CSCE 5222: Feature Engineering

Project Report – Group 2

(Final Project)

Project Title: "Power Outage Insights: Feature Engineering in Time Series Analysis"

Team Members:

1. Sravani Katlaganti (11560617)
2. Panduga Raja Tejasvi Prasad (11414926)
3. Yasmeen Haleem (11462753)
4. Varun Mohan (11615500)

GitHub Link: https://github.com/yasmeenha/Featureengg

Google Drive Links:

1. Python_colab_county_Lee
2. Python_colab_county_Miami

Idea Description

As time series feature engineering deals with lags and temporal and spatial analysis, the team decided to work on the Eagle-I power dataset from the SMC dataset challenge 2023, and after studying the IEEE paper [1] the team decided to apply feature engineering techniques like lags, autocorrelation, dickey fuller test to find anomalies and seasonal decomposition, and the final propose is to see the improvement in power outage prediction model using LSTM, SVM, linear regression, xgboost and cat boost regressor.

Motivation

Power outages are a common and bothersome concern which impacts individuals, firms, and critical infrastructure. The need to address the issues that are due to power outages and ultimately improve the dependability of the electric system proved to be the highlight of the project. These outages may be caused by a variety of reasons, such as hardware malfunctions, severe weather, natural disasters, and even cyberattacks. We have to comprehend these factors and work towards reducing the incidences and duration of power outages by enhancing methods of feature engineering for time series data pertaining to power outages [2].

By performing feature engineering on time series and taking into account the effect of weather correlation on the power outages we try to see the improvement in the power outage prediction and offer utility companies, enterprises, and governments useful information to help them make decisions, prioritize grid enhancements, and allocate resources more effectively.

Significance

The significance of this project is diverse, involving a number of important elements that have an impact on both individuals and society as a whole. Firstly, this study has the potential to significantly improve the precision and efficacy of predictive analytics by concentrating on feature engineering for time series data related to power outages. These more accurate forecasts can result in earlier alerts and more effective preventive measures, which will eventually reduce the incidence and length of power outages. By decreasing downtime, minimizing equipment damage, and raising overall output, this increase in grid stability and resilience offers a superior quality of life for people and businesses [3].

The financial consequences are additionally crucial. Due to business being disrupted and the requirement for backup power sources, power outages frequently result in large financial losses. By using advanced feature engineering to estimate the economic cost of power outages, this study can offer utilities and companies useful information to help them make informed choices about grid upgrades, resource allocation, and investment. In turn, this results in significant cost savings and more effective resource usage, which is advantageous to the economy and consumers.

Furthermore, it impacts healthcare facilities, communication networks, and emergency response procedures. The project directly contributes to public safety and the efficient operation of vital services during time-sensitive situations by improving grid dependability through improved feature engineering. The relevance for crucial services by making timely and accurate predictions that can make difference between life and death in healthcare situations and prompt reactions to emergencies [4].

Objectives

As the power outage dataset from Eagle-I only has the given outages as sum as the column with the power outages of 92% of U.S county with 15-minute time stamp, the team decided to do feature engineering on time series Eagle-I dataset, and to do temporal and spatial analysis on the power outage data. The objective is to combine the given power outage data from 2014 -2022 in the Eagle-I dataset and to combine it with the U.S weather dataset from Kaggle that gives the weather information for U.S counties from 2016 to 2022, such as rain, fog, snow, wind speed, etc. One aim is to find correlation between weather and power outages and after doing time series feature engineering the team aims to see an improvement in the accuracy of the power outage prediction. The other aim is to create a multivariate power outage prediction model by creating rainfall, windspeed, humidity, different days of the week, months as features and try to see if the power outage prediction accuracy improves in a multivariate model.

Related Work

The study of time series data and feature engineering come together crucially in the predictive modeling of power outages, highlighting the importance of managing temporal data skillfully and creating features that reflect the inherent dynamics and patterns in the data. In this regard, a thorough examination of numerous feature extraction and selection techniques as well as forecasting methodologies offers a fundamental comprehension of the dominant tactics and difficulties in this field.

Sandya H.B. et al. illuminate the importance of intelligent feature extraction, specifically employing Fuzzy Logic and GARCH techniques in time-series signal processing. Although the methodologies presented may not directly relate to the prediction of power outages, they do highlight a key idea in feature engineering: the extraction of pertinent features that accurately `capture the temporal dependencies and volatilities present in electrical grid data. These features may be useful in improving predictive models for power outages.

Khalid Ijaz et al. present a novel approach to feature selection in the temporal domain, proposing an LSTM model integrated with a distinctive temporal feature selection technique for short-term electrical load forecasting. Particularly in the field of power outage prediction, where temporal dependencies are critical, the emphasis on choosing salient temporal features that can accurately describe the sequential data highlights the pivotal importance of feature selection in increasing model performance and generalization.

Vivian Do et al. delve into the spatiotemporal aspects of power outages, exploring the intricate relationships with climate events and social vulnerability. This research highlights the possibilities of integrating temporal elements, together with social and climatic factors, into predictive models in order to help them recognize and respond to anomalies and patterns connected to weather conditions and social interactions. By offering a multidimensional approach that incorporates temporal, geographical, and external elements, this broadens the scope of feature engineering.

Finally, a comparison of feature extraction and selection methods for time-series data classification can shed light on the advantages and disadvantages of various approaches, laying the groundwork for effective feature engineering approaches for power outage prediction [4]. Exploration and adaptation of these approaches may open the door to more sophisticated and effective models that can recognize and respond to trends and abnormalities in data from the electrical grid.

Model

Architecture of ARIMA:

As shown in Figure-1, ARIMA model architecture shows the process of time series forecasting. It starts with identifying the model and its parameters, followed by differentiating the data to ensure its stationarity. Once the model is estimated, it undergoes diagnostic checks to confirm its validity. If the model passes, it is then used for forecasting future data points. Errors from predictions are used to improve the model iteratively [8].

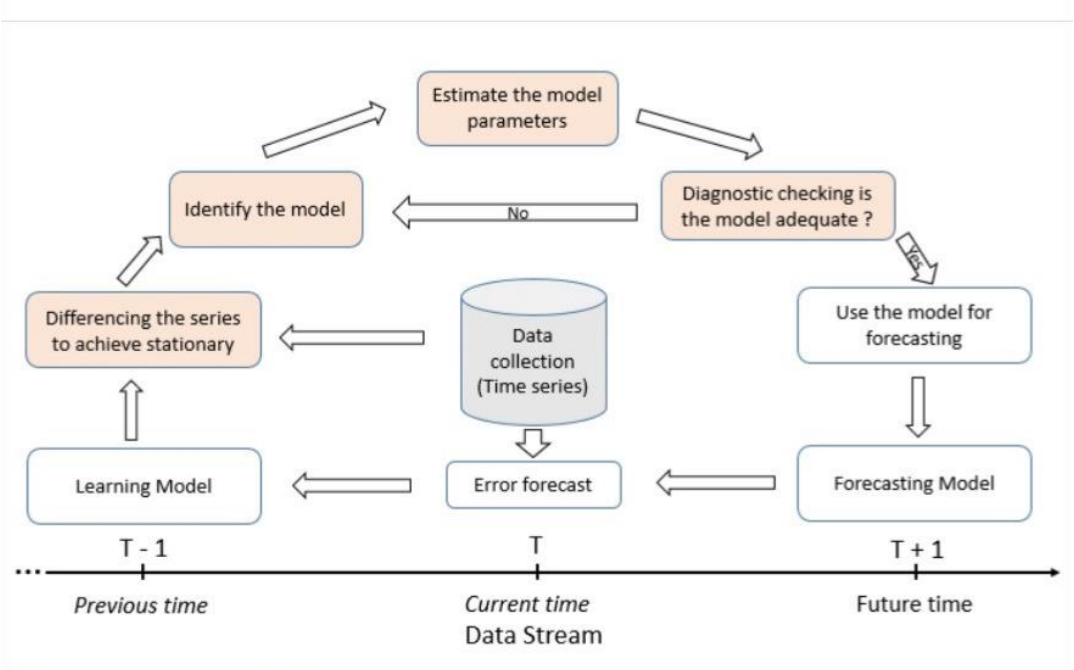

Figure-1: ARIMA Model

Architecture of SARIMAX:

The model is similar to ARIMA the addition is the seasonality in this model. Since seasonality is a significant factor influencing the time series behavior the model basically allows seasonality differentiating. It captures both the non-seasonal and seasonal aspects of the series.

The process involves reading original time data. Then we are applying the regular and seasonal differentiating to make it stationary. The next module involves autoregressive components for both non-seasonal and seasonal parts. Later module involves the moving average and estimation parameters. Finally forecasting the values.

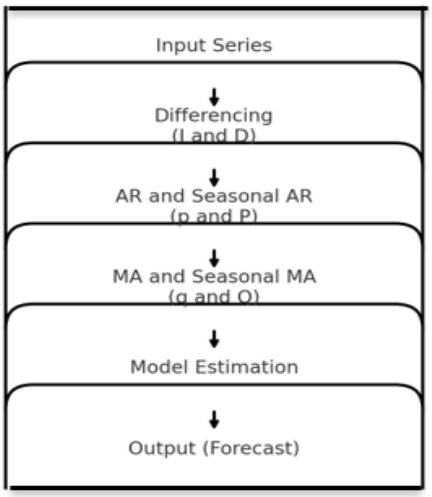

Figure-2: SARIMAX Model

Workflow:

In the project, the first step is to collect appropriate dataset for the problem statement as seen in Fig-3. The datasets used in the project are two different Outage and Weather datasets from the years 2018-2022 in Lee and Miami counties. Then, the next step is to preprocess these datasets to clean the data: to filter out data for a particular county, to process any null values, to remove any unwanted columns, and changing frequency of outage data to every day etc.,

