Energy Demand Forecasting using Machine Learning (Delhi)

by

SUJAYKUMAR REDDY M
Vellore Institute of Technology, Vellore

Report submitted to the
Center of Excellence in Logistics and Supply Chain Management (CoELSCM)
National Institute of Technology, Calicut
In fulfillment of the requirements
For the Pragati Internship, part of PM Gati Shakti Scheme

Advised by
Dr Gopakumar G, CSE Department
National Institute of Technology, Calicut

Abstract

The National Capital Territory (NCT) of India, namely Delhi, is home to New Delhi, the country's capital. Being a significant political and economic hub, Delhi experiences unique considerations in terms of power consumption. As of May 31st, 2023, the highest recorded power demand in Delhi occurred on June 29th, 2022, reaching a peak of 7770 MW. Unlike traditional sectors, the Power and Energy industry does not deal with physical inventory. Instead, it focuses on the generation, transmission, and distribution of electricity, as well as other energy sources.

Electricity forecasting plays a very significant role in the power grid as it is required to maintain a balance between supply and load demand at all times, to provide a quality electricity supply, for financial planning, generation reserve, and many more. Forecasting helps not only in Production Planning, but also in Scheduling like Import / Export which is very often in India and mostly required by the rural areas and North Eastern Regions of India. As Electrical Forecasting includes many factors which cannot be detected by the models out there, We use Machine Learning to extract patterns from the daily data of Maximum Demand for the Union Territory Delhi.

This research contributes to the power supply industry by helping to reduce the occurrence of disasters such as blackouts, power cuts, and increased tariffs imposed by regulatory commissions. The forecasting techniques can also help in reducing OD and UD of Power for different regions. We use the Data provided by a department from the Ministry of Power and use different forecast models including Seasonal forecasts for daily data.

This report is made accordingly to the given template. It begins with the Introduction that highlights the recent problems in Delhi Power Sector, Delhi DISCOMs, and Power Generation in Delhi. The Background analysis chapter surveys the Potential Forecasting Techniques by Indian authors and reports from different ministries. The Methodology chapter gives the framework for further understanding of my work. The Data Generation chapter provides a clear understanding of how the data is taken, processed, and ready to use.The Forecasting Models chapter gives a technical understanding of the models and the parameters which capture the interactions. The Results chapter tabulates the results of the forecasting models, finally the Conclusion.

List of Abbreviations

AI	Artificial Intelligence.
ML	Machine Learning
TSA	TimeSeries Analysis
DL	Deep Learning
ANN	Artificial Neural Networks
RES	Renewable Energy Sources
IAEA	International Atomic Energy Agency
UNDESA	United Nations Department of Economic and Social Affairs
IEA	International Energy Agency
EEA	Eurostat and European Environmental Agency
CAGR	Compound Annual Growth Rates
YoY	Year over Year
PSP	Power Supply Position
NLDC	National Load Despatch Centre
MU	Million Units
MW	MilliWatts
OD	Overdrawl
UD	Underdrawl
DISCOM	Distribution Companies
NCT	National Capital Territory
MOP	Ministry of Power
POSOCO	Power System Operation Corporation Limited

Table of Contents

Chapter 1

Introduction

This chapter formalizes the problem statement which I want to solve using Machine Learning and the major advantages of Machine Learning in the context of Demand Forecasting and also briefly explains the Power situation in delhi, what are the stakeholders which are involved in the Delhi's Power Generation and the Perspectives of Political Parties, Types of Energy available in Delhi and finally how PM gati-shakti scheme will help the Demand Forecasting in further aspects through-out the journey.

1.1 Indian Energy GRID

The Indian Energy GRID is maintained by POWERGRID [12] which has the objectives of running the GRID efficiently and installing transmission lines etc... and second one is the National Load Dispatch Center (NLDC) [13] which concentrates on Supervision over the Regional Load Despatch Centres, Scheduling and despatch of electricity over inter-regional links in accordance with grid standards specified by the Authority and grid code specified by Central Commission in coordination with Regional Load Despatch Centres, Monitoring of operations and grid security of the National Grid etc... Our main focus is on this NLDC which is a Division of Ministry of Power.

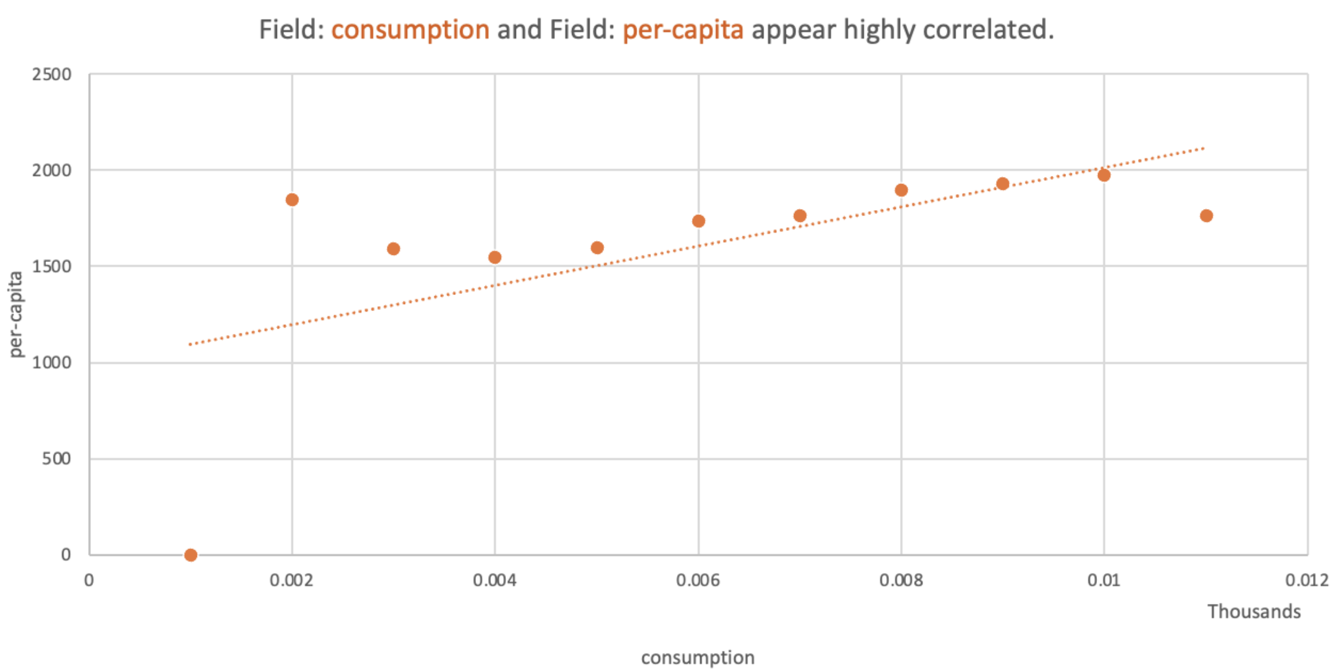
Fig 1.1 [16] describes the Yearly Installed Power Capacity in Delhi.The highest installed Capacity was 8,346.72 MW at Fiscal year 2016. which is responsible for sending the energy from Stations to sub-stations and to discoms and then to homes, industries, commercials etc... As of 21-06-2023 The installed Capacity Sector-wise data [7] gives an overview of what type of Thermal Plants which are present in Delhi and also what types of Energy resources present in Delhi. There are 11 Thermal Stations in Delhi with 4 400KV Substations and 42 200 KV Substations [4].

1.2 Statistics on the Delhi's Power

To further Analyse the Delhi's Power we use the [16] RBI's handbook which contains yearly state-wise data to analyze the Delhi v/s Whole India's data. Fig 1.2 gives Delhi v/s whole

India's Statewise Per-capita of Power which raises to 1974.4 Kilo-Watt Hour at Delhi on 2018-2019 and in India 1115.3 Kilo-Watt Hour on 2021-2022 Fig 1.3 gives a Availability of Power at Delhi raises to 3308 net Crore Units on 2019-2020 and 137402 net crore units on 2021-2022 for All India , Fig 1.4 gives Total Installed Capacity of RES Power which raises to 245 Mega Watt at 2021 for Delhi and 94434 MegaWatt at 2021 for all India. Fig 1.5 depits the Power Requirement for Delhi which has the maximum reading at 3309 net crore units on 2019-2020 and for all India 137981 net crore units on 2021-2022.

1.3 Relation to Demand Forecasting



As we are speaking we can see that there are Correlation between these variables with the Consumption of Electricity of Delhi [3] Now, as we speak Consumption by the consumers is based on the Maximum Demand which has been recorded per-day in the National Load Dispatch Center (NLDC) As, all these entities are highly correlated we push our limits to only use univariant analysis of Maximum Demand attained in the Daily data produced by the Delhi Consumers.

1.4 Perspective of Machine Learning

Statistically speaking many papers have proven that the Economic Factors also have been an useful data for Maximum Demand Forecasting Although their findings have been sucessfull we present a baseline characteristic model which can only concentrate of one variable and captures the Auto-Correlations between the days, weeks, months and even years. This is to create a baseline Performance models for the reason of Seasonal Dependence, we use different pre-processing techniques including with an Handful Machine Learning and Time Series Forecasting models which have been used before in-order to create a Baseline model.

1.5 An PM-Gati Scheme Initiative

An Integration of Ministries with 7 drivers railways, Ports, waterways, Logistic Infrastructure, Mass Transport, Airport and Roads which aims to improve connectivity and make Indian businesses more competitive has six pillars of Gati-Shakti known as Connectivity for Productivity.

1. Comprehensiveness
2. Prioritization
3. Optimization
4. Synchronization
5. Analytical
6. Dynamic

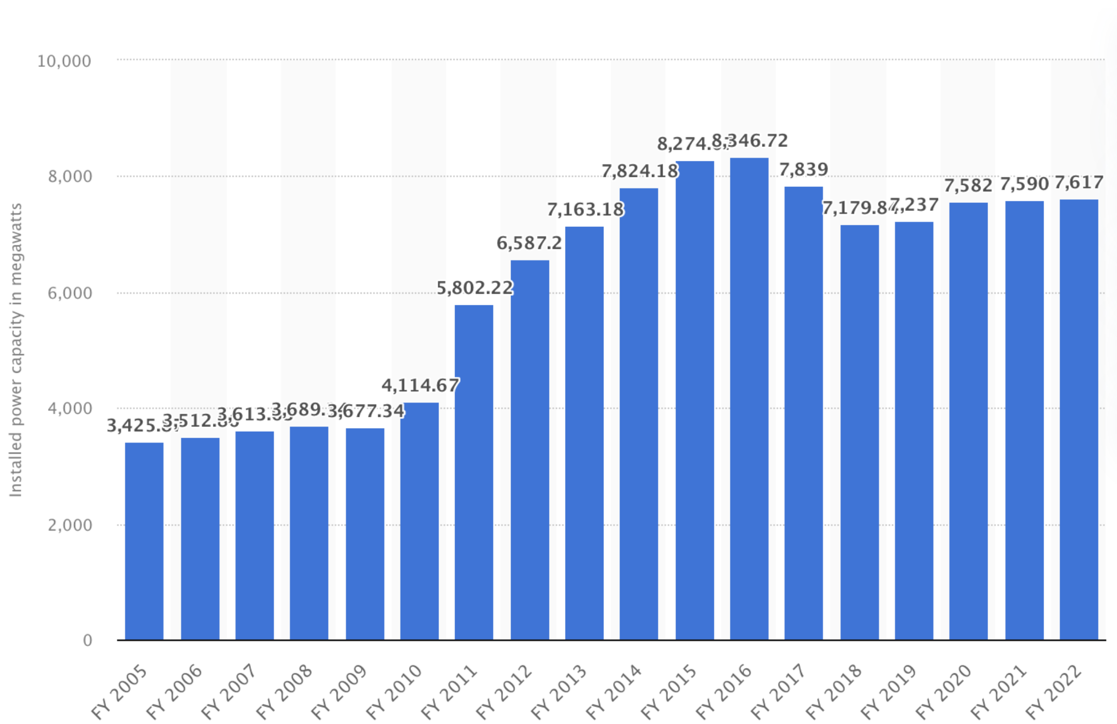
Given these Pillars, we try to formalize and study the use-cases which are provided for this problem Statement, "Maximum Demand Forecasting in Delhi"

1.6 Case Study of Maximum Demand Forecasting

Lets say that Ministry of Steel[10] created an initiative to increase domestic production that would lower dependence on imported Steel and would result in considerable saving of foreign exchange. As this seems reasonable, there are some pros and cons related to this one being less Foreign Exchange obviously and a con being Increased Electricity Consumption. A data which explains from the Ministry of Coal to Ministry of Power in-order to increase the installed Capacity or Power Generation can be captured through Comprehensiveness.

As there's increase in demand of Electricity, the Ministry of Power tries to solve this problem but as the transparency of Ministries in PM-Gati-Shakti provides to consolidate the increase in demand like the Ministry of Railways tries to optimize routes on weekends and majorly metro trains which are in Delhi can be optimized to consolidate this increase in demand which can be captured through Prioritization, Optimization while maintaining an Holistic Approach.

Note that this is a Scenerio which can be predicted but there will be many scenerios as possible in-order facilitate the Maximum Demand Forecasting.



