Time Series Analysis and Forecasting

Submitted by

Ankit Pandey
Roll No: 22223018
MCA- V Semester
(Session: 2022-2025)

Submitted to

Dr Priyanka Tripathi

Department of Computer Applications
National Institute of Technology Raipur

Dr. Tanmoy Kanti Das

Supervisor

Department of Computer Applications
National Institute of Technology Raipur

Department of Computer Applications
National Institute of Technology, Raipur

DECLARATION

I, **Ankit Pandey** roll No. **22223018**, hereby declare that the work done in the project entitled **Time series Analysis and Forecasting** is done on my own.

I confirm that:

- The work contained in this report is original and has been done by me under the guidance of Assistant Professor Dr. Tanmoy Kanti Das, Department of Computer Applications, National Institute of Technology Raipur.
- The work has not been submitted to any other institute for any other degree or diploma;
- I have followed the guidelines provided by the institute in preparing the project report;
- I have conformed to ethical norms and guidelines while writing the project report.
- Whenever I have used materials such as data, model, figures, and text from other source, I have given due credit to them by citing them in the text of the report and giving their details in the references.

Place : Raipur

Date : 10/11/2024

Name : Ankit Pandey
Roll No: 22223018
MCA- V Semester

CERTIFICATE FROM THE SUPERVISOR

This is to certify that the project entitled **Time series Analysis and Forecasting** has been carried out by **Ankit Pandey** roll No. **22223018** , MCA Vth Semester, under my guidance.

The matter embodied in this project has not been submitted for the award of any other degree or diploma, to the best of my knowledge.

Place: Raipur (Signature)

Date:10/11/2024

Acknowledgement

I would like to express my sincere thanks to Assistant Professor Dr Tanmoy Kanti Das, National Institute of Technology Raipur, for his valuable guidance and support in completing my project. I would also like to express my gratitude towards our Professor & Head of Department Dr Priyanka Tripathi for giving me this great opportunity to do a project on **Time series Analysis and Forecasting**. Without their support and suggestions, this project would not have been completed.

Ankit Pandey

22223018

Index

Sl. No.	Section Name	Page no.
1	Introduction	1
2	Project Overview	2
3	System model	3-4
4	Methodology	5-6
5	Implementation	7-8
6	Results and discussion	9-10
7	Conclusion and Future work	10
9	Biblography \ References	10

Table of Content

1. **Introduction** 1
   - 1.1. Overview of Time Series Forecasting 1
   - 1.2. Importance of Forecasting Electricity Usage 1
   - 1.3. Problem Statement 1
2. **Project Overview** 2
   - 2.1. Objective of the Project 2
   - 2.2. Scope of the Forecasting System 2
   - 2.3. Key Technologies Used (Prophet, Streamlit, etc.) 2
3. **System Model** 3-4
   - 3.1. Data Flow and Architecture of the Forecasting System 3
   - 3.2. Overview of Machine Learning Models Used (SARIMA, XGBoost, Prophet)...... 4
4. **Methodology** 5-6
   - 4.1. Data Collection and Preprocessing 5
   - 4.2. Exploratory Data Analysis (EDA) 5
   - 4.3. Model Training Process 6
   - 4.4. Choosing Prophet for Forecasting 6
5. **Implementation** 7-8
   - 5.1. Building the Forecasting Model with Prophet 7
   - 5.2. Creating a Streamlit Web Application for User Interaction 7
   - 5.3. Deploying the Application for Online Access 8
6. **Results and Discussion** 9-10
   - 6.1. Model Performance and Evaluation 9
   - 6.2. Visualizing the Forecasts and Comparing Results 9
   - 6.3. Insights from the Forecasts 10
7. **Conclusion and Future Work** 10
   - 7.1. Key Findings and Insights 10
   - 7.2. Opportunities for Model Improvement 10
   - 7.3. Future Enhancements and Application Areas 10
8. **Biblography / Reference**.................... 10

Time Series Analysis and Forecasting

1. Introduction

1.1. Overview of Time Series Forecasting

Time series forecasting is a statistical technique used to predict future values based on previously observed values. The primary goal is to use historical data to make informed predictions about future trends and patterns. This is essential in many fields such as finance, economics, healthcare, and energy consumption. Time series forecasting models aim to capture the temporal dependencies in data to generate accurate predictions. Common methods for time series forecasting include statistical models like ARIMA, machine learning models like XGBoost, and more recently, deep learning approaches.

