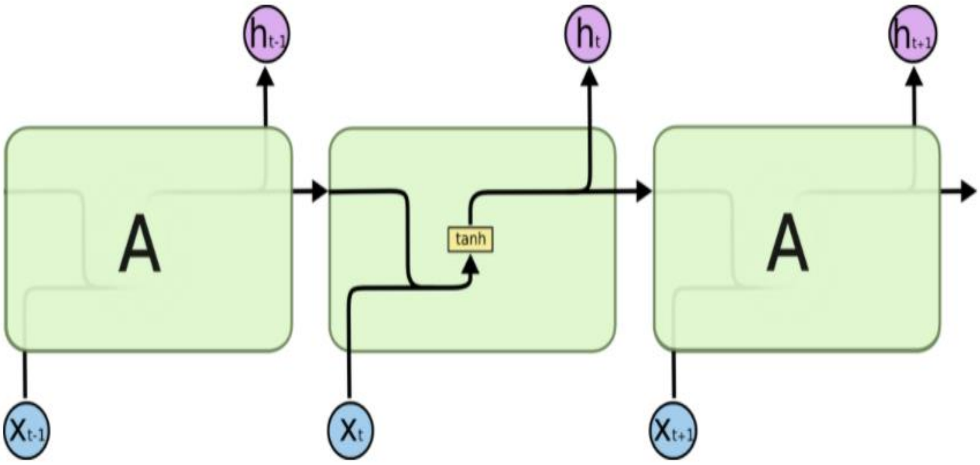


Figure 5.5. Repeating Module in a standard RNN

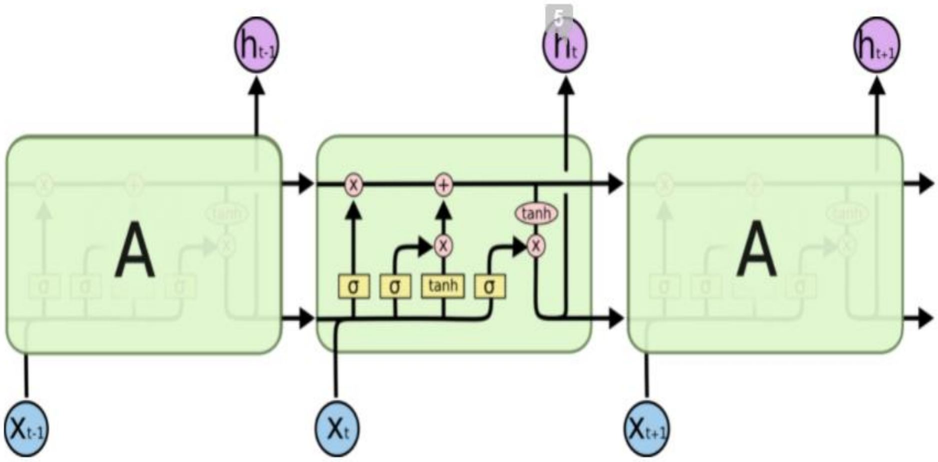


Figure 5.6. Repeating module in LSTM with 4 layers

There are a series of colored blocks and lines that are flowing from one point to the others their interpretations can be explained simply in figure 5.7 below.

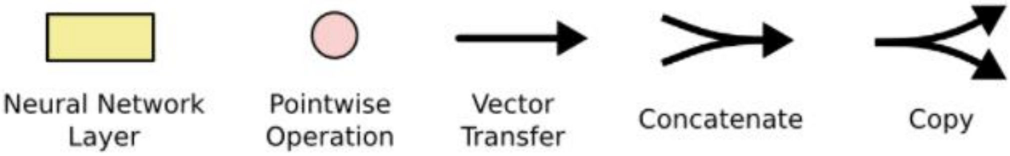


Figure 5.7. Components' meanings

Where the orange boxes presented in figure 5.6 are neural network layers that are responsible for applying a specific function to the data, pink circles represent the mathematical operations that the data will undergo, black lines with arrowheads represent vectors moving towards a specific direction, two arrowheads meeting means they concatenate while two vectors parting ways means that they were copied and proceeded towards two different directions.

Looking at figure 5.6, the horizontal line at the top is considered the most important key in an LSTM module, it is known as a cell state and it can be seen as a conveyer belt that runs above the whole operation chain in the module. In this line, minor linear interactions are applied such as multiplication and addition, but mainly it serves as a non-congested path from the beginning to the ending of the module where information can flow easily without facing any changes. In this cell state, the previous output vector is passing and denoted by C_{t-1}.

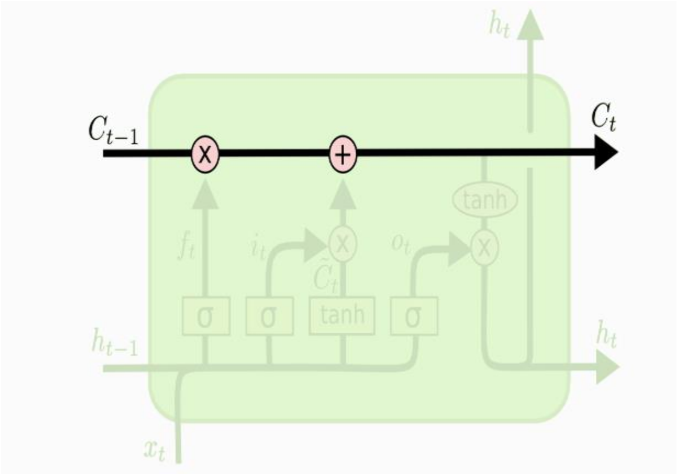


Figure 5.8. LSTM Cell State

The initial step in any LSTM module is to decide what information is to be kept and what is to be forgotten from the resulted output of the previous neighbor module. This decision is taken in what is known as the forget gate Layer. It is important to note that for every module an output vector is produced and is copied and sent to two directions, first is as an input to the other module (towards the forget layer) and will be denoted as h_{t-1} and the other direction is towards the cell state and in that case, it will be denoted as C_{t-1}. This procedure can be illustrated in figure 5.9 below.

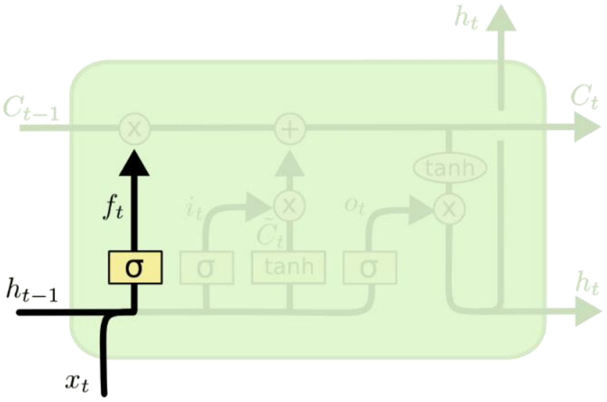


Figure 5.9. Forget Gate Layer

This forget layer consists of a sigmoid function, this layer is fed with both the input vector X_t and the previous output vector from the module h_{t-1} and maps it to the range between 0 to 1 where 0 is to be forgotten and 1 is to be of the highest importance and the rest are scaled accordingly. The resulting vector is denoted by f_t and can be shown algebraically in equation 5.8

$$f_t = \sigma\left(W_f . |h_{t-1}, X_t| + b_f\right) \qquad \text{(Eq. 5.8)}$$

Where W_f is the weight of forget gate neurons, b_f are biases of the forget gate and σ is the sigmoid function.

After filtering the unneeded information from h_{t-1}, LSTM turns its focus to what is the new information that needs to be obtained from the input vector X_t and stored in the cell state. This operation is carried out in the input gate layer.

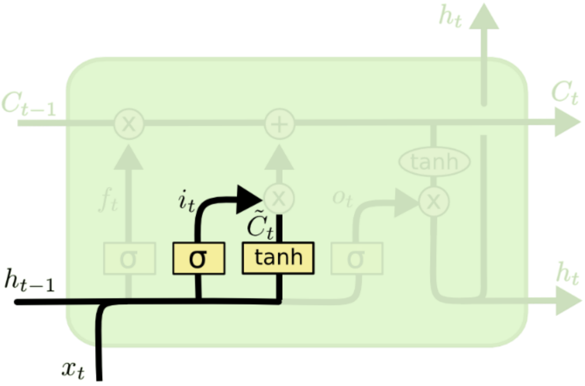


Figure 5.10. Input Gate Layer

As indicated in figure 5.10, this layer consists of two neural network layers, a sigmoid and a tanh. A copy of the input vector and the output of the previous module is again recopied to two directions one entering the sigmoid function again to assign weights for the importance of the data and the result is a vector i_t whose values are between the range of 0 and 1, and towards the tanh function the creates a new candidate input $\widetilde{C_t}$ whose vector values are between 1 and -1. The neural networks layers' equations are shown below:

$$i_t = \sigma(W_i . |h_{t-1}, X_t| + b_i) \qquad \text{(Eq. 5.9)}$$

$$\widetilde{C_t} = \tanh(W_C . |h_{t-1}, X_t| + b_C) \qquad \text{(Eq. 5.10)}$$

After these output vectors C_t , i_t , f_t are obtained, the LSTM will combine them, update the cell state $C_{t-1}$and apply the needed adjustments. Figure 5.11 shows that first the cell state is multiplied with the forget gate's vector output and then multiply the outputs vectors from the input gates together, and the product resulting from this multiplication will be added to the cell state.

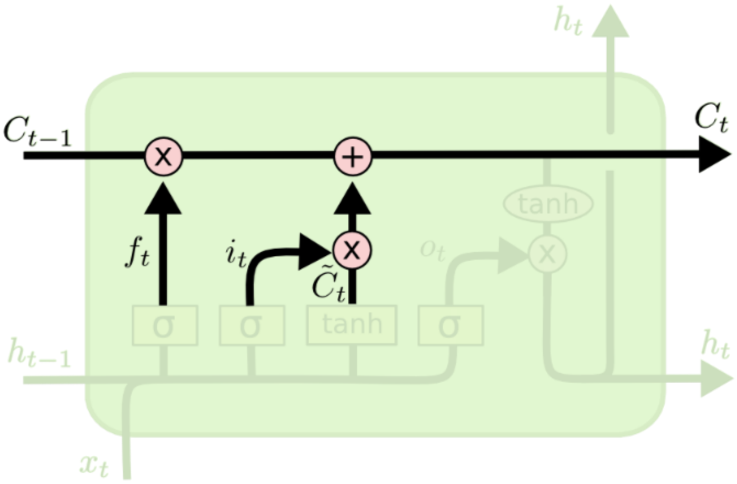


Figure 5.11. Linear operation application

$$C_t = f_t * C_{t-1} + i_t * \widetilde{C}_t \qquad \text{(Eq. 5.11)}$$

By successfully multiplying the forget gate with the cell state, the resulting vector will include the needed information from the previous output and the unnecessary information will be dropped. Later on, when the product vector from the input gate will be added to the cell state, the newly gathered information from the input will be added to the cell state.

After the new cell state is updated, the decision of what to output comes next, at this point the cell state is copied towards two directions one which is along with the cell state towards the second module, and it will start to be considered as C_{t-1} and the other will pass towards the output gate which includes a tanh function where the vector will be multiplied by the output of a sigmoid function that will result in a filtered form of the updated cell state and will be referred to as h_t in this current module and h_{t-1} in the next. This can be seen below in figure 5.12.

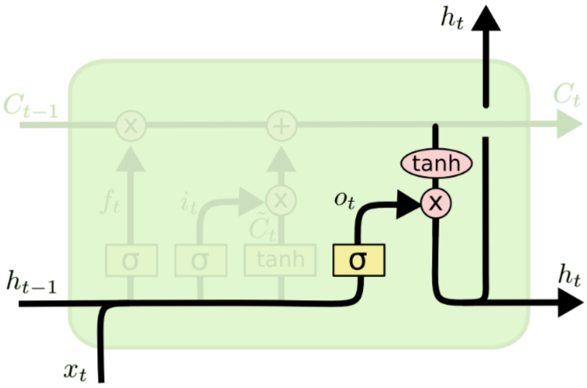


Figure 5.12. LSTM Output Gate

By passing it through the tanh function, it will be remapped to the range of 1 and -1 and the resulting vector will be multiplied by the output of the sigmoid function to ensure that the output includes only the information that was deemed beneficial. The governing equations for the output gate are given below.

$$o_t = \sigma(W_o.|h_{t-1}, X_t| + b_o) \quad \text{(Eq. 5.12)}$$

$$h_t = o_t * \tanh(C_t) \quad \text{(Eq. 5.13)}$$

This chain of operation was in one module, the LSTM keeps on repeating this for every piece of new information until the prediction needed is reached. By understanding this model, it can be said that LSTM is one of the closest NNs that can mimic what a human brain does, LSTM basically remembers every important piece of information fed to it and forgets what is not useful leading to a continuous freeing of space for new data to be stored. Moving to the upcoming models that will be applied, Support Vector Regression and ANN will be discussed next.

5.3 *ε*-Support Vector Regression

SVR models are arguably one of the most powerful commonly applied models for predictions. SVR is from the Support Vector Machine (SVM) family, SVMs are used for classification problems which means that they deal with categories and classes consisting of discrete values whereas, on the other hand, SVR deals with numerical continuous data. Both of these models have proved their accuracy and precision since they were first introduced by Vapnik [81].

SVM constructs their predictions by creating hyperplanes to classify data using the available inputs. When it comes to the methodology of solving a problem, SVM operates completely differently than ANNs, SVMs use linear constraints to solve a quadratic programming problem, unlike ANN which works by minimizing the loss without any constraints. SVMs only being able to deal with classification problems, led to the novelty of *ε*-SVR which deals with non-linear regression problems. The *ε*-SVR has one main principle; that is to obtain an *ε*- insensitive loss function. The goal of the algorithm is to reach a regressive function that has a distance less than a predefined ε value from actual values, which results in a tube with an ε radius around the regression line formed. A graphical representation of an ε-SVR can be shown below in figure 5.13 [82]

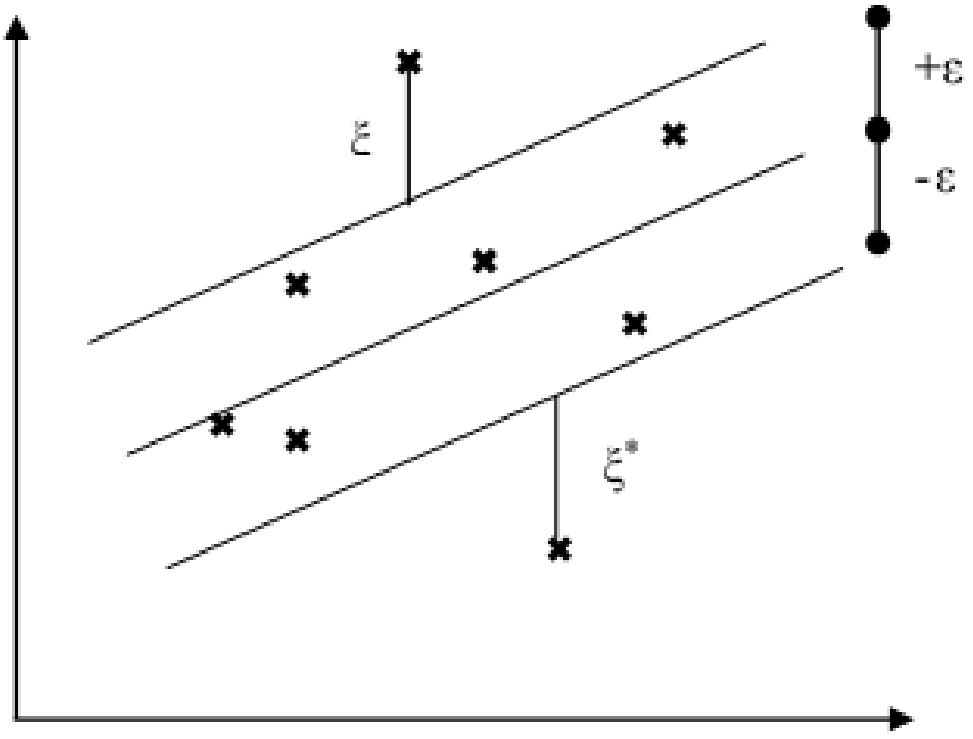


Figure 5.13. ε-SVR example

Generally, for any given problem, there are a set of independent variables (inputs) $x_1, x_2, x_3, \dots\dots, x_n$ and their corresponding dependant variables of interest $y_1, y_2, y_3, \dots\dots, y_n$. Their correlation varies from one situation to the other, The non-linear solving ability of ε-SVR lies in its ability to map each input into a higher dimension F where a linear relation with the target value exists [57], this is achieved by utilizing non-linear mapping functions as shown in Eq. 5.14 below. In this higher dimension, since a linear relationship is reached, a linear regression function is then fitted to the inputs as seen in Eq. 5.15.

$$\phi(\boldsymbol{x_i}) : \mathbb{R}^n \rightarrow \mathbb{R}^F$$

$$\boldsymbol{x_i} \in \mathbb{R}^n \rightarrow \phi(\boldsymbol{x_i}) = [\,\phi(x_{i1})\phi(x_{i2}) \ldots\ldots\ldots.. \phi(x_{in})]^T \in \mathbb{R}^F \qquad \text{(Eq. 5.14)}$$

$$f(\boldsymbol{x_i}) = \boldsymbol{w}^{\boldsymbol{T}} . \phi(\boldsymbol{x_i}) + b \qquad \text{(Eq. 5.15)}$$

In the equations above, $\phi(\boldsymbol{x_i})$ represents the mapping function employed on the inputs, $\mathbb{R}^n$ represents the original dimension where the input exists whereas $\mathbb{R}^F$ is the new dimension to which the input has been transformed. $\boldsymbol{w}$ is the weight vector and b is a predefined threshold value. The purpose of the model is to try to minimize $\boldsymbol{w}$ while the output of the function $f(\boldsymbol{x_i})$ has a maximum error of ε. Having such characteristics, the problem then transforms into the following convex optimization problem [57].

$$\min_{\boldsymbol{w}} \quad \frac{1}{2} \|\boldsymbol{w}\|^2$$

$$\begin{aligned} \text{s.to} \qquad & y_i - f(\boldsymbol{x_i}) \le \varepsilon, \quad \forall_i = 1, \ldots., n \\ & f(\boldsymbol{x_i}) - y_i \ge \varepsilon, \; \forall_i = 1, \ldots., n \qquad \text{(Eq. 5.16)} \end{aligned}$$

The above equation is deemed valid, only in the case that the deviation resulting from the function f is less than ε. However, some infeasible constraints may alter the results of the above equation, to account for such mischief, a concept known as the soft margin idea, was introduced and added to Equation 5.16 by Vapnik [81]. Slack Variables ξ and ξ^* were assigned for points that are outside the ε tube as indicated in figure 5.13. As a result, a new form of equation 5.16 was created that controls the tradeoff between the minimization of $\boldsymbol{w}$ and acceptable tolerance for errors. Eq. 5.17

is the new version. “C” is an introduced constant known as the cost of error coefficient.

$$\min_{\boldsymbol{w}} \quad \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{n}(\,\xi_i + \xi_i^*)$$

$$y_i - f(\boldsymbol{x_i}) \leq \varepsilon, \quad \forall_i = 1, \dots., n$$

$$f(\boldsymbol{x_i}) - y_i \geq \varepsilon, \; \forall_i = 1, \dots., n$$

$$\xi_i, \xi_i^* \geq 0, \; \forall_i = 1, \dots., n \qquad \text{(Eq. 5.17)}$$

To solve this complex optimization problem in an easier approach, the following lagrangian function is applied.

$$L:\; = \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{n}(\,\xi_i + \xi_i^*) - \sum_{i=1}^{n}(\eta_i\xi_i + \eta_i^*\xi_i^*)$$

$$= \sum_{i=1}^{n}\alpha_i(\varepsilon + \xi_i + f(\boldsymbol{x_i}) - y_i)$$

$$= \sum_{i=1}^{n}\alpha_i^*(\varepsilon + \xi_i^* + y_i - f(\boldsymbol{x_i})) \qquad \text{(Eq. 5.18)}$$

Where η_i, η_i^*, α_i, α_i^* are Langrangian multipliers and they abide by the following constraint.

$$\eta_i, \eta_i^*, \alpha_i, \alpha_i^* \geq 0 \qquad \text{(Eq. 5.19)}$$

Saddle point optimality conditions are illustrated below starting from Eq. 5.20 till Eq.5.23. These are basically derivatives of the Langrangian function in Eq.5.18 with respect to the main variables in the primal function $(\boldsymbol{w}, b, \xi_i, \xi_i^*)$ [57]

$$\frac{\partial L}{b} = \sum_{i=1}^{n}(\alpha_i^* - \alpha_i) = 0 \qquad \text{(Eq. 5.20)}$$

$$\frac{\partial L}{\boldsymbol{w}} = \boldsymbol{w} - \sum_{i=1}^{n}(\alpha_i - \alpha_i^*)\boldsymbol{x_i} = 0 \qquad \text{(Eq. 5.21)}$$

$$\frac{\partial L}{\xi_i} = c - \alpha_i - \eta_i = 0 \qquad \text{(Eq. 5.22)}$$

$$\frac{\partial L}{\xi_i^*} = c - \alpha_i^* - \eta_i^* = 0 \qquad \text{(Eq. 5.23)}$$

The dual problem is constructed using the four equations above which then replaces the $\phi(\boldsymbol{x_i}, \boldsymbol{x_j})$ with $K(\boldsymbol{x_i}, \boldsymbol{x_j})$, where “K” is known as a kernel function which will be explained later on in this section. The dual problem is represented below[57]:

$$\max \frac{-1}{2} \sum_{i,j=1}^{n}(\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*)K(\boldsymbol{x_i}, \boldsymbol{x_j}) - \varepsilon \sum_{i=1}^{n}(\alpha_i - \alpha_i^*) + \sum_{i=1}^{n}(\alpha_i + \alpha_i^*)$$

$$\text{s.to} \quad \sum_{i=1}^{n}(\alpha_i - \alpha_i^*) = 0$$

$$\alpha_i, \alpha_i^* \geq 0,\ \forall_i = 1, \ldots., n$$

$$\alpha_i, \alpha_i^* \leq C\,,\ \forall_i = 1, \ldots., n \qquad \text{(Eq. 5.24)}$$

By following the constraint that for a Lagrangian multiplier, α_i^* , it has to satisfy the relation $\alpha_i^* \alpha_i = 0$, in addition to that, both of these variables can't be assigned values at the same time, the weight $\boldsymbol{w}$ can be expressed as :

$$\boldsymbol{w} = \sum_{j=1}^{n} \left(\alpha_j - \alpha_j^*\right) \phi\left(\boldsymbol{x_j}\right) \qquad \text{(Eq. 5.25)}$$

This then leads to the last form of the regression function shown below [57]:

$$f(\boldsymbol{x_i}) = \boldsymbol{w^T}.\phi(\boldsymbol{x_i}) + b = \sum_{j=1}^{n} \left(\alpha_j - \alpha_j^*\right) \phi\left(\boldsymbol{x_j}\right) \phi(\boldsymbol{x_i}) + b$$
$$= \sum_{j=1}^{n} \left(\alpha_j - \alpha_j^*\right) K\left(\boldsymbol{x_i}, \boldsymbol{x_j}\right) + b$$

(Eq. 5.26)

As for the Kernel function; a kernel is a dot product in the feature space F. The kernel functions serve one purpose that is to simplify the calculating operations of the mapping functions. Several well-known kernel functions are commonly used such as Radial Basis Function (RBF), Linear function, Homegenuous Polynomial function, and Sigmoid function.

For achieving the best results, the cost of error coefficient C, the width of the ε tube, and the type of kernel function that will be applied in the model are the most vital hyperparameters that determine the performance of the ε-SVR model.

Discussed next, is the last model (ANN) that will be explored in an effort of creating an efficient hybrid model.

5.4 Artificial Neural Network

Most of this subsection's info is retrieved from [85]

Artificial Neural Networks (ANNs) are one of the most famous types of NNs, they are inspired by the structure of a human's central nervous system. ANNs are powerful mathematical and computational modules that have the ability to capture non-linear or implicit relationships between a variety of inputs and target values. ANNs work by performing non-linear mapping of inputs and starts learning their behaviors during a defined training process.

Like many other statistical-based learning models, ANNs use training and testing sets. The common order of operations is as follows; first, the input parameters are fed into the model and fitted using the training data set, afterward, the resultant fitted model is then applied to the testing data followed by obtaining the relevant results, finally, accuracy checks are carried out to evaluate the model's performance.

In the training phase, the general functions and hyperparameters of the ANN are adjusted according to the results of the testing data set to increase the model's performance, however, continuous altering of variables to ensure maximum accuracy may lead to two main obstacles, one which is that; the hyperparameters chosen are only fitting for this specific testing data and if a new testing data is checked, the model's accuracy would drop, also, overfitting is a common consequence of too much manipulation of hyperparameters, this is why for any statistical learning model, a validation set is utilized. A validation set is a set of data points that the model is fitted to and then the fitted model is applied to the testing dataset to overcome the previously mentioned problems.

ANNs consist of interconnected artificial neurons (nodes) as a base layer (input layer), these nodes get inputs as x_i's and processes them by applying random weight coefficients w_i's and a constant bias θ, the assigned weight coefficients together with the bias term enters an activation (transfer) function f which results in an output y. The above explanation can be seen clearly in figure 5.14 below. There are multiple well-known activation functions such as Rectifier (relu), sigmoid, softplus,softsign, tanh, and many more.