21-06-2023	Delhi	STATE SECTOR	Thermal	2100.4
21-06-2023	Delhi	STATE SECTOR	Nuclear	NA
21-06-2023	Delhi	STATE SECTOR	Hydro	NA
21-06-2023	Delhi	STATE SECTOR	RES	NA
21-06-2023	Delhi	PVT SECTOR	Thermal	108
21-06-2023	Delhi	PVT SECTOR	Nuclear	NA
21-06-2023	Delhi	PVT SECTOR	Hydro	NA
21-06-2023	Delhi	PVT SECTOR	RES	302.26

Figure 1.1: Installed Capacity

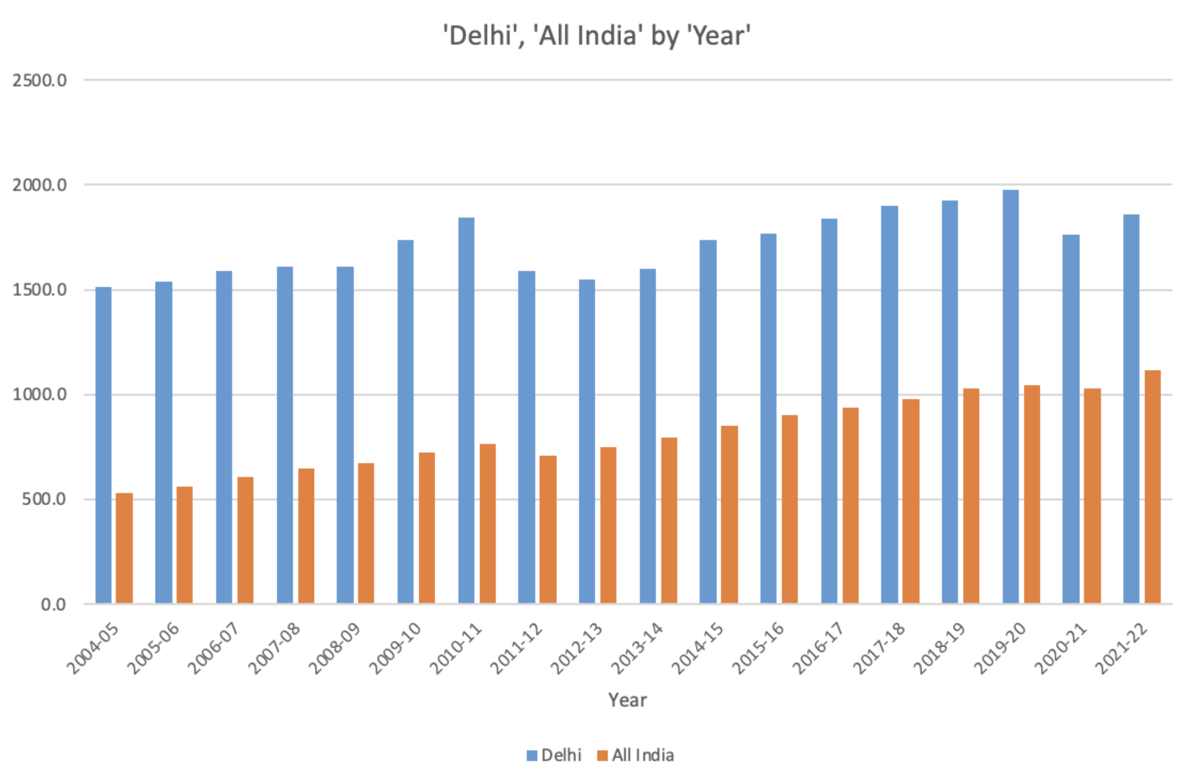


Figure 1.2: Per-capita Availability of Power

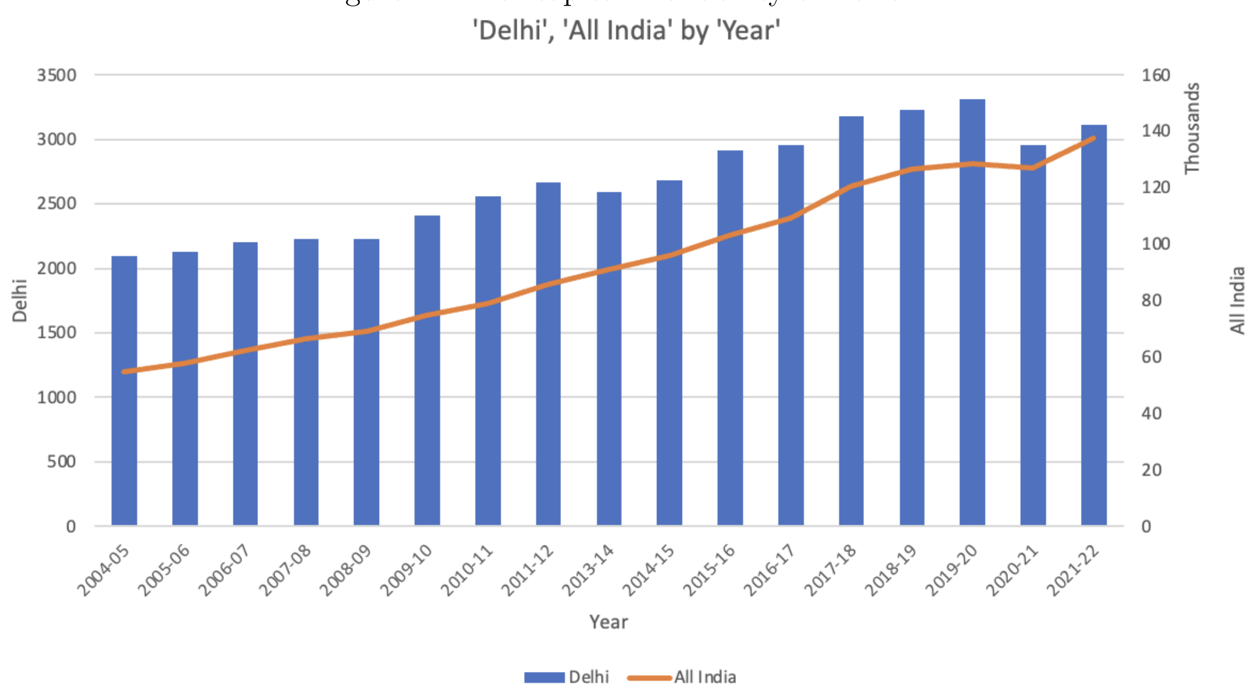


Figure 1.3: Availability of Power

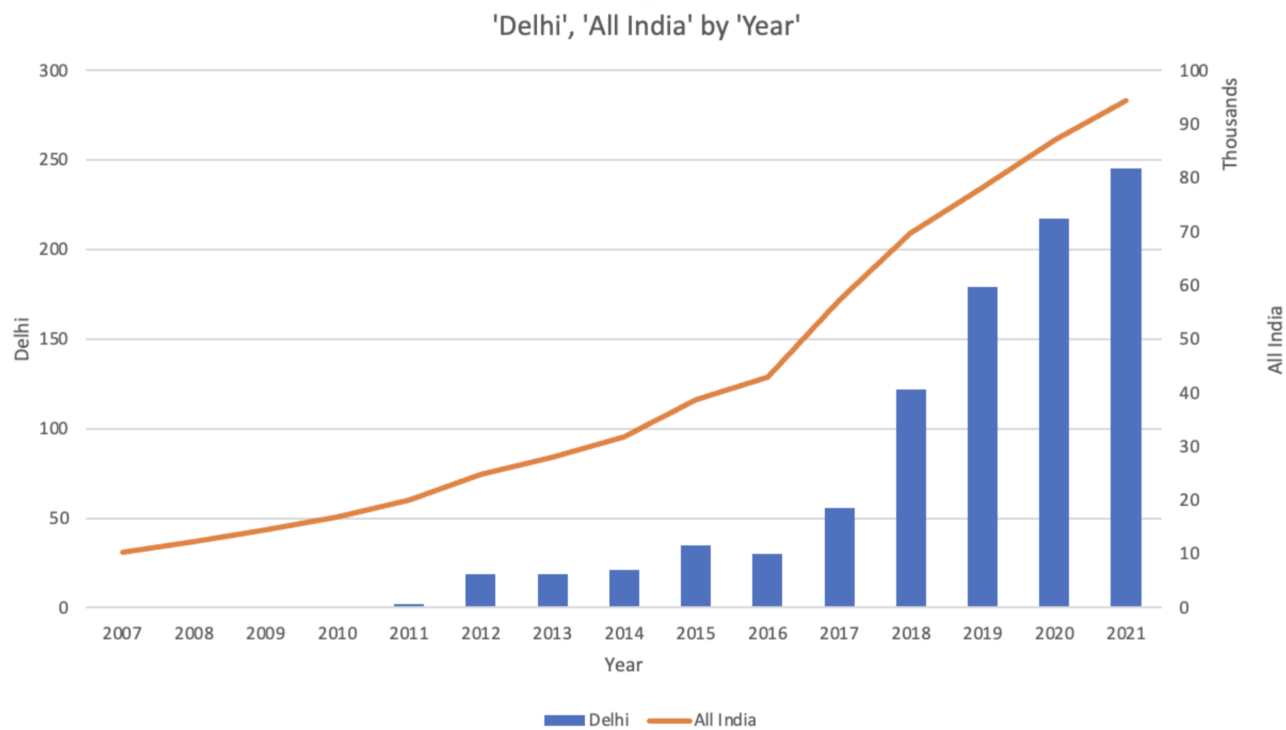


Figure 1.4: Total Installed Capacity of grid Interactive Renewable Power

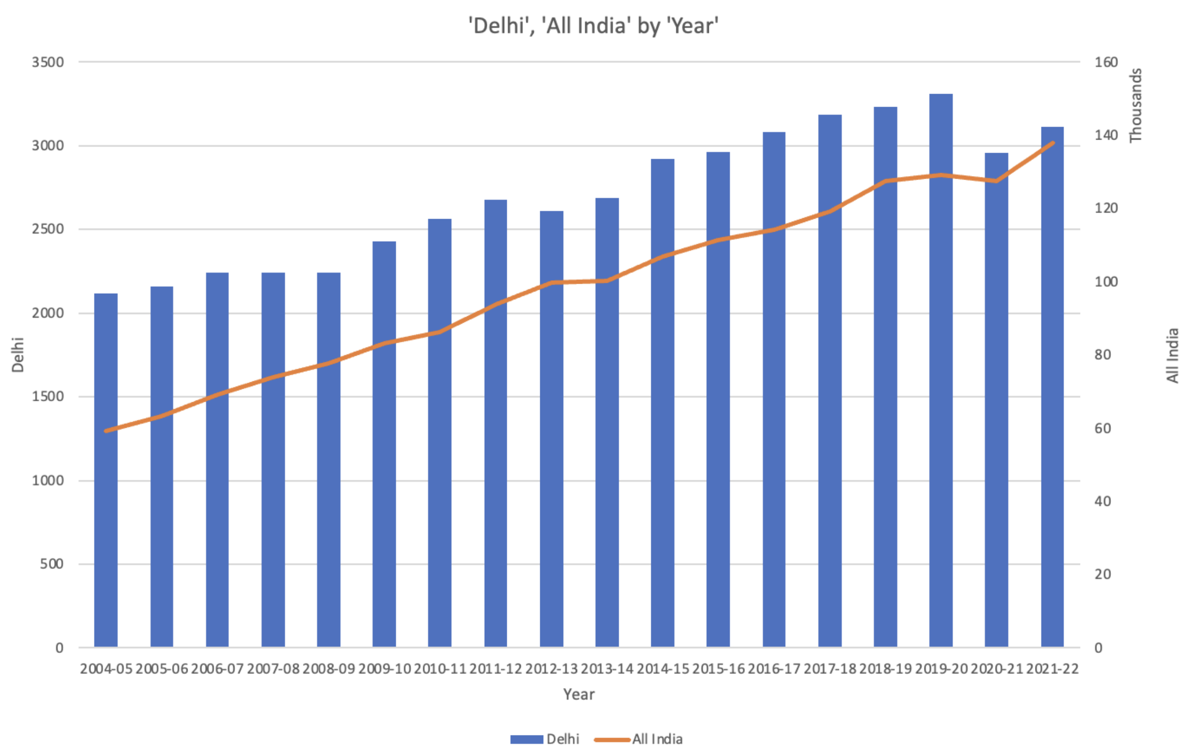


Figure 1.5: Power Requirement

Chapter 2

Background Analysis

This Chapter consists of Background analysis a.k.a Literature review of the Demand Forecasting Techniques adopted and Analyzing the recent reports by the Ministries of Government. This chapter focuses on many different attributes which are needed to be considered by the previous research done by the individuals. We address the following Questions.

1. What are the current major algorithms used from (State-of-the-art to basic)
2. Which Forecasts is better Short term or Long Term
3. Which Country Data sources have they taken into account.
4. Identification of Research gaps

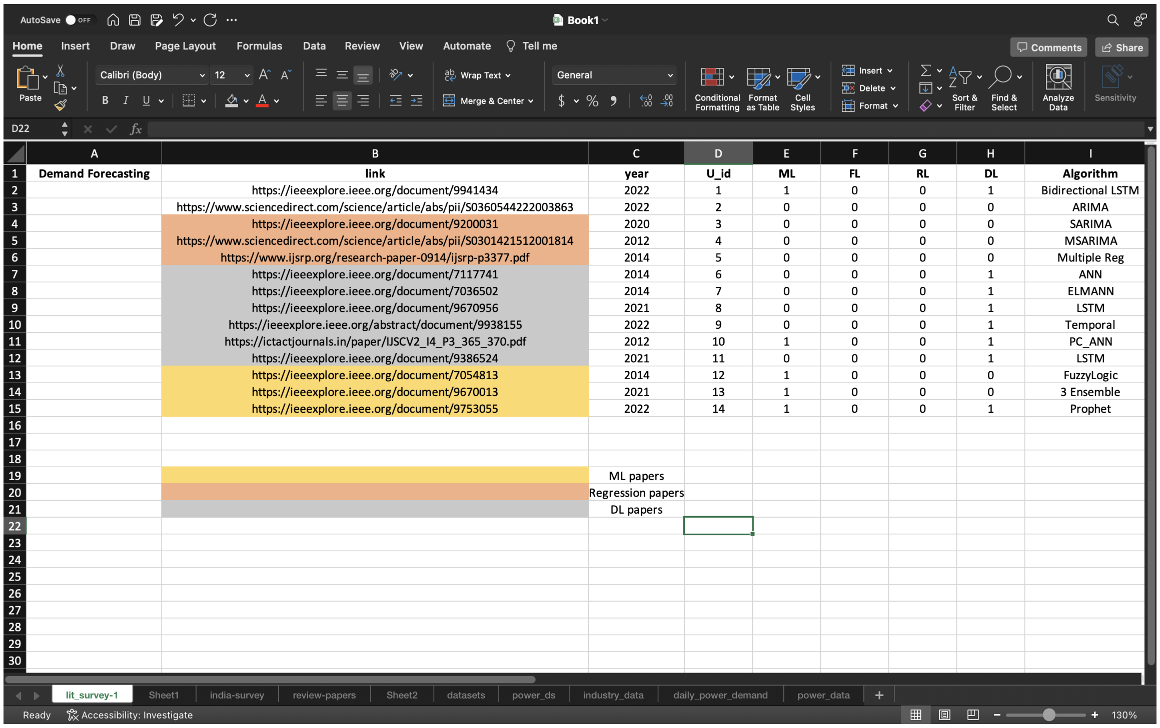
2.1 Literature Review Methodology

We actually formulate a new Literature Review Methodology which gives more information of the paper in an concise way, which is **Feature Extraction and Survey Analysis Framework.** This helps us to do the Literature Survey Fast, Precise and most importantly avoids repetition. As the Demand Forecasting has a large extensive research which has already been done, we only consider the papers which are from 2019-2023 for global papers and 2000-2023 for Indian Forecasting Papers and majorly Delhi.