In time series data, the most crucial feature is the time element itself. Thus, these models focus on recognizing the seasonality, trends, and other patterns that evolve over time. For example, predicting stock prices, weather forecasts, or electricity usage relies on time series data for accurate forecasting.

1.2. Importance of Forecasting Electricity Usage

Electricity usage forecasting plays a significant role in power grid management, optimizing energy resources, and planning for future energy demands. The need for accurate predictions of electricity consumption has grown, driven by the increasing complexity of energy distribution networks and the rise of renewable energy sources, which have variable outputs. Effective forecasting helps power utilities to balance electricity supply and demand, ensuring that energy production aligns with consumption patterns.

We are using PJME hourly energy consumption dataset in this project ,Accurate electricity demand forecasts allow utility companies to plan for peak loads, reduce energy waste, and improve overall grid reliability. For consumers, it can help reduce costs by enabling demand-side management, while also fostering the efficient integration of renewable energy sources into the grid.

1.3. Problem Definition Statement

The problem we aim to solve is:

“Given historical electricity usage data, how can we forecast the load (electricity consumption) for the next upcoming days?”

Accurate forecasts help electricity retailers plan their supply, manage resources, and reduce costs. This project focuses on using historical data from **PJME Dataset**, to forecast electricity consumption for the next year. The goal is to use this data to predict future electricity load and help retailers make better decisions regarding energy procurement.

2. Project Overview

2.1 **Objective of the Project**

The primary objective of this project is to perform data preprocessing and exploratory data analysis (EDA), followed by a comparative study between different time series forecasting techniques such as **XGBoost**, **SARIMA**, and **Facebook Prophet**. The goal is to identify the best model and then use it to develop a **Streamlit app** that allows any user to easily forecast time series data, regardless of the dataset.

In this case, we are focusing on predicting electricity usage for the next upcoming days (lets say 365 days) . By analyzing historical power consumption data, the developed model will assist electricity retailers in forecasting future demand, helping them plan for electricity procurement, pricing strategies, and other operational needs. The model is designed to be accurate, efficient, and scalable, making it suitable for real-world applications.

2.2 Scope of the Forecasting System

The scope of this project includes the development of a time series forecasting system that can predict electricity usage. The system will be capable of handling any time series data and provide users with reliable forecasts. Key functionalities include:

- **Data Preprocessing:** Cleaning and preparing data for analysis.
- **Exploratory Data Analysis (EDA):** Identifying trends, patterns, and correlations in the data.
- **Model Selection:** Comparing different forecasting models like SARIMA, XGBoost, and Prophet to identify the most effective one.
- **Deployment:** Creating a user-friendly application via Streamlit, making the model accessible for anyone with time series data.

The project will focus on using historical electricity consumption data to make predictions, but it will be generalized for use with other types of time series datasets.