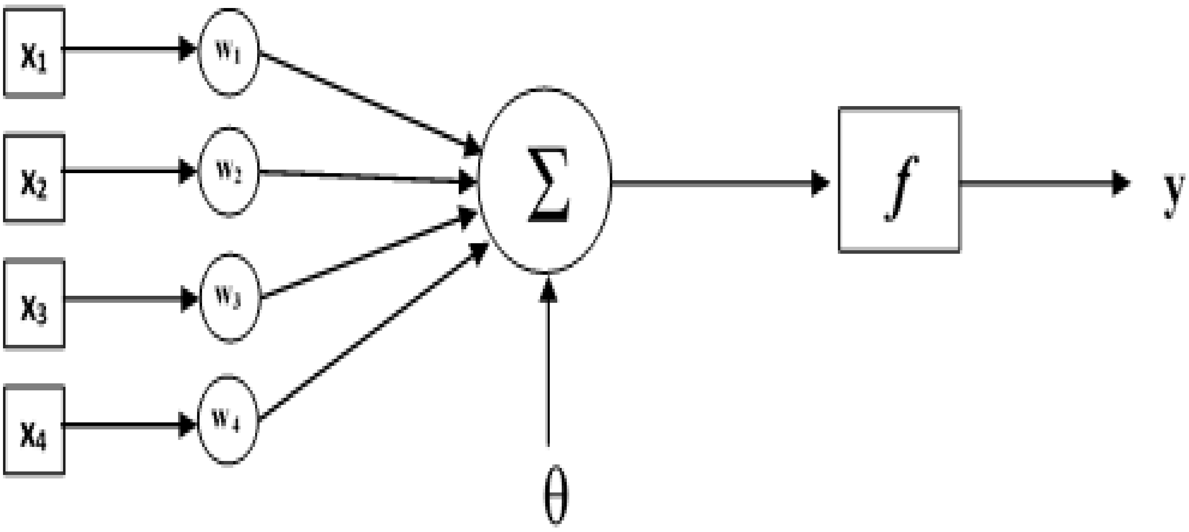


Figure 5.14. Simple ANN

Since the ANN was initially introduced in 1943 [83], there were several versions of ANN that differed in terms of the number of neurons, activation functions, learning rates, and the general architecture of the model. One of the most used types of ANN which will also be applied in this thesis is the Multilayer Perception Network (MLP) [84]. MLP consists of one input layer, one output layer, and n hidden layers. Figure 5.15 shows an MLP with one hidden layer.

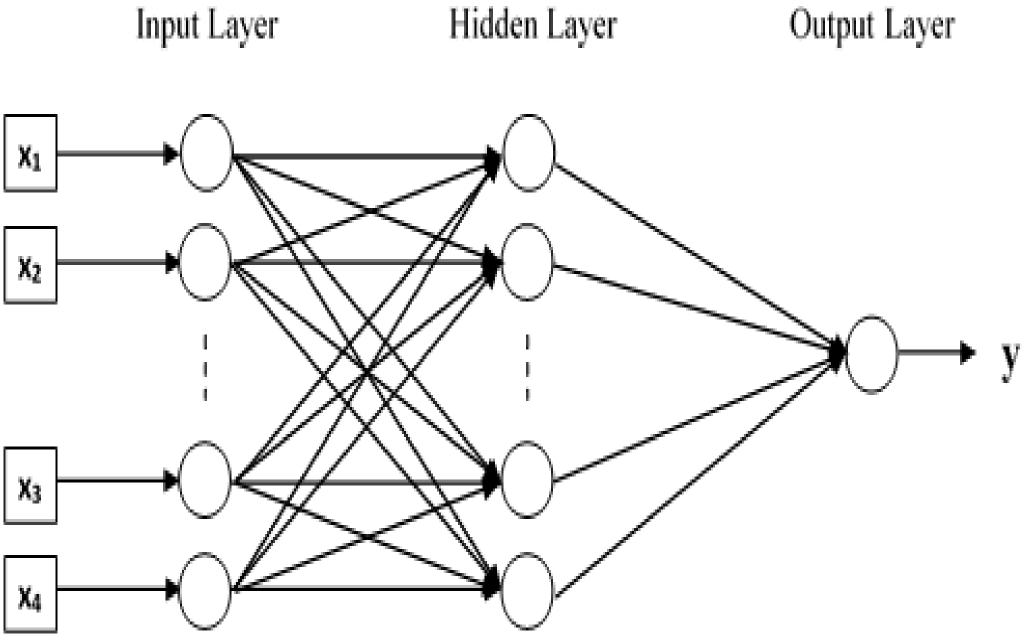


Figure 5.15. An MLP (feed-forward) network with one hidden layer

The hidden layer can consist of any number of neurons depending on the choice of the designer, A module of a NN can be denoted as a feed-forward network when information is fed in only one direction without undergoing any cycles. In the case of the ANN that is illustrated in the figure above and the one used by the thesis, the ANN will be considered as a feed-forward network that has information flowing from the input layer to the hidden layer and consequently from the hidden layer to the output layer.

5.4.1 Processing of Data in ANNs

Each neuron in any of the layers is considered as a summation point for the inputs of that neuron. In a general sense, the inputs are fed into the input layer's nodes and then they are transferred to the first hidden layer, in which the inputs that reached a node in the hidden layer (hidden node) are multiplied by randomized weight coefficients and then added up together, moreover, the bias term is also added and the resultant value is then transformed via an activation function, the final output of this operation is then passed on to the next hidden layer and the procedure continues until it reaches the output layer. For one node, the mathematical procedure of n number of inputs is formulated below:

$$out_j = f\left(\sum_{i=1}^{n} x_i \times \boldsymbol{w}_{ij} + \theta_j\right) \qquad \text{(Eq. 5.27)}$$

This equation's interpretation may vary depending on what it is describing, but the subscripts used are always referring to the following analogy; subscript j refers to the neurons in the current layer of interest whereas i indicates the neurons of the proceeding layer.out_j is the product from the current neuron, x_i is the input being fed to the neuron, if the neuron is in the input layer then x_i represents the model's inputs, while in the case that a hidden layer is of interest then x_i is the output of the previous node. θ_j is the bias term that is added in the current node, and f is the transfer function applied. This procedure is carried out in all nodes in all of the layers in the ANN.

To optimize the model's performance, the designer has to find the optimum; number of neurons in each layer, number of hidden layers, and type of activation function.

5.4.2 ANN's learning methods

In order to obtain the desired output, after deciding the number of neurons in each layer, the number of hidden layers, and the activation function, the randomized weights for the ANN and bias terns are constantly being updated via learning. The learning stage is carried out in the form of epochs (iterations). One epoch is done when all the training datasets (input and output pairs) are fed into the model. In the learning stage, the weights and bias terms are updated after every epoch.

The learning stage can be categorized into two categories; supervised and unsupervised, supervised is when a set of inputs and desired outputs are fed into the network so that the weights are adjusted to minimize the cost function (error). On the other hand, unsupervised learning is carried out by only feeding inputs to the network and the outputs are unknown, in that case, weights are being adjusted to cluster the inputs. Unsupervised learning is widely used for classification problems. In this thesis, **supervised** learning will be used.

Various supervised learning algorithms can be applied to train the ANN, one of the most prominent algorithms is *backpropagation (backward propagation of errors)*. This concept behind backpropagation is to decrease the difference between the actual data and the output of the network by using a *gradient descent algorithm* which is formulated below.

$$\boldsymbol{w}_{\boldsymbol{n+1}} = \boldsymbol{w}_{\boldsymbol{n}} - \lambda_n \nabla f(\boldsymbol{w}_{\boldsymbol{n}}) \qquad \text{(Eq. 5.28)}$$

In Eq. 5.28, λ_n represents the learning rate, $\boldsymbol{w}_{\boldsymbol{n}}$ is the weight at the nth iteration or epoch and f is the loss function. When the network starts operating, the inputs are assigned an initial weight of $\boldsymbol{w}_{\boldsymbol{0}}$ which will result in a correspondent gradient of the loss function. As iterations continue the weights are adjusted in the direction where f is decreasing.

The loss function used in backpropagation is the sum of squared errors (SSE) and it is computed using the following equation.

$$f(\boldsymbol{w}) = \frac{1}{2}\sum_{t \in T} \; y_t - \hat{y}_t \qquad \text{(Eq. 5.29)}$$

Where T indicated the training set, $\boldsymbol{w}$ is one complete set of weights, y_t is the actual data and $\hat{y}_t$ is the predicted data.

After obtaining the error function, the change of each weight $\Delta\boldsymbol{w}_{ij}$ is calculated by multiplying, the gradient of the error function with respect to w_{ij} (weight vector) with the negative magnitude of the learning rate λ as seen below in Eq.5.30.

$$\Delta w_{ij} = -\lambda \frac{\partial f(\boldsymbol{w})}{w_{ij}} \qquad \text{(Eq. 5.30)}$$

w_{ij} is the weight between nodes i and j. All of the previously assigned weights are adjusted with every epoch and newly updated weights are assigned.

For this type of prediction model, It is important to identify when should the epochs stop. With every iteration, the cost function is decreasing and the weights are being updated, which consequently leads to the improvement of the model's performance, however, this trend of improvement stops at a certain point and overfitting occurs, so it is important to know when to stop the epochs. There are various criteria for stopping the epochs, one of which is the "*Minimum validation Error"* and this is basically telling the ANN to stop iterations when the first minimum is reached and another criterion is *"Maximum number of Validation Failure"* and this method follows a different approach that is to stop epochs after an n number of iterations where no decrease of the loss function is observed.

As the size of datasets increases or the number of neurons increases, or the number of hidden layers increases, the computational complexity will also increase leading to an increase in total time to get results, that is why these models have a predefined maximum number of epochs or maximum amount of waiting time where the model executes the operation if exceeded.

All of the parametric and non-parametric models that will be used have been discussed and this chapter is concluded. The following chapter will discuss how was the dataset available handled and how were the models applied while illustrating the results of each model. Important conclusions and remarks will also be demonstrated.

CHAPTER 6

EMPIRICAL RESULTS

In this chapter, all the empirical results will be discussed from data processing to each model being implemented and their corresponding results, also the resultant hybrid model will be explained and its results will be presented. Finally, important remarks about the methodology and future work will be mentioned. The software used for all the upcoming demonstrations is python version 3.9.

Before the discussion, a flowchart describing the flow of the work will be presented below to familiarize the reader with the upcoming operations

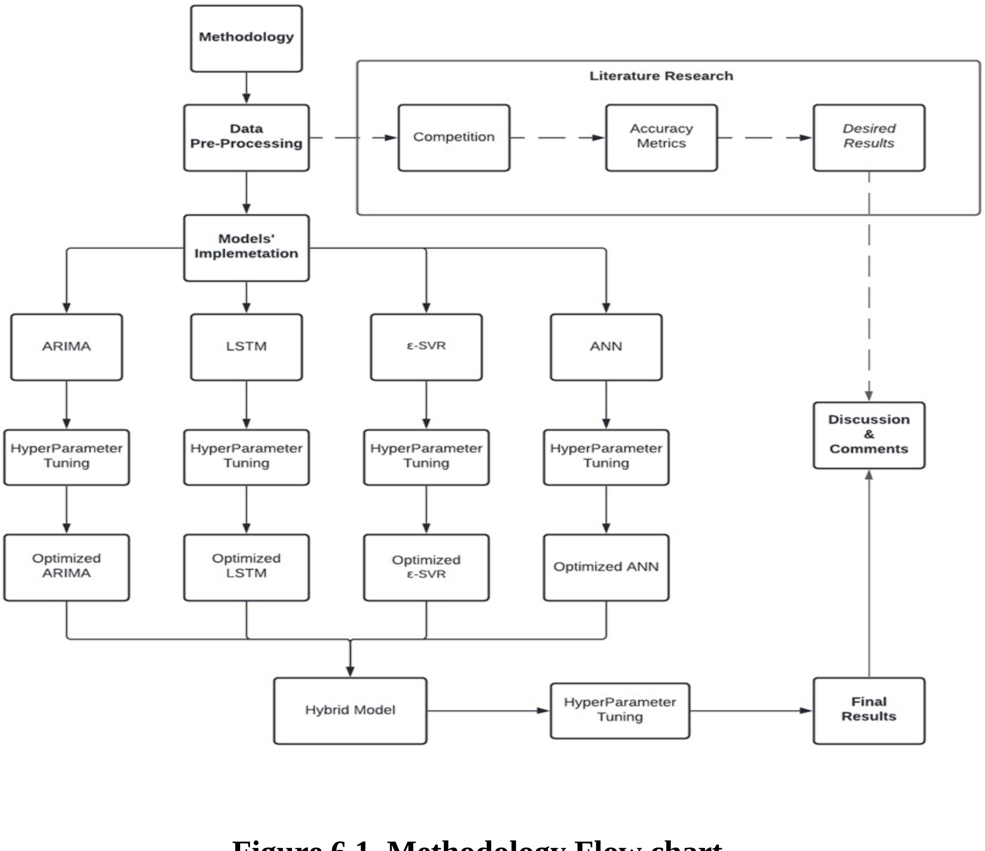


Figure 6.1. Methodology Flow chart

6.1 Accuracy Metrics

Usually, the accuracy of any operation can be calculated by dividing the difference between the output and input of the operation by the input and then multiplying it by 100, however, this is only valid for discrete, singular values. In regression problems such as the one presented in this thesis, the inputs and outputs are continuous, therefore different metrics are used to grade the models' performances. There are numerous accuracy metrics available in the literature, this thesis will use Mean Absolute Percentage Error (MAPE) and Mean Absolute Error (MAE).

6.1.1 Mean Absolute Percentage Error

This accuracy metric measures how accurate the forecast of the model is in terms of a percentage. The general scale of this metric is; the smaller the better. MAPE is formulated below [86] :

$$M = \frac{1}{n}\sum_{t=1}^{n}\left|\frac{A_t - F_t}{A_t}\right| * 100 \qquad \text{(Eq. 6.1)}$$

Where n is the number of fitted points, A_t is the actual value, and F_t is the forecasted value. The general scale is the smaller the better, 0 meaning accuracy of 100 percent whereas 100 means accuracy of 0 percent.

6.1.2 Mean Absolute Error

MAE is more or less a measure of errors between paired observations, it can be formulated using Eq.6.2 below:

$$MAE = \frac{\sum_{i=1}^{n}|y_i - x_i|}{n} \qquad \text{(Eq. 6.2)}$$

Where y_i is the predicted value, x_i is the true value and n is the total number of data points. The general scale is the same as MAPE, where the smaller the MAE the better [87].

6.2 The competition

Before trying to apply the models proposed, it is important to know what is the thesis trying to beat, so this thesis is going against ENSTO-E predictions that were obtained using temperature regression and load projection model that incorporates uncertainty analysis under various climate conditions. This model's predictions will be assessed according to the previously mentioned metrics starting from the date 15th of August 2018 till the end of the year 2018.

After applying the accuracy metrics it was found that ENSTO-E was able to forecast the demand with a **MAPE** of **0. 878** and **MAE** of **255.27**. which means that according to MAPE the forecast has an accuracy of **99.122 percent** and in the case of MAE, it means that there is an average of **255.27 MWh** error for every prediction.

The Accuracy may seem to be too good to be true, however, this is the accuracy of the last quarter of the year 2018. Since the beginning of this thesis, the importance of accurate electricity demand predictions has been emphasized and the sensitivity of error is very high, so it is expected to have such a percentage when dealing with data that was used to predict an important and advanced country such as Spain.

The goal of the thesis is to come up with a hybrid model that will outperform these results by applying different techniques that weren't used together before. Also, if the goal wasn't achieved and the proposed model failed to get the desired results, the thesis's results will guide future work into not taking the same path and if other researchers were to take the same path, they would have guidance on what to do and what to avoid, which will in return save time and create more room for optimization in the available literature.

After deciding on the accuracy metrics and identifying the accuracy that this thesis is trying to beat, the application starts to take place. The methodology is as follows, the available datasets will be processed before feeding them into the models, then each of the previously explained models will be applied and optimized. The results will then be obtained. Afterward, the best models in terms of performance will be combined and a hybrid model will be achieved.

6.3 Data Pre-Processing

As mentioned earlier in chapter 4, two data sets were obtained from reliable sources, one that is illustrated in figure 4.1 and the other in figure 4.3. Since this thesis is only focused on forecasting the electricity demand, in the dataset presented in figure 4.1, the columns containing the price information were dropped.

As for the other data set that includes the weather variables, this thesis will focus on the quantitative data only that is constantly changing with time, as a result, the columns including Rain in the last hour, Rain in the last three hours, Snow in the last three hours, Clouds, and Weather id were dropped.

Later on, the values in the datasets were set to appropriate types to ensure that none of the values lose their significant figure order while undergoing computations. For future reference, the dataset that is including the electricity demand will be referred to as **df2** and the other dataset including the weather conditions will be mentioned as **df1**. The datasets information are tabulated below in table 6.1 and 6.2 respectively.

Table 6.1. df2 datatypes

Column Name	Data Type
Time	Datetime64 [ns,UTC]
Total load actual	Float64
Total load Forecast	Float64

Table 6.2. df1 datatypes

Column Name	Data type
City name	object
Temp	Float64
Temp_min	Float64
Temp_max	Float64
Pressure	Float64
Humidity	Float64
Wind_speed	Float64
Wind_deg	Float64

After assigning the appropriate datatype for each variable, the dataset was checked for any missing data, and it was found that there are a total of 36 missing data points in df2 in the “Total actual load” column which accounted for 0.112 percent out of all the data points available. The number of missing data points in every month was plotted against time and is presented below in figure 6.1.

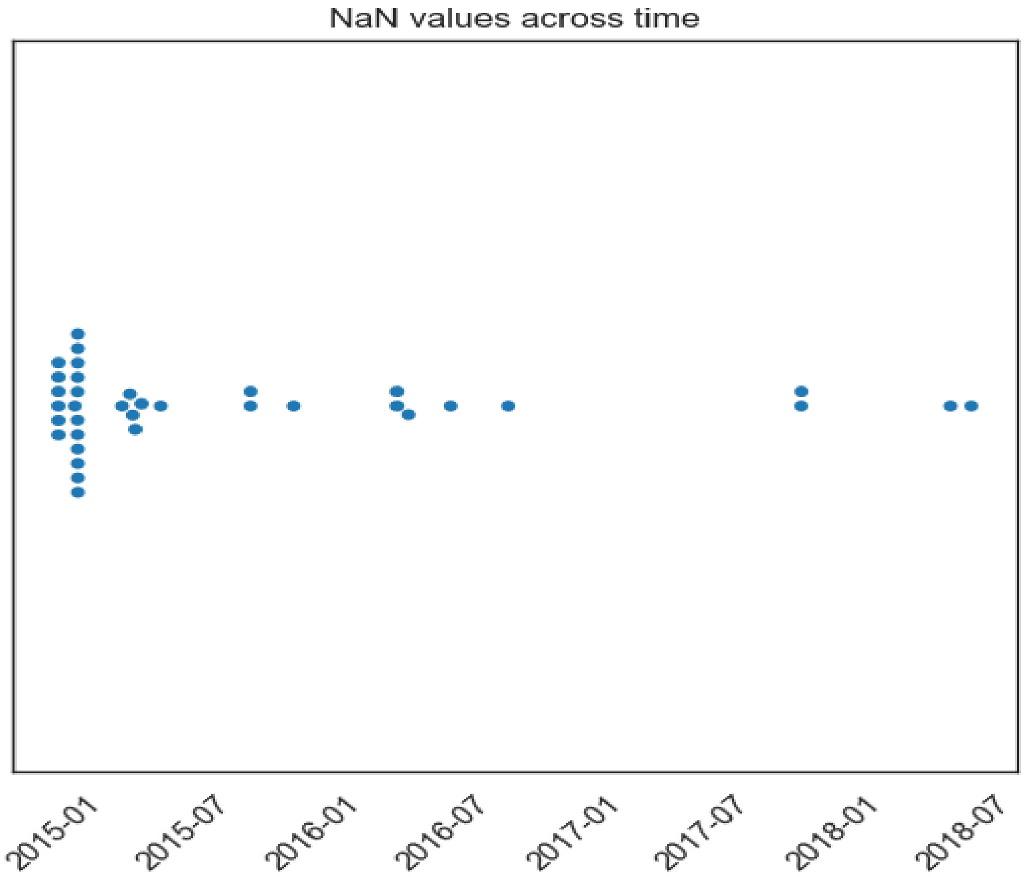


Figure 6.2. Number of missing data points in every month

Another illustration of the numbers presented in the figure above is shown below in table 6.3.

Table 6.3. Number of missing data points per month

Year-Month	Number of missing data points
2015-01	7
2015-02	11
2015-04	4
2015-05	2
2015-10	2
2015-12	1
2016-04	2
2016-05	1
2016-07	1
2016-09	1
2017-11	2
2018-06	1
2018-07	1

As the figure and table above illustrated, it can be seen in the timeline of the event, the frequency of NaN values is scattered around while the data are being recorded. There are several clusters early on, such as the ones around January 2015 and February 2015 whereas fewer records of missing values are recorded recently. It is possible that the instruments were being set up to measure those energy generation data early on, and as time progressed, a more healthy infrastructure was in place to collect necessary data.

Any machine learning algorithm will experience a decrease in its performance if there are missing data points that are included in the dataset, and to overcome such an obstacle, interpolation was used to reasonably assign values to the missing data points. Two different methods were used for interpolation, their usage depended mainly on the span of the NAN values, If there were multiple NaN values in the close span, quadratic interpolation with the order of four was used to account for the movement of the curve when calculating the standard deviation, mean, etc.., and in the case of a singular NaN value in the close span, a CubicSpline method was used to interpolate that specific NaN value [88].

At this point, the datasets are clear of any missing values and are in the correct format, before further processing, the two datasets are to be connected to have one whole uniform dataset that includes the inputs and the outputs. Since the two datasets have one column in common, that is; the time, which is in the same format (hourly basis from 2015 till 2018), the two datasets were combined using the method proposed by Dimitrios Roussi [89]. It is important to remind the reader that df1 had weather variables for 5 of the largest cities which as a result, when combining both datasets, for every weather variable, 5 columns were added each representing a different city.

To have a better understanding of the variation of each weather variable in each city, histograms, and distribution plots were plotted, all of them are available in Appendix A. Also, box plots were drawn for every variable, and outliers' existence was checked and removed to prevent any errors in future calculations. One of the multiple box plots that are also available in Appendix A is given below.

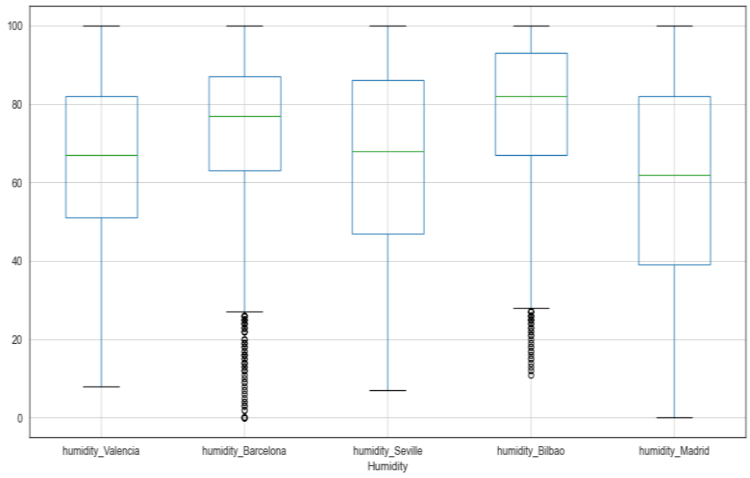


Figure 6.3. Humidity Box Plots for every city before removing outliers.

The black circles presented in the figure above represent outliers, and after removing these outliers as mentioned before, the new humidity box plot is shown below.

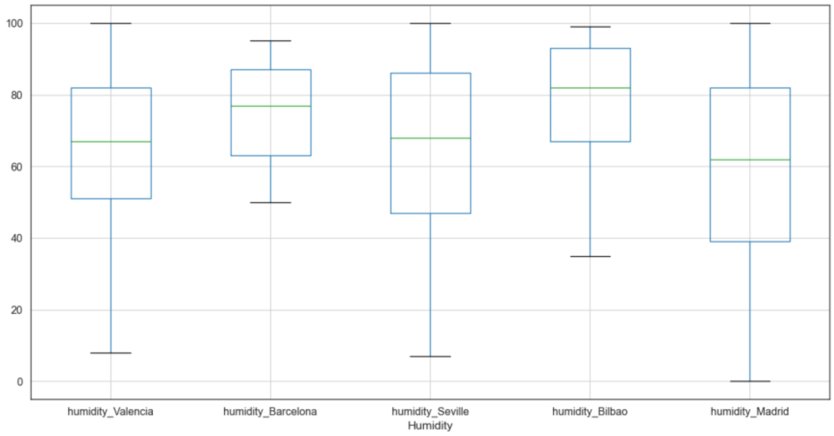


Figure 6.4. Humidity Box Plots for every city after removing outliers.