2.2 Feature Extraction

Given a paper retrieved from SCOPUS database, we extract the Features which is some of the major questions asked in a paper, like what is accuracy, which dataset, pre-processing techniques. In this research we extract 17 Features from each paper and then we analyze through different graphs.

The features being ML(1/0), FL(1/0), RL(1/0), DL(1/0), Algorithm(object), Regression(object), Industry(power/energy), dataset(object), production(predictor),



Demand Forecasting	link	year	U_id	ML	FL	RL	DL	Algorithm
	https://ieeexplore.ieee.org/document/9941434	2022	1	1	0	0	1	Bidirectional LSTM
	https://www.sciencedirect.com/science/article/abs/pii/S0360544222003863	2022	2	0	0	0	0	ARIMA
	https://ieeexplore.ieee.org/document/9200031	2020	3	0	0	0	0	SARIMA
	https://www.sciencedirect.com/science/article/abs/pii/S0301421512001814	2012	4	0	0	0	0	MSARIMA
	https://www.ijsrp.org/research-paper-0914/ijsrp-p3377.pdf	2014	5	0	0	0	0	Multiple Reg
	https://ieeexplore.ieee.org/document/7117741	2014	6	0	0	0	1	ANN
	https://ieeexplore.ieee.org/document/7036502	2014	7	0	0	0	1	ELMANN
	https://ieeexplore.ieee.org/document/9670956	2021	8	0	0	0	1	LSTM
	https://ieeexplore.ieee.org/abstract/document/9938155	2022	9	0	0	0	1	Temporal
	https://ictactjournals.in/paper/IJSCV2_I4_P3_365_370.pdf	2012	10	1	0	0	1	PC_ANN
	https://ieeexplore.ieee.org/document/9386524	2021	11	0	0	0	1	LSTM
	https://ieeexplore.ieee.org/document/7054813	2014	12	1	0	0	0	FuzzyLogic
	https://ieeexplore.ieee.org/document/9670013	2021	13	1	0	0	0	3 Ensemble
	https://ieeexplore.ieee.org/document/9753055	2022	14	1	0	0	1	Prophet
		ML papers						
		Regression papers						
		DL papers						

Figure 2.1: Feature Extraction Methodology

demand_freq(daily/monthly/yearly), MAPE/RMSE(metric), acc(metric), NN(object), Cross - Validation CV (1/0), ARIMA(1/0), country(object).

The Figure 2.1 explains the Feature Extraction Methodology of this research. The color codes explain the clusters of the Papers like Gray cluster is totally based on Deep Learning papers where Orange cluster and Yellow cluster is based on Machine Learning and Regression papers. After doing the Feature Extraction Methodology we proceed to answer all the features by doing an Iteration 1 for each paper which take 5-10 minutes. We now proceed to survey analysis.

2.3 Analysis Survey

During the Analysis For each paper we extract more papers which are surveyed and clustered by the Feature Extraction process. Now, the data which is retrieved by reading the major paper without reading the cited paper is known to be meta information like in what cluster does that paper belong to, the year published and the results.

The Figure 2.2 explains depicts the level-set 1 (main paper to cited paper) from this data we sort it through years and take the relevant papers and move it to Feature Extraction and rest papers meta information is retrieved and saved. This process is said to be Iteration-1 Level Set-1 of Literature Survey. Now in the Upcoming sections with this framework, I will

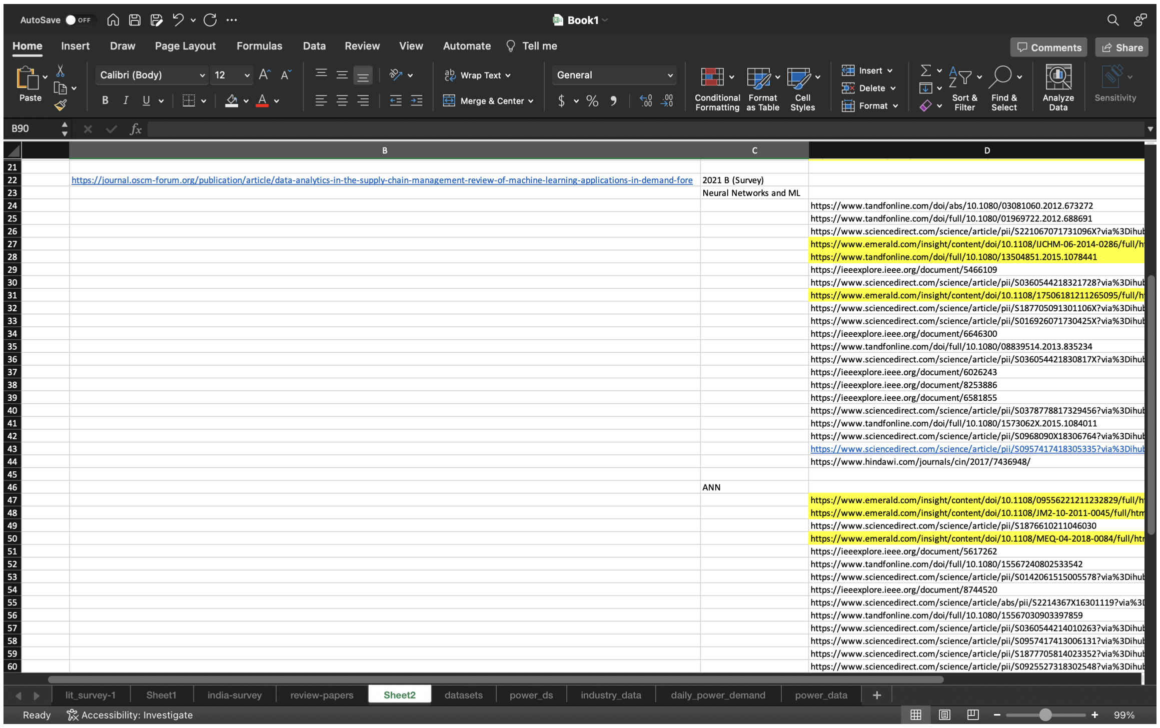
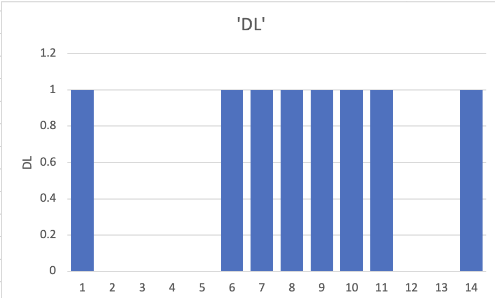


Figure 2.2: Survey Analysis Methodology

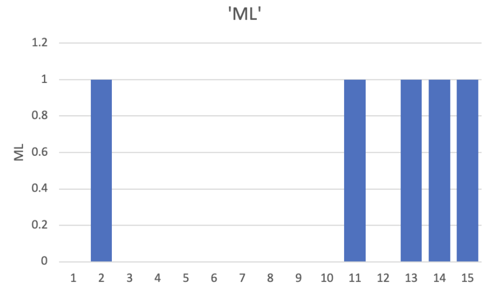
produce my findings according to each cluster.

2.4 Deep Learning

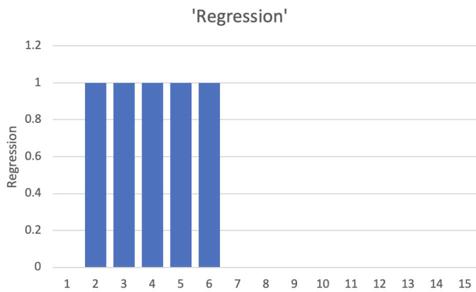
Fig 2.3 (a) gives us the clear understanding of number of Deep Learning papers which is mostly based on Indian Authors published in IEEE. Anil et al [11] uses Levenberg-marquardt back propagation algorithm ANN on day ahead Short term load Forecasting on the state of Uttar Pradesh trained on hourly data with the MAPE score of average MAPE 3.05, This work suggests to use the ANN model to check with our dataset too. Navneet et al [19] uses the New Delhi ADEL data to forecast the load by using different Neural Network Architectures in which ELMANN Neural Network Architecture has given the good accuracy. Dharmoju et al [5] provided a sector of Residental buildings by the United States Dataset by using LSTM (Long Short Term Memory) model for monthly forecasting. Shaswat et al [8] uses a Temporal Fusion Architecture to capture the interactions which are scaled between 0 and 1 for daily data which acheives 4.15% more than the existing models and this is for the whole India which is not region specific. Saravanan et al [17] uses the economic factors like GDP, national income, consumer price index etc.. with that they used Principle component Analysis following with ANN which gives the highest accuracy of MAPE score 0.43. Vishnu et al [9] concentrates on the work on Renewable Energy Resources devised two major LSTM (Long Short Term Memory) models.



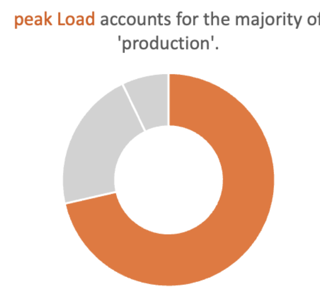
(a) Deep Learning Papers Analysis



(b) Machine Learning Papers Analysis



(c) Regression based Papers Analysis



(d) Predictors used in the Dataset Papers

Figure 2.3: Literature Survey Framework Analysis

2.5 Machine Learning

Fig 2.3 (b) gives us a clear understanding of number of Machine Learning papers and their analysis in the all the papers (as ids). Christos et al [1] which also forecasts the Peak Demand in the electrical sector of Producers side. They have used a Netherland dataset from 3 regions and analyzes comparatively with all the existing Forecasting methods such as ARIMA, Ridge and Lasso Regression and the results suggested that the Bi-directional LSTM. Saravanan et al [18] formalizes a set of 64 if-else statements and the variables includes per-capita GDP, population and Import/Export and they have acheived with the MAPE of 2.3. Mannish et al [20] devised an Ensemble Approach for the DISCOMs in the region delhi for the Post Covid Scenerio and their proposed model includes by combining XGBoost, LightGBM, CatBoost algorithms which acheived an average MAPE of 5.0. Banga et al [2] compared many Machine Learning Algorithms for the dataset which considers 29 attributes and the Facebook Prophet model outscores daily and hourly datasets with the MAPE scores of 0.4 and 0.2.

2.6 Regression Based Learning

Fig 2.3 (c) gives us a clear understanding of number of Regression papers or Auto Regression papers compared with all the papers (as ids). note these papers are based completely on uni-variant/single-variant datasets. Carlos et al [21] analyzed a time series datasets for Brazilian Electricity Demand Forecasting and divides the Brazil into 2 regions and forecasts the Electricity demand according by using ARIMA models. Kakoli et al [6] forecasted the electricity demand for the state Assam in the Northeast Region and the results suugest that to use the Seasonal ARIMA model with the formula given below SARIMA(0,1,1)(0,1,1,7) with the MAPE of 10.7. Srinivasa et al [15] provided a forecasting method which is formulated monthly for the whole India without considering the states and regions. It has been found that MSARIMA model outperforms CEA forecasts both in-sample static and out-of-sample dynamic forecast horizons in all five regional grids in India.

2.7 Summary of Literature Survey

Clusters	**Paper Citations**
Machine Learning	[1] [18] [20] [2]
Deep Learning	[11] [19] [5] [8] [17] [9]
Auto-Regressive Models	[21] [6] [15]

Chapter 3

Methodology

This chapter gives a context and framework of further research work that has been focussed on. This chapter continues to explain the where has the data been taken and what is the previous research which has been adopted to focus on the forecasting techniques. In this chapter we address the following questions

1. What is the Dataset that has been used for Machine Learning.
2. How many variables has been devised to solve this problem
3. Considering Ministries in India, How does combining Ministries would help to solve this problem.
4. How would you select the best model from the all the Machine Learning Models.

3.1 Framework for my research Methodology

To effectively select a Machine Learning Model, we usually divide the dataset into 2 parts Train and Test but in this case we divide the dataset into 3 parts Train, Test and Validation We train our model using the Training dataset and We use all the confined space models into the test dataset by using the Evaluation metric We select the best model and from those Evaluation metric, top 3 models with less MAPE will be selected for the validation set and the top MAPE will be selected as our model from the Validation set. This is how the framework has been set and in the next section we will analyze how to split the data.