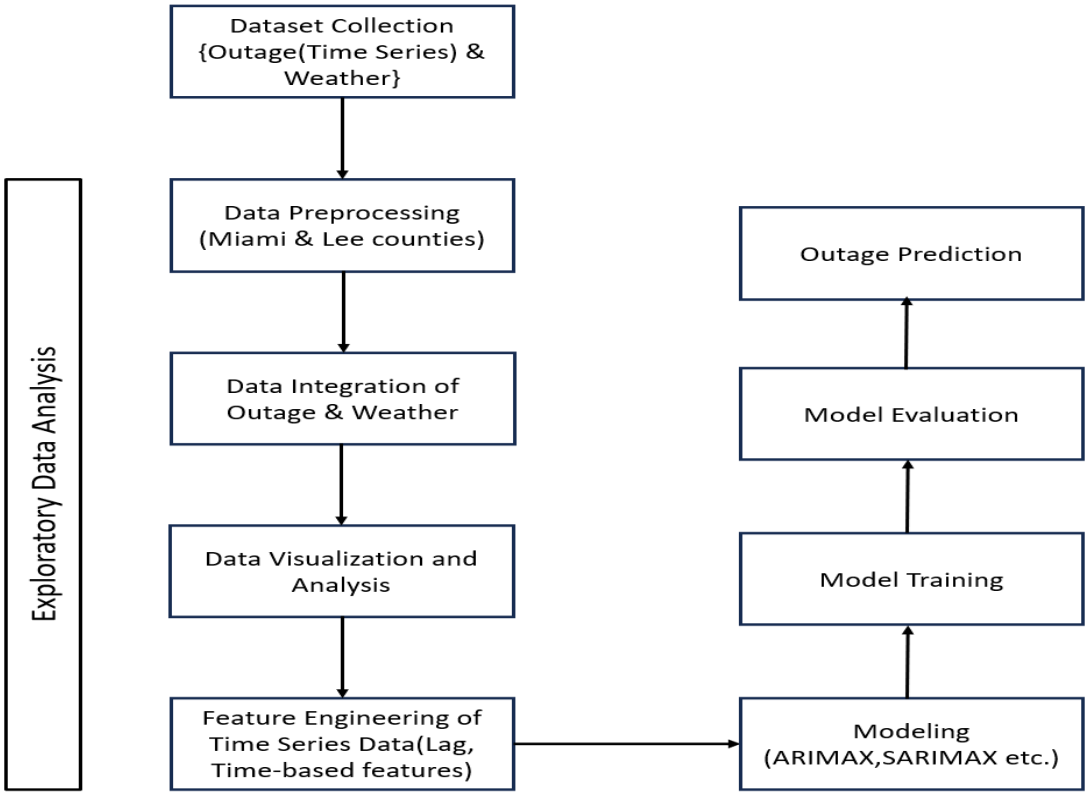

Figure-3: Workflow diagram

And in the next step merging outage and weather datasets based on state and county for each day. Then, the next step is to visualize the merged dataset, it includes visualizing the distribution outages corresponding to different weather conditions, trends in the outages and correlation between various features. And in the next step of feature engineering, extracting time-based features like day, month, year, lag terms, rolling statistics, cyclic features and visualizing these plots along with patterns and seasonality decomposition, auto and partial correlations and Fourier transform. Then all this preprocessed data with new features is used for modeling various models of ARIMA, SARIMAX, Linear Regressor, XGBoost, ARIMAX, CatBoot Regressor, SVM and LSTM. Then these models are trained with training dataset and then evaluated on testing dataset. From these evaluations, calculation of performance metrics like RMSE, MSE and R2 scores. Then these trained and evaluated models are used for forecasting outages.

Dataset

Description of Dataset:

The dataset that we are going to use in this project is the Eagle-I power dataset from the SMC dataset challenge 2023[5]. The dataset consists of 8 years of power outage data between 2014 and 2022 for each county of united states with an interval of 15 mins which is derived by the EAGLE-I program at ORNL. They have collected it through a process of ETL from public outage maps of utility. The columns present in this dataset are 'fips_code', 'county', 'state', and 'sum' indicating the power outage and the total number of rows are 1689460. The other dataset that we will use to integrate with the former dataset is the weather dataset from Kaggle dataset of US Weather Events (2016 - 2022) [6]. This dataset consists of 7 years of weather events data between January 2016 and December 2022 for each county of 49 states in the US which is collected from 2071 airport-based weather stations nationwide. It consists of events of about 8.6 million including general rain and snow to severe storms and freezing conditions. The columns include 'ZipCode', 'County', 'AirportCode', 'EventId'. 'Type', 'Severity' etc. We'll be integrating these 2 datasets for further use in the feature engineering and modeling processes.

Detail design of Features:

Outage Dataset:

	EventId	Type	Severity	StartTime(UTC)	EndTime(UTC)	Precipitation(in)	TimeZone	AirportCode	LocationLat	LocationLng	City	County	State	ZipCode
0	W-4275658	Rain	Light	2016-01-01 07:26:00	2016-01-01 07:36:00	0.00	US/Eastern	KHST	25.4949	-80.3732	Homestead	Miami-Dade	FL	33033.0
1	W-4275659	Fog	Severe	2016-01-01 07:36:00	2016-01-01 07:40:00	0.00	US/Eastern	KHST	25.4949	-80.3732	Homestead	Miami-Dade	FL	33033.0
2	W-4275660	Rain	Light	2016-01-01 07:40:00	2016-01-01 08:09:00	0.01	US/Eastern	KHST	25.4949	-80.3732	Homestead	Miami-Dade	FL	33033.0
3	W-4275661	Fog	Severe	2016-01-01 08:09:00	2016-01-01 09:23:00	0.00	US/Eastern	KHST	25.4949	-80.3732	Homestead	Miami-Dade	FL	33033.0
4	W-4275662	Fog	Severe	2016-01-01 09:53:00	2016-01-01 10:23:00	0.00	US/Eastern	KHST	25.4949	-80.3732	Homestead	Miami-Dade	FL	33033.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
22324	W-8480168	Rain	Light	2022-12-26 12:48:00	2022-12-26 13:53:00	0.04	US/Eastern	KMIA	25.7880	-80.3169	Miami	Miami-Dade	FL	33122.0
22325	W-8480169	Rain	Light	2022-12-26 14:53:00	2022-12-26 16:53:00	0.07	US/Eastern	KMIA	25.7880	-80.3169	Miami	Miami-Dade	FL	33122.0
22326	W-8480170	Rain	Light	2022-12-26 21:53:00	2022-12-27 07:36:00	0.41	US/Eastern	KMIA	25.7880	-80.3169	Miami	Miami-Dade	FL	33122.0
22327	W-8480171	Rain	Light	2022-12-27 08:45:00	2022-12-27 08:53:00	0.02	US/Eastern	KMIA	25.7880	-80.3169	Miami	Miami-Dade	FL	33122.0
22328	W-8480172	Rain	Light	2022-12-27 09:53:00	2022-12-27 13:43:00	0.04	US/Eastern	KMIA	25.7880	-80.3169	Miami	Miami-Dade	FL	33122.0

22329 rows × 14 columns

Screenshot-1: Outage Dataset

Weather Dataset:

	fips_code	county	state	sum	run_start_time
0	12086	Miami-Dade	Florida	658.0	2018-01-01 00:00:00
1	12086	Miami-Dade	Florida	899.0	2018-01-01 00:15:00
2	12086	Miami-Dade	Florida	683.0	2018-01-01 00:30:00
3	12086	Miami-Dade	Florida	2043.0	2018-01-01 00:45:00
4	12086	Miami-Dade	Florida	54.0	2018-01-01 01:00:00
...	...	...	...	...	...
79057	12086	Miami-Dade	Florida	414.0	2022-06-07 09:00:00
79058	12086	Miami-Dade	Florida	345.0	2022-06-07 09:15:00
79059	12086	Miami-Dade	Florida	344.0	2022-06-07 09:30:00
79060	12086	Miami-Dade	Florida	337.0	2022-06-07 09:45:00
79061	12086	Miami-Dade	Florida	337.0	2022-06-07 10:00:00

79062 rows × 5 columns

Screenshot-2: Weather Dataset

Exploratory Data Analysis

Data Preprocessing:

For the purpose of analysis and predicting power outages we are making use of two datasets, the EagleI Outage dataset from the Oak Ridge National Laboratory and the US Weather Events dataset from Kaggle.

EagleI Outage Dataset

Overview: This dataset contains 8 years of power outage out of which we will be using 4 years from 2018 - 2022 at the county level in intervals of 15 minutes. These 4 years are available as 4 separate datasets.
Merging: We are concatenating the 4 datasets of 4 years into a single data frame for ease of analysis.
Filtering: The dataset is filtered and only the Lee County of Florida is being extracted to be used in this project.
Preprocessing Steps: We are dropping irrelevant columns like fips_code and converting run_start_time to a datetime format which we then split to give date and time columns. We are discarding the run_start_time and the time column retaining only the date column. We are then grouping the data by date and taking the maximum of the outage 'sum' per each day. We are doing this to minimize the dataset size which initially was in intervals of 15 minutes. We are also creating a complete date range from 2018 to 2022 and merging it with this outage dataset we have to make sure all dates are accounted for. The final features or columns in this data frame are county, state, sum, date.

Weather Severity Dataset

This dataset has weather events for 49 states of the US from 2016 - 2022 like rain, fog, snow, storms and other conditions along with the severity of each event.
Filtering: For the sake of this project, we are filtering events only from 2018-2022. We are also dropping irrelevant columns like 'Precipitation(in)', 'TimeZone', 'AirportCode', 'LocationLat', 'LocationLng', 'City', 'ZipCode', 'EventId', 'County', 'State'. Since we already know we are using Lee County of Florida state, we will just extract for it and also localise the timezone before removing the irrelevant columns.
Preprocessing Steps: We are first creating 5 new columns equal to the number of unique events in the Type column and labelling it as such. We then encoding the different severity in the Severity column to numeric values. We are using this severity mapping to fill in for the 5 new columns we created before. We are then grouping them by date and in case a day has more than one weather event, we consider the maximum severity for it.

Data Integration

	county	state	sum	Date	Rain	Fog	Cold	Precipitation	Storm
0	Miami-Dade	Florida	2408.0	2018-01-01	1	3	0	0	0
1	Miami-Dade	Florida	3129.0	2018-01-02	1	3	0	0	0
2	Miami-Dade	Florida	9059.0	2018-01-03	2	0	0	0	0
3	Miami-Dade	Florida	1485.0	2018-01-04	0	0	0	0	0
4	Miami-Dade	Florida	399.0	2018-01-05	0	0	3	0	0
...	...	...	...	...	...	...	...	...	...
1821	Miami-Dade	Florida	0.0	2022-12-27	1	0	0	0	0
1822	Miami-Dade	Florida	0.0	2022-12-28	1	3	3	0	0
1823	Miami-Dade	Florida	0.0	2022-12-29	1	3	0	0	0
1824	Miami-Dade	Florida	0.0	2022-12-30	1	3	0	0	0
1825	Miami-Dade	Florida	0.0	2022-12-31	1	3	0	0	0

1826 rows × 9 columns

Screenshot-3: Data integration

The integration of the processed power outage dataset and the weather dataset is a crucial step. We first make sure that the date column in both the dataset is in date format and is uniform. We then use an outer join merge to merge the dataset on their respective date columns. We then replace 'NaN' values in weather columns with 0s since 0 indicates the absence of any abnormal weather event. Finally, we make sure the merged dataset has no missing values and it is of an appropriate type and formal for further analysis and forecasting [7].