2.3 Key Technologies Used

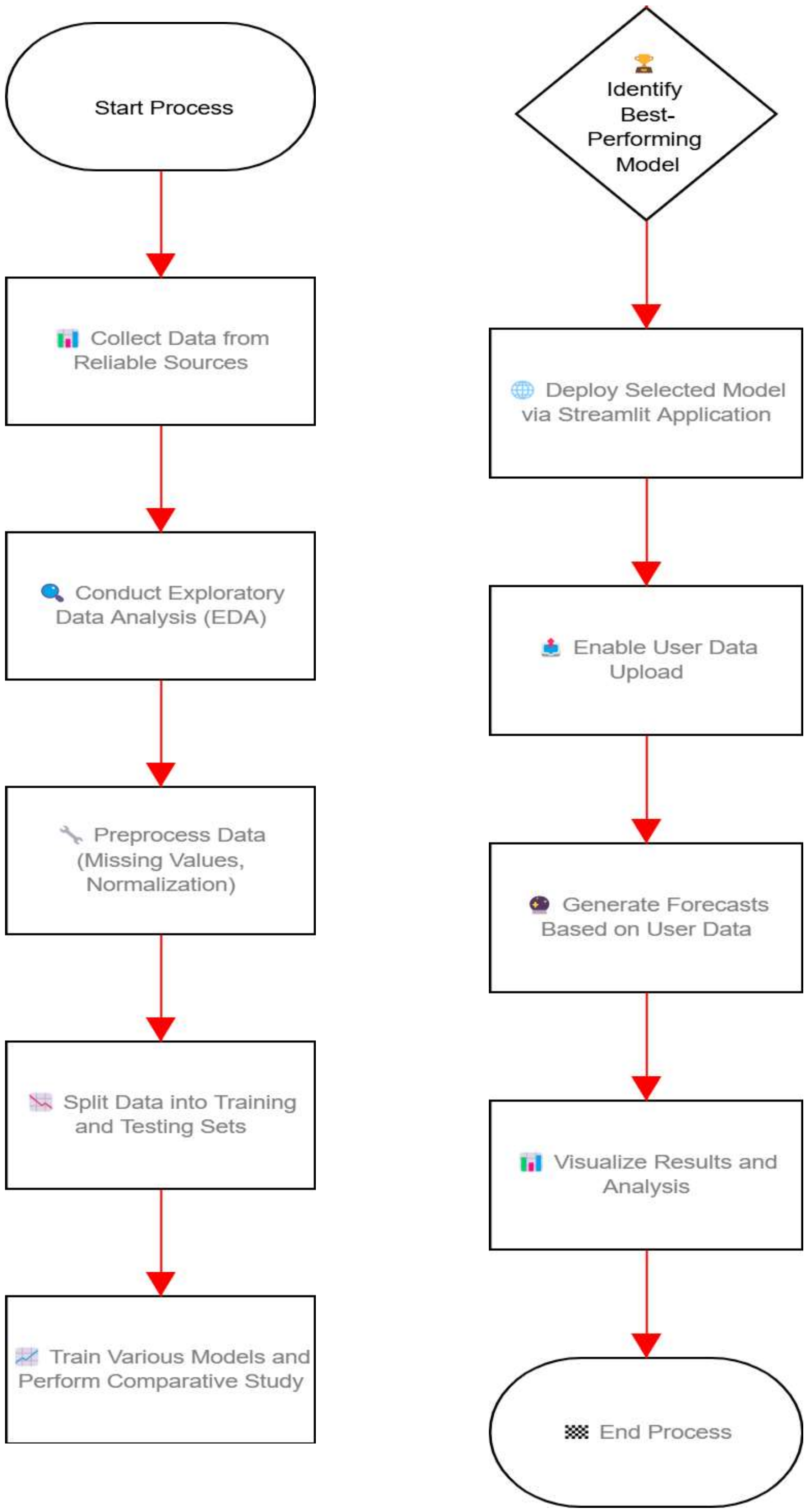
The project uses the following key technologies:

- **Prophet:** A forecasting tool developed by Facebook that is particularly good for handling time series data with trends and seasonality.
- **Streamlit:** A Python framework used to quickly build and deploy data applications. It will be used to create an interactive web interface for users to input their time series data and obtain forecasts.
- **XGBoost:** A popular machine learning algorithm that can be used for time series forecasting when the data is transformed appropriately.
- **SARIMA:** A statistical model used for forecasting time series data that accounts for seasonality and trends.
- **Python Libraries:** Libraries like Pandas, NumPy, and Matplotlib will be used for data preprocessing, analysis, and visualization.

3. System Model

3.1 Data Flow and Architecture of the Forecasting System

The system’s architecture follows a structured pipeline to ensure efficient data processing, forecasting, and visualization. The flow is designed to handle the entire process—from data collection to delivering the forecast via a web interface.



3.2 Overview of Machine Learning Models Used

SARIMA (Seasonal Autoregressive Integrated Moving Average):
SARIMA is a traditional statistical model widely used for time series forecasting, especially for data with seasonality. It incorporates autoregressive (AR) and moving average (MA) components, along with differencing to make the data stationary and adjustments for seasonality.

XGBoost (Extreme Gradient Boosting): XGBoost is a powerful machine learning model effective for regression tasks. By transforming time series data into supervised learning problems (e.g., using lag features), XGBoost can predict future values based on patterns found in historical data.

Facebook Prophet: Prophet is an open-source tool designed for forecasting time series with daily seasonal patterns and missing data. It is robust to outliers and particularly useful for datasets that exhibit seasonality and trends.

The comparative study of these models helps determine the best approach for forecasting electricity usage in this project.