3.2 Data

Considering the Energy and Power sector of India we have a handfull number of stakeholders which are involved we considered a few in this research work as being the first work in Electricity Demand Forecasting of Delhi region, we consider only one main ministry which is Ministry of Power and a department of ministry of Power which is GRID INDIA- National load dispatch centre [13] formerly known as POCOSO. They have produced a daily generation reports for every region and every state so we take the data from these reports. As the Figure

3.1 shows that we spilt the data from April 2013 to April 2023 as Training set which is of 10 years and Test set to be May 2023 and the Validation as June 2023. We divide it accordingly to follow the Machine Learning Techniques to devise a model.

3.3 Pre-Processing Techniques

As seen that the real world data is messy, A lot of reports from the PSP data is missing and we include those as Null Values subjected as "NA" and now we use preprocessing techniques to devise different datasets from the original dataset which has been provided.

3.4 Time Series Forecasting

We develop a Time series forecasting models ARIMA which is known to be as Auto-Regressive Integrated Moving Average. the models which includes like AR, MA and ARMA and ARIMA we develop a model list from these regression types using the parameters.

3.4.1 Parameters which Include ARIMA

The major parameters which include in the arima model are p, q and d where p is the parameter for Auto-regressive co-efficient which says about the how many days have the co-relation between today's date.

3.5 Machine Learning

We extract Features from the Univariant Data which has been provided by the different datasets.We use all Regression types like Ridge and Lasso Regression to train the model.

3.6 Deep Learning

single layer LSTM, double layer LSTM and Bi-directional LSTM are used in-case of Time Series Forecasting with and without Feature Extraction.

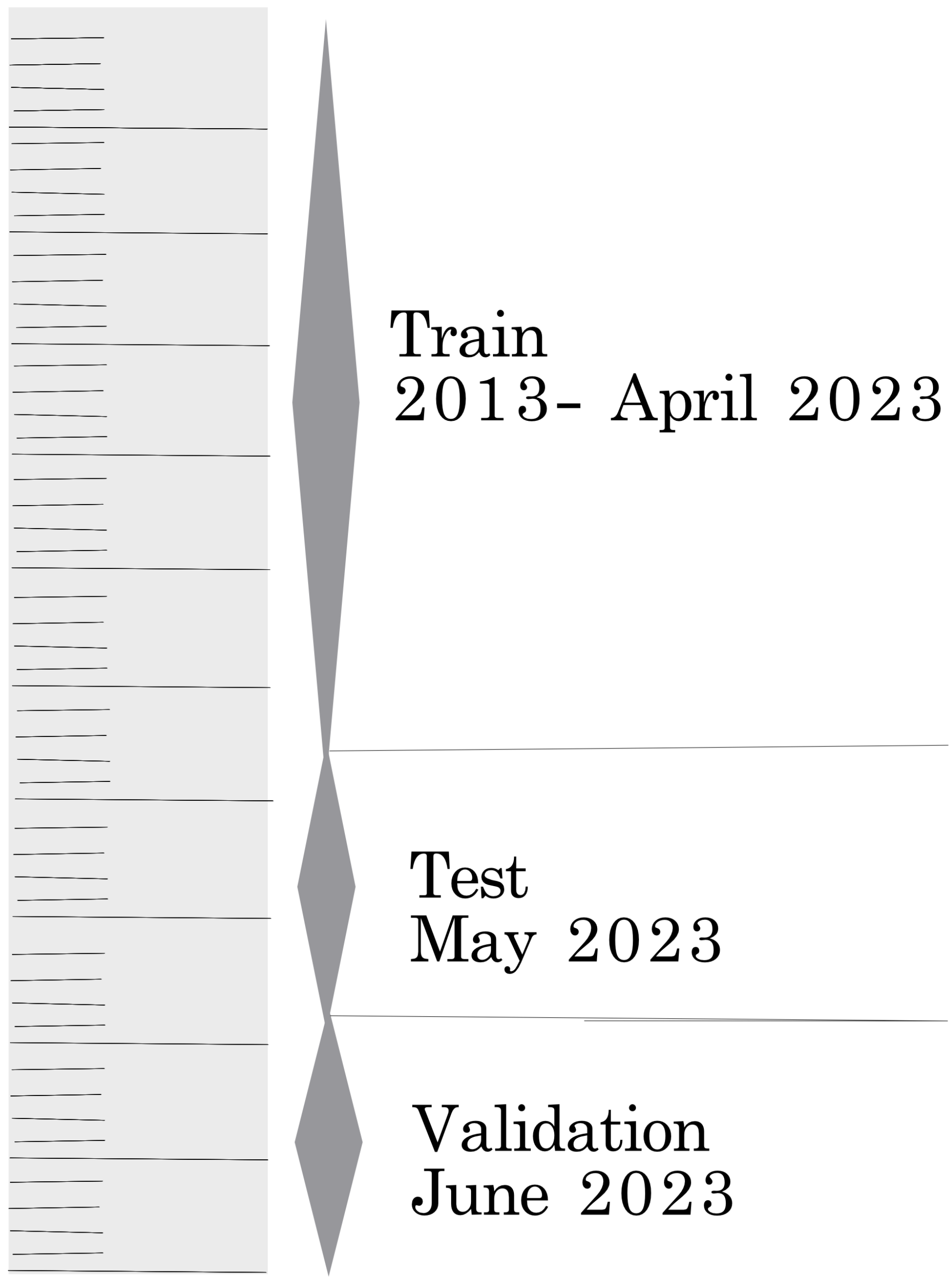


Figure 3.1: Dividing dataset Methodology

Chapter 4

Data Generation

This Chapter focusses on generating Data from the reports and an inclusion of an Automation Software which converts all pdf files into data.

4.1 Automation Software

```


'''
Author :  Sujay
Definition : An Automation System to convert PDF's to Data

Data Extracted :

    1. Date (DD/MM/YYYY)
    2. Max.Demand met during the day (MW)
    3. Shortage during maximum Demand (MW)
    4. Energy Met (MU)
    5. Drawal Schedule(MU)
    6. OD(+)/UD(-) (MU)
    7. Max OD (MW)
    8. Energy Shortage (MU)

'''

# Update - 0.1
'''
    - No need to convert to txt file
    - Extracting Region Data
'''

import pandas as pd
import numpy as np
import PyPDF2
import os


```

```python
# Defining File Paths Here
folder_path = "raw_data/"
processed_path = "processed_data/"


# State for which demand can be found
state = "Delhi"


'''
    - Converts the PDFs to .txt files
    - Only converts 2nd page of PDF to text (less space)
    - Path From  : raw_data/* (all)
    - Path to : processed_data/* (all)
'''
def pdf_conversion(file):
    org_path = folder_path+file
    processing = processed_path+file[0:8]+".txt"
    pdffileobj=open(org_path,'rb')
    pdfreader=PyPDF2.PdfReader(pdffileobj)
    x=len(pdfreader.pages)
    if x>1:
        pageobj=pdfreader.pages[1]
        pageobj1 = pdfreader.pages[0]
        text=pageobj.extract_text()
        text = text + pageobj1.extract_text()
    else:
        pageobj = pdfreader.pages[0]
        text=pageobj.extract_text()

    # Make a Txt File
    file1=open(processing ,"a")
    file1.writelines(text)
    file1.close()
    pdffileobj.close()



'''
    - Extract the Data from the .txt files (Constant Time Complexity - reduced file size)
    - Only extracts the particular data from the length from x to x'
    - Path From  : processed_data/* (all)
    - Returns an array of Strings : []
'''
def extract_info(file):
    contents = ""
    file_path = processed_path+file
    myfile = open(file_path, "rt")
    contents = myfile.read()
    myfile.close()
    # Find the Substring index from the file
    x = 0
    if "Delhi" in contents:
```

```
        x = contents.index(state)

    stri = ""
    s = []
    s.append(file[0:8].replace('.','/'))
    sd = ""
    while stri != "\n" and x<len(contents):
        if stri == " ":
            s.append(sd)
            sd = contents[x]
        else:
            sd = sd + contents[x]
        stri = contents[x]
        x = x+1
    sd = sd[:4]
    s.append(sd)
    return s



files = os.listdir(folder_path)
data = []

# This is only for MAC's to remove the .DStore file
files.remove(".DS_Store")

print(len(files))

# # First : Conversion
for file in files:
    pdf_conversion(file)

processed_files = os.listdir(processed_path)

# This is only for MAC's to remove the .DStore file
processed_files.remove(".DS_Store")

# Second : Extraction
for file in processed_files:
    data.append(extract_info(file))


# Check for the length equal to size of the folder
print(len(data),len(files),len(processed_files))
assert len(data) == len(files)

df = pd.DataFrame(data)

df.to_csv("main/ndlc_psp.csv")
```

4.2 Data Overview

The Features of the data which has been provided below

1. Date (DD/MM/YYYY)
2. Max.Demand met during the day (MW)
3. Shortage during maximum Demand (MW)
4. Energy Met (MU)
5. Drawal Schedule (MU)
6. OD(+)/UD(-) (MU)
7. Max OD (MW)
8. Energy Shortage (MU)

Energy Shortage (MU) feature is not available for Everyday. This feature is recorded from 2017-05-09 as per the reports generated by the PSP by POSOCO. The data is available here [14]. As we focussing on Univariant Analysis not to make it as complex. We only take the Max Demand met during the day (MW) as a single column.

As this feature "Max Demand met During the day" is the major feature which we consider to focus on. As the data starts from 2013-04-01 to 2023-05-31 which consists of 3713 days but the data points only consists of 3640 with the missing data we use Imputation Techniques dataset and Non-Imputation Techniques dataset (no_null).

For Imputation Techniques, we take Mean, Median, Mode and Interpolation Imputation data and there are many types of Interpolation Imputation we choose linear because for most of the types the data which has been imputed is same as linear-interpolation Techniques. So, as combined we generate 5 datasets where we use the same algorithms for every data-sets to compare the performance of which Imputation is good.

As we explained in the Methodology Chapter we divide the dataset into train and test to take the MAPE score. For ARIMA models we generate train MAPE and test MAPE to check the overfitting criteria also with AIC, BIC, log(p) etc... For Machine Learning models we take down the MAPE for train divide(we divide the train into 75 and 25 percentage again to check the overfitting criteria) and whole test has been passed into the model which is captured as Model's MAPE score. Same for used in the Deep Learning methods like LSTM, dual-LSTM, Triple-LSTM and Bi-directional LSTM.

Chapter 5

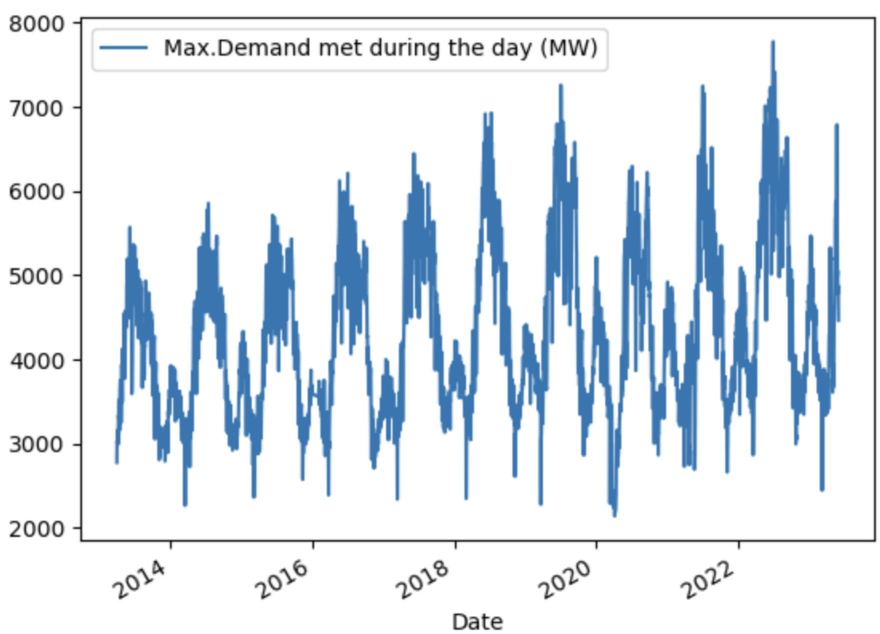
Forecasting Models

5.1 Augmented Dickey Fuller Test

The real world data tends to be always non-Stationary. A signal is said to be stationary if their statistical properties like mean, standard deviation, trend etc... doesn't change overtime. To check if the time-series is stationary or not, we use Augmented-Dickey Fuller Test where the null hypothesis is "the time series contains a unit root and is non-stationary". In the below subsections we will see the results of Augmented Dickey Fuller Test for each of the Imputation datasets.

5.1.1 No Imputation

The Figure below gives a plot of the whole dataset without dividing into train and test.



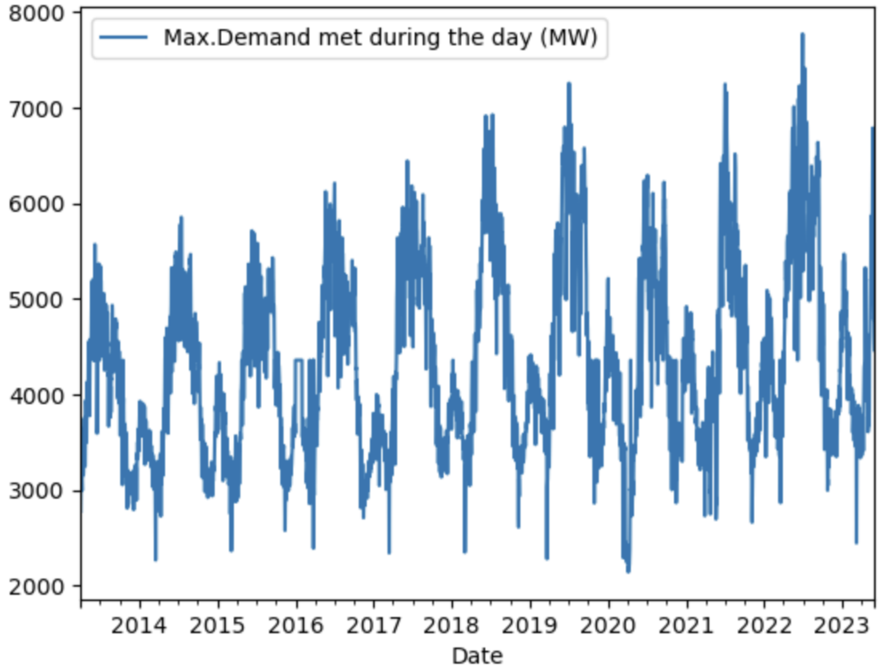
Now, we use the ADF test for no-imputation dataset test-statistic = -5.45 p-value = 2.55e-06

Now, we use the ADF test for no-imputation dataset for first difference to try to change the time-series to stationary. test-statistic = -10.403 p-value = 1.88e-18

Now, we use the ADF test for Second Difference, which is not suggested as the p-value is zero (over-differencing) test-statistic = -21.152 p-value = 0.0

5.1.2 Mean Imputation

The Figure below gives a plot of the whole dataset without dividing into train and test.