Analysis and Visualization:

The Analysis and Visualization of the merged dataset of weather and power outages is very crucial in drawing meaningful insights and correlation among the power outages due to the various event types of weather dataset. With the help of graphs, various trends and distribution of each variable were understood and analyzed. Below is the detailed explanation of these visualizations and analysis.

i. **Time Series Visualization:**

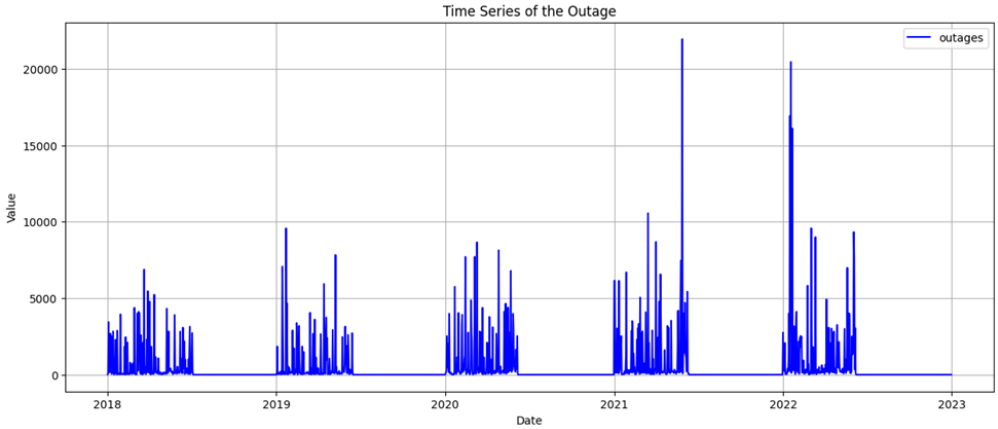

Figure-4: Time Series Visualization

Figure -4 is the visualization of the 'sum' column of outage dataset which is basically the power outage values. The outage values are plotted against the time period of 2018-2022, which has various fluctuations across different time periods even with spikes at some period of time. From the graph in Figure-1, temporal variations of the sum column i.e., the power outages can be seen across the given timeframe. It is noticed that there are higher power outages in the end of the year 2021 and beginning of the year 2022. These patterns reveal some periods of interest where we can use it for further analysis.

ii. **Correlation Analysis:**

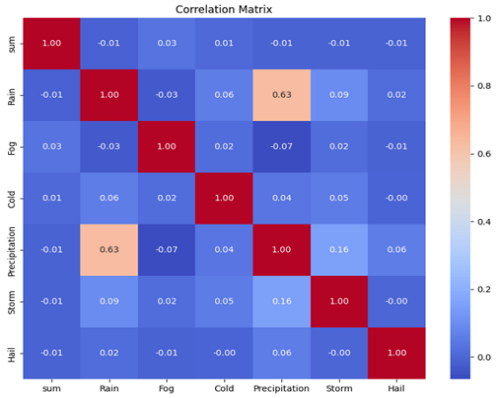

Figure-5: Correlation Matrix

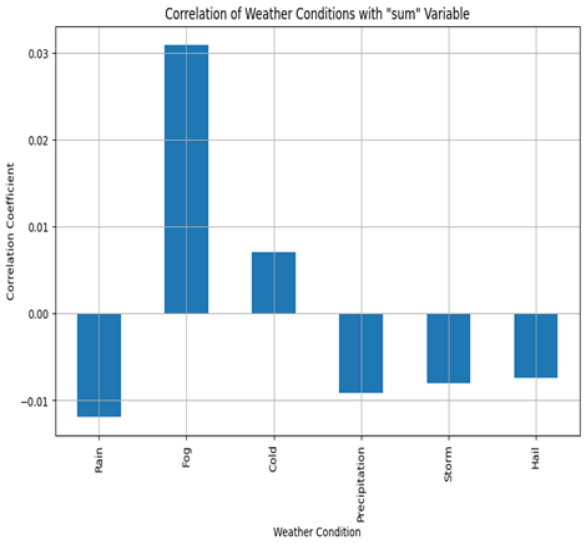

Figure-6: Correlation

The correlation analysis gives more information on how the different event types in weather are related to the outages in the power. For this, generated a correlation matrix that has all the correlation coefficients of each weather condition and power outages. Then this correlation matrix is represented through a heatmap for convenient visualization as in Figure-5. Also, the correlation coefficients are plotted against each weather condition as we can see in Figure-6. Among the correlations coefficients that are generated, the correlations between power outages

and each weather condition are of most importance as this is the whole concept of the project. The smaller the coefficient values the lesser the relationship between the two variables. From the correlation matrix values, we see that there is very less dependence of power outages on weather conditions individually. So, it can be understood that there is a non-linear relationship between power outages and weather events, or it can be related to combination of weather events.

iii. **Distribution Analysis:**

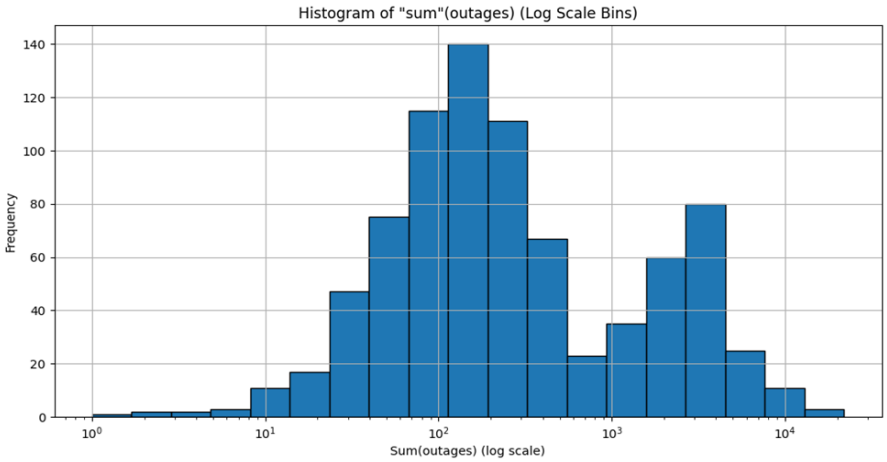

Figure-7: Frequency Distribution of Power Outages

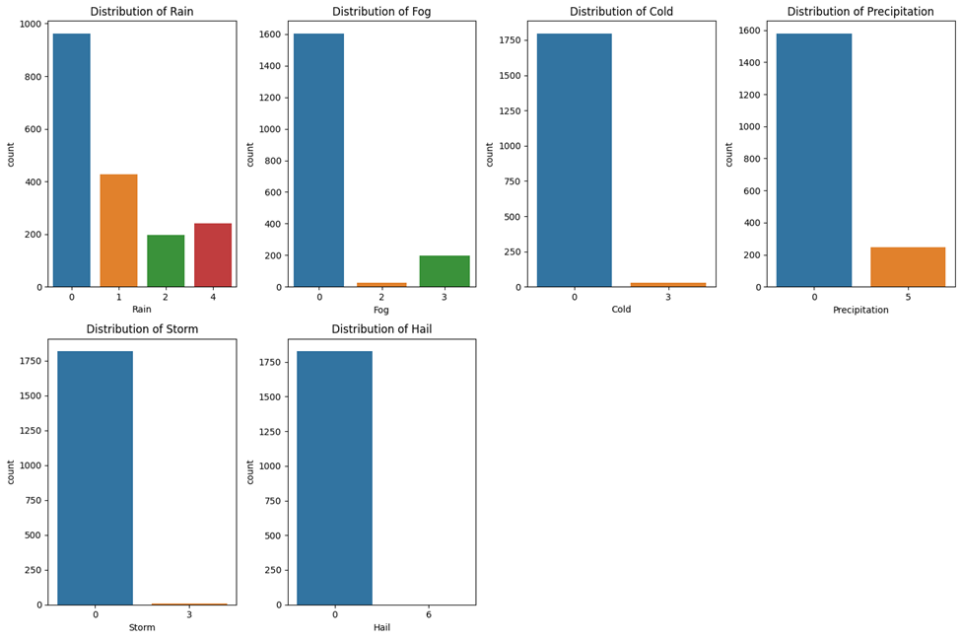

Figure-8: Frequency Distribution of Weather Conditions

The distribution of outage's frequencies is plotted using the histogram plots as in Figure-7. This gives the presence of any rare events in the data distribution and retrieving frequently appearing data. The distribution of power outages is done after converting it into log values as this normalizes that data if it contains any outliers or skewness in the distribution. This also helps in identifying patterns in the power outages. The distribution plots of weather conditions as in Figure-8 shows each event type occurrence against their severity levels of low, moderate, severe and heavy represented by numeric values of 1 to 4. It can be seen from the graphs that most of the weather conditions are almost at normal levels, that is in the low severity conditions. This shows the severe weather conditions occur only on rare counts.

iv. **Impact of various weather conditions on average power outage:**

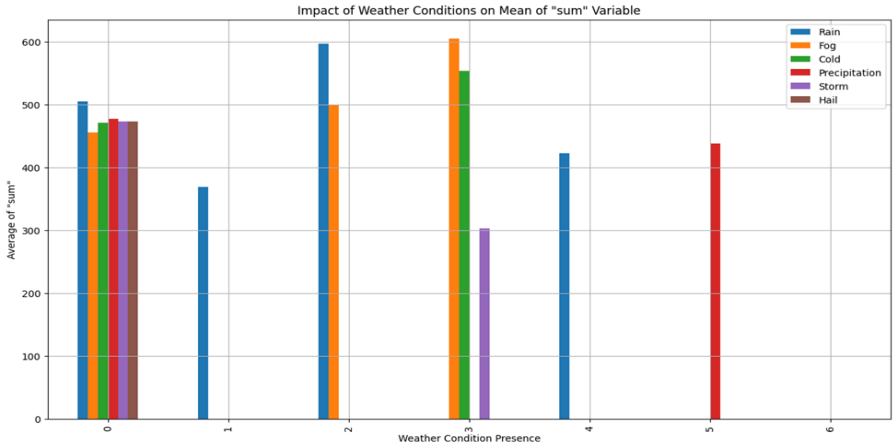

Figure-9: Distribution of average power outage across various weather conditions

For analyzing the impact of various severity levels of each weather event on the average power outage, plotted the bar charts as in Figure-9 where each colored bar represents each weather event and the average power outages of these each weather event is plotted against the severity level range of 0 to 6 representing no to severe and other unknown conditions. This graphical representation helps in identifying which weather event with which severity has more impact on the average power outages. From the plot, it is noted that the moderate fog and severe rains has caused the highest average power outages. Severe and low rainfall also has higher impact on power outage and all the normal weather events also have caused the considerate amounts of average power outages. This shows that there is no linear relationship between weather conditions and power outages and also there is no definite one weather event that has the most impact on the outage. Power outages are dependent on the group of weather conditions, or they are linearly related.