3.2.1 Platforms

Python Environment: All data processing, model training, and forecasting will take place in a Python environment in kaggle notebook available at github link :- https://github.com/ankit1576/Time-Series-Forecasting, leveraging libraries like Pandas, NumPy, and Scikit-learn.

Streamlit Platform: The user interface is built using Streamlit, which will host the forecasting model and allow users to interact with it by uploading their own time series data.

Web Hosting/Deployment: The application will be deployed on a Streamlit hosting service. Users can input their data and get forecasts via a simple web interface.

3.2.2 Libraries

The system utilizes several key Python libraries to streamline the data processing, modeling, and visualization tasks. **Pandas** and **NumPy** are used for data manipulation and numerical computations. For time series forecasting, we leverage **Prophet** and **XGBoost** to model and predict future values, while **Statsmodels** is used for traditional statistical models like SARIMA. Visualization is handled with **Matplotlib** and **Seaborn** to create interactive plots. Finally, the user interface is built using **Streamlit**, which enables seamless integration of the forecasting models into a web application for easy access.

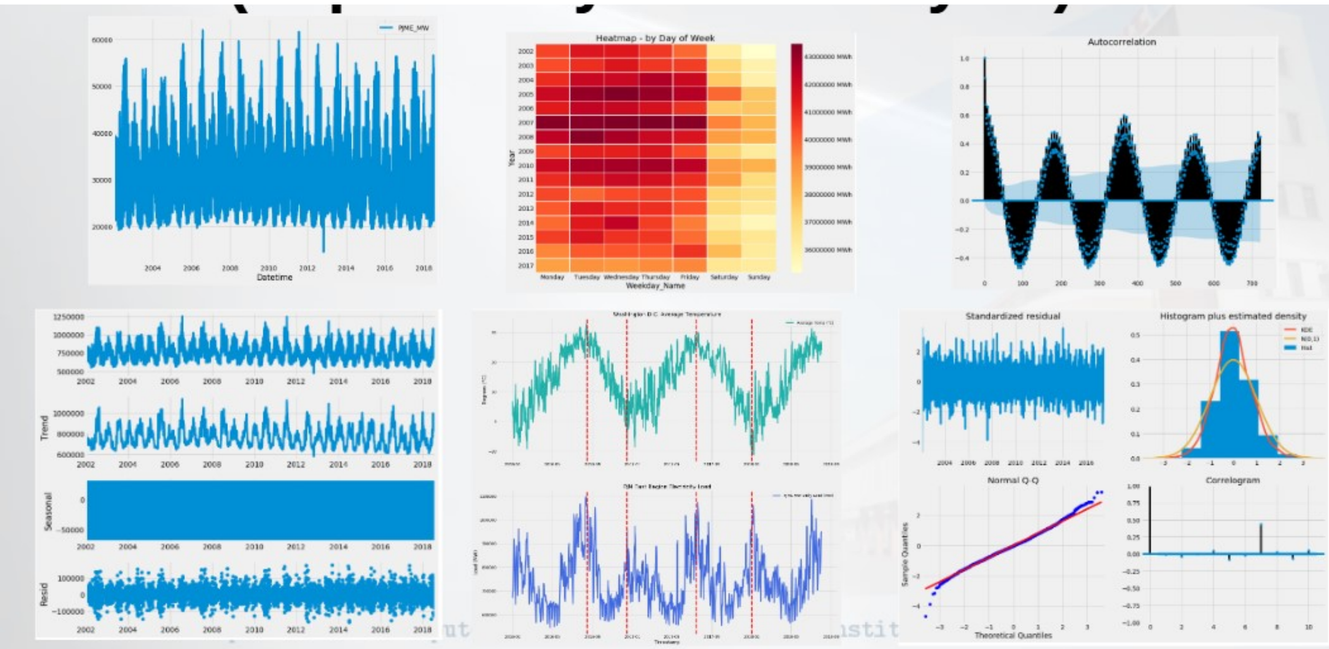
4. Methodology

4.1 **Data Collection and Preprocessing**

The data used in this project is sourced from the PJME hourly energy consumption, which provides hourly electricity consumption data for the PJM East Region. This dataset consists of historical power consumption data in megawatts (MW). The preprocessing phase includes handling missing values, dealing with outliers, and transforming the data into a structured time series format. Data normalization and other necessary transformations are applied to prepare the dataset for model training.