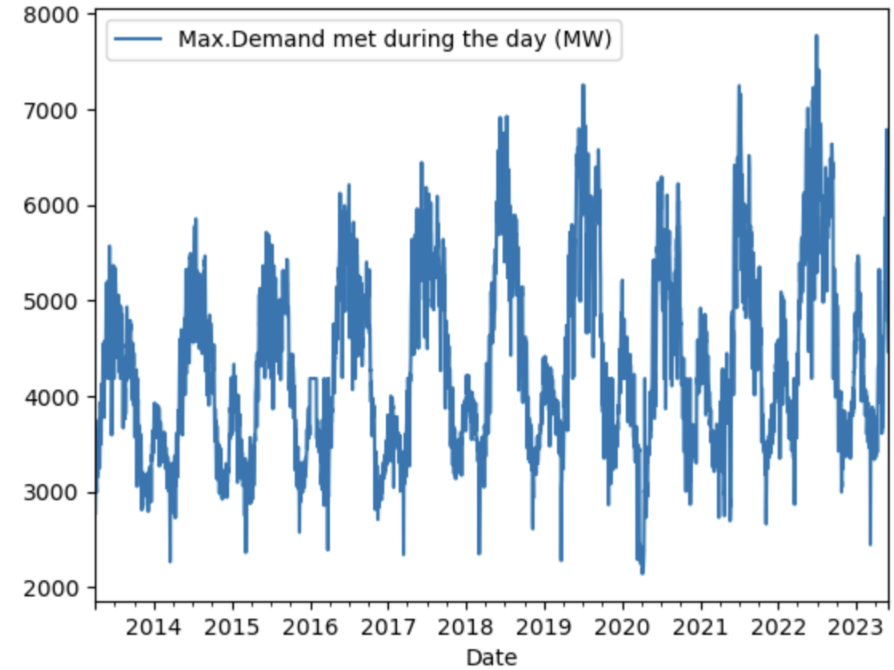
Now, we use the ADF test for Mean imputation dataset test-statistic = -5.393 p-value = 3.49e-06

Now, we use the ADF test for first difference to try to change the time-series to stationary. test-statistic = -10.073 p-value = 1.23e-17

Now, we use the ADF test for Second Difference, which is not suggested as the p-value is zero (over-differencing) test-statistic = -21.617 p-value = 0.0

5.1.3 Median Imputation

The Figure below gives a plot of the whole dataset without dividing into train and test.



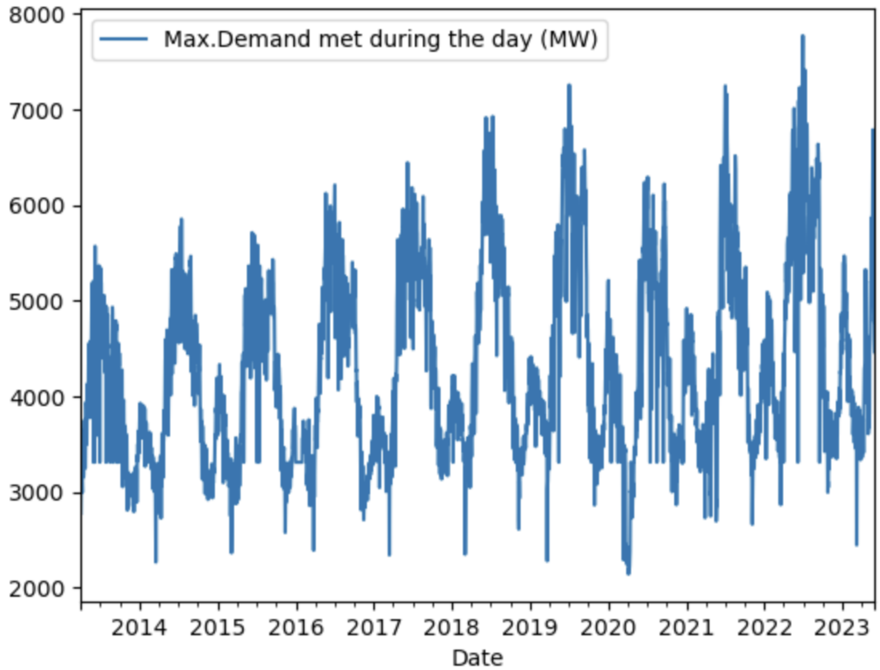
Now, we use the ADF test for Median imputation dataset test-statistic = -5.363 p-value = 4.042e-06

Now, we use the ADF test for first difference to try to change the time-series to stationary. test-statistic = -10.075 p-value = 1.223e-17

Now, we use the ADF test for Second Difference, which is not suggested as the p-value is zero (over-differencing) test-statistic = -21.686 p-value = 0.0

5.1.4 Mode Imputation

The Figure below gives a plot of the whole dataset without dividing into train and test.



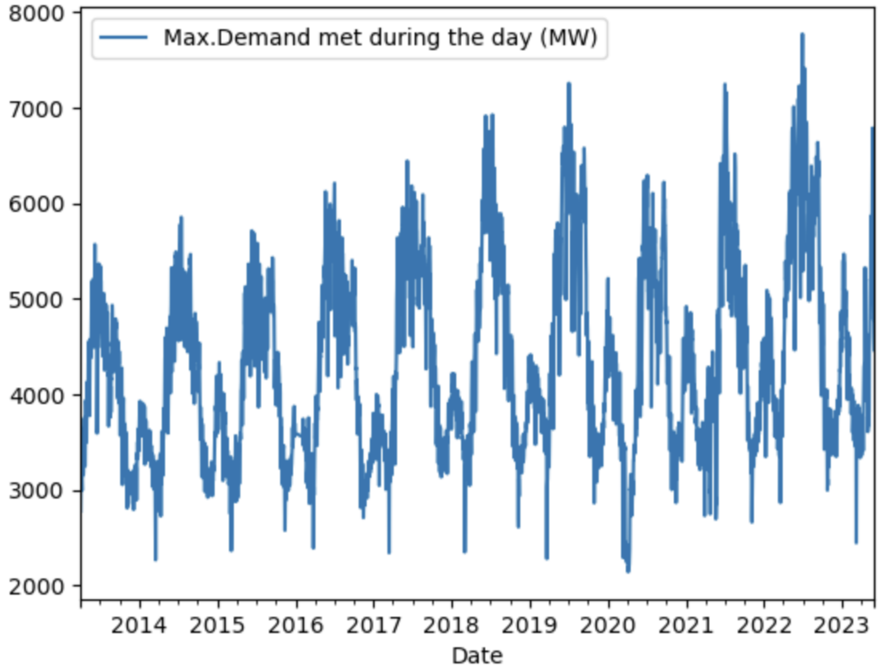
Now, we use the ADF test for Mode imputation dataset test-statistic = -5.227 p-value = 7.73e-06

Now, we use the ADF test for first difference to try to change the time-series to stationary. test-statistic = -10.258 p-value = 4.31e-18

Now, we use the ADF test for Second Difference, which is not suggested as the p-value is zero (over-differencing) test-statistic = -22.121 p-value = 0.0

5.1.5 Interpolation Imputation

The Figure below gives a plot of the whole dataset without dividing into train and test.



Now, we use the ADF test for Linear Interpolation imputation dataset test-statistic = -5.390 p-value = 3.53e-06

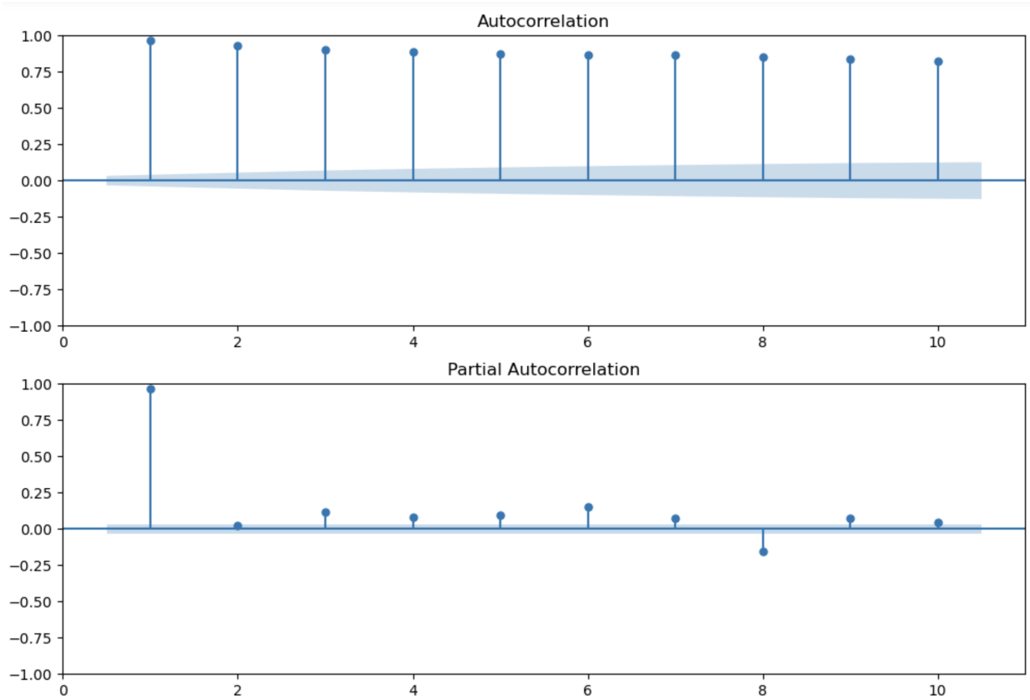
Now, we use the ADF test for first difference to try to change the time-series to stationary. test-statistic = -10.072 p-value = 1.24e-17

Now, we use the ADF test for Second Difference, which is not suggested as the p-value is zero (over-differencing) test-statistic = -21.44 p-value = 0.0

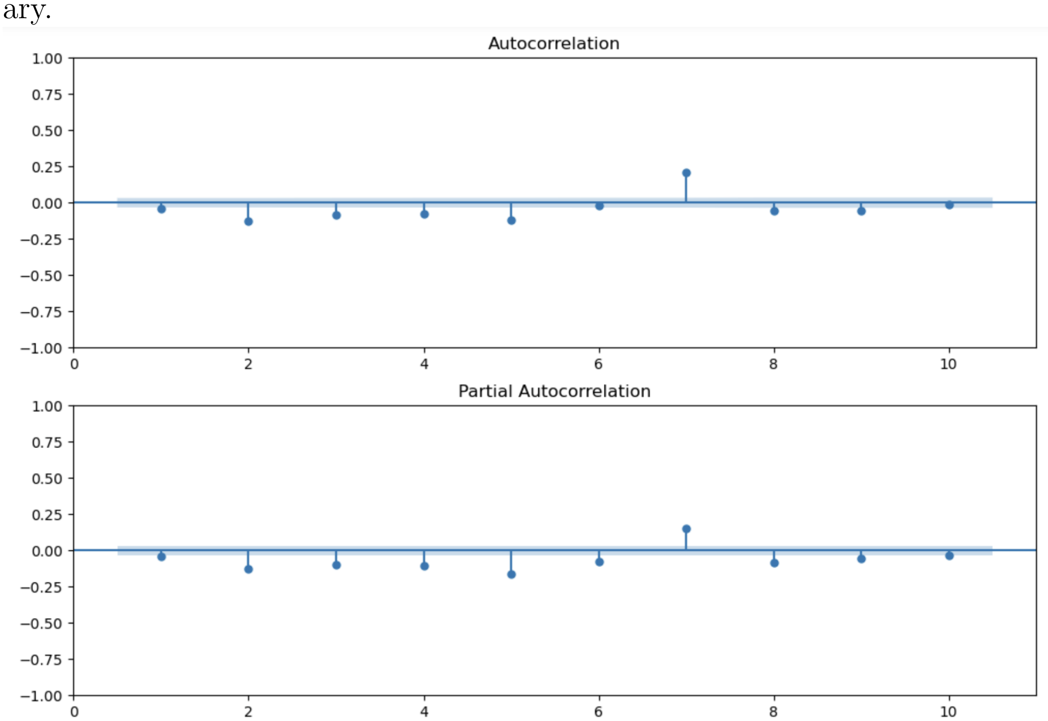
5.2 Analysis using ACF and PACF plots

5.2.1 No Imputation

Now, we plot the Auto-Correlation Function and Partial Auto-Correlation Function Graph for the original dataset

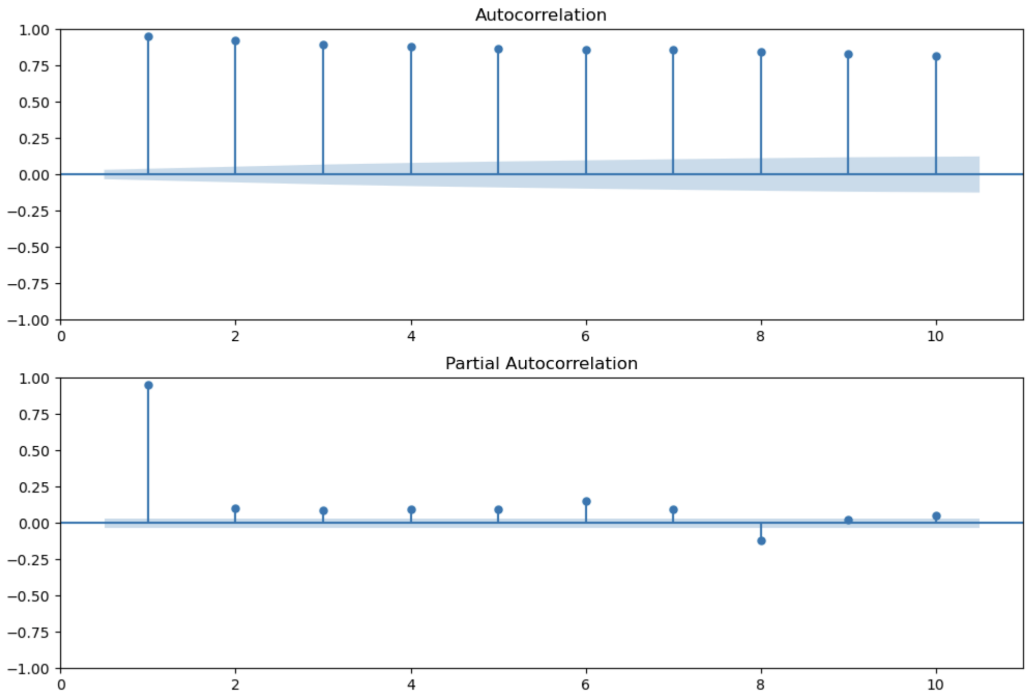


Now, we use the ADF test for first difference to try to change the time-series to stationary.