Box plots were plotted for all variables and were checked and then adjustments for outliers' removal were applied accordingly. One of the main variables in df1 is "wind degree" it was realized that the wind degrees are recorded in a cartesian coordinate system (0 to 360 degrees), since this is the case, angles such as 360 and 0 degrees for example are wind speeds pointing in the same direction but may have different interpretations by the machine learning algorithms, therefore, the wind degree column was omitted and instead wind components in the x and y direction were added as 2 new columns to prevent such possible misinterpretations using the equations below.

$$w_x = wind\_speed \ \cos(wind\_degree)$$

$$w_y = wind\_speed \ \sin(wind\_degree) \qquad \text{(Eq. 6.3)}$$

Where w_x is the x-component of wind speed and w_y is the wind speed in the y-direction. The resultant combined dataset includes a total of 42 columns which are tabulated below.

Table 6.4. Variables included in the combined Dataset

Column Number	Column	Datatype
1	Time	Datetime[ns,UTC]
2	Total load actual	Float64
3	Temp_Barcelona	Float64
4	Temp_min_Barcelona	Float64
5	Temp_max_Barcelona	Float64
6	Pressure_Barcelona	Float64
7	Humidity Barcelona	Float64
8	Wind_Speed_Barcelona	Float64
9	Wx_Barcelona	Float64
10	Wy_Barcelona	Float64
[11-18]	.	Weather_variables_Bilbao
[19-26]	.	Weather_variables_Valencia
[27-34]	.	Weather_variables_Seville
35	Temp_Madrid	Float64
36	Temp_min_ Madrid	Float64
37	Temp_max_ Madrid	Float64
38	Pressure_ Madrid	Float64
39	Humidity Madrid	Float64
40	Wind_Speed_ Madrid	Float64
41	Wx_ Madrid	Float64
42	Wy_ Madrid	Float64

The current problem that this thesis is dealing with is considered to be a time-series forecasting problem and one of the main characteristics of the target value that needs to be known is its trend, however, the trend is not easily recognized in complicated variations.

As can be seen below in figure 6.4, it is difficult to identify what is the actual trend of the total actual load due to its shape.

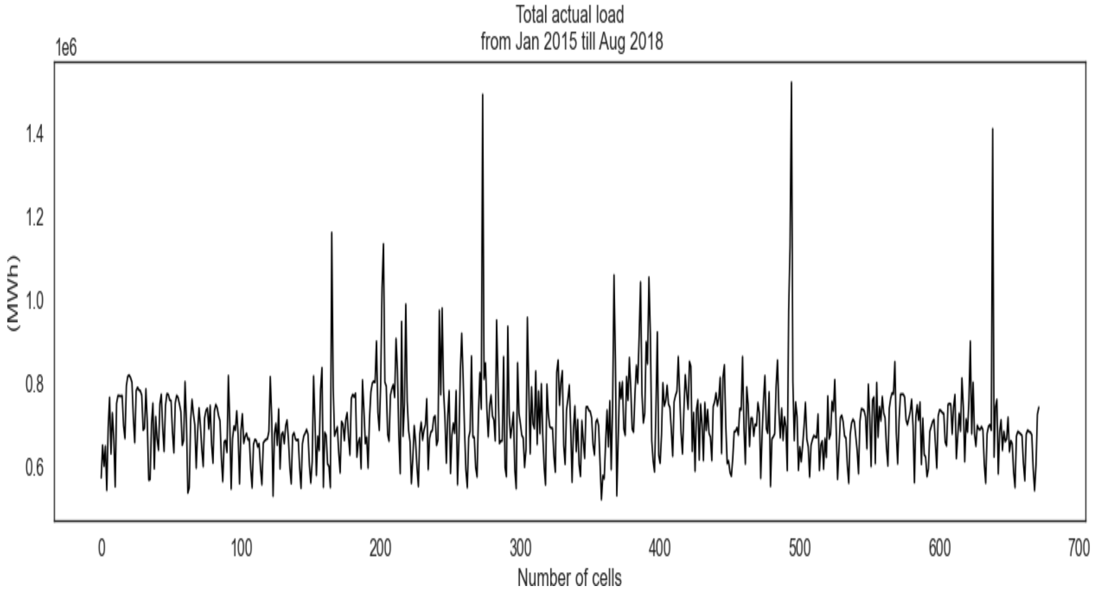


Figure 6.5.Total actual load variation

In figure 6.4, the total actual load is plotted on the y-axis and the number of data points is plotted on the x-axis. In the figure above to demonstrate the data in a clearer version, the data points were grouped to have a better scale of demonstration.

As it can be seen, no clear trend can be concluded from the figure, however, if this trend is decomposed into seasonal and residual, a resultant trend can be easily seen as illustrated below in figure 6.5.

As mentioned earlier in chapter 3, under the Trend Variation section, multiplicative seasonality proved to be superior to additive seasonality and as a result, it was used to decompose the trend in figure 6.4 for optimum results.

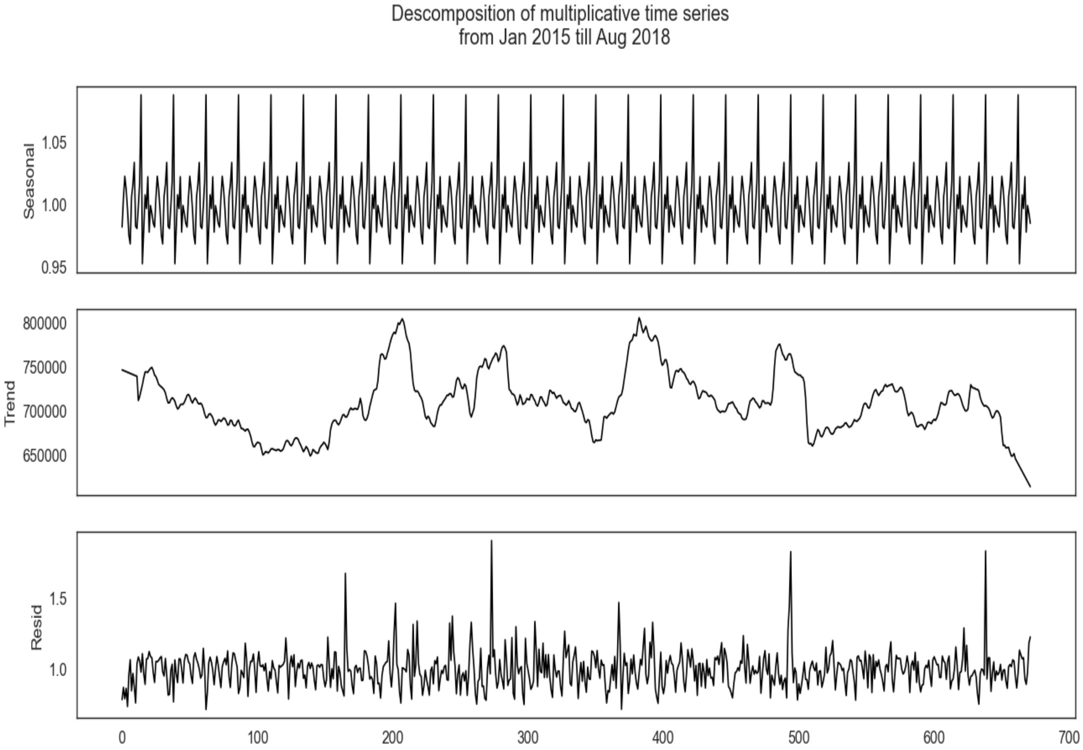


Figure 6.6. Decomposition of multiplicative time series

Resid is more or less, noises (unwanted spikes). Seasonality and residuals are unwanted information in any trend variation, decomposing them will facilitate the finding of a clearer relationship between input (weather) and output variables.

Other than the removal of seasonality, checking for the randomness of the dataset is a vital and critical step to ensure the possibility of machine learning models' application. Randomness is an unwanted trait in any dataset in an AI application because if the dataset was random, the possibility of reaching useful conclusions would be impossible, since no relationships or underlying connections can be made. Random datasets prevent the machine learning algorithms from achieving their optimal performance for mapping the data because relationships can't be established.

To check for randomness, correlograms are one of the most popular methods used. A correlogram depicts the auto-correlation between data pairs at various time intervals. Correlograms are tools for determining randomness in a dataset by computing auto-correlations and partial auto-correlations for data values at various time lags. Correlograms are simply computing the relation between two variables and if the results were zero, that means that there is no direct relation between them and if the results of their computations were non-zero, the opposite would be true.[90]

A correlogram was applied on the total load actual(dependant variable) only, to check if its variation is random or there is an underlying relationship that the applied models can achieve. Since the correlogram was applied on one variable only, the relationships created by the correlogram is between every data point and its lag, in other words; $Total\ actual\ load_t$ and $Total\ actual\ load_{t-1}$

If the time lag separation is random, both correlation graphs are near zero, but if it is not, one or more of the auto-correlations will be non-zero. Autocorrelation and partial autocorrelation graphs are plotted below for the combined dataset.

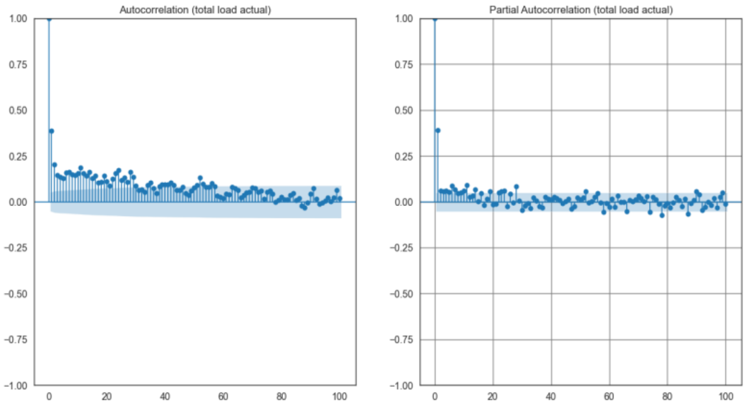
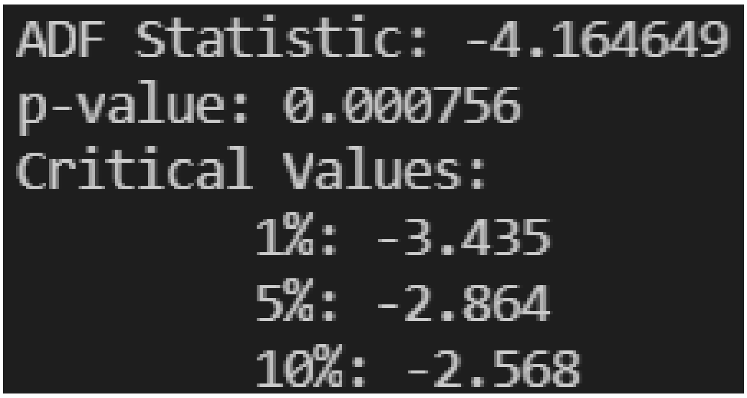


Figure 6.7. Autocorrelation and partial autocorrelation graphs.

In figure 6.6, the time lags are presented on the x-axis and their correlation magnitudes are on the y-axis, the blue region accounts for values that can be considered as 0. It can be seen in the figure above that there are lag values which are larger than 0 indicating a non-random behavior in both; autocorrelation (left) and partial autocorrelation (right).

Before concluding the data preprocessing, one final check is to be made which is checking if the dataset is stationary. “Stationary” is a term that is usually used with time series data like the one present in this thesis. Time-series data can be considered stationary if its statistical properties do not change with time [91]. Stationarity is vital in ML and AI applications because these applications are constantly using statistical tools and tests that rely mainly on the statistical properties of the data.

To check for stationary, Several tests are conducted in the literature, one of the famous methods is the Augmented Dickey-Fuller (ADF) test. This test operates on what is known as a “null hypothesis” which is the assumption that the time series contains a unit root and is non-stationary. Resulting in the following criterion; if the P-value in ADF is less than the significance level of (0.05) and the ADF test statistics value is smaller than the critical values at different percentages, the time series can be considered as stationary [91]. In python, there are specific functions that can calculate the ADF statistic which was applied, and the following results were obtained.



```
ADF Statistic: -4.164649
p-value: 0.000756
Critical Values:
        1%: -3.435
        5%: -2.864
        10%: -2.568
```

Figure 6.8. ADF test results

The P-value is smaller than the predefined significance level and the ADF statistic is smaller than several critical values, concluding that the dataset presented is stationary.

The data-preprocessing stage has been concluded at this point and the models' application will be initiated. After discussing in the previous chapter, the different models that will be utilized, and the complexity of the governing equations used, when it is time for application, the current technology facilitated the process with the aid of different softwares. Therefore, simple and precise steps were conducted to achieve reliable results using Python.

6.4 ARIMA MODEL

There exist different ways of implementing an ARIMA model on a dataset, the classical way of doing this is following the Box-Jenkins Methodology [72]. The implementation procedure is as follows:

6.4.1 Model Identification:

This stage involves the usage of plots and statistical summaries to differentiate different trends, seasonality, and autoregression elements to reach a conclusion on the amount of differencing and size of the lags that are required.

This stage was completed in the data-preprocessing phase. According to the theory explained in chapter 5 and the results reached in the previous section, the dataset is stationary (no seasonality) and no differencing is needed which means that the **ARIMA (p,d,q)** model will be used with a “**d**” term equal to **0**. Moreover, the dataset had to be normalized by changing the data to fit within a common scale, without distorting differences in the ranges of values or losing information.

6.4.2 Parameter Estimation:

In this phase, the model is fitted to the dataset and the other parameters which resemble the Autoregression and Moving Average, are to be found. In python, there are libraries and built-in functions that calculate the optimum p and q values for the specific data set that is used. The functions obtained **p** and **q** values of **3** and **2** respectively. Consequently, **ARIMA (3,0,2)** model will be implemented.

6.4.3 Model Checking

After choosing the parameters for ARIMA, the model is then constantly receiving adjustments to reach the optimum level of fit for the in-sample or out-sample of the model (training and testing candidates).

Again this procedure could be achieved using built-in features in python while obtaining the results. The fitting summary can be shown below.

Table 6.5. Summary of Convergence of the ARIMA model

N	*Tit*	*Tnf*	*F*
6	42	45	8.2965

Where *N* accounts for the number of steps ahead for which prediction is required, *Tit* is the number of iterations, *Tnf* is the total number of function evaluations and *F* is the final function value with which the model is fitted on the data. The dataset was divided into a training phase that starts from the beginning of the dataset till the 14th of August 2018 at 23:00 and a testing phase that begins with the 15th of August 2015 at 00:00 till the end of 2018.

As mentioned earlier, the ARIMA model doesn’t involve any inputs but rather focuses on prediction using lags of the value of interest. As a result, only the total load actual will be fed into the model and forecasted values will be generated.

6.4.4 Results

After the model was successfully fitted, graphs demonstrating the results were plotted, as seen below in the figure.

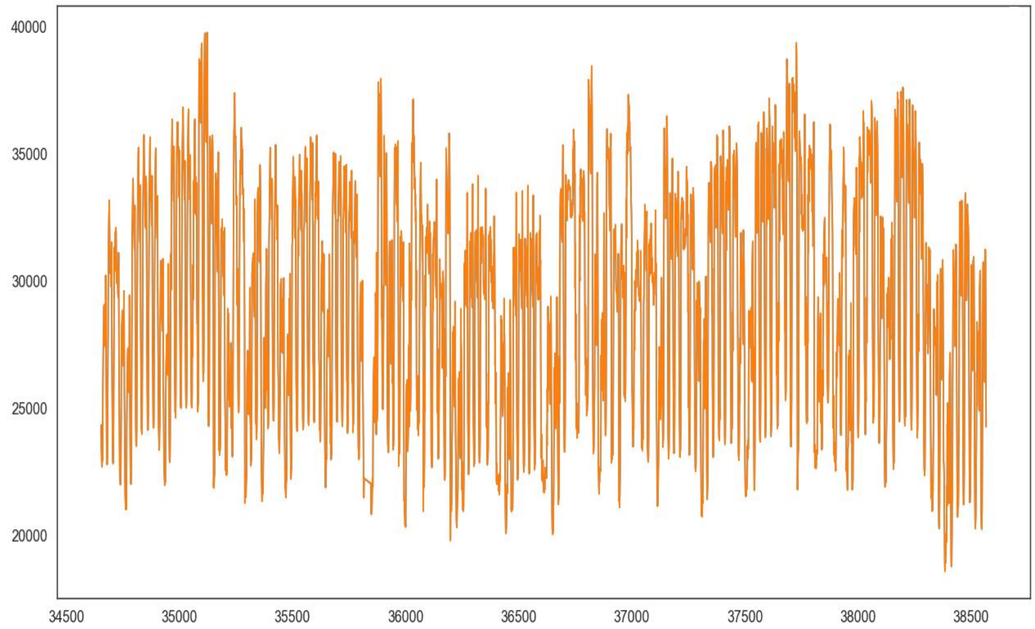


Figure 6.9. ARIMA Forecasted load

In the figure above the forecasted demand is colored orange. On the y-axis, the values of the demand in MW are listed, and on the x-axis, the number of data points present in the dataset which is around 36000 is plotted. When the forecasted load is added to the actual load in one graph, the following demonstration is obtained in figure 6.9. Where the blue-colored trend represents the actual load as indicated in the legend. It can be seen that the orange line starts from a specific region which is the forecasted time domain that starts from the 15th of August 2018 till the end of 2018.

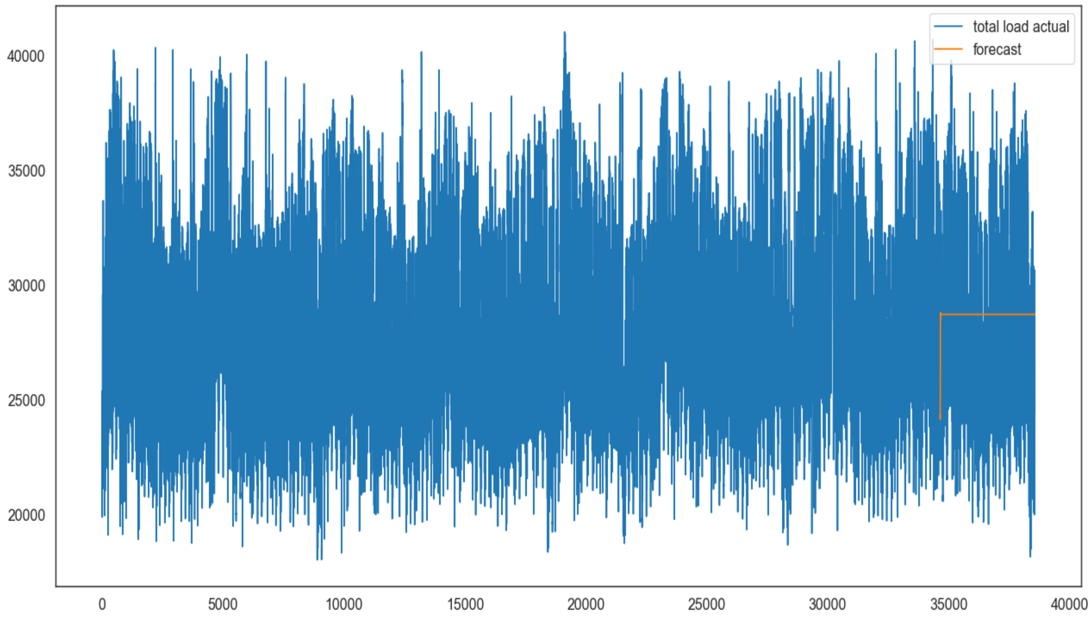


Figure 6.10. Actual vs. ARIMA forecasted Load

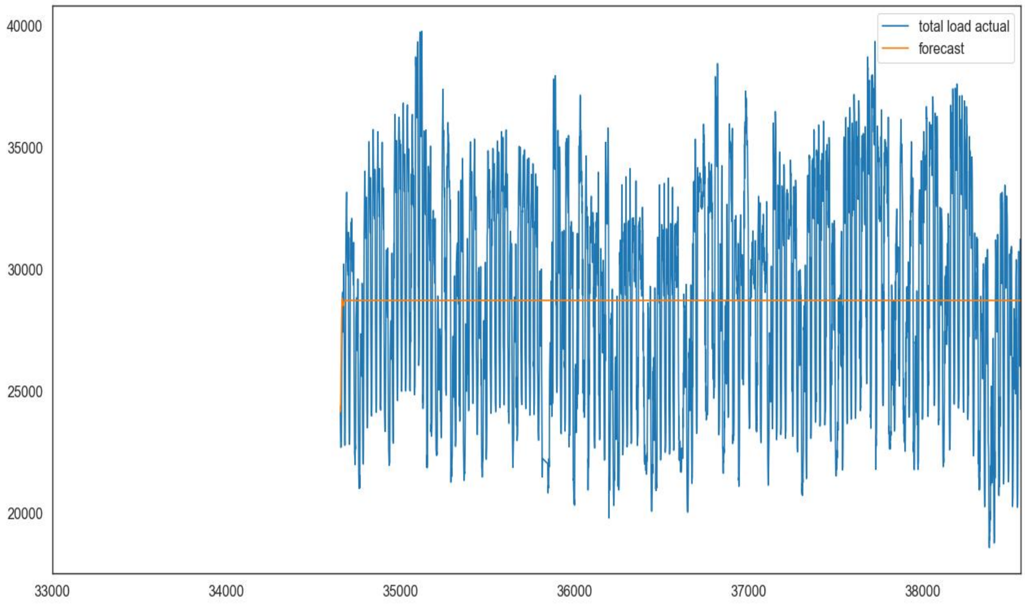


Figure 6.11. Total vs. Actual load for the forecasted region

Unfortunately, the ARIMA model performed poorly to the extent that the predictions were just the average of the total actual load.

After applying the accuracy metrics priorly mentioned, the ARIMA model achieved a **MAPE** of **14.12** and an **MAE** of **3938.59**. which means that according to MAPE the forecast has an accuracy of **85.88 percent** and in the case of MAE, it means that there is an average of **3938.59 MWh** error for every prediction.

6.5 LSTM

The second model that is applied to the dataset is the long short-term memory, unlike ARIMA, LSTM is more complicated and it contains several "*hyperparameters",* hyperparameters are specific variables in the model's algorithm that need to be adjusted to fit the dataset in order to achieve optimum results. In other words, the tuning of these parameters would help the model to best fit the dataset. The model contains the following hyperparameters:

- ***Lookback:***
  This term represents the number of data points that will be taken into consideration while predicting one value.
- ***The number of modules:***
  As explained earlier, LSTM consists of n number of modules, so identifying the optimum number of modules for the best results and simultaneously, having considerable computational complexity with a suitable run-time is the goal of this choice.
- ***Batch size:***
  The number of data points that are fed to the system at a time
- ***Epochs:***
  Number of iterations that the whole data will be fed into the total number of modules to give out predictions.

- ***Optimizer type:***
  These are functions that are used to fit the parameters to the model, there is a total of 7 functions known as: *"Adam", "Rmsprop", "Adadelta", "Adagrad", "Adamax", "Nadam", and "Ftrl".*

The importance of these variables depends on their contribution to the prediction, so their adjustments will be done one at a time according to their vitality. The hyperparameters' tuning will be done manually.

To prevent any overfitting and to make sure that the model would work similarly on any new data i.e. the hyperparameters would be tuned perfectly for that specific testing data, the dataset will be divided into a training dataset that acquires 70 percent of the whole data, validation dataset of 19 percent, and testing of 11 percent. These portions were chosen according to the testing data's initial date which is the 15th of August which represents the last 11 percent of the data.

A validation dataset was added to the LSTM and not ARIMA, due to the fact that no tuning was done manually and the hyperparameters weren't adjusted specifically to the dataset. The LSTM model that will be used is a univariate LSTM with total load actual as its variable. In other words, LSTM will be doing the exact same thing as ARIMA but using a different algorithm. In addition, the data set has to be normalized before any tuning.

The number of modules was deemed to be the most crucial out of all the hyperparameters so its tuning was done first. The other parameters were set to the following values.

- Look back: 1
- Batch size: 10
- Epochs:10
- Optimizer: "Adam"

The epochs were chosen to be a small value to decrease the computational time and according to the previous chapter and the literature, as the number of epochs increases, the learning of the algorithm will also get better to a certain extent. To grade the values inserted for the hyperparameters, MAPE will be chosen as the criteria of acceptance. The initial values are chosen randomly. Optimizer type and epochs will be the last hyperparameters that will be adjusted because they depend on the other parameters.

Table 6.6. Hyperparameter tuning (Number of Modules)

Trial	Number of Modules	MAPE
1	*100*	*3.3824*
2	*200*	*3.3912*
3	150	3.4043
4	120	3.4068
5	*50*	*3.3906*
6	70	3.4373
7	**80**	**3.3554**
8	*60*	*3.3916*
9	20	3.4094
10	10	3.5304

Initially, 100 modules were chosen then the model's MAPE was taken as a reference, larger modules were then assigned and their MAPEs were worse than the reference, so the search shifted direction towards smaller values. It can be seen at this stage of optimization from the table that the optimum number of modules is **80**. Moving on to the next hyperparameter; Look Back.

Table 6.7. Hyperparameter tuning (Look Back) with number of modules as 80

Trial	Look Back	MAPE
1	1000	11.294
2	500	6.4605
3	100	4.424
4	80	3.1848
5	60	2.6605
6	40	2.4407
7	*20*	*2.3264*
8	*10*	*2.2757*
9	5	2.4319
10	6	2.4235
11	8	2.4524
12	**11**	**2.2236**
13	*15*	*2.2783*

Since there are around 36000 data points, it was seen that it is possible to use 1000 historical data points for the prediction of one, so the initial value was taken as 1000 and a descending value order was taken since the MAPE was continuously getting better. According to the table above, the optimum number of lookbacks is **11**. Next, is the batch size.