Overall, from this correlation analysis and visualization it can be drawn that the relationship between the different types of weather events and power outages is non-linear and depends on the combination of various weather events. Also, it can be understood that weather event of 'Fog' has the highest impact by considering the correlation coefficient as it is highest for this event among all other weather event types.

Feature Engineering

Time-Based Features

In the time-based feature, we are extracting the important features such as year, month, day and weekday from the merged dataset. These features are significant for analysis of the time series. It will help us to understand the temporal trends and patterns in the dataset. We can also analyze the variations in power outages across the various time scales.

Binary Weekend Feature

From extracting the Binary weekend features we can analyze the patterns across the weekends and weekdays. We can get insights into whether the outages are more frequent on the weekends or weekdays due to different usages by the customers on weekdays and weekends.

Rolling Statistics

Later, we calculated the rolling statistics which is the rolling means and standard deviations of the 'sum' attribute that represents power outages over seven and thirty days. The rolling statistics will highlight the long-term patterns by reducing the short-term noise. These statistics are thus more useful in identifying the low and high outage values. We can easily track anomalies.

Lag Features

Further, we have added the lag features to the impact of the previous power outage data on the future patterns. We have introduced three types of lag features here: 1-day, 7-day, and 30-day lags for the 'sum' attribute. This allows us to determine the temporal correlations which allow the model to use past outage data to forecast future predictions. Thus, improve the accuracy of our time series analysis.

Cyclical Features

We have transformed the day and month elements using sine and cosine to account for time's cyclical structure. These transformations are important for capturing the cyclic trends in the power outages including mainly weekly or monthly variations. This will help our model in adjusting to the recurring patterns which are frequently impacted by seasonal variations.

Handling Missing Values

We have handled the missing values in the dataset which have been created from obtaining the rolling and lag features. The missing values have been filled with zeros in order to ensure there is no discrepancy in our dataset. Handling the missing values will thus eliminate the probability of bias and inaccurate results which might affect the performance of the model.

Visualization of time series features:

i. **Yearly average of sum:**
   In Figure**-10**, the graph shows the yearly average of the sum of power outages. We can find that the most occurrence of power outages was observed in 2022 followed by 2021 and least in 2019.

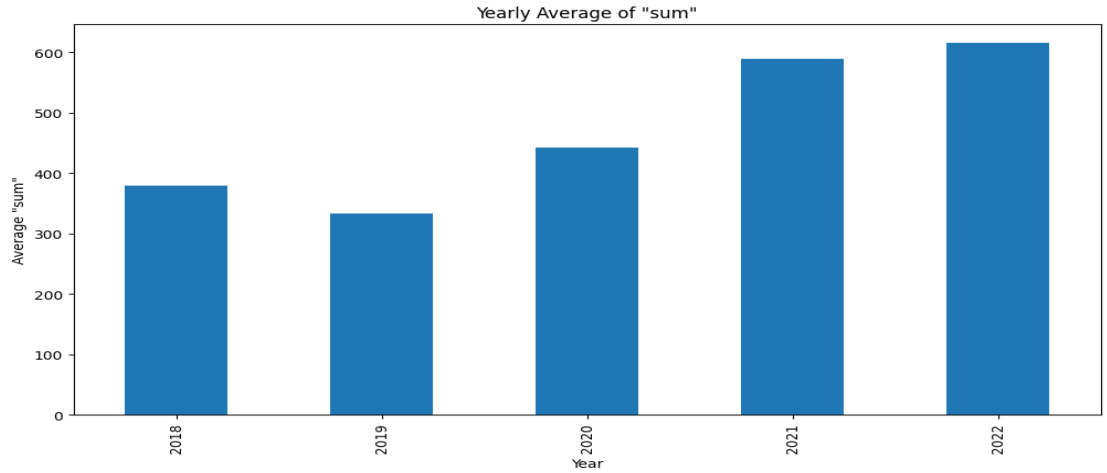

Figure-10: Yearly average of sum

ii. Monthly average:

In **Figure-11**, the graph shows the monthly average of the sum of power outages. From the bar plot we can see that the greatest number of power outages occur in January followed by May and the lease occurrences from July to very less in December.

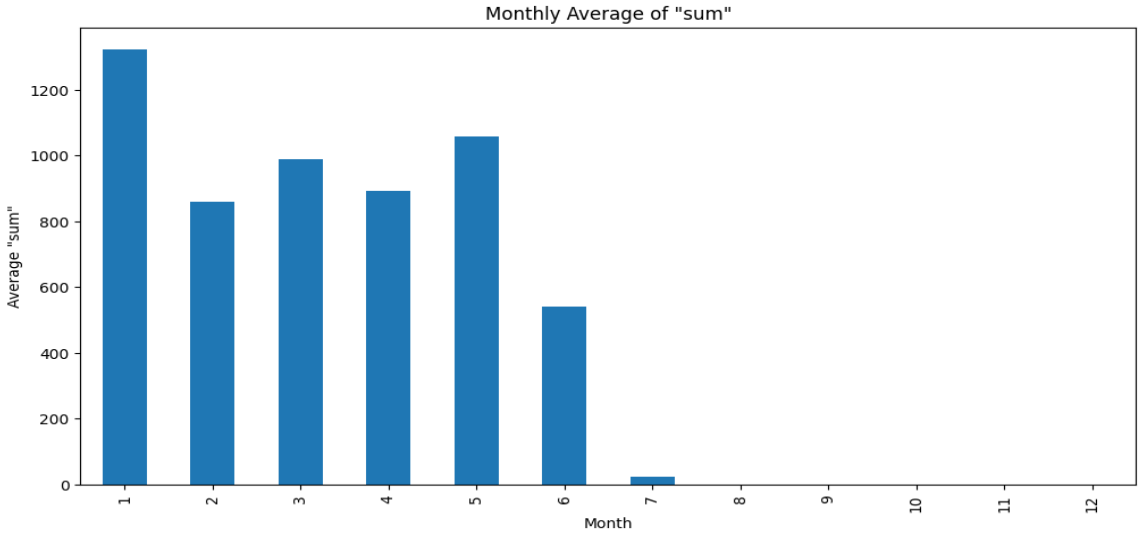

Figure-11: Monthly average of sum

iii. Daily average:

The line plot shows the average power outages across the days. We can explore if the power outages are higher or less at the end of the month. For example, we can see that the outages keep decreasing at the end of the month.

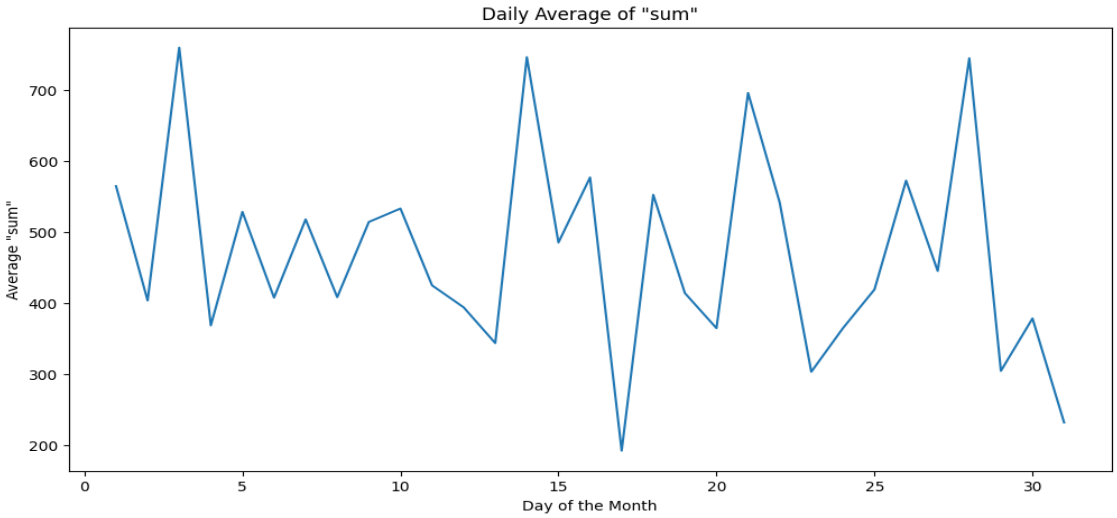

Figure-12: Daily average of sum

iv. Rolling Statistics:

In **Figure-13** we have the rolling statistics and standard deviation for 7 and 30 days. We can observe in the plot that rolling mean and standard deviation is higher in 2022 and 2021 compared to other years.

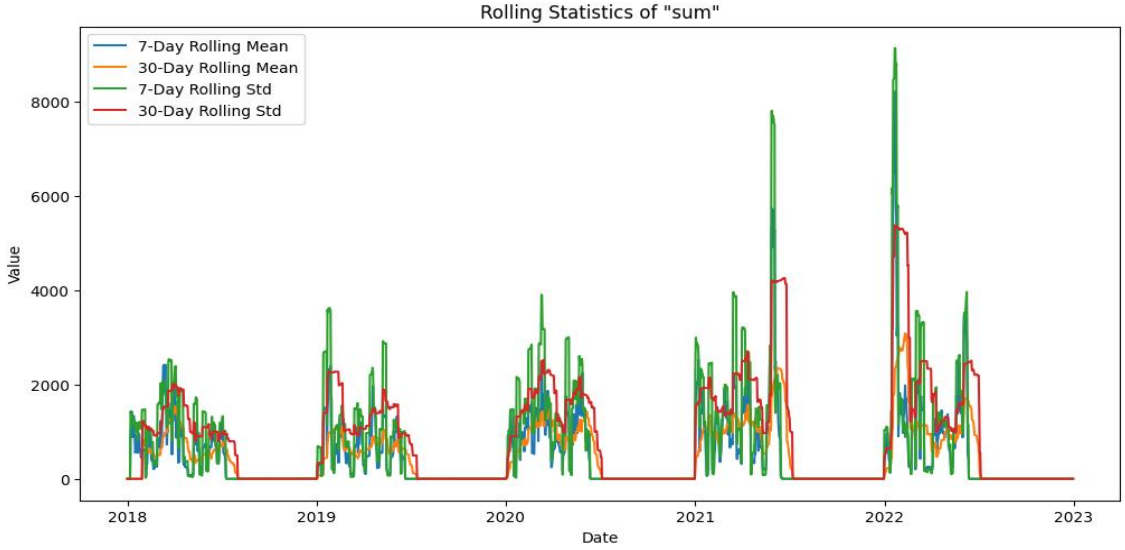

Figure-13: Rolling statistics

v. **Lagged Features:**

Figure–14 gives an overview of how past outages data affect the current data. In the data if the 1 day lagged line followed the original dataset that means there is daily pattern for the outage to take place. As we can see in the graph, the lagged line follows the original line which diminishes over time. The 7 day and 30 days don't follow the 1-day lag and are less aligned which indicates that the decrease in correlation as the time goes by.