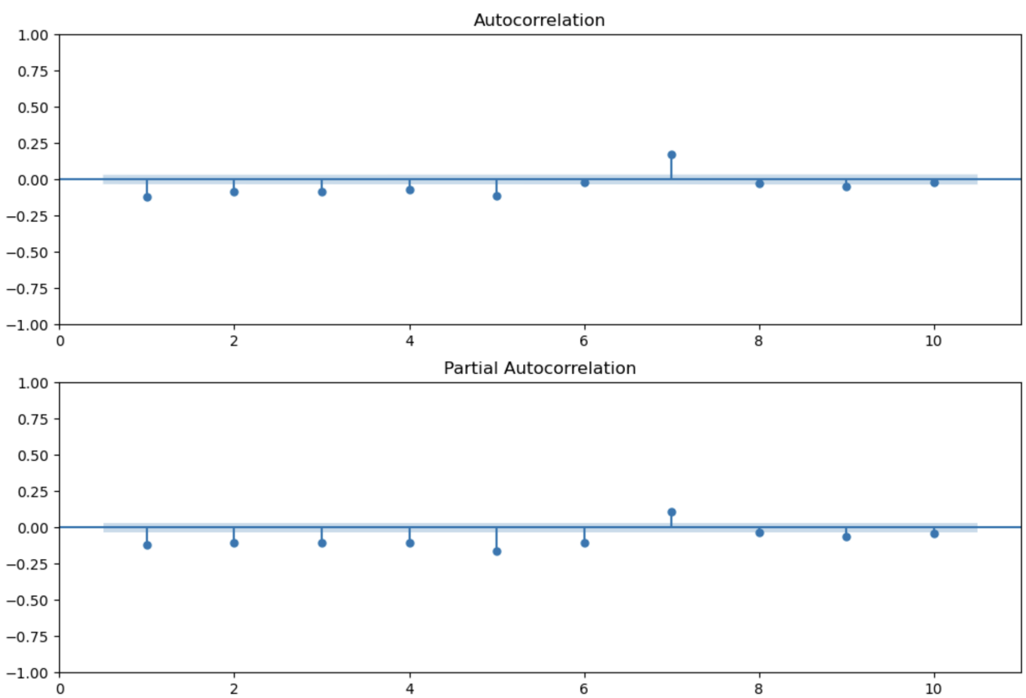


5.2.2 Mean Imputation

Now, we plot the Auto-Correlation Function and Partial Auto-Correlation Function Graph for the original dataset

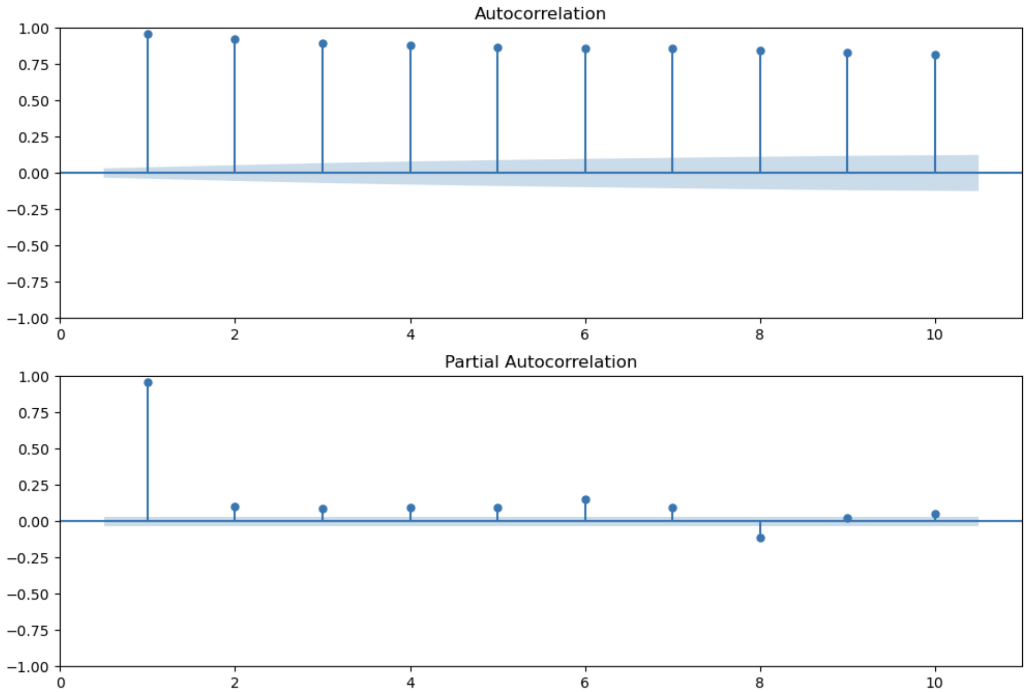


Now, we use the ADF test for first difference to try to change the time-series to stationary.

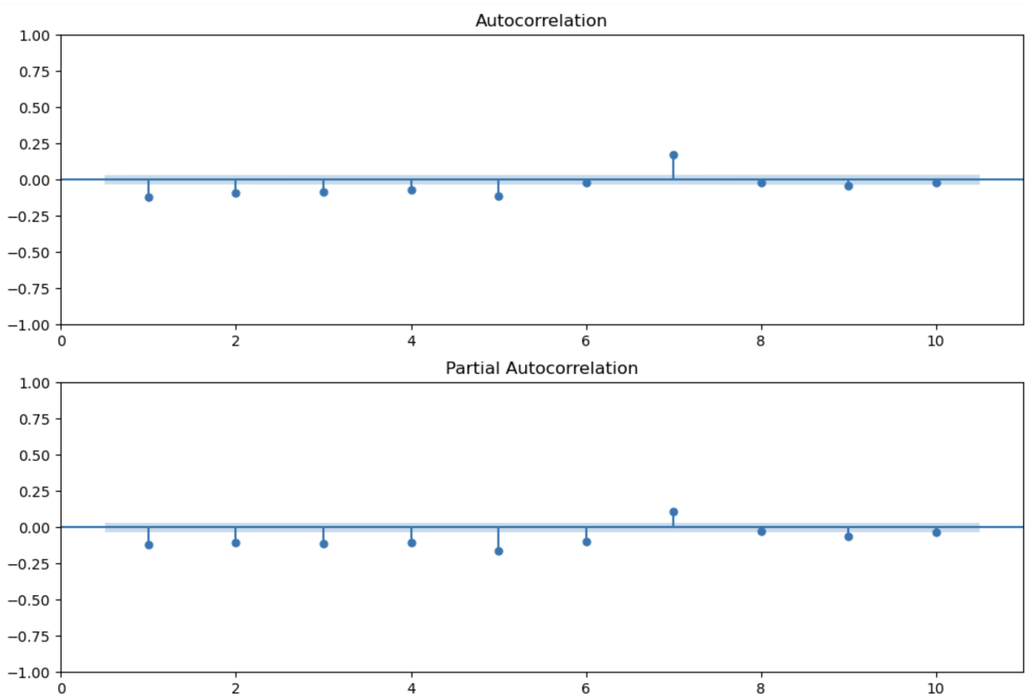


5.2.3 Median Imputation

Now, we plot the Auto-Correlation Function and Partial Auto-Correlation Function Graph for the original dataset

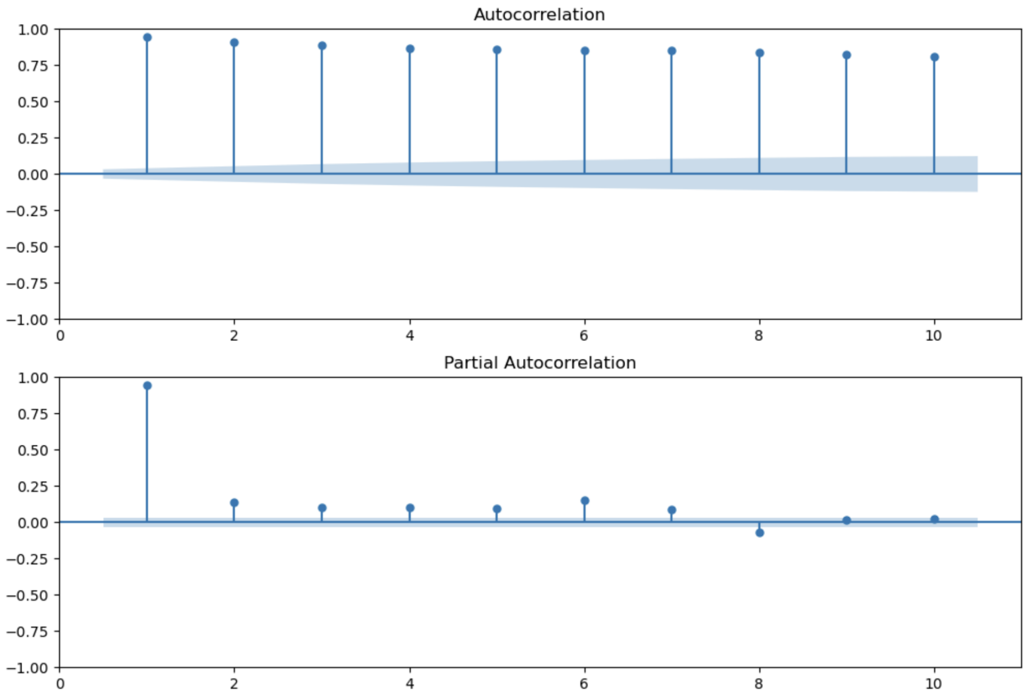


Now, we use the ADF test for first difference to try to change the time-series to stationary.

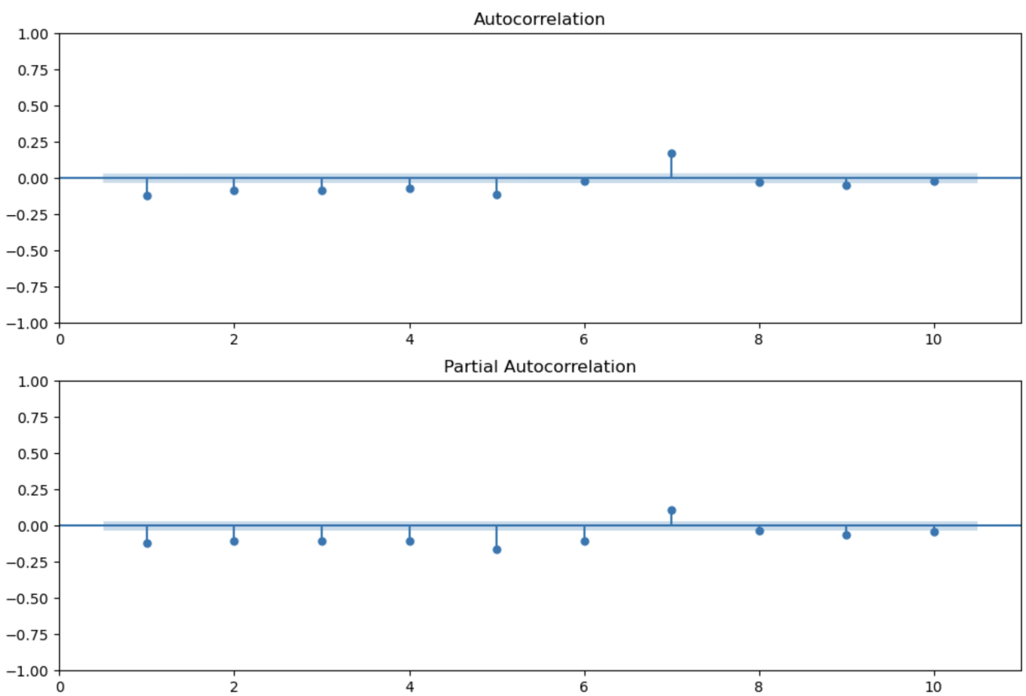


5.2.4 Mode Imputation

Now, we plot the Auto-Correlation Function and Partial Auto-Correlation Function Graph for the original dataset

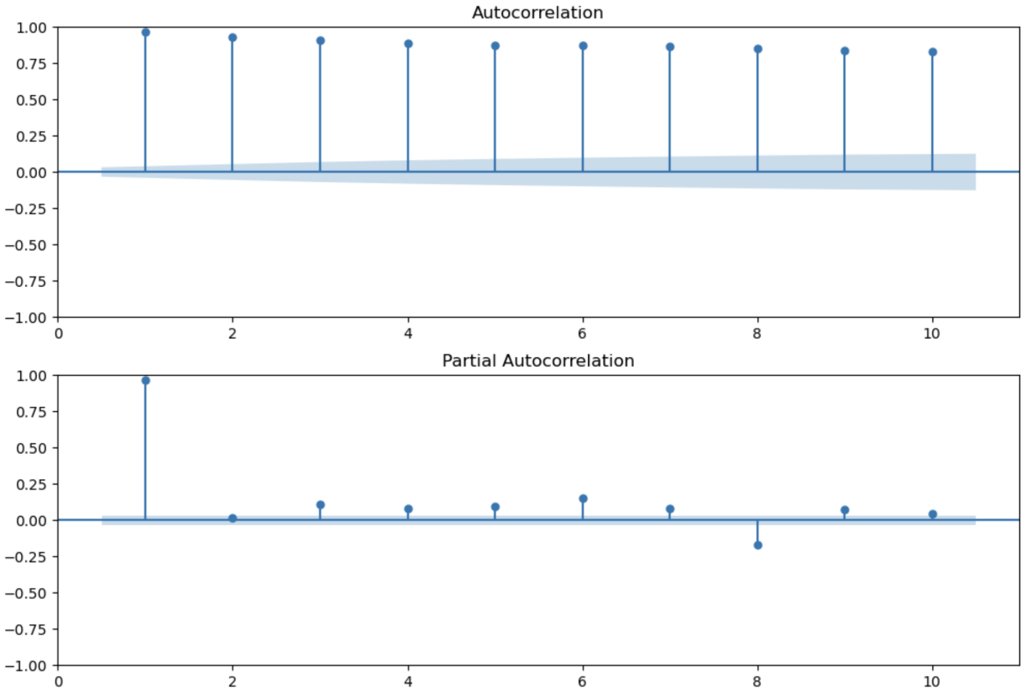


Now, we use the ADF test for first difference to try to change the time-series to stationary.