Since predictions are hourly based, and there are 4 years worth of data, the batch size estimation was initiated having multiples of 24 which is the number of data points per day. The initial batch size was chosen as the maximum value which is 32850. This value is equivalent to the number of data points present until the 15th of August 2018.

Table 6.8. Hyperparameter tuning (Batch size) with number of modules as 80 and look back as 11

Trial	Batchsize	MAPE
1	32850	11.9403
2	26280	13.8612
3	17520	11.9003
4	8760	13.1479
5	4380	9.7234
6	2190	8.1350
7	720	3.5703
8	360	3.1538
9	168	2.6184
10	72	2.4000
11	24	2.4307
12	*12*	*2.2611*
13	6	2.4107
14	*10*	*2.2477*
15	**8**	**2.2165**

According to the table above, the optimum batch size is **8**, moving forward, the type of optimizing function will be decided.

Table 6.9. Hyperparameter tuning (Optimizer) with number of modules as 80, look back as 11, and batch size 8

Trial	Optimizer	Mape
1	**Adam**	**2.2165**
2	Rmsprop	2.5162
3	Adadelta	11.7162
4	Adagrad	6.8793
5	Adamax	2.3079
6	*Nadam*	*2.2246*
7	Ftrl	10.2459

According to table 6.9 above, the best optimizer for the current hyperparameter values seems to be “Adam”. So far the hyperparameter tuning was done manually and at this point, A rough estimate for the range of each parameter is known. In python software, a loop was created to try each possible combination of the top parameters that are listed in bold or italic or underlined in the tables above. A total of **120** different combinations was checked to ensure that the manual combination found is actually the optimum combination. A sample of the highest results is attached in Table 1 in Appendix B.

The final result that was obtained was as follows.

- **Number of Modules:100**
- **Look back:11**
- **Batch size:10**
- **Optimizer: “Adam”.**

This combination is different from the manually chosen one indicating that there is a possibility that there are different combinations out there that would yield better results. Nevertheless, trying all the possible combinations is not feasible, consequently, no more trials will be conducted. This optimum combination resulted in a **MAPE** of: **2.200** which is equivalent to **97.8 percent** accuracy.

So far all of these hyperparameter tunings were done on the validation set. Now, these final parameters will be applied to the testing data set and a final prediction will be obtained. The number of epochs will be adjusted in an ascending manner until “the test loss” stays constant or starts increasing to prevent overfitting as mentioned prior.

A model’s loss is a number indicating the errors in the prediction. As the number of iterations increases the training loss together with the testing loss is expected to decrease respectively, yet till a certain point. The testing loss may stop varying and remain constant and sometimes, the loss starts increasing again.

A graph demonstrating the training and testing loss was plotted against the number of epochs used and the figure below was obtained.

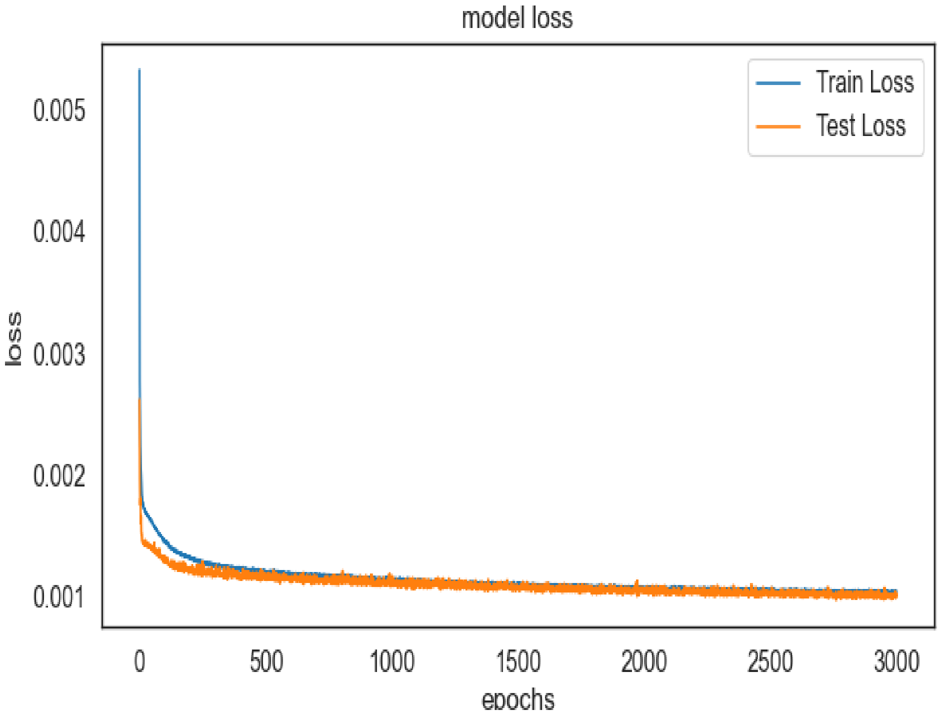


Figure 6.12. Model Loss

A total of 3000 epochs was conducted and the training and testing loss started to show constant variation, indicating that increasing the number of iterations more than that would not change the final accuracy much and will be more time-consuming.

Even though it can also be argued that stopping at 1500 iteration could be sufficient, since the targeted problem is of high vitality, more iterations were conducted to have higher confidence in the obtained results.

The predicted load vs the actual load were then plotted on the same graph to see the difference in value as illustrated below in figure 6.12.

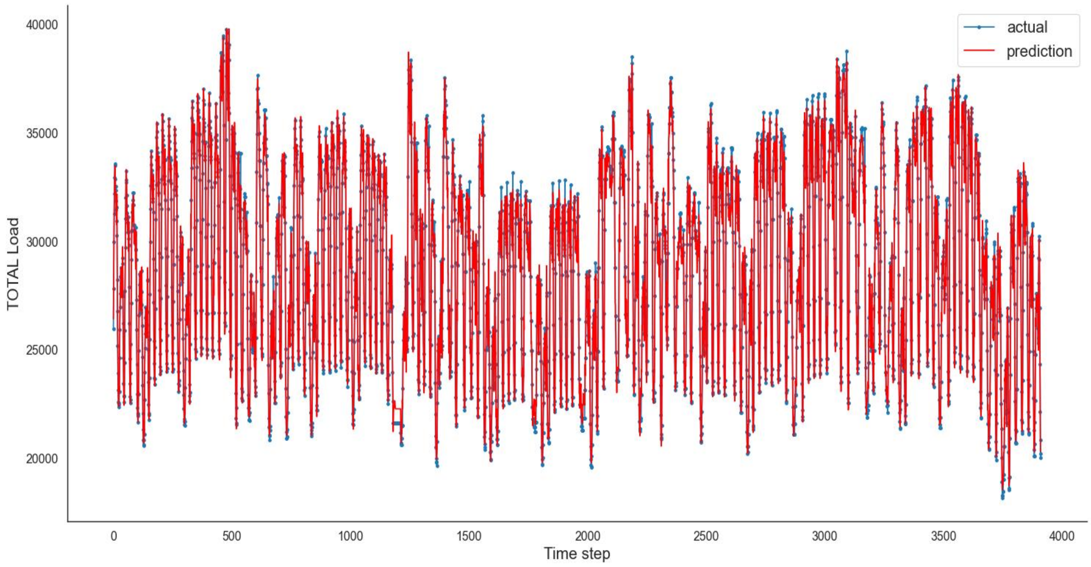


Figure 6.13. Actual vs. LSTM predicted load

The legend above states that the red trend is the LSTM predicted load whereas the blue is the actual total load. Minimal differences were seen indicating that the accuracy of the model is high.

Later on, the accuracy metrics were applied and the final results were obtained. The LSTM model acquired a **MAPE** of **1.7703** and an **MAE** of **504.756**. which means that according to MAPE the forecast has an accuracy of **98.23 percent** and in the case of MAE, it means that there is an average of **504.756 MWh** error for every prediction.

6.6 *ε*-SVR

This model will start incorporating the weather variables as inputs meaning that *ε*-SVR will be used as a multivariate model. The same methodology is applied as the LSTM, the data set was first normalized and then divided into three main bulks training, validation, and testing portions with the same percentages for consistencies and ease of comparison.

In the case of ε-SVR, the main hyperparameters that need tuning are:

- The number of inputs
- Type of Kernel
- Cost Function
- *ε*

These variables were explained earlier in the previous chapter so no further explanation is needed. As specified earlier, there are a total of 40 different variables that represent each weather input for the five major cities and one extra variable that is time. The literature suggested that regardless of the problem that is aimed to be solved, the less the number of variables that can be fed to an ε-SVR model the better, given that no important information is lost. [92]

In an attempt to filter the number of inputs, a correlation matrix was drawn to check for any of the variables that have high collinearity. A correlation matrix operates by connecting all variables using a linear relationship. These proportionality constants are in the range of 0 to 1. Zero meaning; not correlated and one indicates that the variables are the same. The correlation matrix between the variables are is shown below in figure 6.13

.

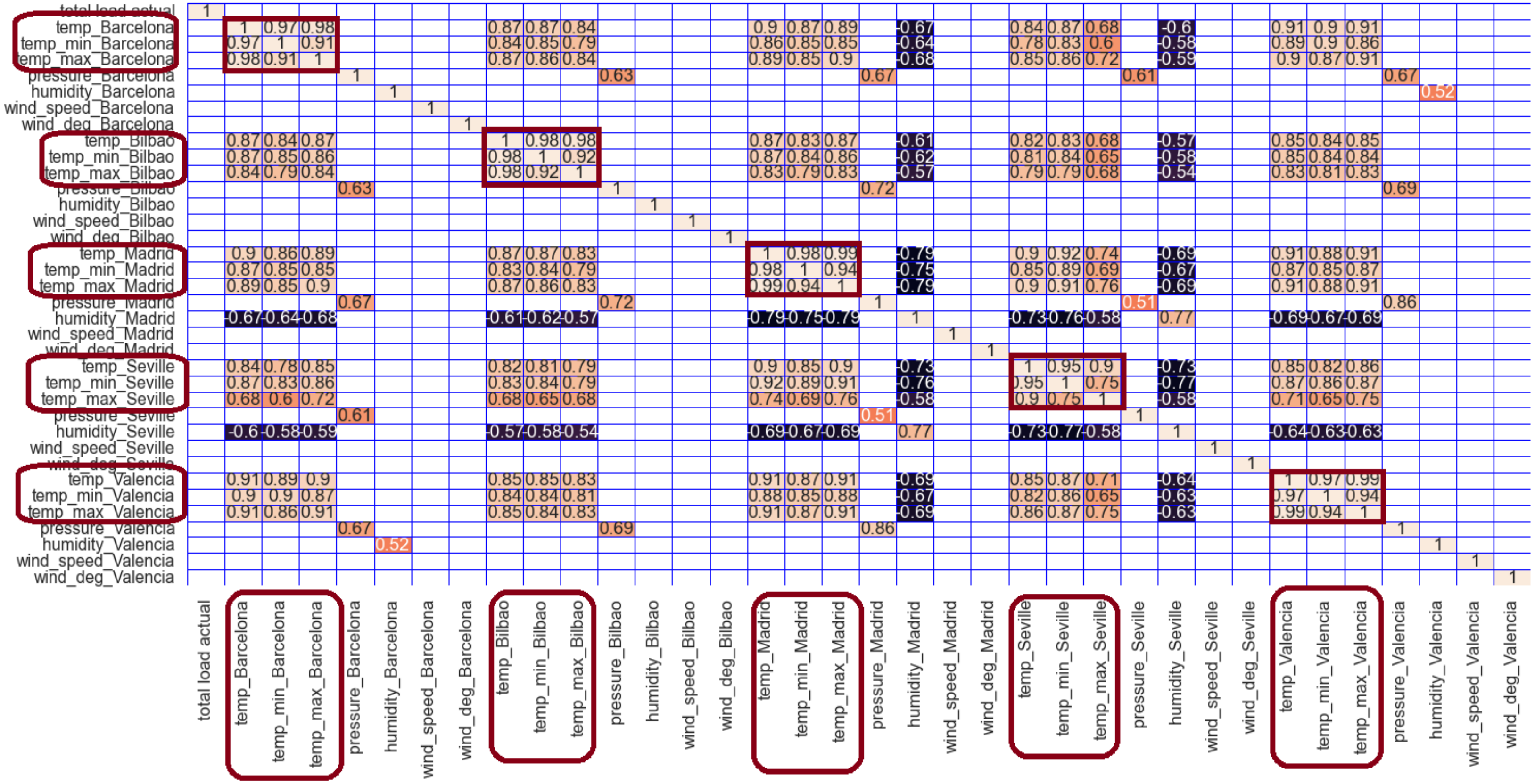


Figure 6.14. Correlation Matrix

Figure 6.13 is a correlation matrix between all 40 weather variables. If a closer observation is taken, it can be found that temp_min and temp_max are highly correlated with each other and also with the actual temp for each city, to prevent any multicollinearity conundrum, the temp_min, and temp_max columns were dropped.

The number of input variables decreased from 40 to 30. In an attempt to decrease the number of weather inputs further, the possibility of adding weighted coefficients was explored. These weighted coefficient values were chosen in accordance with the previously stated assumption that the 5 major cities in terms of the population can provide acceptable accurate results. Having said this, weighted coefficients were assigned values with respect to the cities' population in the specific years, resulting in one combined column for all 5 cities for a specific weather input [93]. For example, the combined temperature for 2015 will be as follows:

$$Combined\ temp\_2015 = w\_2015_{barc} * T_{barc} + w\_2015_{bilb} * T_{bilb} + w\_2015_{mad} * T_{mad} + w\_2015_{sev} * T_{sev} + w\_2015_{val} * T_{val} \qquad \text{(Eq. 6.4)}$$

Where;

$$w\_2015_{barc} = \frac{Barcelona\_population\_2015}{Total\ Population\_2015}$$

$$w\_2015_{bilb} = \frac{Bilbao\_population\_2015}{Total\ Population\_2015}$$

$$w\_2015_{mad} = \frac{Madrid\_population\_2015}{Total\ Population\_2015}$$

$$w\_2015_{sev} = \frac{Seville\_population\_2015}{Total\ Population\_2015}$$

$$w\_2015_{val} = \frac{Valencia\_population\_2015}{Total\ Population\_2015} \qquad \text{(Eq. 6.5)}$$

Proceeding to apply equations 6.4 and 6.5 to all the dataset, the combined dataset with the weighter parameters includes 6 major inputs (dependant variables), all of which are listed below in table 6.10.

Table 6.10. Combined Dataset with weighted parameters

Column	Data type
Time	Float64
Combined Temperature	Float64
Combined Humidity	Float64
Combined Pressure	Float64
Combined Wind Speed	Float64
Combined Wx	Float64
Combined Wy	Float64
Total Load Actual	Float64

Following the decrease of the number of inputs, the other main parameters are to be chosen there are a total of 3 different kernels namely; Radial Basis Function(RBF), Linear function, Homegenuous Polynomial function. Each of these kernels includes other main parameters such as C and ε which are common in all functions and a specific parameter for polynomial which is its degree. In other words; SVR is a function of ;

$$SVR(kernal\colon (RBF, Linear), C, \varepsilon)$$

$$SVR(kernal\colon Polynomial, Degree, C, \varepsilon) \qquad \text{(Eq. 6.6)}$$

The width of the ε -insensitive zone mentioned in the previous chapter, which is used to fit the training data, is controlled by ε. The number of support vectors needed to generate the regression function is affected by this value The larger the ε, the fewer support vectors are generated. Larger ε, on the other hand, results in more flat estimations. However, if the ε is too low, for example, 0, overfitting is expected. Choosing a valid value for ε becomes the most vital step after choosing the type of kernel which will be used.

In that sense a series of ε were chosen to be [0.01,0.05,0.1,0.4,0.6,0.8]. All the values were are to chosen to be less than 1 because while normalizing the data, the data points were set in the range of 0 to 1. As for C, which is the “Regularization” parameter. The strength of the regularization process is inversely proportional to C, adding a constraint of preferring small positive C values. C values were chosen to be one of the following: [0.001,0.005,0.01,0.05,0.1,0.4,0.8,1,2,4]. Finally, for the polynomial kernel, different degrees were plugged in ranging from 2 to 8.

Following the same procedure, all the possible combinations were tried on the validation dataset and a total of **540** different models were generated with MAPE as the grading matrix and it was found that the best combination that acquire a **MAPE** of **12.977** which is equivalent to an accuracy of **87.023** percent is:

- **Type of Kernel: Linear**
- **C: 4**
- ***ε:* 0.05**

Samples of all the combinations are provided in table 2 and table 3 in Appendix B. The resultant model was then applied to the testing dataset. The forecasted load was plotted on top of the actual load to visualize the prediction error in figure 6.11 below with the forecasted demand in red.

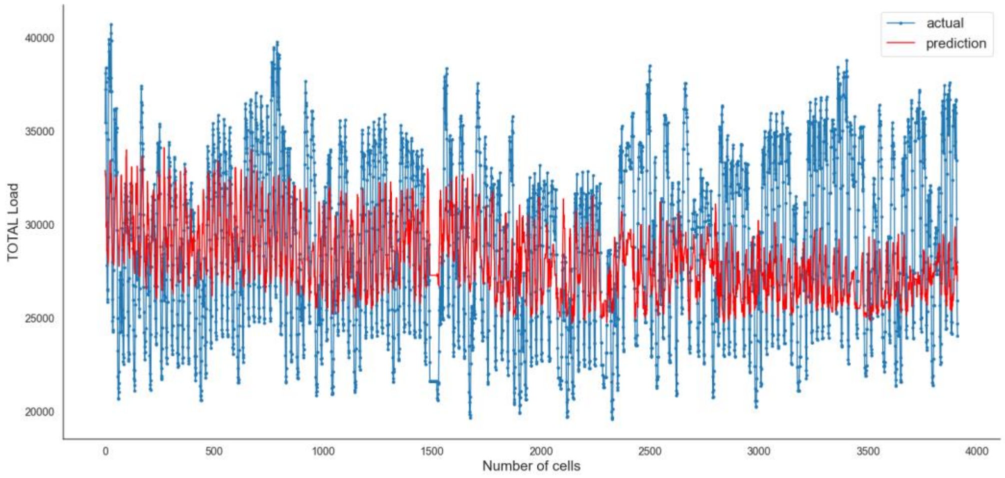


Figure 6.15. Actual total load vs. ε -SVR forecast

Figure 6.14 only demonstrates the forecasted region which as a result, the forecasted load is spread through all the x-axis. The optimum parameters were then applied to the testing dataset and the results were as follows, the model ε -SVR acquired a **MAPE** of **12.886** and an **MAE** of **3731.5**. which means that according to MAPE the forecast has an accuracy of **87.114 percent** and in the case of MAE, it means that there is an average of **3731.5 MWh** error for every prediction.

6.7 ANN

Artificial Neural Networks like ε -SVR and LSTM will need the same procedure for implementation. That includes the normalizing of the dataset, division of the dataset into training, validation, and testing.

ANN is also known to have an increase in computational complexity and duration as the number of inputs increases [92], therefore, the weighted inputs that are tabulated in table 6.10 will be fed into the ANN. This machine learning model requires multiple parameters that require tuning in order to achieve optimum results. Using python as the execution platform, the parameters needed to be adjusted are as follows:

- Number of neurons (nodes) in each layer
- Number of hidden layers
- The gradient of neurons increments throughout the network

  This parameter represents the nature of the number of nodes increments in each layer i.e. converging (number of neurons are maximum in the input layer and then continues to decrease until the output layer), diverging or linear(constant through our the layers).
- Activation functions:

  There is a total of 8 activation functions available; Rectified Linear Activation (*ReLU*), Logistic (*Sigmoid*), Hyperbolic Tangent (*Tanh*), *Softplus*, *Softsign*, Scaled Exponential Linear Unit (*selu*), Exponential Linear Unit (*ELU*), and *Exponential*.
- Optimizing Functions

  Same as the ones presented in table 6.9, a total of 7 functions;
  Adam, Rmsprop, Adadelta, Adagrad, Adamax, *Nadam,* and Ftrl.
- Epochs
- Batch size

Having multiple hyperparameters that can take numerous values, the decision of choosing the right combination was complicated. To simplify the faced problem and to save time; 2 hidden layers were chosen with a converging gradient of 4n for the input layer, 2n for the first hidden layer and n or the second hidden layer (where n represents the number of neurons), and finally, 1 for the output layer since the model is trying to predict one output value.

As an initiation of the filtration process, the number of epochs was set to 10 and batch sizes were set to be [12,24,48], number of neurons was set to take values equal to [2,4,8,16,32,64] and all activation and optimization functions were used to find the top possible combinations. A total of **1008** different models were checked according to their MAPE's. A sample of these combinations is attached in Table 4 in Appendix B and the top combinations are tabulated below.

Table 6.11. Top ANN models (Initial Filtration).

Trial-Number	Parameters	Accuracy
303	neurons: 64 - Act: **softsign** - Opt:**RMSprop** - batch_size:12 - Epochs:10	70.20198375
639	neurons: 64 - Act: **softsign** - Opt:**RMSprop** - batch_size:24 - Epochs:10	70.12225229
583	neurons: 32 - Act: **softsign** - Opt:**RMSprop** - batch_size:24 - Epochs:10	70.09178252
296	neurons: 64 - Act: softplus - Opt:**RMSprop** - batch_size:12 - Epochs:10	70.08514573
191	neurons: 16 - Act: **softsign** - Opt:**RMSprop** - batch_size:12 - Epochs:10	69.91640711
506	neurons: 16 - Act: *relu* - Opt:**RMSprop** - batch_size:24 - Epochs:10	69.90841512
198	neurons: 16 - Act: tanh - Opt:**RMSprop** - batch_size:12 - Epochs:10	69.9075389
618	neurons: 64 - Act: *relu* - Opt:**RMSprop** - batch_size:24 - Epochs:10	69.89030986
226	neurons: 32 - Act: *relu* - Opt:**RMSprop** - batch_size:12 - Epochs:10	69.87402996
646	: neurons: 64 - Act: tanh - Opt:**RMSprop** - batch_size:24 - Epochs:10	69.7950178
653	neurons: 64 - Act: selu - Opt:**RMSprop** - batch_size:24 - Epochs:10	69.66521784
898	neurons: 32 - Act: *relu* - Opt:**RMSprop** - batch_size:48 - Epochs:10	69.60468832
247	neurons: 32 - Act: **softsign** - Opt:**RMSprop** - batch_size:12 - Epochs:10	69.58311479

Table 6.11 lists the top 13 combinations in terms of MAPE. As the table shows, the optimizing function that proved to work best in most of the different combinations is Rmsprop. And in the case of the activation functions, “relu”, “softsign”, and“selu” have the highest frequency in terms of repetition in the top results. Also, “softplus” was in the top three models so it can’t be ignored. As for the batch sizes, 24 and 12 gave the best results. No conclusions were reached for the number of neurons, thus the first filtration process is terminated with the highest model scoring a **MAPE** of **29.80**. Next, the following combinations will be checked:

- The number of neurons (nodes) in each layer: 64 or 128 or 256
- Number of hidden layers:2
- The gradient of neurons increments throughout the network: converging
- Activation functions: “relu”, “softsign”, and“selu” and “softplus”
- Optimizing Functions: Rmsprop
- Epochs: 10
- Batch size: 12 and 24

These combinations result in **24** different models, once more, the models were graded according to their mape and the highest are tabulated below.

Table 6.12. ANN best models (Second Filteration Process)

Trial-Number	Parameters	Accuracy
17	**17 Parameters: neurons: 128 - Act: relu - Opt:RMSprop - batch_size:24 - Epochs:10**	**70.64658936**
6	6 Parameters: neurons: 128 - Act: softplus - Opt:RMSprop - batch_size:12 - Epochs:10	70.16981432
5	5 Parameters: neurons: 128 - Act: relu - Opt:RMSprop - batch_size:12 - Epochs:10	70.06811113
1	1 Parameters: neurons: 64 - Act: relu - Opt:RMSprop - batch_size:12 - Epochs:10	70.02923484
9	9 Parameters: neurons: 256 - Act: relu - Opt:RMSprop - batch_size:12 - Epochs:10	70.01887144

From the second filtration process, the activation function can be narrowed down to only relu and softplus. It can be seen that even after the narrowing down, the accuracy of the model didn't change much, the highest **MAPE** achieved was **29.35** indicating a possibility that ANN is not suitable, to make sure that this intuition is correct, more narrowed down combinations were carried out.

- The number of neurons (nodes) in each layer: 64 or 96 or128
- Number of hidden layers:2
- The gradient of neurons increments throughout the network: converging
- Activation functions: "relu" and "softplus"
- Optimizing Functions: Rmsprop
- Epochs: 40
- Batch size: 12,18, 24

The number of epochs was increased to have an understanding of whether the change in accuracy is heavily dependent on the increase in epochs or not. These combinations resulted in a total of **18** different combinations the highest 8 of which are tabulated below.