4.2 **Exploratory Data Analysis (EDA)**

In this step, exploratory data analysis (EDA) is performed to better understand the patterns and trends in the electricity usage data. This includes visualizations like line plots, seasonal plots, and correlation heatmaps. The purpose of EDA is to identify trends, detect seasonality, and check for any anomalies in the dataset. These insights help in choosing the right model for forecasting and improve the understanding of the underlying structure of the data.



Go through github [‘PJME_Forecasting.ipynb’] file for the detailed EDA

4.3 **Model Training Process**

Once the data is preprocessed and analyzed, it is split into training and testing datasets. The training data is used to fit the forecasting models, while the testing data is used to evaluate the performance of the models. During this phase, the three models (SARIMA, XGBoost, and Prophet) are trained, and hyperparameter tuning is performed to optimize their performance. Various evaluation metrics, such as Mean Absolute Error (MAE) and Mean Absolute Percentage Error (MAPE), are used to assess the model's predictive accuracy.

4.4 **Choosing Prophet for Forecasting**

After comparing the results from all three models, Prophet is selected as the most suitable model for forecasting electricity usage. Prophet excels in handling seasonality, trends, and missing data, making it a strong choice for this project. Its robustness to outliers and its ability to provide uncertainty intervals around forecasts are key advantages. Prophet's flexibility and simplicity also allow it to be easily integrated into a user-friendly web application for real-time forecasting.

PROPHET

See Official Documentation for Fb prophet here :- Fb Prophet Docs

5. Implementation

5.1 **Building the Forecasting Model with Prophet**

Using Prophet, we create a forecasting model designed for accurate, trend-based electricity usage predictions by capturing seasonality and handling outliers.

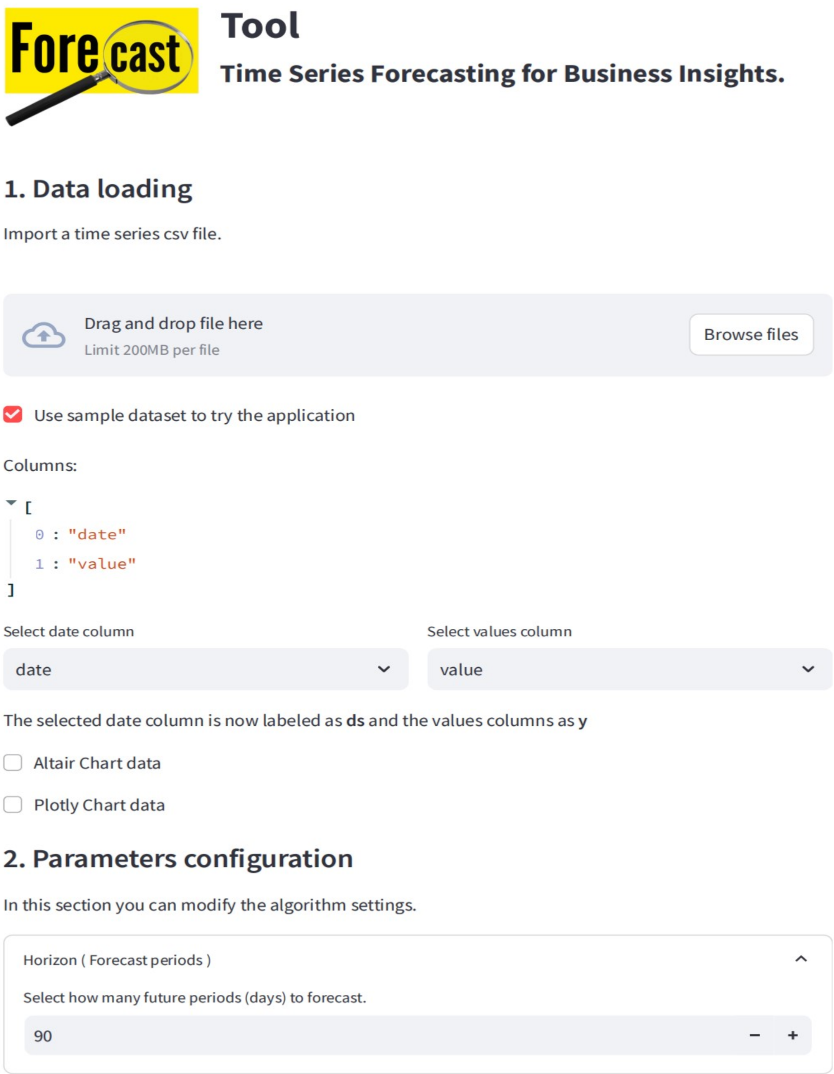
5.2 **Creating a Streamlit App for User Interaction**

We develop an interactive Streamlit app, allowing users to upload their own time series data, specify forecast periods, and visualize results with ease. This makes forecasting accessible even to non-technical users.