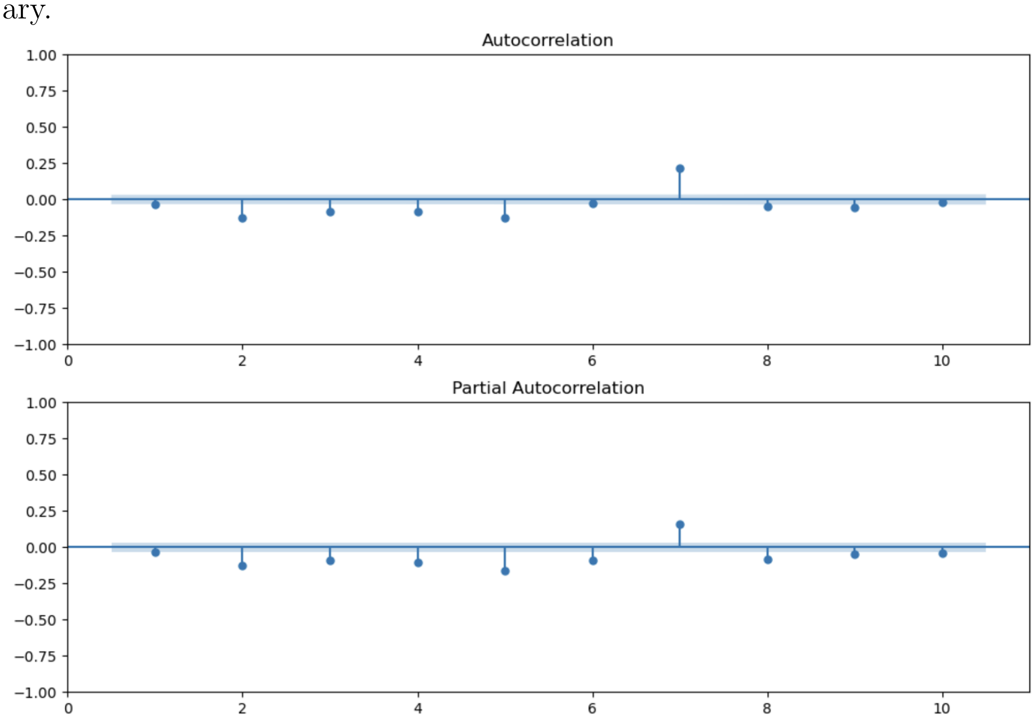


5.2.5 Interpolation Imputation

Now, we plot the Auto-Correlation Function and Partial Auto-Correlation Function Graph for the original dataset



Now, we use the ADF test for first difference to try to change the time-series to stationary.



5.3 Auto Regression and Moving Average model

In time series forecasting, the autoregressive moving average model of order (p, q), denoted as ARMA(p, q), is a popular approach. The ARMA(p, q) model combines the autoregressive

(AR) model of order p and the moving average (MA) model of order q. The ARMA(p, q) model assumes that the value of the time series at a given point is linearly dependent on the previous p values of the series and the previous q error terms. The formula for the ARMA(p, q) model is as follows:

$$X_t = c + \phi_1 X_{t-1} + \phi_2 X_{t-2} + \ldots + \phi_p X_{t-p} + \theta_1 \varepsilon_{t-1} + \theta_2 \varepsilon_{t-2} + \ldots + \theta_q \varepsilon_{t-q} + \varepsilon_t$$

In this formula:

- X_t represents the value of the time series at time t.
- c is the intercept or constant term.
- $\phi_1, \phi_2, \ldots, \phi_p$ are the coefficients of the autoregressive terms that capture the relationship between the current and previous values.
- $X_{t-1}, X_{t-2}, \ldots, X_{t-p}$ represent the lagged values of the time series.
- $\theta_1, \theta_2, \ldots, \theta_q$ are the coefficients of the moving average terms that capture the relationship between the current value and the previous error terms.
- $\varepsilon_{t-1}, \varepsilon_{t-2}, \ldots, \varepsilon_{t-q}$ represent the lagged error terms of the time series.
- ε_t is the error term at time t, which represents the random fluctuations or noise in the series.

To estimate the parameters $(\phi_1, \phi_2, \ldots, \phi_p, \theta_1, \theta_2, \ldots, \theta_q)$ and the intercept (c) of the ARMA(p, q) model, various estimation techniques can be used, such as maximum likelihood estimation.

Once the parameters are estimated, the ARMA(p, q) model can be used for forecasting by substituting the lagged values and lagged error terms of the time series into the formula to predict future values.

Note that the ARMA(p, q) model assumes stationarity of the time series, and it is a flexible model that can capture both autoregressive and moving average components in the data.

5.4 Seasonal Auto-Regressive Models

The Seasonal Autoregressive Integrated Moving Average (SARIMA) model is a time series forecasting model that extends the Autoregressive Integrated Moving Average (ARIMA) model to account for seasonality. SARIMA combines the components of ARIMA with seasonal differencing and seasonal autoregressive and moving average terms.

The SARIMA(p, d, q)(P, D, Q, s) model is defined by the following equations:

Autoregressive (AR) component: AR(p): $Y_t = \phi_1 Y_{t-1} + \phi_2 Y_{t-2} + \ldots + \phi_p Y_{t-p} + \varepsilon_t$

Integrated (I) component: I(d): $Y'_t = (1 - B)^d Y_t$, where B is the backshift operator $(BY_t = Y_{t-1})$

Moving Average (MA) component: MA(q): $Y_t = \theta_1\varepsilon_{t-1} + \theta_2\varepsilon_{t-2} + \ldots + \theta_q\varepsilon_{t-q} + \varepsilon_t$

Seasonal Autoregressive (SAR) component: SAR(P): $Y_t = \Phi_1 Y_{t-s} + \Phi_2 Y_{t-2s} + \ldots + \Phi_P Y_{t-Ps} + \varepsilon_t$

Seasonal Moving Average (SMA) component: SMA(Q): $Y_t = \Theta_1\varepsilon_{t-s} + \Theta_2\varepsilon_{t-2s} + \ldots + \Theta_Q\varepsilon_{t-Qs} + \varepsilon_t$

where: Y_t is the observed time series at time t ε_t is the error term (also known as the residual) at time t p, d, q are the non-seasonal AR, I, MA orders, respectively P, D, Q are the seasonal SAR, I, SMA orders, respectively s is the seasonal period or frequency (e.g., 12 for monthly data, 4 for quarterly data, etc.) $\phi_1, \phi_2, \ldots, \phi_p$ are the non-seasonal autoregressive coefficients $\theta_1, \theta_2, \ldots, \theta_q$ are the non-seasonal moving average coefficients $\Phi_1, \Phi_2, \ldots, \Phi_P$ are the seasonal autoregressive coefficients $\Theta_1, \Theta_2, \ldots, \Theta_Q$ are the seasonal moving average coefficients

Chapter 6

Comprehensive Results

6.1 ARIMA model Results

6.1.1 No Imputation

Models	Order	test_MAPE	test_MAE	train_MAPE	train_MAE	AIC	BIC
AR/ MA models							
	1,0,0	19.483	190.131	26.125	190.125	50534.716	50553.290
	2,0,0	19.516	190.126	26.126	190.126	50534.362	50559.127
	1,1,0	21.903	191.545	26.602	191.545	50575.065	50587.447
	1,2,0	90.594	248.288	27.274	248.28	52837.437	52829.454
	0,0,1	16.617	483.421	21.619	483.421	56254.968	56273.542
	0,1,1	21.811	191.678	26.594	191.678	50572.554	50584.936
	0,2,1	21.9037	191.545	26.602	191.545	50578.459	50590.841
ARMA models							
AIC and BIC	8,0,8	18.157	175.183	26.332	175.183	50067.375	50178.816
	8,1,8	18.814	180.292	26.453	180.292	50148.885	50254.131
	9,0,7	18.201	174.845	26.226	174.845	50067.729	50179.170
	9,1,7	18.332	175.647	26.453	175.647	50055.038	50160.283
	8,0,9	18.135	175.271	26.280	175.271	50068.705	50186.337
	8,1,9	18.274	175.240	26.464	175.240	50059.330	50170.766
auto-arima	5,1,3	18.844	183.187	26.454	183.187	50190.242	50245.961

6.1.2 Mean Imputation

Models	Order	test_MAPE	test_MAE	train_MAPE	train_MAE	AIC	BIC
AR/MA models							
	1,0,0	19.186	198.187	25.749	198.187	52184.042	52202.676
	2,0,0	19.352	198.835	25.756	198.835	52144.192	52169.037
	1,1,0	21.628	200.508	26.276	200.508	52188.319	52200.741
	1,2,0	86.037	260.44	27.046	260.44	54529.566	54542.004
	0,0,1	16.616	491.065	21.290	491.065	57600.735	57619.368
	0,1,1	21.322	201.046	26.248	201.04	52171.044	52183.466
	0,2,1	21.628	200.508	26.276	200.508	52243.646	52256.067
ARMA models							
AIC and BIC	9,0,8	18.266	180.575	26.027	180.575	51635.298	51753.311
	9,1,8	18.069	184.905	26.135	184.905	51687.745	51799.541
	8,0,9	18.400	182.063	25.965	182.063	51663.391	51781.404
	8,1,9	18.366	184.291	26.173	184.291	51695.921	51807.718
	8,0,8	18.518	181.670	25.977	181.670	51670.186	51781.988
	8,1,8	18.324	181.714	26.093	181.714	51637.395	51742.981
auto-arima	5,1,3	18.563	191.315	26.085	191.315		

6.1.3 Median Imputation

Models	Order	test_MAPE	test_MAE	train_MAPE	train_MAE	AIC	BIC
AR/MA models							
	1,0,0	19.203	198.168	25.740	198.168	52180.550	52199.184
	2,0,0	19.363	198.758	25.748	198.758	52142.980	52167.825
	1,1,0	21.636	200.371	26.268	200.371	52187.390	52199.812
	1,2,0	86.201	260.024	27.038	260.024	54527.913	54540.351
	0,0,1	16.646	492.204	21.271	492.204	57600.941	57619.575
	0,1,1	21.336	200.903	26.241	200.903	52170.583	52183.005
	0,2,1	21.636	200.371	26.268	200.371	52239.944	52252.365
ARMA models							
AIC and BIC	9,0,8	18.288	180.307	26.032	180.307	51635.740	51753.753
	9,1,8	18.115	184.513	26.121	184.513	51681.885	51793.682
	8,0,8	18.506	181.701	25.958	181.701	51674.400	51786.202
	8,1,8	18.421	181.122	26.104	181.122	51641.077	51746.663
	8,0,9	18.407	182.128	25.939	182.128	51668.680	51786.693
	8,1,9	18.502	184.418	26.152	184.418	51701.538	51813.335
auto-arima	5,1,3	18.676	190.163	26.098	190.163	51805.777	51861.675

6.1.4 Mode Imputation

Models	Order	test_MAPE	test_MAE	train_MAPE	train_MAE	AIC	BIC
AR/MA models							
	1,0,0	19.203	198.168	25.740	198.168	52180.550	52199.184
	2,0,0	19.363	198.758	25.748	198.758	52142.980	52167.825
	1,1,0	21.636	200.371	26.268	200.371	52187.390	52199.812
	1,2,0	86.201	260.024	27.038	260.024	54527.913	54540.351
	0,0,1	16.646	492.204	21.271	492.204	57600.941	57619.575
	0,1,1	21.336	200.903	26.241	200.903	52170.583	52183.005
	0,2,1	21.636	200.371	26.268	200.371	52239.944	52252.365
ARMA models							
AIC and BIC	9,0,8	18.288	180.307	26.032	180.307	51635.740	51753.753
	9,1,8	18.115	184.513	26.121	184.513	51681.885	51793.682
	8,0,8	18.506	181.701	25.958	181.701	51674.400	51786.202
	8,1,8	18.421	181.122	26.104	181.122	51641.077	51746.663
	8,0,9	18.407	182.128	25.939	182.128	51668.680	51786.693
	8,1,9	18.502	184.418	26.152	184.418	51701.538	51813.335
auto-arima	5,1,3	18.676	190.163	26.098	190.163	51805.777	51861.675