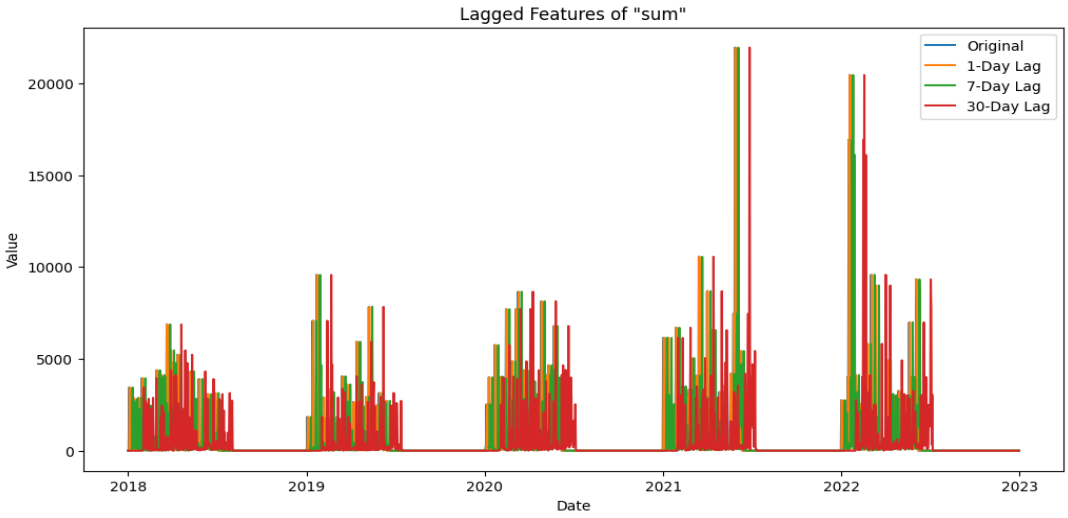

Figure-14: Lagged Features

vi. Cyclic Features:

In Figure**-15**, Day Sine, Day Cosine, Month Sine, and Month Cosine are the four lines that fluctuate between -1 and 1. While the Month Sine and Cosine lines show the lengthier yearly cycles, each completing a full oscillation over the course of a year, the Day Sine and Cosine lines capture the daily cycles, completing a full oscillation approximately every month. For example, the month's beginning may see the peak of the day cosine and its midpoint, the day sine.

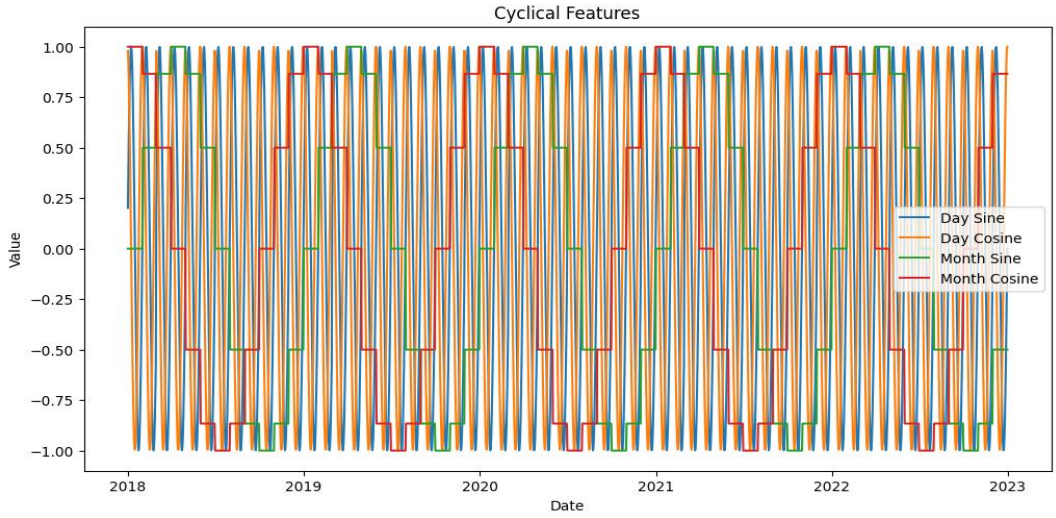

Figure-15: Cyclic Features

Seasonal Decomposition

The graphs show the collection of original data, trend seasonality, and residuals components. The original time series represents the raw data that may be used to analyze the power outage occurrences over time. Greater and evident spikes in the data indicate times of heavy outages. The second graph shows the long-term patterns which includes the rising or falling of the trends by

mainly separating the trend and smoothing out anomalies. The third graph represents the seasonality which shows regular cycles or patterns in the data, such as weekly, monthly or quarterly variations. Lastly, after removing the trend and seasonal influence from the raw data, the residuals graph shows the noise or random changes which can be further analyzed for presence of anomalies not described by the model.

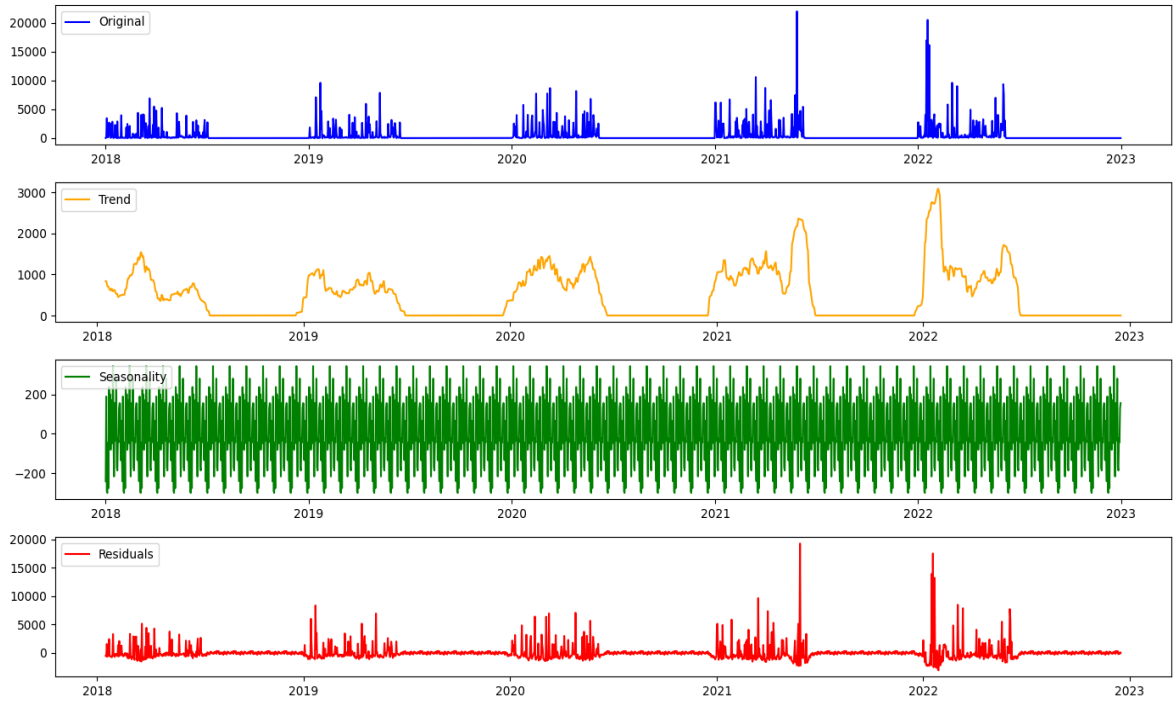

Figure-16: Seasonal Decomposition Graph

Autocorrelation and Partial Auto Correlation Analysis (ACF and PACF)

The **Autocorrelation** graph on the left displays the correlation between the time series at various lags and itself. The first lag has the highest correlation as we can see in the plot which indicates that it is very close to original dataset. The correlation values show fluctuations indicating that the past values have significant impact on the current value, but that impact is not as strong over time.

The **Partial Auto Correlation** graph on the right depicts the partial correlation of the time series with itself at various lags while also controlling for the values of the time series at all shorter lags. The drastically dropping of the values after the first lag indicates that there is not much correlation between the time series points that have been affected by past value.

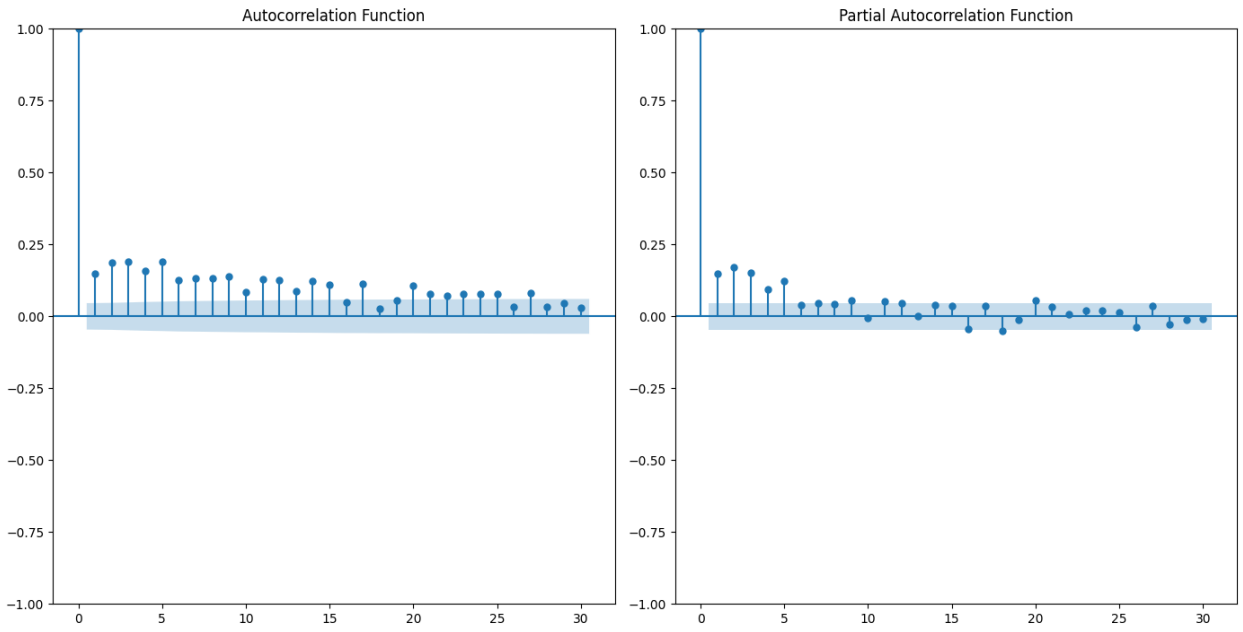

Figure-l7: Autocorrelation and Partial Auto Correlation Plot

Fourier Transform

The graph shows the Fourier Transform Power Spectrum of a time series which signifies the power of various frequencies within the data. The peak at zero frequency indicates that there is a strong constant component in the time series and also that the data might have a mean around it that varies. The absence of other peaks suggests that there are no other seasonal effects present. This is used for analyzing the periodic behaviors or for filtering out noise.