5.3 **Deployment on Streamlit**

The app is deployed online at forecastify.streamlit.app, providing users with seamless, anytime access to the forecasting tool. This deployment ensures easy, real-time interaction with the model in a scalable, user-friendly setup.

Below are screeshots of the website having url :forecastify.streamlit.app



Seasonality

Default seasonality used is additive

Seasonality

- (●) additive
- ( ) multiplicative

Trend components

Add or remove components:

- [ ] Daily
- [ ] Weekly
- [ ] Monthly
- [ ] Yearly

Growth model

By default a linear growth model is Applied.

Growth model

- (●) linear
- ( ) logistic

Holidays

Select country

Country name

- [ ] Add country holidays to the model

3. Forecast

Fit the model on the data and generate future prediction.

- [ ] Initialize model (Fit)
- [ ] Generate forecast (Predict)
- [ ] Show components

4. Model validation

In this section it is possible to do cross-validation of the model.

The Prophet library makes it possible to divide our historical data into training data and testing data for cross validation. The main concepts for cross validation with Prophet are:

Training data (initial): The amount of data set aside for training. The parameter is in the API called initial.

Horizon: The data set aside for validation.

Cutoff (period): a forecast is made for every observed point between cutoff and cutoff + horizon.

Cross validation

initial

365 − +

period

90 − +

horizon

90 − +

Here we do cross-validation to assess prediction performance on a horizon of **90 days** days, starting with **365 days** days of training data in the first cutoff and then making predictions every **90 days**.

6. Results and Discussion

6.1 **Model Performance and Evaluation**

The performance of each forecasting model was evaluated using Mean Absolute Error (MAE), which measures the average magnitude of prediction errors. A lower MAE indicates higher accuracy. Here’s a summary of the models' performance based on MAE:

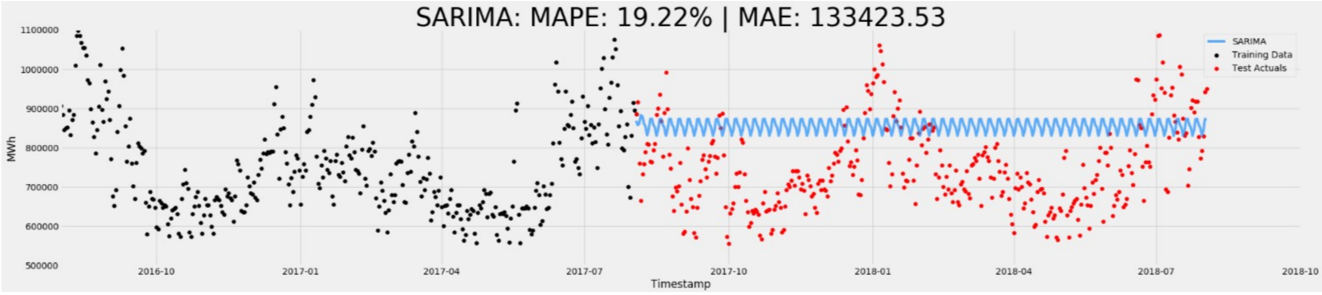
As shown, **Prophet achieved the lowest MAE values**, demonstrating better accuracy compared to other models. XGBoost also performed well, with a slightly higher MAE.