6.1.5 Linear Interpolation Imputation

Models	Order	test_MAPE	test_MAE	train_MAPE	train_MAE	AIC	BIC
AR/MA models							
	1,0,0	19.552	186.541	26.051	186.541	51429.997	51448.631
	2,0,0	19.573	186.540	26.051	186.540	51430.941	51455.786
	1,1,0	21.936	187.723	26.516	187.723	51472.149	51484.571
	1,2,0	90.903	242.157	27.171	242.157	53740.953	53753.391
	0,0,1	16.678	480.814	21.549	480.814	57337.414	57356.048
	0,1,1	21.870	187.838	26.510	187.838	51470.598	51483.020
	0,2,1	21.936	187.723	26.516	187.723	51473.203	51485.624
ARMA models							
AIC and BIC	8,0,8	18.409	167.296	26.271	167.296	50801.251	50913.053
	8,1,8	18.683	168.762	26.385	168.762	50838.995	50944.581
	9,0,8	18.223	167.355	26.175	167.355	50814.366	50932.379
	9,1,8	18.440	169.103	26.385	169.103	50807.144	50918.941
	9,0,7	18.157	168.363	26.114	168.363	50836.848	50948.650
	9,1,7	19.432	168.581	26.363	168.581	50840.746	50946.332
auto-arima	5,1,4	19.114	177.509	26.389	177.509	51004.208	51066.318

6.2 Machine Learning Feature Extraction model Results

6.2.1 No Imputation

No Imputation

	window_size	prev_days_data	Ridge_MAPE	Lasso_MAPE	GBR_MAPE	SVR_MAPE
shift_features	10	30	0.187	0.178	0.131	0.186
shift+date_features	10	30	0.187	0.178	0.131	0.186
shift_features	0	30	0.187	0.178	0.131	0.186
shift+date_features	0	30	0.187	0.178	0.131	0.186
shift_features	10	7	0.172	0.150	0.152	0.204
shift+date_features	10	7	0.172	0.152	0.144	0.213
shift_features	0	7	0.061	0.061	0.059	0.120
shift+date_features	0	7	0.060	0.060	0.059	0.134
percentile_features	10	30	0.127	0.159	0.182	0.226
percentile_features	0	30	0.136	0.136	0.141	0.147
percentile_features	10	7	0.161	0.108	0.159	0.208
percentile_features	0	7	0.090	0.090	0.081	0.097
basic_features	10	30	0.185	0.127	0.181	0.226
basic_features	0	30	0.185	0.127	0.180	0.226
basic_features	10	7	0.175	0.102	0.152	0.212
basic_features	0	7	0.175	0.102	0.152	0.212

6.2.2 Mean Imputation

Mean Imputation

	window_size	prev_days_data	Ridge_MAPE	Lasso_MAPE	GBR_MAPE	SVR_MAPE
shift_features	10	30	0.182	0.173	0.132	0.184
shift+date_features	10	30	0.182	0.173	0.132	0.184
shift_features	0	30	0.182	0.173	0.132	0.184
shift+date_features	0	30	0.182	0.173	0.132	0.184
shift_features	10	7	0.172	0.152	0.145	0.201
shift+date_features	10	7	0.173	0.154	0.131	0.209
shift_features	0	7	0.065	0.065	0.063	0.120
shift+date_features	0	7	0.063	0.063	0.058	0.134
percentile_features	10	30	0.131	0.186	0.078	0.226
percentile_features	0	30	0.134	0.134	0.129	0.147
percentile_features	10	7	0.164	0.122	0.160	0.205
percentile_features	0	7	0.089	0.089	0.079	0.098
basic_features	10	30	0.185	0.139	0.156	0.220
basic_features	0	30	0.185	0.139	0.154	0.220
basic_features	10	7	0.174	0.103	0.175	0.205
basic_features	0	7	0.174	0.103	0.175	0.205

6.2.3 Median Imputation

Median Imputation

	window_size	prev_days_data	Ridge_MAPE	Lasso_MAPE	GBR_MAPE	SVR_MAPE
shift_features	10	30	0.182	0.173	0.134	0.181
shift+date_features	10	30	0.182	0.173	0.134	0.181
shift_features	0	30	0.182	0.173	0.134	0.181
shift+date_features	0	30	0.182	0.173	0.134	0.181
shift_features	10	7	0.173	0.153	0.140	0.206
shift+date_features	10	7	0.174	0.154	0.137	0.214
shift_features	0	7	0.065	0.065	0.064	0.119
shift+date_features	0	7	0.063	0.063	0.061	0.134
percentile_features	10	30	0.132	0.184	0.084	0.229
percentile_features	0	30	0.136	0.136	0.136	0.147
percentile_features	10	7	0.164	0.121	0.158	0.211
percentile_features	0	7	0.089	0.089	0.078	0.098
basic_features	10	30	0.187	0.137	0.124	0.223
basic_features	0	30	0.187	0.137	0.124	0.223
basic_features	10	7	0.176	0.105	0.124	0.212
basic_features	0	7	0.176	0.105	0.122	0.212

6.2.4 Mode Imputation

Mode Imputation

	window_size	prev_days_data	Ridge_MAPE	Lasso_MAPE	GBR_MAPE	SVR_MAPE
shift_features	10	30	0.186	0.176	0.139	0.183
shift+date_features	10	30	0.186	0.176	0.139	0.183
shift_features	0	30	0.186	0.176	0.139	0.183
shift+date_features	0	30	0.186	0.176	0.139	0.183
shift_features	10	7	0.177	0.157	0.144	0.209
shift+date_features	10	7	0.178	0.160	0.130	0.217
shift_features	0	7	0.067	0.067	0.064	0.120
shift+date_features	0	7	0.065	0.065	0.062	0.136
percentile_features	10	30	0.136	0.155	0.099	0.227
percentile_features	0	30	0.136	0.136	0.131	0.145
percentile_features	10	7	0.164	0.112	0.146	0.211
percentile_features	0	7	0.091	0.091	0.079	0.097
basic_features	10	30	0.187	0.130	0.159	0.230
basic_features	0	30	0.187	0.130	0.159	0.230
basic_features	10	7	0.177	0.103	0.133	0.213
basic_features	0	7	0.177	0.103	0.136	0.213

6.2.5 Linear Interpolation Imputation

Linear Interpolation Imputation

	window_size	prev_days_data	Ridge_MAPE	Lasso_MAPE	GBR_MAPE	SVR_MAPE
shift_features	10	30	0.185	0.175	0.133	0.183
shift+date_features	10	30	0.185	0.175	0.133	0.183
shift_features	0	30	0.185	0.175	0.133	0.183
shift+date_features	0	30	0.185	0.175	0.133	0.183
shift_features	10	7	0.173	0.152	0.163	0.207
shift+date_features	10	7	0.175	0.154	0.143	0.216
shift_features	0	7	0.061	0.061	0.060	0.119
shift+date_features	0	7	0.060	0.060	0.058	0.134
percentile_features	10	30	0.128	0.155	0.118	0.228
percentile_features	0	30	0.136	0.136	0.137	0.147
percentile_features	10	7	0.162	0.109	0.171	0.211
percentile_features	0	7	0.090	0.090	0.079	0.097
basic_features	10	30	0.185	0.126	0.171	0.228
basic_features	0	30	0.185	0.126	0.171	0.228
basic_features	10	7	0.176	0.103	0.163	0.214
basic_features	0	7	0.176	0.103	0.165	0.214

6.3 Deep Learning model Results

	no_null_MAPE	mean_MAPE	median_MAPE	mode_MAPE	Inter-linear_MAPE
1-layer LSTM	99.076	98.647	99.058	98.784	98.447
1-Dense Layer(50)	18.571	18.340	18.325	18.529	18.468
Bi-directional LSTM	97.395	97.078	97.729	97.095	97.769
2-Dense Layer (50)	18.560	18.329	18.448	18.516	18.478
4-Dense Layer(50)	18.909	18.365	18.267	18.602	18.414

Chapter 7

Conclusion

As the Base models concluded we try to integrate more data into the PM Gati-Shakti Scheme to validate the case study given in the section of Introduction. The future work also focusses on using Reinforcement Learning for model selection for a much larger data with integrated ministries in the Union Territory of Delhi.

In conclusion, our study highlights the importance of integrating more data into the PM Gati-Shakti Scheme in order to validate the findings presented in the Introduction section. The base models provide a preliminary understanding of the scheme's potential, but further data incorporation is crucial for robust conclusions. By expanding the scope of our analysis to encompass a wider range of variables and factors, we can enhance the accuracy and reliability of the case study.

Furthermore, our future work will focus on employing Reinforcement Learning techniques for model selection. This approach is particularly relevant when dealing with a larger dataset that integrates ministries within the Union Territory of Delhi. Reinforcement Learning algorithms can effectively evaluate and select the most suitable models by considering the complex interactions and dependencies between different variables. By leveraging the power of machine learning and advanced analytics, we can make informed decisions that lead to better outcomes and enhanced efficiency within the PM Gati-Shakti Scheme.

In summary, our research emphasizes the need for data integration and the application of Reinforcement Learning in the context of the PM Gati-Shakti Scheme. These steps will contribute to a more comprehensive understanding of the scheme's impact and enable evidence-based decision-making for the integration of ministries in the Union Territory of Delhi. By continuously improving our analytical approaches, we can enhance the effectiveness of the scheme and drive positive socio-economic outcomes.

References

[1] Christos L. Athanasiadis, Georgios Tsoumplekas, Antonios Chrysopoulos, and Dimitrios I. Doukas. Peak demand forecasting: A comparative analysis of state-of-the-art machine learning techniques. In *2022 2nd International Conference on Energy Transition in the Mediterranean Area (SyNERGY MED)*, pages 1–6, 2022.

[2] Alisha Banga and SC Sharma. Electricity demand forecasting models at hourly and daily level: A comparative study. In *2022 International Conference on Advanced Computing Technologies and Applications (ICACTA)*, pages 1–5, 2022.

[3] Central Electricity Authority, Government of India. All india electricity statistics. Report, May 2022.

[4] Delhi Transco Limited. Dtl substations, Year of publication (if available). Accessed on Day Month Year.

[5] Pavan Kumar Dharmoju, Karthik Yeluripati, Jahnavi Guduri, and Kowstubha Palle. Forecasting electrical demand for the residential sector at the national level using deep learning. In *2021 International Conference on Artificial Intelligence and Machine Vision (AIMV)*, pages 1–6, 2021.

[6] Kakoli Goswami and Aditya Bihar Kandali. Electricity demand prediction using data driven forecasting scheme: Arima and sarima for real-time load data of assam. In *2020 International Conference on Computational Performance Evaluation (ComPE)*, pages 570–574, 2020.

[7] Government of India. Daily data: Sector-wise and mode-wise installed capacity, Year of publication (if available). Accessed on Day Month Year.

[8] Shashwat Jha and Vishvaditya Luhach. Indian peak power demand forecasting: Transformer based implementation of temporal architecture. In *2022 IEEE Global Conference on Computing, Power and Communication Technologies (GlobConPT)*, pages 1–5, 2022.

[9] Vishnu Vardhan Sai Lanka, Millend Roy, Shikhar Suman, and Shivam Prajapati. Renewable energy and demand forecasting in an integrated smart grid. In *2021 Innovations in Energy Management and Renewable Resources(52042)*, pages 1–6, 2021.

[10] Ministry of Steel, Government of India. Ease of doing business, Year of publication (if available). Accessed on Day Month Year.

[11] Anil K Pandey, Kishan Bhushan Sahay, M. M Tripathi, and D Chandra. Short-term load forecasting of uppcl using ann. In *2014 6th IEEE Power India International Conference (PIICON)*, pages 1–6, 2014.

[12] Power Grid Corporation of India Limited. Power grid corporation of india limited, Year of publication (if available). Accessed on Day Month Year.

[13] Power System Operation Corporation Limited (POSOCO). POSOCO official website. https://posoco.in, 2023.

[14] Power System Operation Corporation Limited (POSOCO). Daily report, Year of publication. Accessed on Day Month Year.

[15] Srinivasa Rao Rallapalli and Sajal Ghosh. Forecasting monthly peak demand of electricity in india—a critique. *Energy Policy*, 45:516–520, 2012.

[16] Reserve Bank of India. Title of the document, 2022. Available at: https://rbidocs.rbi.org.in/rdocs/Publications/PDFs/0HBS19112022_FLFE4F2F9158294692B030A251E00555F8.PDF.

[17] Saravanan S, Kannan Subramanian, and C. Thangaraj. India's electricity demand forecast using regression analysis and artificial neural networks based on principal components. *ICTACT Journal on Soft Computing*, 2:365–370, 07 2012.

[18] S. Saravanan, S. Kannan, R. Nithya, and C. Thangaraj. Modeling and prediction of india's electricity demand using fuzzy logic. In *2014 International Conference on Circuits, Power and Computing Technologies [ICCPCT-2014]*, pages 93–96, 2014.

[19] Navneet Kumar Singh, Asheesh Kumar Singh, and Manoj Tripathy. A comparative study of bpnn, rbfnn and elman neural network for short-term electric load forecasting: A case study of delhi region. In *2014 9th International Conference on Industrial and Information Systems (ICIIS)*, pages 1–6, 2014.

[20] Manish Uppal, Rumita Kumari, and Saurabh Shrivastava. An ensemble approach for short-term load forecasting for discoms of delhi across the covid-19 scenario. In *2021 9th IEEE International Conference on Power Systems (ICPS)*, pages 1–6, 2021.

[21] Carlos E. Velasquez, Matheus Zocatelli, Fidellis B.G.L. Estanislau, and Victor F. Castro. Analysis of time series models for brazilian electricity demand forecasting. *Energy*, 247:123483, 2022.