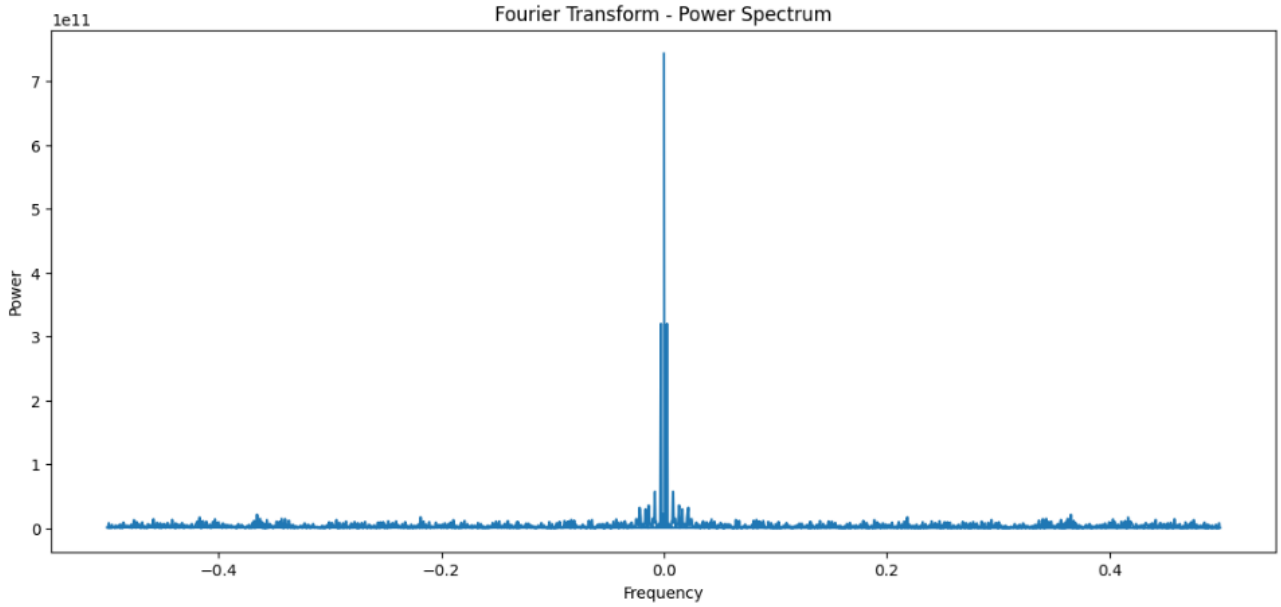

Figure-18: Fourier transform power spectrum

Implementation

The Eagle dataset which provides the power outages for 92% of U.S counties and the weather dataset providing the information of severe weather conditions like Rain, Snow, Hail, Fog etc., has been used to create power outage prediction models. To create the power outage prediction model using ARIMA, ARIMAX, SARIMA, and SARIMAX, the power outage for LEE county in Florida from 2018-2022 has been selected along with weather information for LEE county between these years. The two datasets have been merged and then data preprocessing and time series feature engineering was performed on the merged data frame.

In the merged data frame, some columns named start time, end time, and state columns were dropped and the run_start_time was converted using Date time stamp into Date format and maximum outages for each day were calculated. In the weather dataset the severity for different weather conditions like Rain, Snow, Hail etc., were marked with indices as Light: 1, Moderate: 2, Severe: 3, Heavy: 4, UNK: 5, Other: 6. Thus the merged data frame now shows the columns for Rain, Snow etc. and with severity indices for each day.

Pseudocode implementation of models:

ARIMA model

###Train-Test Data Split

```
# Prepare the data for time series forecasting
from sklearn.model_selection import train_test_split

# Load your dataset
merged_df = pd.read_csv('merged_df.csv')
merged_df['Date'] = pd.to_datetime(merged_df['Date'])
merged_df.set_index('Date', inplace=True)

# Prepare the dataset for linear regression
X = merged_df.drop(columns=['sum','county', 'state'], axis=1)
y = merged_df['sum']


# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.2, random_state=42)
```

###ARIMA

```
# Forecasting using ARIMA

from statsmodels.tsa.stattools import adfuller, acf, pacf
from statsmodels.tsa.arima.model import ARIMA


# Taking out Time Series column
```

Screenshot-4: ARIMA Pseudo code

The screenshot shows code for setting up a time series forecasting task using an ARIMA model. Firstly, loading a dataset and preparing it by setting the dates as an index and dropping the unnecessary columns. A train-test split is performed, keeping 20% of the data for testing. In the ARIMA model, the code includes a stationarity check using the Dickey-Fuller test, which the time series passes, indicating that it is suitable for modeling. It also shows functions to plot Autocorrelation (ACF) and Partial Autocorrelation (PACF) charts, which help determine the ARIMA model parameters.

SARIMAX model

The screenshot shows the code for a SARIMAX model, that accounts for seasonality, to training data. After fitting the model, the summary of the model's performance is output. It then compares original data with the model's fitted values through a plot, with the original data. labeled 'Original' and the fitted data labeled 'Fitted Values' in red.

```
###SARIMAX

from statsmodels.tsa.statespace.sarimax import SARIMAX

# Fitting the SARIMAX Model
sarimax_model = SARIMAX(y_train, order=(1, 0, 1), seasonal_order=(0,
0, 0, 0))
sarimax_result = sarimax_model.fit()

# Printing the summary of the fit
print(sarimax_result.summary())

# Plotting the original series and the fitted values
merged_df['sum'].plot(label='Original')
sarimax_result.fittedvalues.plot(color='red', label='Fitted Values')
plt.legend()
plt.show()

/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/
tsa_model.py:473: ValueWarning: A date index has been provided, but it
has no associated frequency information and so will be ignored when
e.g. forecasting.
  self._init_dates(dates, freq)
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model
.py:473: ValueWarning: A date index has been provided, but it is not
```

Screenshot-5: SARIMAX Pseudo code

XGBoost Model

```
import pandas as pd
from sklearn.model_selection import train_test_split
from xgboost import XGBRegressor
from sklearn.metrics import mean_squared_error
import matplotlib.pyplot as plt

# Loading dataset
merged_df = pd.read_csv('merged_df.csv', parse_dates=['Date'],
index_col='Date')

# Creating time-based features
merged_df['Year'] = merged_df.index.year
merged_df['Month'] = merged_df.index.month
merged_df['Day'] = merged_df.index.day
merged_df['DayOfWeek'] = merged_df.index.dayofweek

# Preparing the features and target
X1 = merged_df[['Year', 'Month', 'Day', 'DayOfWeek']]
y1 = merged_df['sum']

# Splitting the data into training and testing sets
X1_train, X1_test, y1_train, y1_test = train_test_split(X1, y1,
test_size=0.2, random_state=42)

# Initializing and fitting the XGBoost regression model
xgb_model = XGBRegressor(objective='reg:squarederror')
xgb_model.fit(X1_train, y1_train)

XGBRegressor(base_score=None, booster=None, callbacks=None,
             colsample_bylevel=None, colsample_bynode=None,
             colsample_bytree=None, device=None,
early_stopping_rounds=None,
             enable_categorical=False, eval_metric=None,
feature_types=None,
             gamma=None, grow_policy=None, importance_type=None,
             interaction_constraints=None, learning_rate=None,
max_bin=None,
             max_cat_threshold=None, max_cat_to_onehot=None,
             max_delta_step=None, max_depth=None, max_leaves=None,
             min_child_weight=None, missing=nan,
monotone_constraints=None,
             multi_strategy=None, n_estimators=None, n_jobs=None,
             num_parallel_tree=None, random_state=None, ...)
```

Screenshot-6: XGBoost Pseudo code

The screenshot shows the code that loads a time series dataset, and here features are created by extracting year, month, day, and day of the week from the 'Date' column, and these are set as new columns. Then these time-based features are designated and the 'sum' column as the predictors and target for XGBoost regression model. The dataset is split into training and test sets, and xgboost regressor model is trained to predict the target variable.

Implementation of models:

1. ARIMA (AutoRegressive Integrated Moving Average)

The ARIMA model was applied to the sum column, representing power outages. The important steps included checking for stationarity using dickey fuller test, determining ARIMA parameters (p, d, q), and fitting the model to the data. It was observed that ARIMA models are well-suited for time series data without external influences. The model could capture temporal patterns in the data but lacked the ability to incorporate external factors like weather conditions.

2. ARIMAX (ARIMA with eXogenous variables)

Here the ARIMA model is extended to include the exogenous/external variables (weather conditions). ARIMAX provided a better detailed understanding by including the influence of weather on the power outages.

3. SARIMAX (SARIMA with eXogenous variables)

SARIMA (Seasonal ARIMA) usually adds a seasonal component to the ARIMA model, making it suitable for any data with seasonal patterns. Parameters in SARIMA included both non-seasonal (p, d, q) and seasonal (P, D, Q, S) components. ARIMA was found to be effective in modeling both the trend and seasonality in power outage data. SARIMAX is combined to the seasonal features of SARIMA with the ability to incorporate exogenous variables. It required careful selection of both seasonal and non-seasonal parameters, along with relevant external predictors.

4. Linear Regression

Linear Regression is a statistical method used to model the relationship between a dependent variable (power outage sum) and one or more independent variables (like weather conditions). This model assumes a linear relationship between the input variables and the target. These models may not be good in accuracy compared to more complex models, particularly when dealing with non-linear and high-dimensional data.

5. XGBoost (eXtreme Gradient Boosting)

XGBoost is implementation of gradient boosting algorithms in advanced way, and is known for its efficiency, flexibility, and portability and it makes multiple decision trees sequentially, where each tree tries to correct the errors of its predecessor. This model can handle various types of data, and non-linear relationships and interactions between variables. It usually offers high predictive accuracy and is efficient on large datasets.