Table 6.13. ANN best model (Third Filteration Process)

TrialNumber	Parameters	Accuracy
4	**neurons: 96 - Act: relu - batch_size:24 - Epochs:40**	**70.64183**
10	neurons: 64 - Act: relu - batch_size:32 - Epochs:40	70.40984
6	neurons: 64 - Act: softplus - batch_size:24 - Epochs:40	70.38702
13	neurons: 64 - Act: relu - batch_size:64 - Epochs:40	70.38487
3	neurons: 64 - Act: relu - batch_size:18 - Epochs:40	70.37013
2	neurons: 128 - Act: relu -batch_size:18 - Epochs:40	70.31201
1	neurons: 64 - Act: softplus -batch_size:32 - Epochs:40	70.19577
17	neurons: 64 - Act: relu -batch_size:24 - Epochs:40	70.18829

It can be seen that the activation relu is appearing the most in the top combinations, thus it can be concluded that "RELU" is the most appropriate activation function for the current dataset. The increase in the number of epochs using a different number of neurons resulted in no felt change in the MAPE that remained equal to 29.35 in terms of significant figures. Another attempt to increase the accuracy of the model was to increase the number of hidden layers and observe the change in MAPE.

After conducting **1050** combinations these are the current combinations for the next trial.

- The number of neurons (nodes) in each layer: m*96
- Number of hidden layers = **unknown**
- The gradient of neurons increments throughout the network: converging
- Activation functions: “relu”
- Optimizing Functions: Rmsprop
- Epochs: 10
- Batch size: 24

In total **6** trials were conducted varying the number of hidden layers from 1 to 6. As for the gradient it was set to be in a descending order i.e. for 4 layers the input would have 5n, hiddenlayer1 will have 4n and the pattern would continue until the output layer with an n equal to 1. The results are given on the next page

Table 6.14. Best ANN models (Number of hidden layers

Trial	Number of hidden layers	n in the Input layer	n in the 1st hidden layer	n in the 2nd hidden layer	n in the 3rd hidden layer	n in the 4th hidden layer	n in the 5th hidden layer	n in the 6th hidden layer	Accuracy
1	1	2n	n	**X**	**X**	**X**	**X**	**X**	55.8
2	2	3n	2n	n	**X**	**X**	**X**	**X**	56.94
3	3	4n	3n	2n	n	**X**	**X**	**X**	57.13
4	4	5n	4n	3n	2n	n	**X**	**X**	57.7
5	**5**	**6n**	**5n**	**4n**	**3n**	**2n**	**n**	**X**	**59.20**
6	6	7n	6n	5n	4n	3n	2n	n	59.16

It can be seen that the accuracy increases as the number of layers increases. After the last combinations are made, the number of hidden layers will be chosen as 5 to decrease the complexity since no huge difference in terms of accuracy was seen.

Next, the number of neurons in each layer will be experimented with, these experiments will be multiples of the neuron configuration presented in table 6.14 for trial 5. The current combination will be as follows:

- The number of neurons (nodes) in each layer: m*96
- Number of hidden layers: 5
- The gradient of neurons increments throughout the network: converging
- Activation functions: “relu”
- Optimizing Functions: Rmsprop
- Epochs: 40
- Batch size: 24

Table 6.15. ANN models (number of neurons within a layer)

Trial	multiples	Accuracy
1	2	66.6
2	4	64.4
3	6	58.4

Even though the number of the layers was increased together with the number of neurons, positive results were not achieved, It was seen unnecessary to check for the diverging and linear variations for the gradient since the accuracy of all the combinations didn’t break the 71 percent accuracy mark for the validation set. the best outcome out of all the **1059** combinations was:

- The number of neurons (nodes) in each layer: 384,192,96,1
- Number of hidden layers: 2
- The gradient of neurons increments throughout the network: converging
- Activation functions: “relu”
- Optimizing Functions: Rmsprop
- Epochs: 40
- Batch size: 24

To denote the final result for this model, these parameters were applied on the testing set followed by plotting the forecast vs actual load on the same plot as explained earlier.

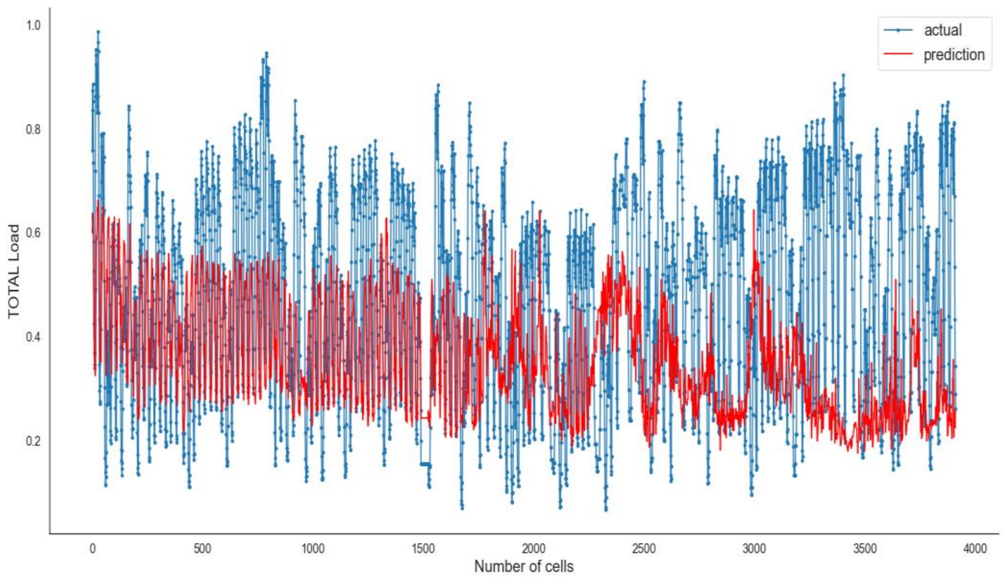


Figure 6.16. Actual vs. ANN forecasted load

The Final ANN model acquired a **MAPE** of **28.5299** and an **MAE** of **11225.68**. which means that according to MAPE the forecast has an accuracy of **71.470 percent** and in the case of MAE, it means that there is an average of **11225.68**. **MWh** error for every prediction.

6.8 Hybrid Model

Up to this point, 4 models have been modified and optimized to be implemented on the specific problem. The final results were :

Table 6.16. Results Summary

Model	MAPE	MAE
ARIMA	14.12	3938.59
LSTM	1.7703	504.756
ε-SVR	12.886	3731.5
ANN	28.5299	11225.68

This thesis will try to combine the top 2 models in terms of MAPE and MAE to create one hybrid model that outperforms both models and decrease the gap between the proposed method and the already applied methodology in the literature.

There are multiple ways to combine models and reach a final ensemble (hybrid model) two of the main methods will be used in this thesis [94]:

- Cascaded methods
- Boosting methods

6.8.1 Cascaded methods

As the word cascaded is defined in a dictionary it is to “pass (something) on to a succession of others” [95]. For using such a method, the application would be feeding an output of model A as an input to model B. In that case model B will operate on optimizing the output even further. In this thesis, the best predictions resulted from the model LSTM (model A), in an attempt to further get more accurate forecasts, the best output obtained will be fed as input to the second-best model which is ε-SVR. To ensure that this resultant hybrid model is working at the maximum efficiency, again the hyperparameter tunings that were mentioned in section 6.6 were applied resulting in 540 different variations, in other words, 540 different hybrid models. A sample of these models is tabulated in table 6 in appendix B. The best combination for the cascade method is as follows:

- Model A [LSTM]:
  - **Number of Modules:100**
  - **Look back:11**
  - **Batch size:10**
  - **Optimizer: “Adam”.**

- Model B [ε-SVR]:
  - **Type of Kernel: RBF**
  - **C: 2**
  - ***ε:* 0.01**

The prior parameters were then applied to the testing portion of the dataset, followed by plotting the actual vs hybrid forecasted load in one graph. The resultant figure is demonstrated below.

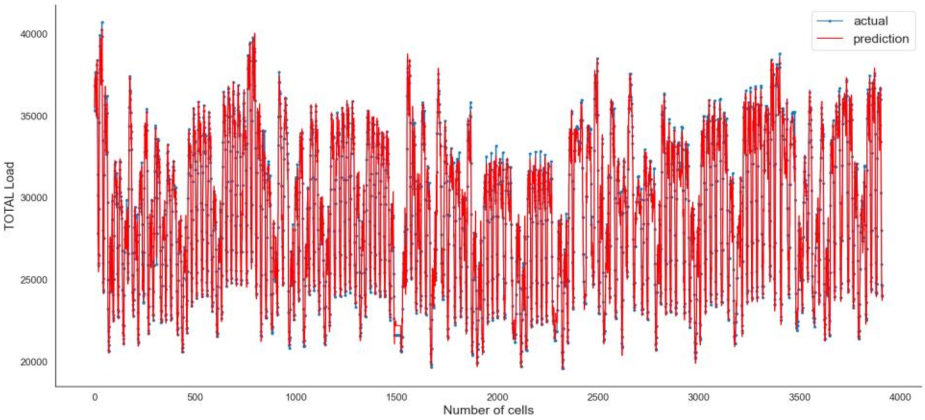


Figure 6.17. Actual vs Hybrid [1] Forecasted Load

The resultant Hybrid model outperformed its formers with a **MAPE** of **1.71157** rather than the MAPEs for LSTM and ε-SVR. that were equal to 1.7703 and 12.886 respectively. The hybrid model achieved an **MAE** of **488.2369**. which also outperformed its formers that had MAEs of 504.756 and 3731.5. These outcomes can be interpreted as follows; the resultant hybrid model that was formed with the cascade method between LSTM and ε-SVR achieved an accuracy of **98.28 percent** and in the case of MAE, it means that there is an average of **488.2369 MWh** error for every prediction.

6.8.2 Boosting methods

Using this method, weighted coefficients are to be added to the forecasts resulting from the two top models to form one final forecast. These weights serve as ratios resembling the importance of each model's prediction while contributing to a new final. The formulation for this algorithm is equated below:

$$\widehat{Y_t} = w_i * \widehat{y_i} + w_j * \widehat{y_j} \qquad \text{(Eq. 6.7)}$$

Where $\widehat{Y_t}$ is the new hybrid model's forecast, w_i and w_j are the weight coefficients assigned for models i and j respectively, and $\widehat{y_i}$ and $\widehat{y_j}$ are the original forecasts resulting from the models. The weight coefficients assigned in that case abide by the following restrictions

$$0 < w_i , w_j < 1$$

$$w_j = 1 - w_i \qquad \text{(Eq. 6.8)}$$

With the aim of finding the ideal combination of weight coefficients for the current dataset, the initial weight coefficient w_i was assigned as 0.01 and constantly increased with increments of 0.01 until it reached its limit of 1, resulting in a total of 100 combinations; meaning 100 different hybrid models using the boosting method, for the validation dataset. A sample of the combinations is tabulated in table 7 in appendix B. The best combination in terms of MAPE emerged from the following weights:

Table 6.17. Best weight coefficients for Hybrid model (2)

Trial Number	Modle (I) LSTM	Model (J) ε-SVR	Mape
	w_i	w_j	
100	1	0	1.7703

The best combination indicates that using LSTM alone would be the optimum solution, concluding that the boosting method is not an appropriate way of forming a hybrid model for the current dataset.

6.9 Discussion

As a summary, a total of 2405 different combinations of 6 different models namely, ARIMA, LSTM, ε-SVR, ANN, Hybrid Model(1), and Hybrid Model(2).were explored. This thesis concluded its analysis with the following model parameters that were considered to be the best fit out of the different combinations tried.

- ➢ ARIMA:
  - p: 3
  - q: 2
  - d: 0

- LSTM:
    - Number of Modules:100
    - Look back:11
    - Batch size:10
    - Optimizer: “Adam”.

- ε-SVR:
  - Type of Kernel: Linear
  - C: 4
  - *ε:* 0.05

- ANN:
  - The number of neurons (nodes) in each layer: 384,192,96,1
  - Number of hidden layers: 2
  - The gradient of neurons increments throughout the network: converging
  - Activation functions: “relu”
  - Optimizing Functions: Rmsprop
  - Epochs: 40
  - Batch size: 24

- Hybrid Model (1):

    - Model A [LSTM]:
        - Number of Modules:100
        - Look back:11
        - Batch size:10
        - Optimizer: “Adam”.

- ❖ Model B [ε-SVR]:
    - Type of Kernel: RBF
    - C: 2
    - *ε:* 0.01

- ➢ Hybrid Model (2):

    - ❖ Model i [LSTM]:
        - Number of Modules:100
        - Look back:11
        - Batch size:10
        - Optimizer: “Adam”.

    - ❖ Model j [ε-SVR]:
        - Type of Kernel: Linear
        - C: 4
        - *ε:* 0.05
    - ❖ w_i : 1
    - ❖ w_j : 0

Unfortunately, the models proposed in this thesis, couldn’t outperform the ENTSO-E’s prediction, however, promising results are awaiting possible future work. In a real-life application, for predicting Spain's electricity demand, multiples of inputs can be considered which can vary in type from qualitative and quantitative. The more inputs available, the more aware the models can become. Yet, this thesis only utilized quantitative data which mainly focused on weather.

Several assumptions were made along the way to ease the problem since no high computational power is available on hand. Such assumptions were the fact that only the 5 major cities’ weather conditions were used to predict all of Spain’s demand, but in reality, if weather conditions were to be used, all the cities' weather features will be fed as inputs.

Focusing on each model at a time, the probability of the ARIMA model to achieve better predictions is minimal since as mentioned earlier in previous sections, ARIMA works based on linear correlations which is not the case for the current data.

LSTM was used as a univariate model and even though a total of 165 different combinations were explored, the possibility of other untested combinations having better predictions is still very high. If more hyperparameters were checked in different ranges, a more optimum result is not far-fetched. Moreover, LSTM can be used as a multivariate model which can potentially achieve higher forecasts.

In ε-SVR, a colossal assumption was made to ease the computational complexity which is the addition of weighted coefficients based on city population to decrease the number of variables, which as a result decreased the number of inputs that could in return provide better learning for the model. Another possible alteration in the model's usage is the utilization of n number of inputs at a time, in the ε-SVR methodology all of the inputs were implemented within the model at the same instance but there can even be a possibility of more accurate results for the weighted inputs if only the significant inputs were fed rather than all. Same as the LSTM, there was a total of 540 discrete combinations used for the hyperparameters within the model, yet there are infinitely many more that could outperform the small sample explored within the thesis's scope.

ANN is known to provide accurate and reliable results when it comes to forecasting, in spite of that, a maximum of 71 percent accuracy was reached and this comes back to the previously stated reasons, which include computational power, stated assumptions, and more trials.

In the case of the Hybrid Model (1), its performance is 100 percent dependent on how well do the optimum models perform together with the tuning range that is chosen. Also, this method can either be applied between two other models or to all of them. There are numerous ways that this method can be implemented.

Finally, for the Hybrid model (2), it was concluded that the way the weights were assigned in the "Boosting method" doesn't yield any promising results. Nevertheless, logical statements can be added as conditions that may correspond to different results. For instance, the holidays or weekends can be grouped and the models' performance towards these data points would be checked, and consequently, rather than applying general weights between models throughout the dataset, constraints can be added. For example, all data points other than the weekends and holidays are to be predicted using model i whereas weekends and holidays are predicted by utilizing model j.

Weather seasonal variations can also be an important logical statement that should be explored in future results and if conclusions could be drawn, Hybrid models that include different working models for each season (winter, summer, spring, and Autumn) can contribute to a felt increase in overall accuracy.

Another crucial contribution that this thesis has achieved is the general usage of the LSTM model in electricity demand short-term prediction. There are no frequent papers that used LSTM in their methodology and LSTM is more seen in other research domains rather than predictions for electricity markets. With the obtained results, more emphasis can be put towards r-the usage of this model.

It can be simply deduced that the methodology applied in this thesis with all of its assumptions that were needed to be applied due to the lack of computational power, was able to achieve a **98.28** percent accuracy with a difference of **0.832** percent to a model that incorporates all the possible scenarios. Also, In a real-life application, all the previously gatherer historical data was used to predict the last quarter of 2018, and in this thesis, 3 years' worth of data points were only used.

All of the mentioned points above will be considered as future optimizations to the current methodology with the goal of achieving higher accuracy than **99.2** percent.

CHAPTER 7

CONCLUSION

Electric demand forecasting approaches with high accuracy and efficiency have drawn increasing interest from experts and enterprises in recent years. Effective forecasting methods can save a lot of time and money, as well as mitigate a variety of hazards. Furthermore, prediction accuracy has not been ideal, and existing singular approaches for forecasting different types of data are imprecise. Single techniques can only deal with either linearity or non-linearity data; they can't handle both. Some combined (hybrid) procedures can enhance accuracy slightly, but not significantly. As a result, developing methods for projecting electricity consumption with greater precision is extremely desirable.

This thesis has combined LSTM and ε-Support Vector Regression to form one unified hybrid model for predicting electricity demand in Spain during the last quarter of 2018. This model was fed with weather variables from the five largest cities in terms of population and was able to outperform ANN, ARIMA, and both of the combined models separately. The model achieved an accuracy of **98.28** percent, yet wasn't able to break past the already used method in the literature that had acquired an accuracy of **99.122** percent.

Many assumptions were applied in the application to overcome the computational duration which as a result decreased the amount of information that is fed to the model. Nevertheless, promising results are awaiting the current methodology if presented with more data and stronger computational power.

REFERENCES

[1] L. Y. Xiao, J. Z. Wang, R. Hou, and J. Wu, “A combined model based on data pre-analysis and weight coefficients optimization for electrical load forecasting,” Energy, vol. 82, pp. 524–549, 2015.

[2] A. Goia, C. May, and G. Fusai, “Functional clustering and linear regression for peak load forecasting,” Int. J. Forecast., vol. 26, no. 4, pp. 700–711, 2010.

[3] E. Erdogdu, “Electricity demand analysis using cointegration and ARIMA modelling: A case study of Turkey,” Energy Policy, vol. 35, no. 2, pp. 1129–1146, 2007.

[4] S. S. Panesar and W. Wang, “Electricity demand forecasting using neural networks,” in Intelligent Data Engineering and Automated Learning, Berlin, Heidelberg: Springer Berlin Heidelberg, 2003, pp. 826–834.

[5] A. P. Plumb, R. C. Rowe, P. York, and M. Brown, “Optimisation of the predictive ability of artificial neural network (ANN) models: a comparison of three ANN programs and four classes of training algorithm,” Eur. J. Pharm. Sci., vol. 25, no. 4–5, pp. 395–405, 2005.

[6] H. Son and C. Kim, “Forecasting short-term electricity demand in residential sector based on support vector regression and fuzzy-rough feature selection with particle swarm optimization,” Procedia Eng., vol. 118, pp. 1162–1168, 2015.

[7] Y. Yang, Y. Chen, Y. Wang, C. Li, and L. Li, “Modelling a combined method based on ANFIS and neural network improved by DE algorithm: A case study for short-term electricity demand forecasting,” Appl. Soft Comput., vol. 49, pp. 663–675, 2016.

[8] L. Meeus, The evolution of electricity markets in Europe. Cheltenham, England: Edward Elgar Publishing, 2020.

[9] M. Greer, “U.S. electric markets, structure, and regulations,” in Electricity Marginal Cost Pricing, M. Greer, Ed. Oxford, England: Elsevier, 2012, pp. 39–100.

[10] A. A. Sadiq, “A Comparative Study on Implementation of Genetic Algorithm (GA) and ATC to Generator Siting in Nigerian 330KV Power Network,” Aug-2023. .

[11] “Electricity small use customers,” Com.au. [Online]. Available: https://www.erawa.com.au/electricity/electricity-licensing/code-of-conduct-for-the-supply-of-electricity-to-small-use-customers/electricity-small-use-customers. [Accessed: 22-Nov-2021].

[12] A. Creti and F. Fontini, Economics of electricity: Markets, competition and rules. Cambridge, England: Cambridge University Press, 2019.

[13] D. R. Biggar and M. Hesamzadeh, The economics of electricity markets. Nashville, TN: John Wiley & Sons, 2014.

[14] A. G. Ter-Gazarian, Energy storage for power systems. Stevenage, England: Institution of Engineering and Technology, 1994.

[15] A. by D. C. D. George Yarrow, “Bidding in energy-only wholesale electricity markets,” Gov.au, Nov-2014. [Online]. Available: https://www.aemc.gov.au/sites/default/files/content/c196404a-e850-46bd-8ae2-41600f8454bb/Professor-George-Yarrow-and-Dr-Chris-Decker-%28RPI%29-Bidding-in-energy-only-wholesale-electricity-markets-Final-report.PDF. [Accessed: 22-Oct-2021].

[16] H. Daneshi, M. Shahidehpour, and A. L. Choobbari, “Long-term load forecasting in electricity market,” in 2008 IEEE International Conference on Electro/Information Technology, 2008, pp. 395–400.

[17] G. R. T. Esteves, B. Q. Bastos, F. L. Cyrino, R. F. Calili, and R. C. Souza, “Long term electricity forecast: A systematic review,” Procedia Comput. Sci., vol. 55, pp. 549–558, 2015.

[18] E. J. Anderson and X. Hu, “Forward contracts and market power in an electricity market,” Int. J. Ind. Organ., vol. 26, no. 3, pp. 679–694, 2008.

[19] G. Rothwell and T. Gomez, Electricity economics: Regulation and deregulation. Nashville, TN: John Wiley & Sons, 2003.

[20] ISO New England, “Day-ahead and real-time energy markets,” Iso-ne.com. [Online]. Available: https://www.iso-ne.com/markets-operations/markets/da-rt-energy-markets/. [Accessed: 22-Nov-2021].

[21] Michał, “Imbalance Settlement Period (electricity balancing market) - emissions-EUETS.Com,” Emissions-euets.com, 03-May-2014. [Online]. Available: https://www.emissions-euets.com/internal-electricity-market-glossary/603-imbalance-settlement-period. [Accessed: 22-Nov-2021].

[22] A. O’Neil, “Energy price variation in the domestic energy market,” Gov.uk, Jun-2014. .

[23] ISO New England, “How resources are selected and prices are set in the wholesale energy markets,” Iso-ne.com. [Online]. Available: https://www.iso-ne.com/about/what-we-do/in-depth/how-resources-are-selected-and-prices-are-set. [Accessed: 22-Nov-2021].

[24] Department of Energy and Mineral Engineering, “Basic economics of power generation, transmission and distribution,” Psu.edu. [Online]. Available: https://www.e-education.psu.edu/eme801/node/530. [Accessed: 22-Nov-2021].

[25] “Baseload power,” Energyeducation.ca. [Online]. Available: https://energyeducation.ca/encyclopedia/Baseload_power. [Accessed: 22-Nov-2021].

[26] J. Torriti, Peak energy demand and demand side response. London, England: Routledge, 2017.

[27] A. Nordin, M. Amin, and A. Majid, “Analysis of carbon dioxide emission of gas fuelled cogeneration plant,” IOP Conf. Ser. Mater. Sci. Eng., vol. 50, p. 012054, 2013.

[28] H. Ritchie and M. Roser, “CO_2 and greenhouse gas emissions,” Our World in Data, 2020.

[29] J. Sharp, “Comparative models for electrical load forecasting: D.H. Bunn and E.D. Farmer, eds.(Wiley, New York, 1985) [UK pound]24.95, pp. 232,” Int. J. Forecast., vol. 2, no. 2, pp. 241–242, 1986.

[30] L. Suganthi and A. A. Samuel, “Energy models for demand forecasting—A review,” Renew. Sustain. Energy Rev., vol. 16, no. 2, pp. 1223–1240, 2012.

[31] N. Oskolkov, “Univariate vs. Multivariate Prediction,” Towards Data Science, 06-Jan-2021. [Online]. Available: https://towardsdatascience.com/univariate-vs-multivariate-prediction-c1a6fb3e009. [Accessed: 22-Nov-2021].

[32] D. J. Denis, SPSS data analysis for univariate, bivariate, and multivariate statistics. Hoboken, NJ, USA: John Wiley & Sons, Inc., 2018.

[33] Z. H. Munim, M. H. Shakil, and I. Alon, “Next-day Bitcoin price forecast,” J. Risk Fin. Manag., vol. 12, no. 2, p. 103, 2019.

[34] K. Chen and S.-H. Kung, “Synthesis of qualitative and quantitative approaches to long-range forecasting,” Technol. Forecast. Soc. Change, vol. 26, no. 3, pp. 255–266, 1984.

[35] Australian Bureau of Statistics, “Statistical language - quantitative and qualitative data.” [Online]. Available: https://www.abs.gov.au/websitedbs/D3310114.nsf/Home/Statistical+Language+-+quantitative+and+qualitative+data. [Accessed: 24-Oct-2021].

[36] L. de Mesnard, “A Note on Qualitative Input–Output Analysis,” Econ. Syst. Res., vol. 7, no. 4, pp. 439–448, 1995.

[37] M. I. Fon, “Disposal of E-waste. Case study: University of Bamenda.” Unpublished, 07-Aug-2018.

[38] G. Napier, “Chapter 2 Modelling trends and seasonal patterns,” Bookdown.org, 08-Sep-2020. [Online]. Available: https://bookdown.org/gary_a_napier/time_series_lecture_notes/ChapterTwo.html. [Accessed: 22-Nov-2021].

[39] D. F. Groebner, P. W. Shannon, and P. C. Fry, Business Statistics: International Edition, 9th ed. Upper Saddle River, NJ: Pearson, 2013.

[40] R. Glowinski, “Lectures on Numerical Methods For Non-Linear Variational Problems,” Res.in, 1980. [Online]. Available: http://www.math.tifr.res.in/~publ/ln/tifr65.pdf. [Accessed: 10-Nov-2021].

[41] G. M. Mimmack, D. H. Meyer, and G. J. Manas, Introductory statistics for business: the analysis of business data, 1st ed. Person Education, 2001.