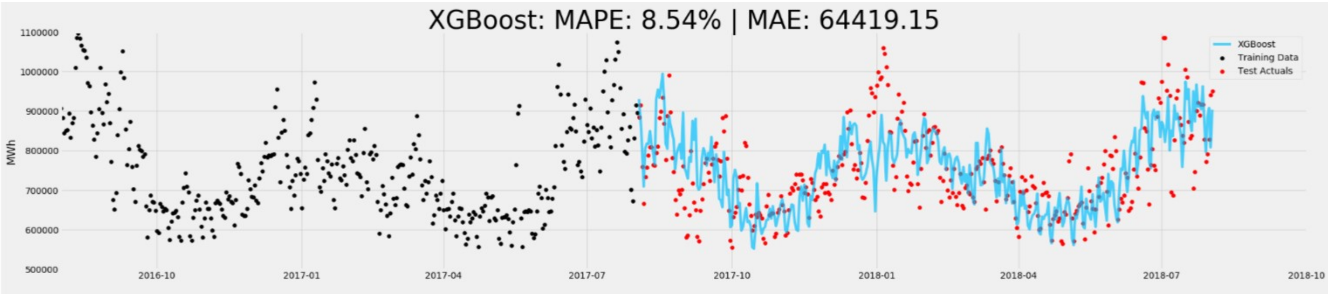
Model	**MAE (%)**
XGBoost	8.54%
SARIMA	19.22%
Prophet	7.94%

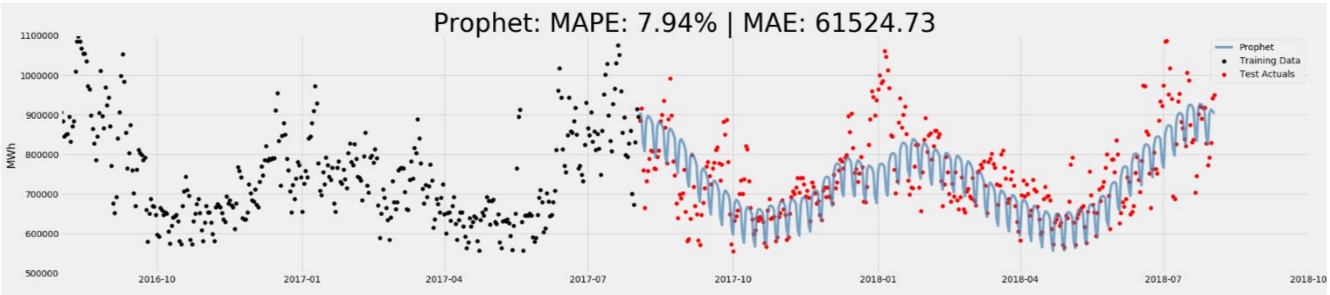
6.2 Visualizing Forecasts and Comparisons

To better understand each model’s predictions, we generated plots comparing the forecasted electricity usage against actual values. These visualizations provided insights into the strengths and limitations of SARIMA, XGBoost, and Prophet:

- **Prophet** showed a strong ability to follow seasonal trends effectively, making it suitable for data with regular patterns.
- **SARIMA** was moderately effective, handling seasonality reasonably well but experiencing limitations during periods with sudden shifts or high volatility.
- **XGBoost** reliably learned from historical data patterns but was slightly less adaptable to extreme fluctuations, making it effective for general patterns but less responsive to sudden changes.







6.3 Insights from Forecasts

The analysis highlights that **model selection impacts forecast accuracy**. Prophet, with its seasonality adjustments, proved useful for data with clear patterns. SARIMA, while effective in steady trends, showed limitations with more complex shifts. XGBoost, though robust, benefits from further fine-tuning to adapt better to seasonal variations. Combining or fine-tuning these models could enhance overall forecasting accuracy.

7. Conclusion and Future Work

7.1 Key Insights

This project showed that different models have unique strengths for time series forecasting. Prophet and XGBoost handled seasonal trends and patterns well. Prophet was strong in capturing seasonality, while XGBoost effectively learned from past data. SARIMA, though useful in some cases, was less reliable with sudden changes, reminding us that model choice matters based on data characteristics.

7.2 Model Improvement

Combining Prophet’s seasonality focus with XGBoost’s pattern recognition might improve accuracy. Also, tuning these models further and exploring other machine learning techniques could help them adapt better to unexpected changes.

7.3 Future Uses and Upgrades

This forecasting tool could be applied across many fields, from retail to finance ,etc. Enhancing the Streamlit app to allow real-time data updates and allowing the user to choose more timeseries model over the app so user can use different models and get the result , more user options would make it even more useful, on-demand forecasts for different users and needs.

8.Bibliography/ References:

[1] S. J. Taylor and B. Letham, "Forecasting at scale," PeerJ Preprints, vol. 5, e3190v2, 2017. https://doi.org/10.7287/peerj.preprints.3190v2.