6. SVM:
Support Vector Machine is useful due to its robustness in classification and regression especially while one is dealing with a high-dimensional dataset. It is very useful in finding the optimal boundary between the outputs to predict the outputs.

7. CatBoost Regressor: It is useful at handling categorical data and complex interactions within the dataset. It can be useful for both regression and classification problems.

8. **LSTM**:
Alongside ARIMA, we implemented Long Short-Term Memory (LSTM) networks, a type of recurrent neural network. LSTMs are designed to recognize the patterns in the sequences of data, making it ideal for time series forecasting like the power outage predictions. It excels in learning from historical data and making predictions about future events, which was crucial in our analysis of the complex relationships between various factors leading to power outages.

Analysis of the Models:

Figure-**19** shows the output of an Augmented Dickey-Fuller (ADF) test and the plots for the Autocorrelation Function (ACF) and for the Partial Autocorrelation Function (PACF), that are used in fitting an ARIMA model to a time series dataset.

The Augmented Dickey-Fuller Test confirms stationarity, that is to proceed with ARIMA modeling without differing the series. The value of -6.433, which is the calculated ADF statistic and a very small p-value (around 1.68e-08), suggests that null hypothesis of a unit root can be rejected. The number of lags used is 1 in this test and the number of observations used in this is 1,860. The critical value thresholds for the test statistic are at 1%, 5%, and 10% levels and as test statistic is more negative than all of these, this data series can be considered a stationary one.
The Autocorrelation Function (ACF) Plot shows the correlation of the time series with lagged values. This plot shows that autocorrelation at lag 0 is 1 and it drops quickly, which is normal for stationary series. The Partial Autocorrelation Function (PACF) Plot shows a significant spike at lag 1 indicating that there is one autoregressive term.

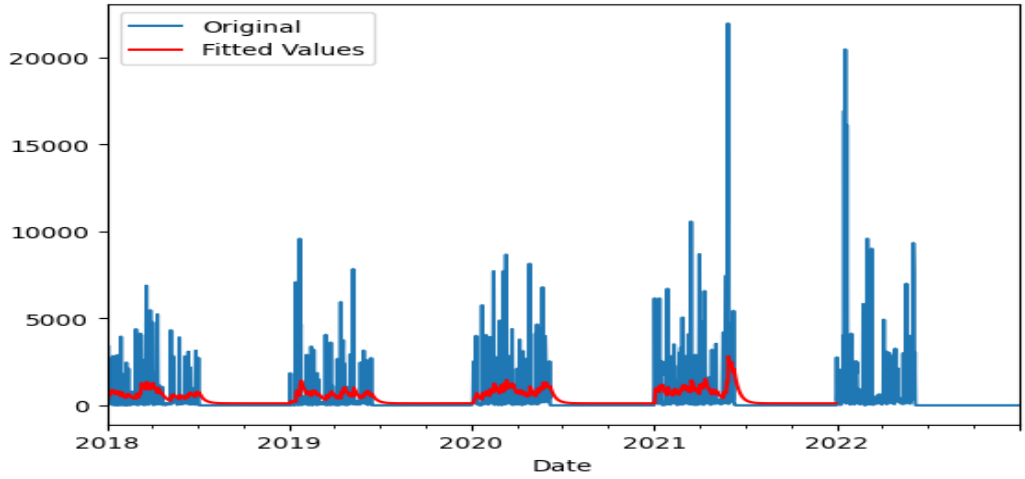

Figure-19: The output of the ARIMA model

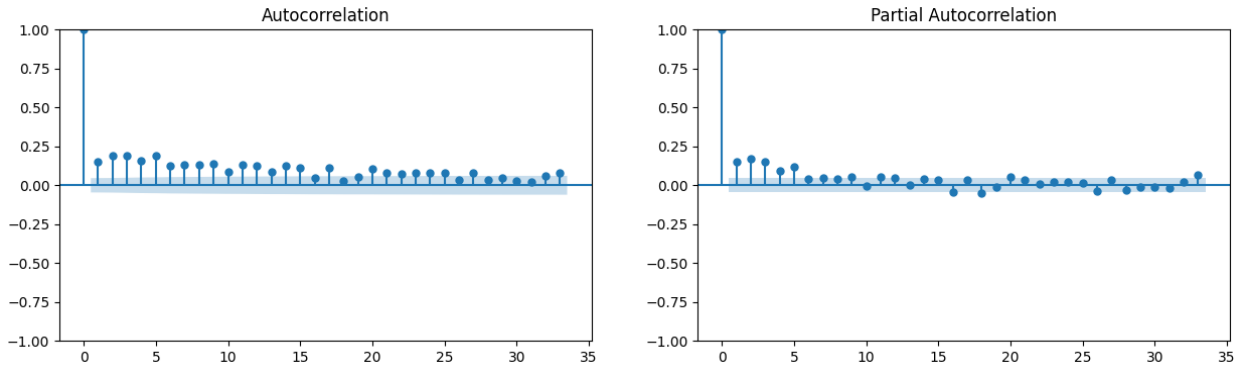

Figure-20: Auto Correlation and Partial Autocorrelation graphs

Figure-20 shows the original data and the fitted values from the ARIMA model over the different Date. The original data shows spikes indicating potential events leading to increased power outages, while fitted values show capturing the central tendency of the data without finding similarity with the spikes. Skewness suggests that the distribution of residuals isn't symmetrical and that it has heavy tails. The ARIMA (1,0,1) model in the figure fits the central trend of the time series data reasonably well but doesn't capture spikes. Significant AR and MA values indicate that the model has identified some temporal structure in the dataset. As the plot shows, while the ARIMA model captures the general pattern, it can't account for the volatility and the spikes. Thus, a more complex model could provide better prediction of power outages.

Results

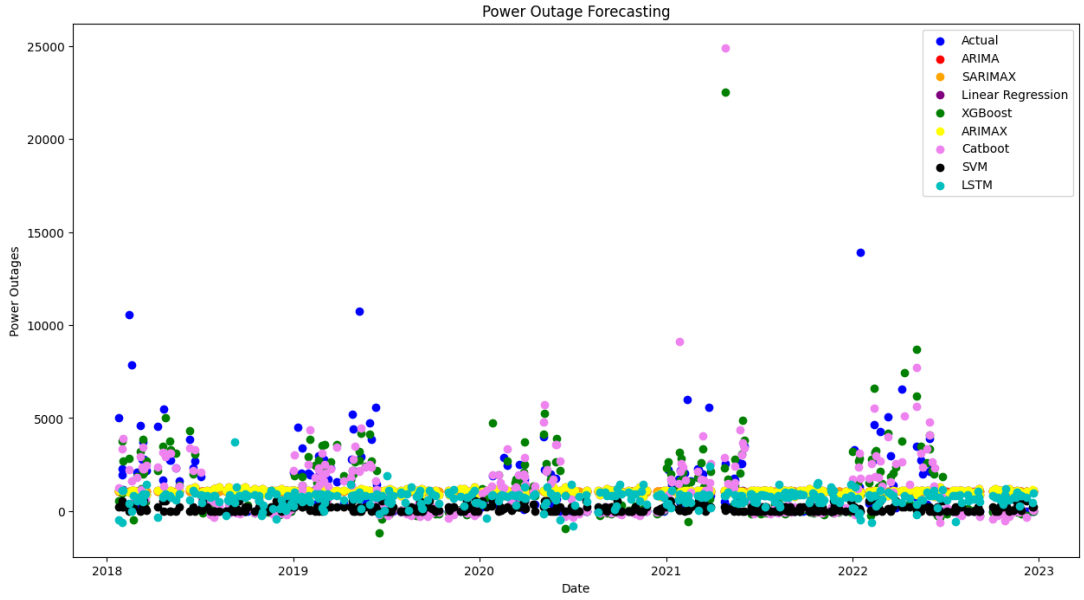

Figure-21: Scatter plot of all the Models

Figure-21 is a scatter plot that compares actual power outage data with forecasts from different models over a time span from 2018 to nearly 2023.The Blue Dots represent the actual power outages over time. Large spikes show the time periods with high number of outages. The orange

dots show the predicted power outages for ARIMA model, and it doesn't capture much. The green dot shows predictions from SARIMA model. The purple dots represent the Linear Regression model's predictions. The yellow dots represent xgboost model predictions, and the spread of predictions is wider than the other models.

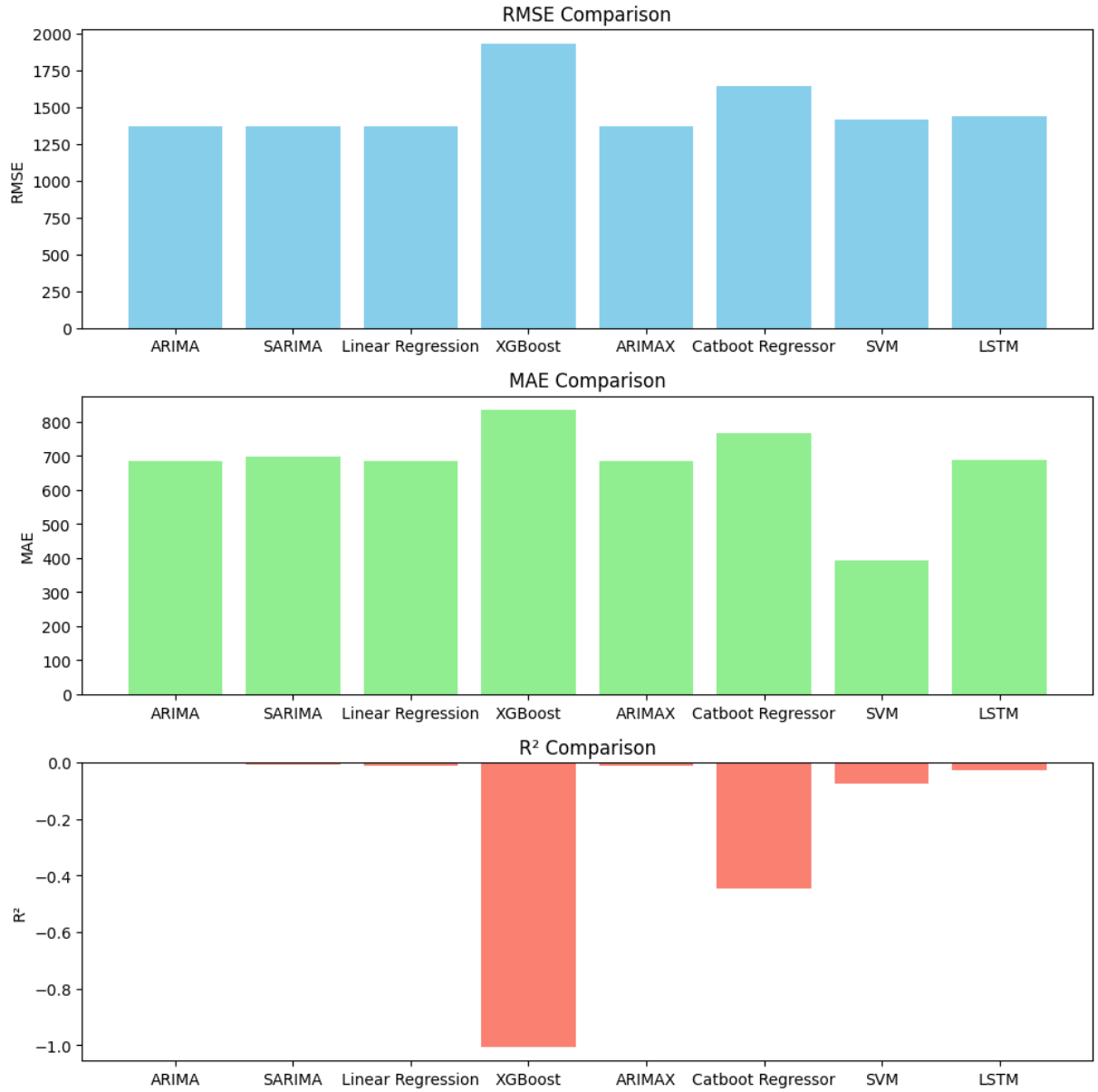

Figure-22: Analysis of the RMSE scores of different prediction models.