[42] J. Hegde and B. Rokseth, “Applications of machine learning methods for engineering risk assessment – A review,” Saf. Sci., vol. 122, no. 104492, p. 104492, 2020.

[43] E. Stoimenova, K. Prodanova, R. Prodanova, and M. D. Todorov, “Forecasting electricity demand by time series models,” in AIP Conference Proceedings, 2007.

[44] T. C. Redman, Data driven: Profiting from your most important business asset. Boston, MA: Harvard Business Review Press, 2008.

[45] W. H. Su and J. Chawalit, “Short-term electricity load forecasting in Thailand: An analysis on different input variables,” IOP Conf. Ser. Earth Environ. Sci., vol. 192, p. 012040, 2018.

[46] S. Gonzales Chavez, J. Xiberta Bernat, and H. Llaneza Coalla, “Forecasting of energy production and consumption in Asturias (northern Spain),” Energy (Oxf.), vol. 24, no. 3, pp. 183–198, 1999.

[47] “1.4 Forecasting data and methods,” Otexts.com. [Online]. Available: https://otexts.com/fpp2/data-methods.html. [Accessed: 22-Nov-2021].

[48] D. W. Boyd, “Systems Modeling Principles,” in Systems Analysis and Modeling, Elsevier, 2001, pp. 35–73.

[49] F. Egelioglu, A. A. Mohamad, and H. Guven, “Economic variables and electricity consumption in Northern Cyprus,” Energy (Oxf.), vol. 26, no. 4, pp. 355–362, 2001.

[50] T. Haida and S. Muto, “Regression based peak load forecasting using a transformation technique,” IEEE Trans. Power Syst., vol. 9, no. 4, pp. 1788–1794, 1994.

[51] D. C. Park, M. A. El-Sharkawi, R. J. Marks, L.E. Atlas, and M. J. Damborg, “Electric load forecasting using an artificial neural network,” IEEE Trans. Power Syst., vol. 6, no. 2, pp. 442–449, 1991.

[52] J. W. Taylor and R. Buizza, “Neural network load forecasting with weather ensemble predictions,” IEEE Trans. Power Syst., vol. 17, no. 3, pp. 626–632, 2002.

[53] H. S. Hippert and J. W. Taylor, “An evaluation of Bayesian techniques for controlling model complexity and selecting inputs in a neural network for short-term load forecasting,” Neural Netw., vol. 23, no. 3, pp. 386–395, 2010.

[54] M. Mohandes, “Support vector machines for short-term electrical load forecasting,” Int. J. Energy Res., vol. 26, no. 4, p. 335, 2002.

[55] M. T. Hagan and S. M. Behr, “The time series approach to short term load forecasting,” IEEE Trans. Power Syst., vol. 2, no. 3, pp. 785–791, 1987.

[56] R. Weron, Modeling and forecasting electricity loads and prices: A statistical approach. Chichester, England: John Wiley & Sons, 2006.

[57] X. Qiu, P. N. Suganthan, and G. A. J. Amaratunga, “Short-term electricity price forecasting with empirical mode decomposition based ensemble kernel machines,” Procedia Comput. Sci., vol. 108, pp. 1308–1317, 2017.

[58] S. Tzafestas and E. Tzafestas, “Computational Intelligence Techniques for Short-Term Electric Load Forecasting,” J. Intell. Robot. Syst., vol. 31, no. 1/3, pp. 7–68, 2001.

[59] W. Charytoniuk, M. S. Chen, and P. Van Olinda, “Nonparametric regression based short-term load forecasting,” IEEE Trans. Power Syst., vol. 13, no. 3, pp. 725–730, 1998.

[60] S. E. Papadakis, J. B. Theocharis, S. J. Kiartzis, and A. G. Bakirtzis, “A novel approach to short-term load forecasting using fuzzy neural networks,” IEEE Trans. Power Syst., vol. 13, no. 2, pp. 480–492, 2016.

[61] A. Abraham and B. Nath, “A neuro-fuzzy approach for modelling electricity demand in Victoria,” arXiv [cs.AI], 2004.

[62] A. A. El Desouky and M. M. El Kateb, “Hybrid adaptive techniques for electric-load forecast using ANN and ARIMA,” IEE Proc. - Gener. Transm. Distrib., vol. 147, no. 4, p. 213, 2000.

[63] A. Kavousi-Fard and F. Kavousi-Fard, “A new hybrid correction method for short-term load forecasting based on ARIMA, SVR and CSA,” J. Exp. Theor. Artif. Intell., vol. 25, no. 4, pp. 559–574, 2013.

[64] “Static Content,” Entsoe.eu. [Online]. Available: https://transparency.entsoe.eu/content/static_content/Static%20content/legacy%20data/legacy%20data2015.html. [Accessed: 28-Nov-2021].

[65] European Network of Transmission System Operators for Electricity, “Demand forecasting methodology,” Entsoe.eu, 28-Aug-2019. [Online]. Available: https://eepublicdownloads.entsoe.eu/clean-documents/sdc-documents/MAF/2020/Demand_forecasting_methodology_V1_1.pdf. [Accessed: 03-Mar-2021].

[66] G. Sousa, “Largest cities in Spain,” Worldatlas.com, 07-Mar-2018. [Online]. Available: https://www.worldatlas.com/articles/the-biggest-cities-in-spain.html. [Accessed: 03-Mar-2021].

[67] “Weather API,” Openweathermap.org. [Online]. Available: https://openweathermap.org/api. [Accessed: 28-Nov-2021].

[68] "Data view," Entsoe.eu. [Online]. Available: https://transparency.entsoe.eu/generation/r2/actualGenerationPerProductionType/show?name=&defaultValue=false&viewType=GRAPH&areaType=BZN&atch=false&datepicker-day-offset-select-dv-date-from_input=D&dateTime.dateTime=29.11.2021%2000:00|CET%7CDAYTIMERANGE&dateTime.endDateTime=29.11.2021%2000:00|CET%7CDAYTIMERANGE&area.values=CTY%7C10YES-REE------0!BZN%7C10YES-REE------0&productionType.values=B01&productionType.values=B02&productionType.values=B03&productionType.values=B04&productionType.values=B05&productionType.values=B06&productionType.values=B07&productionType.values=B08&productionType.values=B09&productionType.values=B10&productionType.values=B11&productionType.values=B12&productionType.values=B13&productionType.values=B14&productionType.values=B20&productionType.values=B15&productionType.values=B16&productionType.values=B17&productionType.values=B18&productionType.values=B19&dateTime.timezone=CET_CEST&dateTime.timezone_input=CET+(UTC+1)+/+CEST+(UTC+2). [Accessed: 30-Nov-2021].

[69] J. Brownlee, Better deep learning: Train faster, reduce overfitting, and make better predictions. Machine Learning Mastery, 2018.

[70] Safari and an O'Reilly Media Company, Training data for machine learning models. O'Reilly Media, 2022.

[71] H. E. Thornton, B. J. Hoskins, and A. A. Scaife, "The role of temperature in the variability and extremes of electricity and gas demand in Great Britain," Environ. Res. Lett., vol. 11, no. 11, p. 114015, 2016.

[72] G. E. P. Box, G. M. Jenkins, and G. C. Reinsel, Time series analysis: Forecasting and control, 4th ed. John Wiley & Sons, 2013.

[73] J. H. Stock and M. W. Watson, Introduction to Econometrics, 4th ed. Upper Saddle River, NJ: Pearson, 2018.

[74] G. Kirchgassner, J. Wolters, and U. Hassler, Introduction to modern time series analysis, 2nd ed. Berlin, Germany: Springer, 2014.

[75] Information Resources Management Association, Ed., Deep learning and neural networks: Concepts, methodologies, tools, and applications. Hershey, PA: Business Science Reference, 2020.

[76] D. Mishra and M. Posts, "Applications of recurrent neural networks (RNNs)," OpenGenus IQ: Computing Expertise & Legacy, 04-Dec-2019. [Online]. Available: https://iq.opengenus.org/applications-of-rnn/. [Accessed: 22-Dec-2021].

[77] J. Schmidhuber, "Untersuchungen zu dynamischen neuronalen Netzen," Idsia.ch, 15-Jun-1991. [Online]. Available: https://people.idsia.ch/~juergen/SeppHochreiter1991ThesisAdvisorSchmidhuber.pdf.

[78] S. Hochreiter and J. Schmidhuber, "Long short-term memory," Neural Comput., vol. 9, no. 8, pp. 1735–1780, 1997.

[79] R. Lee, Ed., Computer and information science, 1st ed. Basel, Switzerland: Springer International Publishing, 2018.

[80] D. Oliva, E. H. Houssein, and S. Hinojosa, Eds., Metaheuristics in Machine Learning: Theory and Applications, 1st ed. Cham, Switzerland: Springer Nature, 2021.

[81] C. Cortes and V. Vapnik, "Support-vector networks," Mach. Learn., vol. 20, no. 3, pp. 273–297, 1995.

[82] A. J. Smola and B. Schölkopf, "A tutorial on support vector regression," Stat. Comput., vol. 14, no. 3, pp. 199–222, 2004.

[83] W. S. McCulloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity," Bulletin of Mathematical Biophysics, vol. 5, no. 4, pp. 115–133, 1943.

[84] H. S. Hippert, C. E. Pedreira, and R. C. Souza, “Neural networks for short-term load forecasting: a review and evaluation,” IEEE Trans. Power Syst., vol. 16, no. 1, pp. 44–55, 2001.

[85] P. J. Braspenning, F. Thuijsman, and A. J. M. M. Weijters, Artificial neural networks: An introduction to ANN theory and practice, 1995th ed. Berlin, Germany: Springer, 1995.

[86] D. Zwillinger, Ed., CRC Standard Mathematical Tables and Formulae. Chapman and Hall/CRC, 2002.

[87] C. Sammut and G. I. Webb, Encyclopedia of machine learning, 2010th ed. New York, NY: Springer, 2010.

[88] H. S. Shryock, J. S. Siegel, E. A. Larmon, and United States. Bureau of the Census, The methods and materials of demography. U.S. Bureau of the Census, 1973.

[89] dimitriosroussis, “Electricity price forecasting with DNNs (+ EDA),” Kaggle.com, 25-Mar-2021. [Online]. Available: https://www.kaggle.com/dimitriosroussis/electricity-price-forecasting-with-dnns-eda. [Accessed: 04-Jan-2022].

[90] Stephanie, “Correlogram / auto correlation function ACF plot: Definition in plain English,” Statistics How To, 23-Aug-2016. [Online]. Available: https://www.statisticshowto.com/correlogram/. [Accessed: 05-Jan-2022].

[91] P. Perron and T. Yabu, “Testing for shifts in trend with an integrated or stationary noise component,” SSRN Electron. J., 2007.

[92] T. Bartz-Beielstein, J. Branke, B. Filipič, and J. Smith, Parallel problem solving from nature -- PPSN XIII: 13Th international conference, Ljubljana, Slovenia, September 13-17,2014, proceedings, 2014th ed. Basel, Switzerland: Springer International Publishing, 2014.

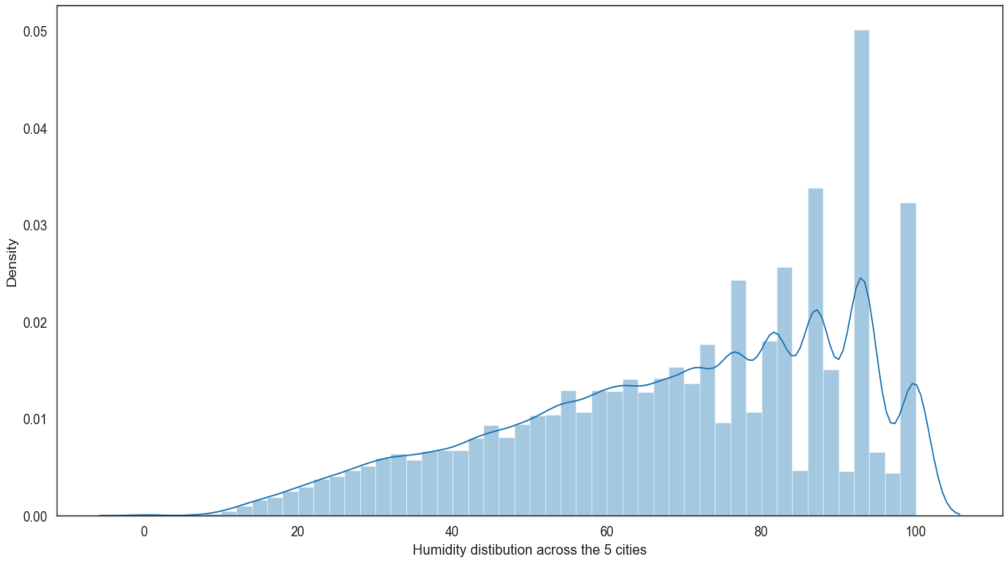
[93] “Spain population 2021 (demographics, maps, graphs),” Worldpopulationreview.com. [Online]. Available: https://worldpopulationreview.com/countries/spain-population. [Accessed: 08-Jan-2022].

[94] J. Brownlee, Ensemble learning algorithms with Python: Make better predictions with bagging, boosting, and stacking. Machine Learning Mastery, 2021.

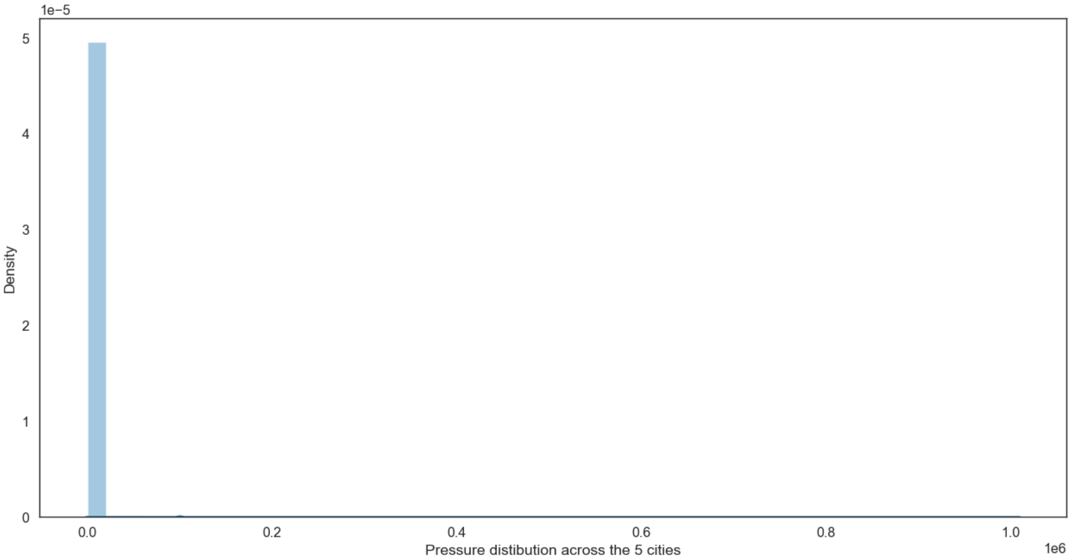
[95] “Cascade,” Cambridge.org. [Online]. Available: https://dictionary.cambridge.org/dictionary/english/cascade. [Accessed: 09-Jan-2022].

APPENDICES

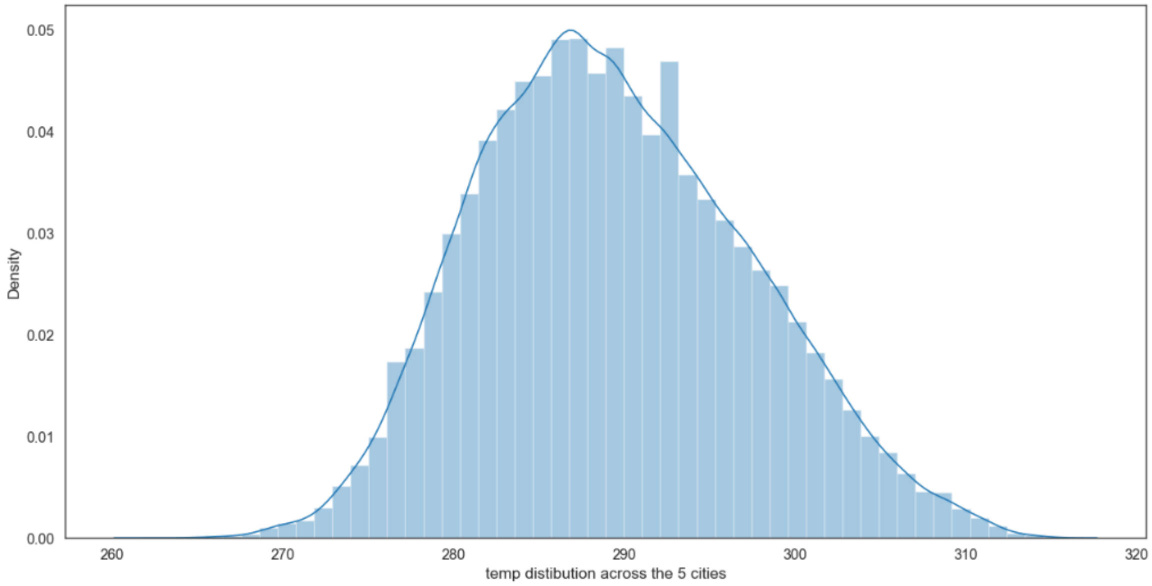
A. Data Pre-processing



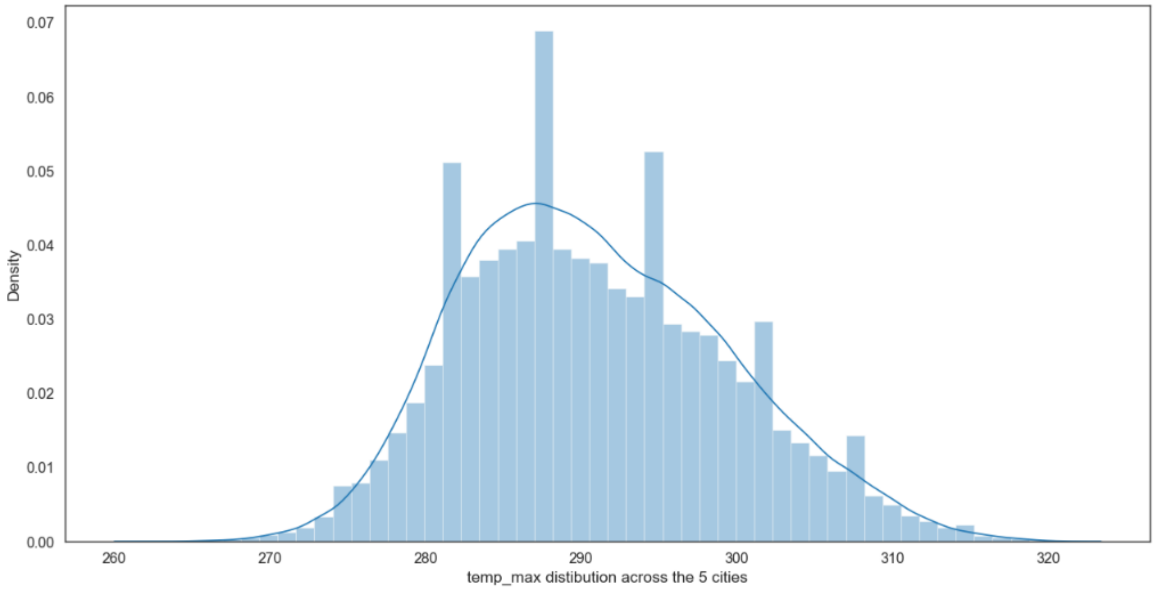
Appendix-A-Fig- 1. Humidity Distribution Plot



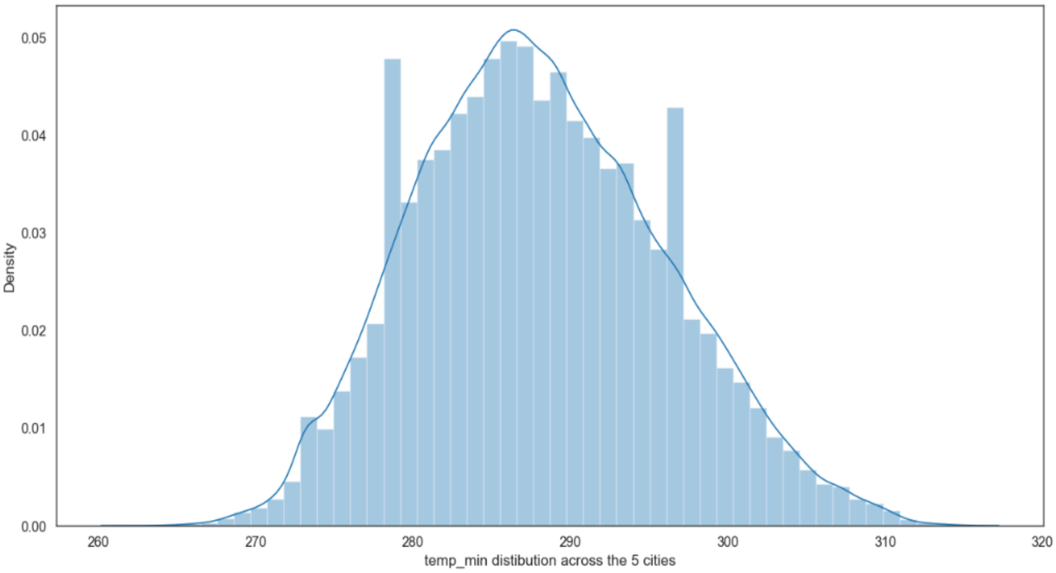
Appendix-A-Fig- 2. Pressure Distribution Plot



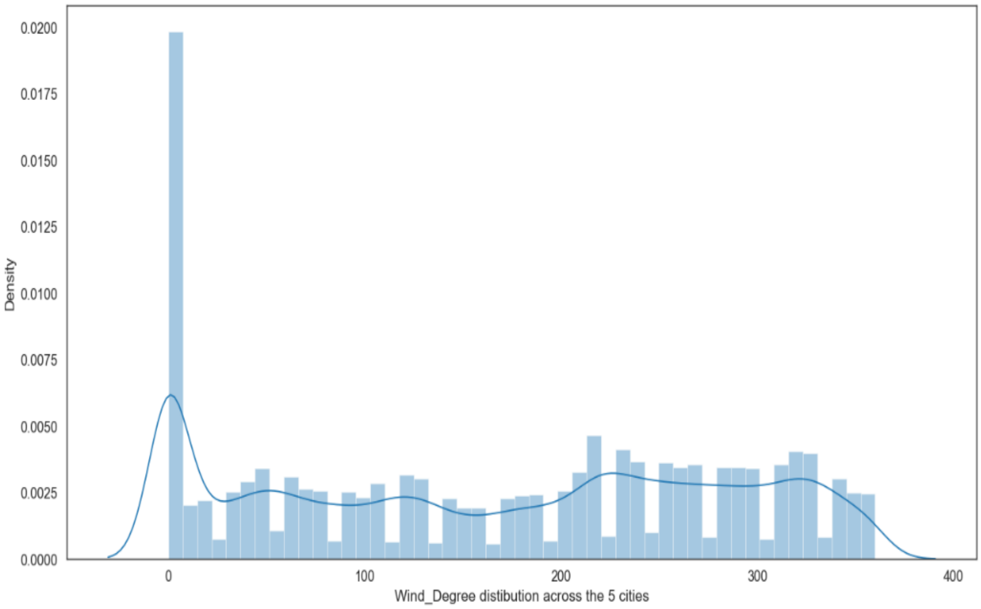
Appendix-A-Fig- 3. Temperature Distribution Plot



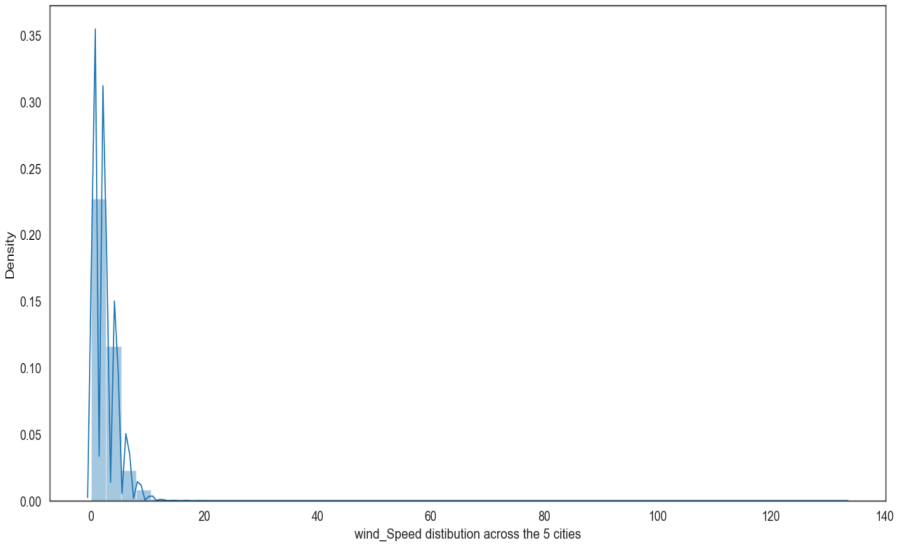
Appendix-A-Fig- 4.Temperature_max Distribution Plot