The **Figure-22** and metrics shows and compares the performance of four predictive models that are ARIMA, SARIMA, Linear Regression, and XGBoost and using the evaluation metrics as Root Mean Squared Error (RMSE), and Mean Absolute Error (MAE), and R-squared (R^2).

Root Mean Squared Error (RMSE) calculates the square root of average squared differences between the predicted and the actual values and it shows the error value and lower the value of

RMSE shows that that model is a better fit. The ARIMA model has the lowest RMSE values, which is followed by Linear Regression, which shows that it predicts values closer to the actual values than SARIMA model and XGBoost model.

Mean Absolute Error measures average absolute differences between the predicted and the actual values, thus provides linear scale of error. SARIMA model has the lowest MAE value, that means it has the smallest errors in prediction.

R-squared shows the proportion of variance in the dependent variable like power outage which is predictable from the independent variables.

Analysis of the model performance:

Phase-1 models:

ARIMA Metrics - RMSE: 1669.87, MAE: 744.96, R^2: -0.01
SARIMA Metrics - RMSE: 1749.92, MAE: 545.90, R^2: -0.11
Linear Regression Metrics - RMSE: 1664.07, MAE: 777.96, R^2: -0.00
XGBoost Metrics - RMSE: 1845.58, MAE: 782.27, R^2: -0.23

ARIMA and Linear Regression have similar RMSE scores, suggesting that they have a similar overall predictive performance in terms of the magnitude of errors. SARIMA has a slightly higher RMSE but the best MAE score, indicating it may make smaller errors on average but is influenced by large outliers. XGBoost has the highest RMSE and a high MAE, which suggests it is less accurate on this task than the other models. Additionally, its R^2 score is the lowest, indicating the poorest explanatory power among the models.

The negative R^2 across all models implies that the models are particularly not well-suited to this dataset, or that the data itself is highly variable and may not be well-represented by these types of models. It could also indicate that important predictive features are missing or that the relationships in the data are non-linear and complex.

Overall, the choice of the best model would depend on the specific requirements of the application. If the goal is to minimize the average error, SARIMA might be preferred. However, if we are concerned with the magnitude of errors, ARIMA or Linear Regression might be better choices, despite no of the models explaining the variance in power outages effectively.

RMSE comparison of all the models:

ARIMA (AutoRegressive Integrated Moving Average)

RMSE value of 1669.87 shows that the average magnitude of the errors in the predictions is high.MAE value of 744.96 shows that the average error is also significant, but lower than the Linear Regression and the XGBoost models.

SARIMA

RMSE value of 1749.92shows that this model has a higher RMSE than ARIMA, that means a larger error.MAE value of 545.90 shows that it has the lowest MAE, thus showing that it outperforms other models.

Linear Regression

RMSE value of 1664.07 shows that it is similar to ARIMA, thus similar predictive errors. MAE value of 777.96 shows that the MAE value is higher than ARIMA.

XGBoost (eXtreme Gradient Boosting)

RMSE value of 1845.58 is the highest RMSE of all models, thus errors for XGBoost are largest MAE value of 782.27 is Similar to Linear Regression, thus it has a high MAE.

Even though SARIMA typically predicts values that are closer to the actual values, the lower MAE in comparison to its RMSE shows that the bigger errors might be the result of a few outliers. Even though the RMSE values of ARIMA and Linear Regression are somewhat similar, ARIMA has a lower MAE value, which shows it may be better on average in predicting values closer to t actual values. XGBoost has the lowest performance across all metrics, which may suggest that the model structure is not appropriate for this specific problem or that the hyperparameters need to be adjusted.

Phase-2 models:

ARIMAX Metrics - RMSE: 1371.78, MAE: 684.93, R^2: -0.01
Catboot Regressor Metrics - RMSE: 1639.76, MAE: 768.09, R^2: -0.45
SVM metrics - RMSE: 1414.74, MAE: 393.90, R^2: -0.08
LSTM Metrics - RMSE: 1442.05, MAE: 688.48, R^2: -0.03

ARIMAX Model:
The ARIMAX model gave RMSE value as 1371.78 and MAE as 684.93 with R square value as -0.01. This indicates that the model is not able to explain the variance that well.

Catboost Model:
Catboost model resulted in RMSE of 1639.76 and MAE of 768.09 which is higher than ARIMAX indicating a higher error in the predictions. R square is much less indicating even lesser explanation of variability of the model.

SVM Model:
SVM RMSE is slightly higher than the ARIMAX showing a lower prediction error than the previous models.

LSTM Model:
LSTM model RMSE and MAE which are 1442.05 and 688.48 are also much higher than other models, it is closer to ARIMAX error.

Comparison of overall models:
From the overall analysis of all the models as shown in we have analyses in phase-1 and phase-2 we can say that the ARIMA model appears to have performance better than other models. It has lower RMSE and MSE compared to other models. The SVM is also consistent in the error predictions. Other models like XGBoost, Catboost have much higher errors and R Square value is very less. It has lower prediction accuracy. We have further tried to analyze the results for a

different County, which is Miami, but the results were similar to those of LEE county. There are ARIMA performed better than other models.

Models' performance metrics:

County: Lee, Florida

Model	RMSE	MAE	R^2
ARIMA	1366.84	684.56	-0.00
SARIMAX	1368.61	697.59	-0.01
Linear Regression	1371.78	684.92	-0.01
XGBoost	1930.88	834.58	-1.01
ARIMAX	1371.78	684.93	-0.01
CatBoost Regressor	1639.76	768.09	-0.45
SVM	1414.74	393.9	-0.08
LSTM	1442.05	688.48	-0.03

County: Miami-Dade, Florida

Model	RMSE	MAE	R^2
ARIMA	1699.93	1230.64	-0.01
SARIMAX	1697.18	1214.14	-0.0
Linear Regression	1703.14	1225.73	-0.01
XGBoost	1879.99	813.0	-0.23
ARIMAX	1703.34	1226.08	-0.01
Catboost Regressor	1946.44	801.32	-0.32
SVM	1861.92	935.47	-0.21
LSTM	2305.32	1106.94	-0.01

Project Management

Work Completed Phase-1:

- Performed Data preprocessing, Exploratory Data Analysis and Feature Engineering.
- Analyzed trends and correlation between the Outage and Weather datasets.
- Modelled using ARIMA, SARIMAX, Linear Regression and XGBoost machine learning algorithms.
- Evaluated using RMSE, MAE and R^2.

Members Responsibility & Contribution

Module	Description	Action Item	Member	%
1	Data Preprocessing	• Data Cleaning • Data augmentation • Data merging with weather dataset • Dropping invalid columns • Coding and Documentation	Varun Mohan	25%
2	Exploratory Data Analysis	• Time series analysis • Weather correlation • Rolling mean • Lags • Seasonal decomposition • Autocorrelation and Partial Auto Correlation Analysis (ACF and PACF) • Fourier Transform • Coding and Documentation	Panduga Raja Tejasvi Prasad	25%
3	Outage Prediction Models	• Test –train split • ARIMA model • SARIMA model • Linear Regression model • Xgboost model • Coding and Documentation	Yasmeen Haleem	25%
4	Detailed Analysis of Outage Prediction Models	• Forecasting • Analysis of RMSE for models • ARIMA model • SARIMA model • Linear Regression model • Xgboost model • Coding and Documentation	Sravani Katlaganti	25%

Work Completed Phase-2:

- We implemented four more prediction models using SVM, catboost regressor, ARIMAX and LSTM.
- We performed analysis for different counties, as we have performed an analysis on 'Lee' county in Florida and then compared the results.
- Additionally, fine-tuned the models and analyzed the effects in each model's performance.

Members Responsibility & Contribution

Module	Description	Action Item	Member	%
1	Analysis for different counties and model prediction	• SVM model • Coding and Documentation	Varun Mohan	25%
2	Other prediction models implementation and analyzing for different county	• Catboost Regression model • Coding and Documentation	Panduga Raja Tejasvi Prasad	25%
3	Modeling for different counties and model prediction	• LSTM Model • Coding and Documentation	Yasmeen Haleem	25%
4	Model prediction and Detailed Analysis of Outage Prediction Models from different models and counties	• ARIMAX model • Coding and Documentation	Sravani Katlaganti	25%

Issues/Concerns

As the Eagle-i dataset contains the power outage data for 92% of U.S counties every 15 minutes, it was difficult to do time series analysis on each county at the same time. So, the approach taken was to select a county and then perform time series feature engineering on that county and we tried to create the power outage prediction models for the county. We tried to combine the outage values from 2018-2022 and the combined weather dataset from Kaggle to study correlation between power outages and weather.

References:

[1] https://www.osti.gov/biblio/1430039

[2] https://ieeexplore.ieee.org/abstract/document/9763125

[3] https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=9849665

[4] https://www.nature.com/articles/s41467-023-38084-6

[5] https://smc-datachallenge.ornl.gov/eagle/

[6]https://www.kaggle.com/datasets/sobhanmoosavi/us-weather-events/data

[7] https://ieeexplore.ieee.org/document/6851555

[8] https://www.researchgate.net/figure/Different-iterative-steps-of-ARIMA-model_fig3_336021949