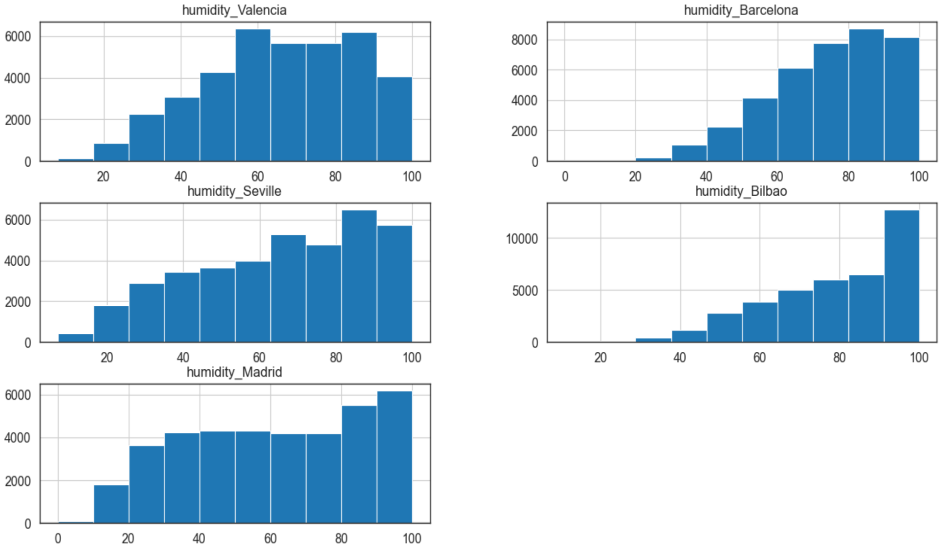
Appendix-A-Fig- 5. Temperature_ min Distribution Plot



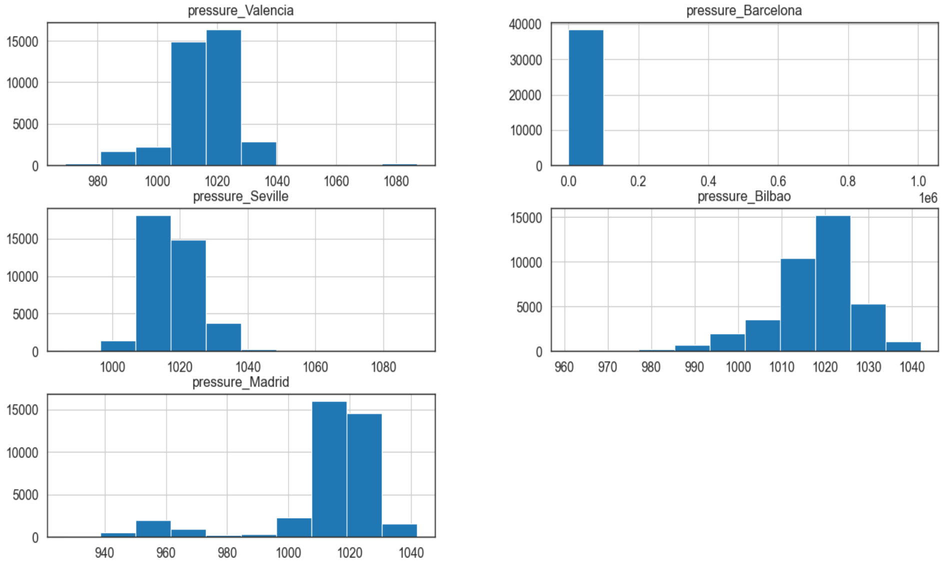
Appendix-A-Fig- 6. Wind Degree Distribution Plot



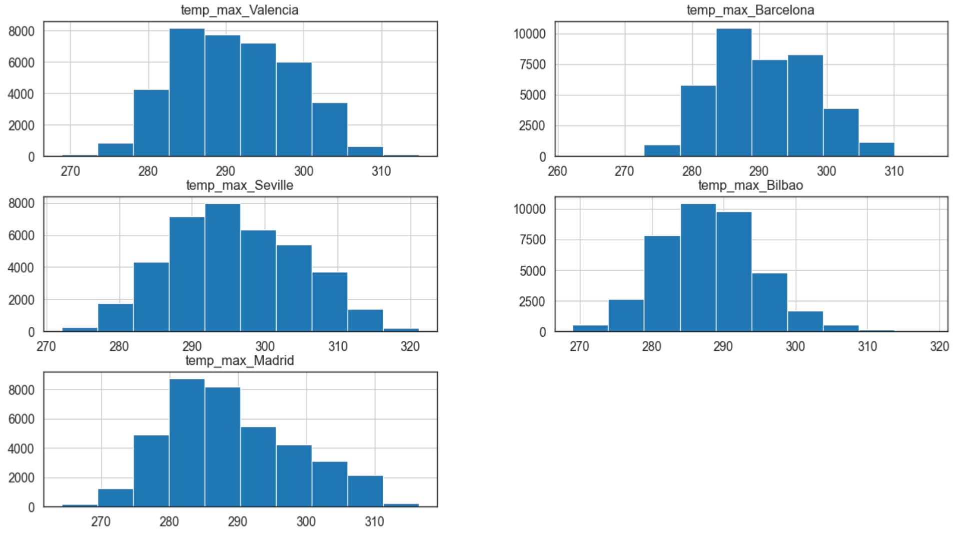
Appendix-A-Fig- 7. Wind Speed Distribution Plot



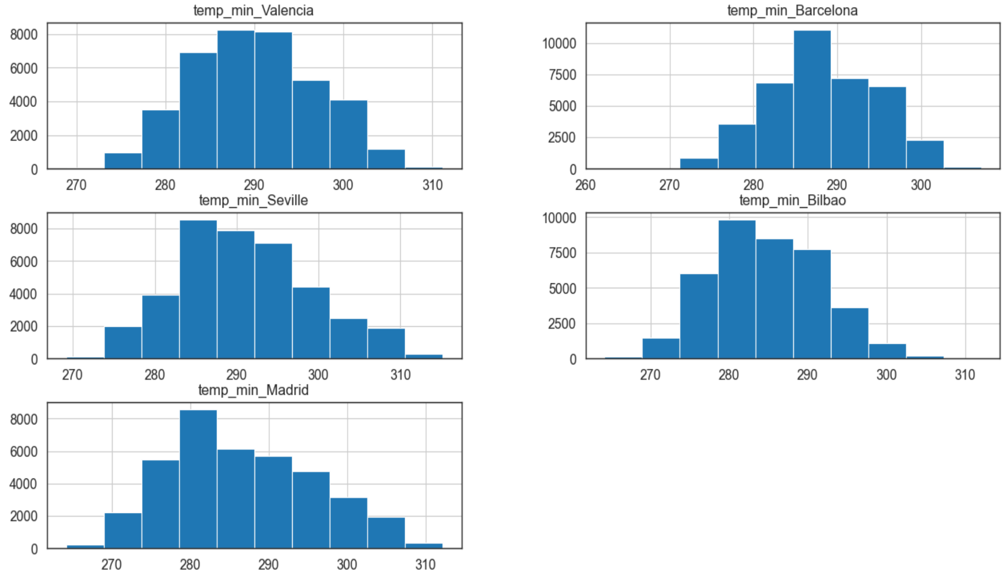
Appendix-A-Fig- 8. Humidity Histogram Plot for each city



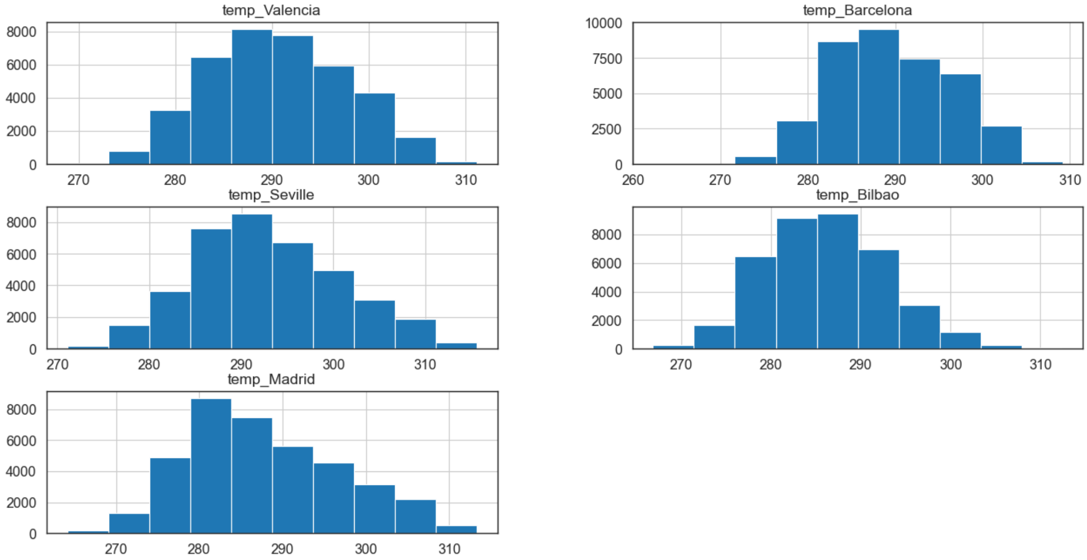
Appendix-A-Fig- 9. Pressure Histogram Plot for each city



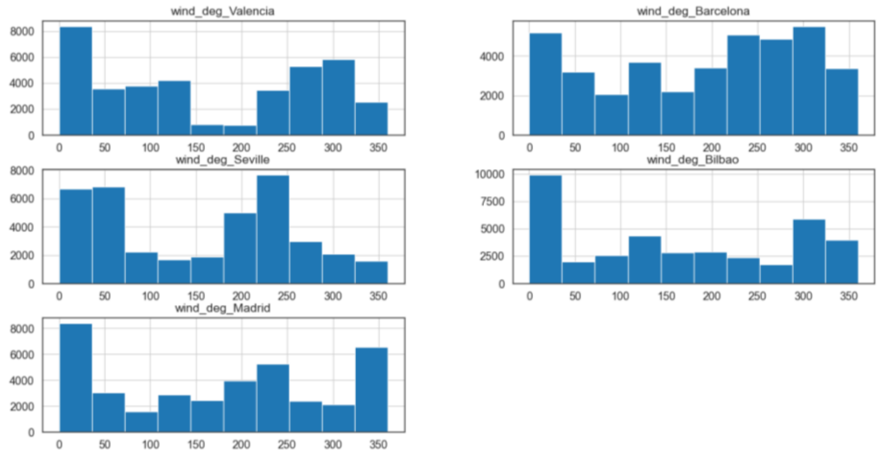
Appendix-A-Fig- 10. Temperature_max Histogram Plot for each city



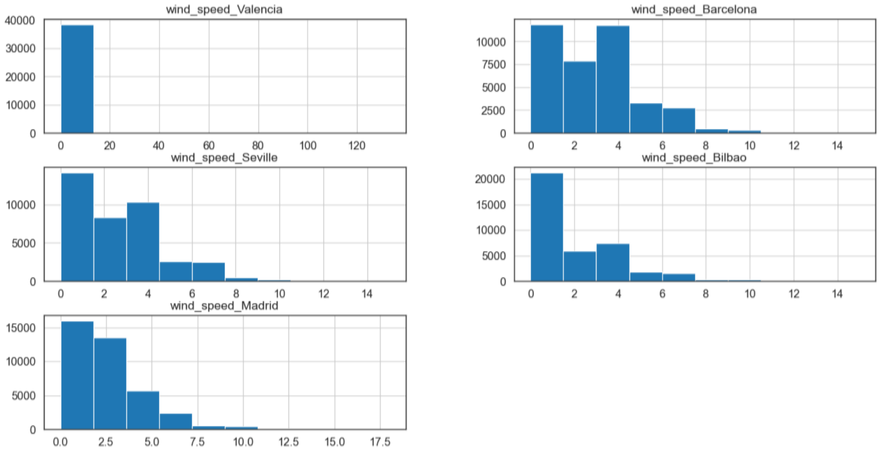
Appendix-A-Fig- 11. Temperature_min Histogram Plot for each city



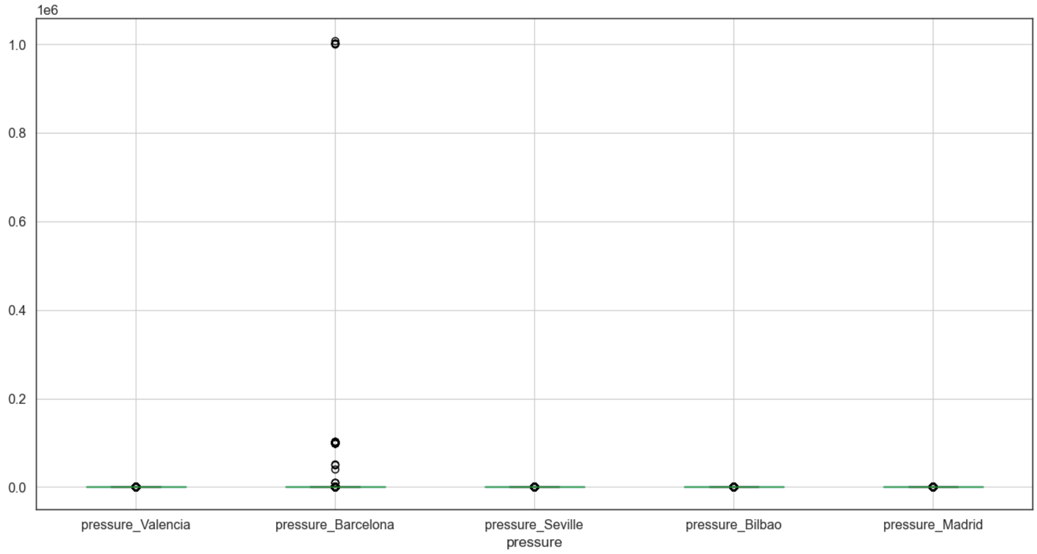
Appendix-A-Fig- 12. Temperature Histogram Plot for each city



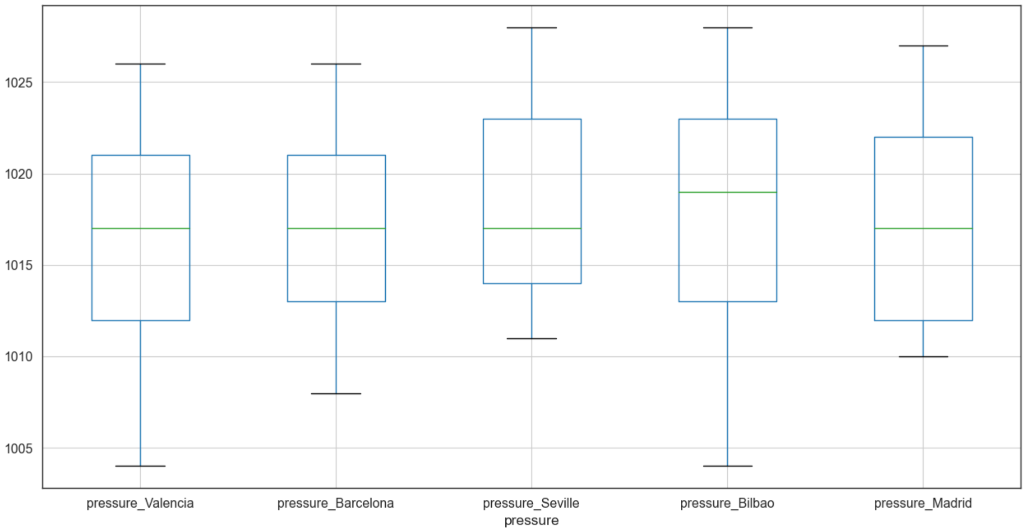
Appendix-A-Fig- 13 Wind Degree Histogram Plot for each city



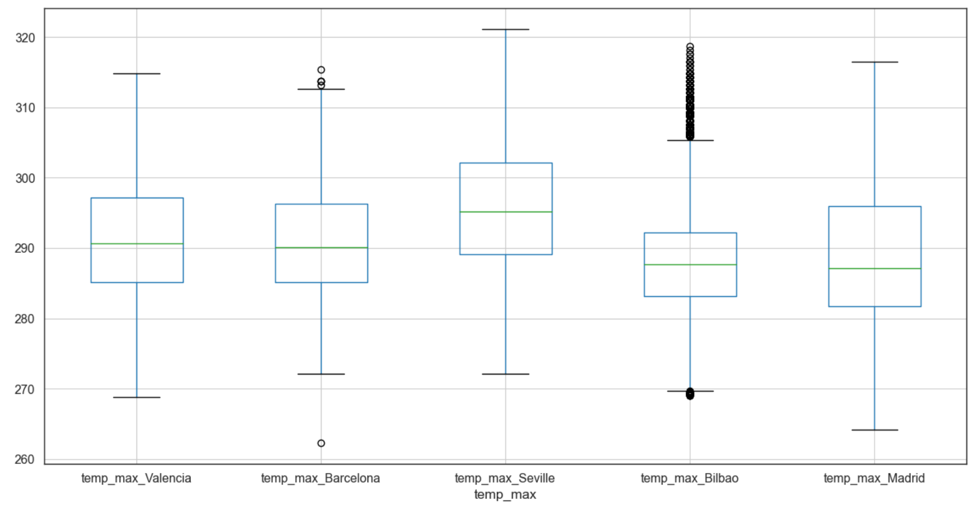
Appendix-A-Fig- 14. Wind Speed Histogram Plot for each city



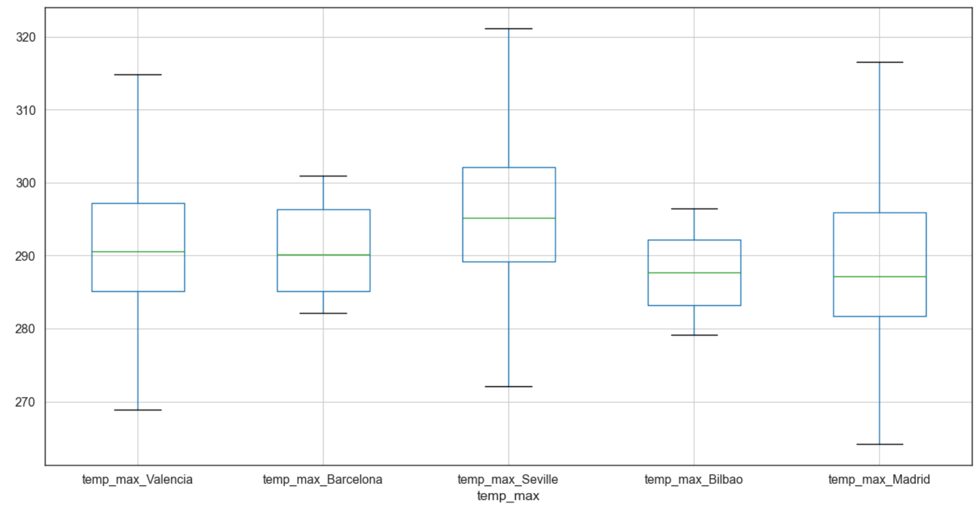
Appendix-A-Fig- 15. Pressure Box Plots for every city before removing outliers.



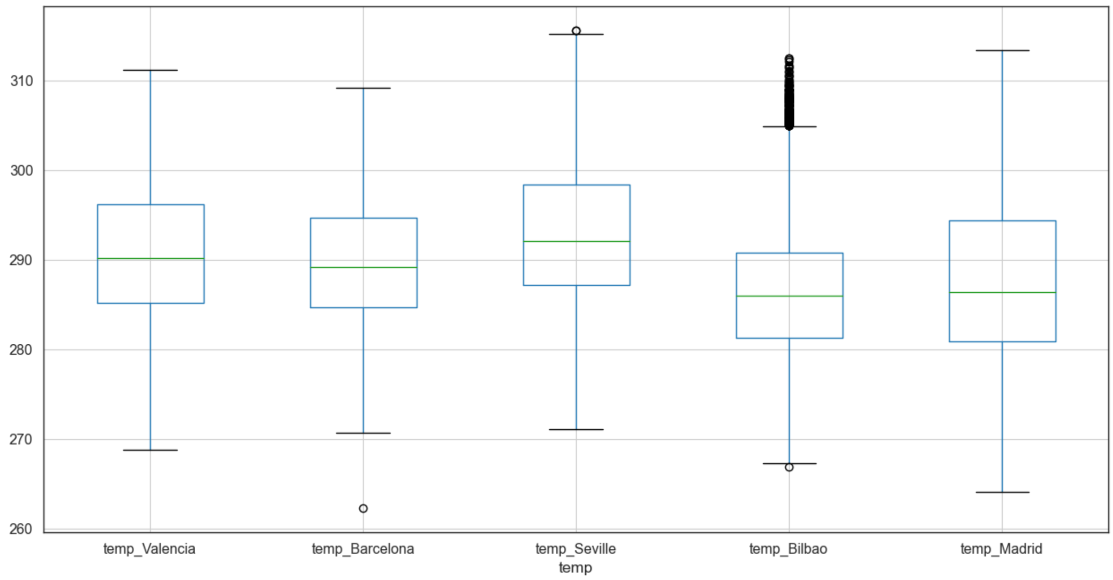
Appendix-A-Fig- 16.Pressure Box Plots for every city after removing outliers.



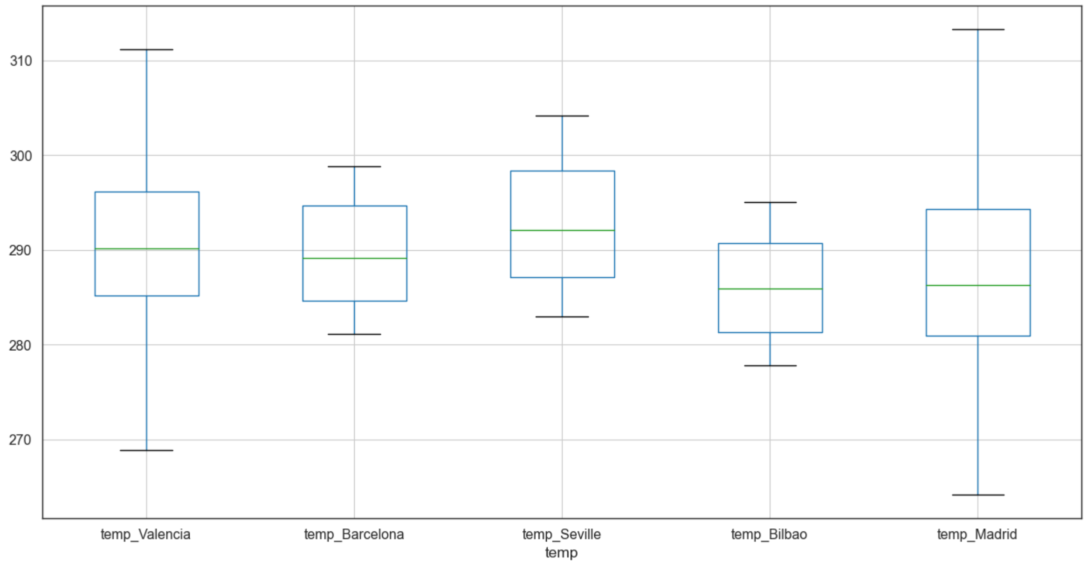
Appendix-A-Fig- 17. Temperature_max Box Plots for every city before removing outliers.



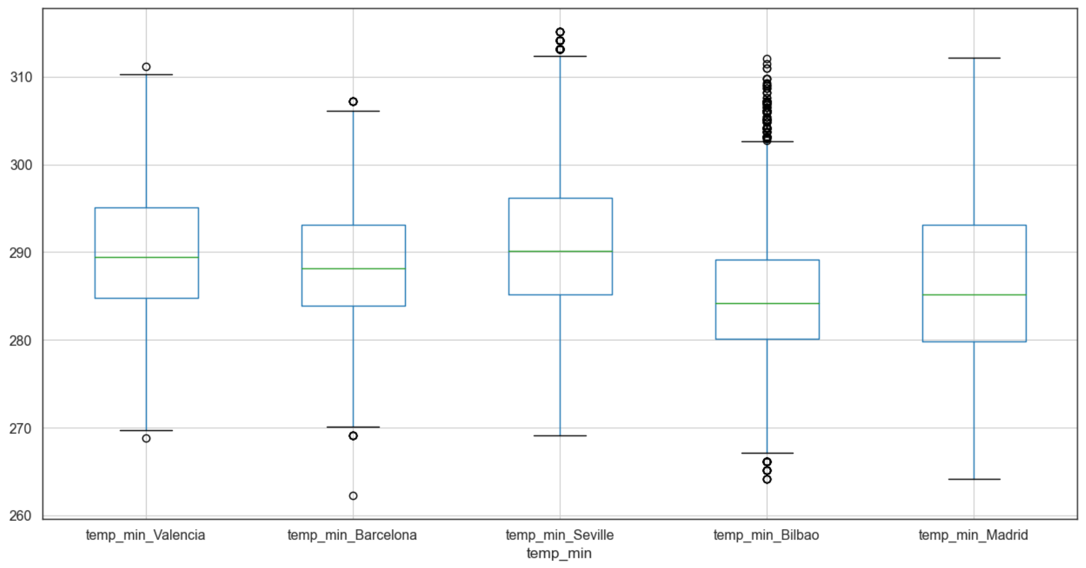
Appendix-A-Fig- 18. Temperature_max Box Plots for every city after removing outliers



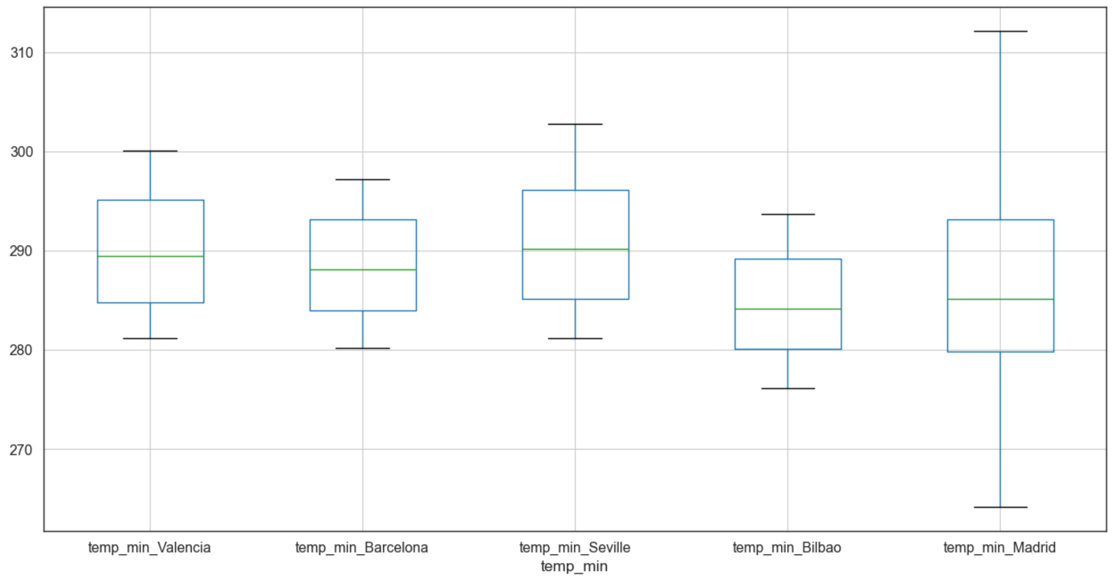
Appendix-A-Fig- 19. Temperature Box Plots for every city before removing outliers.



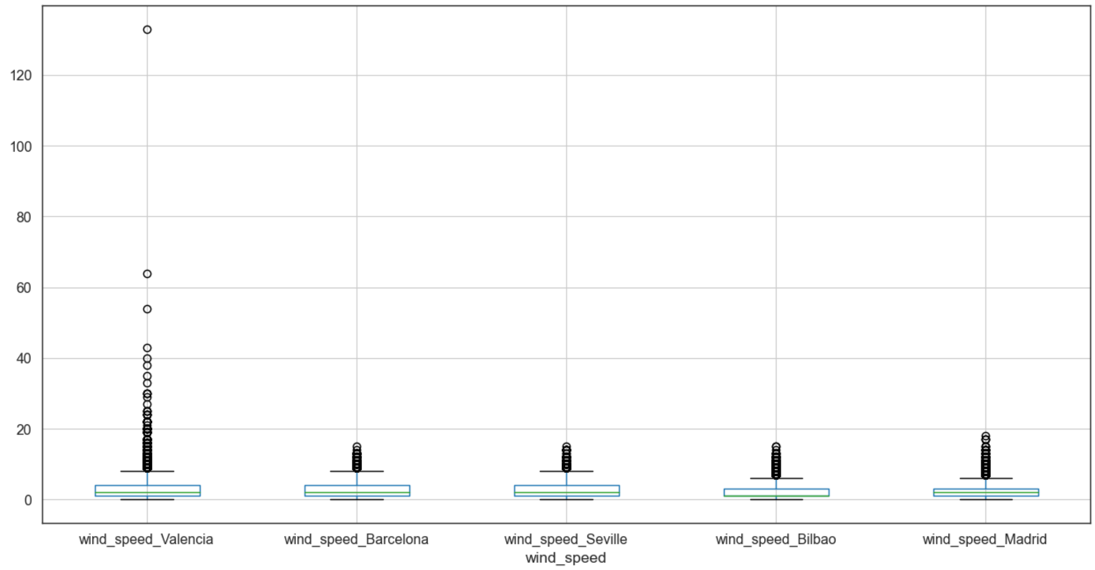
Appendix-A-Fig- 20.Temperature Box Plots for every city after removing outliers.



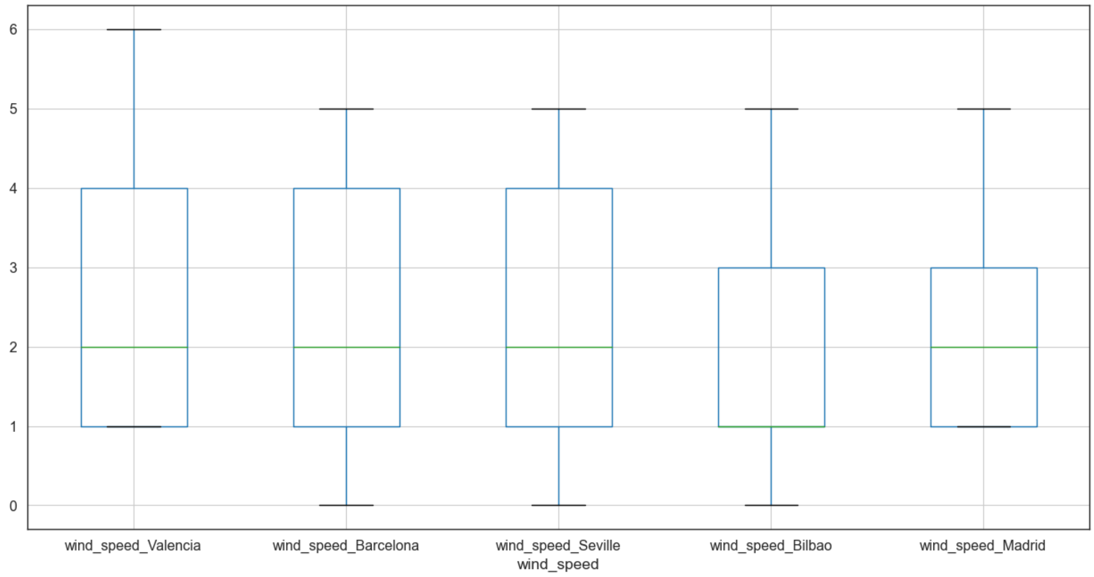
Appendix-A-Fig- 21.Temperature_min Box Plots for every city before removing outliers



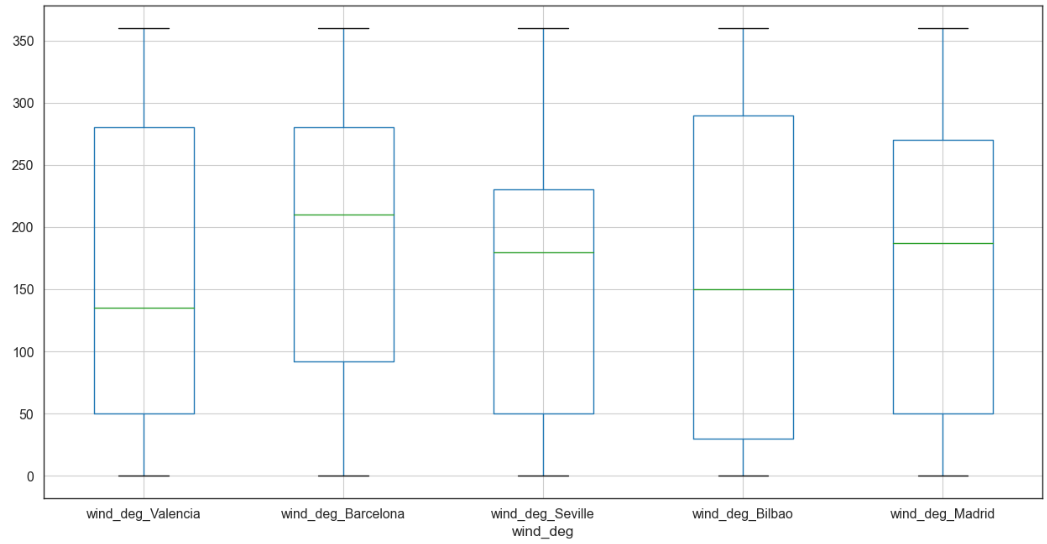
Appendix-A-Fig- 22.Temperature_min Box Plots for every city after removing outliers



Appendix-A-Fig- 23. Wind Speed Box Plots for every city before removing outliers



Appendix-A-Fig- 24.Wind Speed Box Plots for every city after removing outliers



Appendix-A-Fig- 25.Wind Degree Box Plots for every city before removing outliers

B. Hyperparameter Tuning

Appendix-B- Table 1. LSTM Hyper Parameter Tuning

Trial	Parameters	Accuracy
67	**modules: 100 - Lookback: 11 - optimizer:Adam - batch_size:10 - Epochs:10**	**97.80052**
45	modules: 50 - Lookback: 15 - optimizer:Adam - batch_size:10 - Epochs:10	97.78703
59	modules: 80 - Lookback: 11 - optimizer:Adam - batch_size:10 - Epochs:10	97.78447
61	modules: 80 - Lookback: 15 - optimizer:Adam - batch_size:10 - Epochs:10	97.77562
63	modules: 80 - Lookback: 20 - optimizer:Adam - batch_size:10 - Epochs:10	97.77398
51	modules: 60 - Lookback: 11 - optimizer:Adam - batch_size:10 - Epochs:10	97.77158
75	modules: 200 - Lookback: 11 - optimizer:Adam - batch_size:10 - Epochs:10	97.77071
43	modules: 50 - Lookback: 11 - optimizer:Adam - batch_size:10 - Epochs:10	97.76968
92	modules: 60 - Lookback: 11 - optimizer:Nadam - batch_size:12 - Epochs:10	97.76725
66	modules: 100 - Lookback: 10 - optimizer:Nadam - batch_size:10 - Epochs:10	97.76537
106	modules: 100 - Lookback: 10 - optimizer:Nadam - batch_size:12 - Epochs:10	97.76321
35	modules: 200 - Lookback: 11 - optimizer:Adam - batch_size:8 - Epochs:10	97.76248
65	modules: 100 - Lookback: 10 - optimizer:Adam - batch_size:10 - Epochs:10	97.76177
117	modules: 200 - Lookback: 15 - optimizer:Adam - batch_size:12 - Epochs:10	97.75788
73	modules: 200 - Lookback: 10 - optimizer:Adam - batch_size:10 - Epochs:10	97.75772
5	modules: 50 - Lookback: 15 - optimizer:Adam - batch_size:8 - Epochs:10	97.75538
91	modules: 60 - Lookback: 11 - optimizer:Adam - batch_size:12 - Epochs:10	97.75496
25	modules: 100 - Lookback: 10 - optimizer:Adam - batch_size:8 - Epochs:10	97.75252
11	modules: 60 - Lookback: 11 - optimizer:Adam - batch_size:8 - Epochs:10	97.75117
77	modules: 200 - Lookback: 15 - optimizer:Adam - batch_size:10 - Epochs:10	97.75094

Appendix-B- Table 2. SVR Hyperparameter tuning for RBF and Linear Kernels

Trial	Parameters	Accuracy
116	kernel: linear - C values: 4 - Epsilon:0.05	87.02348028
110	kernel: linear - C values: 2 - Epsilon:0.05	87.02335043
92	kernel: linear - C values: 0.4 - Epsilon:0.05	87.02330052
104	kernel: linear - C values: 1 - Epsilon:0.05	87.02329276
98	kernel: linear - C values: 0.8 - Epsilon:0.05	87.02326502
86	kernel: linear - C values: 0.1 - Epsilon:0.05	87.02276336
80	kernel: linear - C values: 0.05 - Epsilon:0.05	87.02276222
109	kernel: linear - C values: 2 - Epsilon:0.01	87.02268905
97	kernel: linear - C values: 0.8 - Epsilon:0.01	87.02267166
91	kernel: linear - C values: 0.4 - Epsilon:0.01	87.0226585
103	kernel: linear - C values: 1 - Epsilon:0.01	87.02253488
115	kernel: linear - C values: 4 - Epsilon:0.01	87.0224246
85	kernel: linear - C values: 0.1 - Epsilon:0.01	87.02147215
79	kernel: linear - C values: 0.05 - Epsilon:0.01	87.02134629
111	kernel: linear - C values: 2 - Epsilon:0.1	87.01809936
87	kernel: linear - C values: 0.1 - Epsilon:0.1	87.017722
117	kernel: linear - C values: 4 - Epsilon:0.1	87.0176689
99	kernel: linear - C values: 0.8 - Epsilon:0.1	87.01762076
93	kernel: linear - C values: 0.4 - Epsilon:0.1	87.01749453
105	kernel: linear - C values: 1 - Epsilon:0.1	87.01748886
74	kernel: linear - C values: 0.01 - Epsilon:0.05	87.01694182
73	kernel: linear - C values: 0.01 - Epsilon:0.01	87.01685412
81	kernel: linear - C values: 0.05 - Epsilon:0.1	87.01660496
68	kernel: linear - C values: 0.005 - Epsilon:0.05	87.01370866
67	kernel: linear - C values: 0.005 - Epsilon:0.01	87.01367237
75	kernel: linear - C values: 0.01 - Epsilon:0.1	87.01319191
69	kernel: linear - C values: 0.005 - Epsilon:0.1	87.00631128

Appendix-B- Table 3. SVR Hyperparameter tuning for Polynomial Kernels

Trial	Parameters	Accuracy
72	**Polynomial: 3 - C values: 0.005 - Epsilon:0.8**	**86.94649395**
71	Polynomial: 3 - C values: 0.005 - Epsilon:0.6	86.92616642
78	Polynomial: 3 - C values: 0.01 - Epsilon:0.8	86.90300109
70	Polynomial: 3 - C values: 0.005 - Epsilon:0.4	86.89908367
77	Polynomial: 3 - C values: 0.01 - Epsilon:0.6	86.87491306
64	Polynomial: 3 - C values: 0.001 - Epsilon:0.4	86.84933533
63	Polynomial: 3 - C values: 0.001 - Epsilon:0.1	86.84862873
62	Polynomial: 3 - C values: 0.001 - Epsilon:0.05	86.84267213
61	Polynomial: 3 - C values: 0.001 - Epsilon:0.01	86.8383234
65	Polynomial: 3 - C values: 0.001 - Epsilon:0.6	86.82421129
69	Polynomial: 3 - C values: 0.005 - Epsilon:0.1	86.79626836
66	Polynomial: 3 - C values: 0.001 - Epsilon:0.8	86.77401251
68	Polynomial: 3 - C values: 0.005 - Epsilon:0.05	86.76572714
76	Polynomial: 3 - C values: 0.01 - Epsilon:0.4	86.75936821
67	Polynomial: 3 - C values: 0.005 - Epsilon:0.01	86.75710503
84	Polynomial: 3 - C values: 0.05 - Epsilon:0.8	86.75649209
186	Polynomial: 5 - C values: 0.001 - Epsilon:0.8	86.63915072
90	Polynomial: 3 - C values: 0.1 - Epsilon:0.8	86.63588554
75	Polynomial: 3 - C values: 0.01 - Epsilon:0.1	86.62130046
74	Polynomial: 3 - C values: 0.01 - Epsilon:0.05	86.62055308
185	Polynomial: 5 - C values: 0.001 - Epsilon:0.6	86.61613342
184	Polynomial: 5 - C values: 0.001 - Epsilon:0.4	86.56366365
73	Polynomial: 3 - C values: 0.01 - Epsilon:0.01	86.56094489
192	Polynomial: 5 - C values: 0.005 - Epsilon:0.8	86.50772534
183	Polynomial: 5 - C values: 0.001 - Epsilon:0.1	86.46343323
182	Polynomial: 5 - C values: 0.001 - Epsilon:0.05	86.45075299
181	Polynomial: 5 - C values: 0.001 - Epsilon:0.01	86.44613149

Appendix-B- Table 4. ANN highest combinations (first filtiration process)

Trial	Parameters	Accuracy
1	neurons: 2 - Act: relu - Opt:Adam - batch_size:12 - Epochs:10	19.74024
2	neurons: 2 - Act: relu - Opt:RMSprop - batch_size:12 - Epochs:10	54.16513
3	neurons: 2 - Act: relu - Opt:Adadelta - batch_size:12 - Epochs:10	8.891686
4	neurons: 2 - Act: relu - Opt:Adagrad - batch_size:12 - Epochs:10	9.835159
5	neurons: 2 - Act: relu - Opt:Adamax - batch_size:12 - Epochs:10	52.74416
6	neurons: 2 - Act: relu - Opt:Nadam - batch_size:12 - Epochs:10	39.2594
7	neurons: 2 - Act: relu - Opt:Ftrl - batch_size:12 - Epochs:10	21.80929
8	neurons: 2 - Act: sigmoid - Opt:Adam - batch_size:12 - Epochs:10	-15.1422
9	neurons: 2 - Act: sigmoid - Opt:RMSprop - batch_size:12 - Epochs:10	64.89809
10	neurons: 2 - Act: sigmoid - Opt:Adadelta - batch_size:12 - Epochs:10	10.89777
11	neurons: 2 - Act: sigmoid - Opt:Adagrad - batch_size:12 - Epochs:10	7.043505
12	neurons: 2 - Act: sigmoid - Opt:Adamax - batch_size:12 - Epochs:10	0.20771
13	neurons: 2 - Act: sigmoid - Opt:Nadam - batch_size:12 - Epochs:10	0.47628
14	neurons: 2 - Act: sigmoid - Opt:Ftrl - batch_size:12 - Epochs:10	27.76777
15	neurons: 2 - Act: softplus - Opt:Adam - batch_size:12 - Epochs:10	-10.6296
16	neurons: 2 - Act: softplus - Opt:RMSprop - batch_size:12 - Epochs:10	61.34832
17	neurons: 2 - Act: softplus - Opt:Adadelta - batch_size:12 - Epochs:10	6.708979
18	neurons: 2 - Act: softplus - Opt:Adagrad - batch_size:12 - Epochs:10	21.47742
19	neurons: 2 - Act: softplus - Opt:Adamax - batch_size:12 - Epochs:10	52.34568
20	neurons: 2 - Act: softplus - Opt:Nadam - batch_size:12 - Epochs:10	-15.2086
21	neurons: 2 - Act: softplus - Opt:Ftrl - batch_size:12 - Epochs:10	0.000932
22	neurons: 2 - Act: softsign - Opt:Adam - batch_size:12 - Epochs:10	3.17126
23	neurons: 2 - Act: softsign - Opt:RMSprop - batch_size:12 - Epochs:10	66.89811

Appendix-B- Table 5. ANN top models (second filtration process)

Trial	Parameters	Accuracy
1	neurons: 64 - Act: relu - Opt:RMSprop - batch_size:12 - Epochs:10	70.02923
2	neurons: 64 - Act: softplus - Opt:RMSprop - batch_size:12 - Epochs:10	67.80174
3	neurons: 64 - Act: softsign - Opt:RMSprop - batch_size:12 - Epochs:10	68.8477
4	neurons: 64 - Act: selu - Opt:RMSprop - batch_size:12 - Epochs:10	69.63384
5	neurons: 128 - Act: relu - Opt:RMSprop - batch_size:12 - Epochs:10	70.06811
6	neurons: 128 - Act: softplus - Opt:RMSprop - batch_size:12 - Epochs:10	70.16981
7	neurons: 128 - Act: softsign - Opt:RMSprop - batch_size:12 - Epochs:10	68.10646
8	neurons: 128 - Act: selu - Opt:RMSprop - batch_size:12 - Epochs:10	68.02608
9	neurons: 256 - Act: relu - Opt:RMSprop - batch_size:12 - Epochs:10	70.01887
10	neurons: 256 - Act: softplus - Opt:RMSprop - batch_size:12 - Epochs:10	69.61939
11	neurons: 256 - Act: softsign - Opt:RMSprop - batch_size:12 - Epochs:10	69.4595
12	neurons: 256 - Act: selu - Opt:RMSprop - batch_size:12 - Epochs:10	68.98504
13	neurons: 64 - Act: relu - Opt:RMSprop - batch_size:24 - Epochs:10	69.50291
14	neurons: 64 - Act: softplus - Opt:RMSprop - batch_size:24 - Epochs:10	69.46919
15	neurons: 64 - Act: softsign - Opt:RMSprop - batch_size:24 - Epochs:10	68.91204
16	neurons: 64 - Act: selu - Opt:RMSprop - batch_size:24 - Epochs:10	67.84672
17	**neurons: 128 - Act: relu - Opt:RMSprop - batch_size:24 - Epochs:10**	**70.64659**
18	neurons: 128 - Act: softplus - Opt:RMSprop - batch_size:24 - Epochs:10	68.98063
19	neurons: 128 - Act: softsign - Opt:RMSprop - batch_size:24 - Epochs:10	67.22923
20	neurons: 128 - Act: selu - Opt:RMSprop - batch_size:24 - Epochs:10	67.32475
21	neurons: 256 - Act: relu - Opt:RMSprop - batch_size:24 - Epochs:10	69.11455
22	neurons: 256 - Act: softplus - Opt:RMSprop - batch_size:24 - Epochs:10	69.45091
23	neurons: 256 - Act: softsign - Opt:RMSprop - batch_size:24 - Epochs:10	68.97506
24	neurons: 256 - Act: selu - Opt:RMSprop - batch_size:24 - Epochs:10	66.98342

Appendix-B- Table 6. SVR Hyperparameter tuning for Hybrid Model (1)

Trial	Parameters	Accuracy
49	**kernel: rbf - C values: 2 - Epsilon:0.01**	**98.36403**
55	kernel: rbf - C values: 4 - Epsilon:0.01	98.36388
43	kernel: rbf - C values: 1 - Epsilon:0.01	98.36376
37	kernel: rbf - C values: 0.8 - Epsilon:0.01	98.36368
31	kernel: rbf - C values: 0.4 - Epsilon:0.01	98.36348
56	kernel: rbf - C values: 4 - Epsilon:0.05	98.36274
44	kernel: rbf - C values: 1 - Epsilon:0.05	98.36270
50	kernel: rbf - C values: 2 - Epsilon:0.05	98.36270
38	kernel: rbf - C values: 0.8 - Epsilon:0.05	98.36266
32	kernel: rbf - C values: 0.4 - Epsilon:0.05	98.36237
25	kernel: rbf - C values: 0.1 - Epsilon:0.01	98.36210
26	kernel: rbf - C values: 0.1 - Epsilon:0.05	98.36127
19	kernel: rbf - C values: 0.05 - Epsilon:0.01	98.36010
20	kernel: rbf - C values: 0.05 - Epsilon:0.05	98.35914
33	kernel: rbf - C values: 0.4 - Epsilon:0.1	98.35760
39	kernel: rbf - C values: 0.8 - Epsilon:0.1	98.35755
45	kernel: rbf - C values: 1 - Epsilon:0.1	98.35743
51	kernel: rbf - C values: 2 - Epsilon:0.1	98.35731
57	kernel: rbf - C values: 4 - Epsilon:0.1	98.35721
27	kernel: rbf - C values: 0.1 - Epsilon:0.1	98.35526
103	kernel: linear - C values: 1 - Epsilon:0.01	98.35278
97	kernel: linear - C values: 0.8 - Epsilon:0.01	98.35276
79	kernel: linear - C values: 0.05 - Epsilon:0.01	98.35275
91	kernel: linear - C values: 0.4 - Epsilon:0.01	98.35275
73	kernel: linear - C values: 0.01 - Epsilon:0.01	98.35272
67	kernel: linear - C values: 0.005 - Epsilon:0.01	98.35271
85	kernel: linear - C values: 0.1 - Epsilon:0.01	98.35270
109	kernel: linear - C values: 2 - Epsilon:0.01	98.35268
115	kernel: linear - C values: 4 - Epsilon:0.01	98.35266
69	kernel: linear - C values: 0.005 - Epsilon:0.1	98.35246
92	kernel: linear - C values: 0.4 - Epsilon:0.05	98.35229
80	kernel: linear - C values: 0.05 - Epsilon:0.05	98.35226
98	kernel: linear - C values: 0.8 - Epsilon:0.05	98.35225
74	kernel: linear - C values: 0.01 - Epsilon:0.05	98.35224
104	kernel: linear - C values: 1 - Epsilon:0.05	98.35224
110	kernel: linear - C values: 2 - Epsilon:0.05	98.35218
68	kernel: linear - C values: 0.005 - Epsilon:0.05	98.35218

Appendix-B- Table 7. Hybrid model (2) Hyperparameter tuning for weights

LSTM Percentage (w_i)	ε-SVR percentage (w_j)	MAPE
0.01	0.99	12.86472163
0.02	0.98	12.74218462
0.03	0.97	12.61965311
0.04	0.96	12.4971346
0.05	0.95	12.3746225
0.06	0.94	12.25213816
0.07	0.93	12.12967379
0.08	0.92	12.00726637
0.09	0.91	11.88489737
0.1	0.90	11.76254545
0.11	0.89	11.64020041
0.12	0.88	11.51786702
0.13	0.87	11.39555025
0.14	0.86	11.2733161
0.15	0.85	11.15110934
0.16	0.84	11.02894351
0.17	0.83	10.90679637
0.18	0.82	10.78472059
0.19	0.81	10.66271721
0.2	0.80	10.54079141
0.21	0.79	10.41896277
0.22	0.78	10.29721675
0.23	0.77	10.17557283
0.24	0.76	10.05400946
0.25	0.75	9.932478555
0.26	0.74	9.810992446
0.27	0.73	9.689581028