FMDB *Transactions on Sustainable Computer Letters*

Time-series Forecasting of Web Traffic Using Prophet Machine Learning Model

Rejwan Bin Sulaiman[1], G. Hariprasath[2,*], P. Dhinakaran[3], Utku Kose[4]

[1]Department of Computer Science and Technology, Arden University, Middlemarch Park, United Kingdom.
[2,3]Department of Computer Science and Engineering, SRM Institute of Science and Technology, Ramapuram, Chennai, Tamil Nadu, India.
[4]Department of Computer Engineering, Suleyman Demirel University, Karasai, Turkey.
rbsulaiman@arden.ac.uk[1], hg8694@srmist.edu.in[2], pp4417@srmist.edu.in[3], utkukose@sdu.edu.tr[4],

Abstract: Forecasting web traffic is critical for website owners, marketers, and organizations to make educated decisions, plan for future development, properly manage resources, and optimize their online presence. Handling online traffic was a simpler and less complicated procedure in the early days of the internet and web development compared to today's standards. The internet was still in its early stages, and websites were simpler. In recent years, handling online traffic with machine learning (ML) time series models have gotten more complex. Machine learning algorithms may give accurate projections and useful insights into online traffic trends. Using Facebook Prophet, a popular forecasting toolkit, this model explains the time series forecasting process and performance evaluation for online traffic data. Prophet's ability to handle complicated time series data with many seasonal components and holidays has won its appeal. Moving Average (MA) models were used for forecasting in time, but there are certain limits and drawbacks to using time series data to capture and forecast underlying trends. MA is specifically designed for short-term forecasting, capturing short-term dependencies and random fluctuations. However, the Prophet model is designed to handle time series data with various seasonal patterns, such as daily, weekly, and annual seasonality. We provide a detailed Explanation of the Prophet Time series model and Evaluation for web traffic data using Facebook Prophet, focusing on understanding model performance and visualization.

Keywords: Moving Average; Prophet of Forecasting; Time Series; Web Traffic; Seasonality of Facebook; Online Traffic Trends; forecasting toolkit; Handling Online Traffic.

Received on: 12/02/2023, **Revised on:** 27/04/2023, **Accepted on:** 01/08/2023, **Published on:** 29/11/2023

Cited by: R. Bin Sulaiman, G. Hariprasath, P. Dhinakaran, and U. Kose, "Time-series Forecasting of Web Traffic Using Prophet Machine Learning Model," FMDB Transactions on Sustainable Computer Letters., vol. 1, no. 3, pp. 161 –177, 2023.

1. Introduction

In today's world, each individual has access to the internet. Everyone can access information through websites, web applications, and other online services. As the number of users increases daily, web traffic is one factor responsible for the success of the internet [13]. People who work for web service providers must know how much traffic a web server receives. If they do not notice the web traffic on the web server, then it may make the people or customers wait longer to access the information and leave the webpage [14]. Web traffic has become essential in today's digital world. Since the World Wide Web was founded, it has transformed the digital ecosystem that powers businesses, transfers information worldwide, and connects people across the globe. As the digital world evolves every day, the ability to understand web traffic has become a crucial part of it [15]. Accurate web traffic forecasting is not easy and needs practicality. Accurate web traffic forecasting help business organization plan their marketing strategies [16].

*Corresponding author.

First, let us understand what web traffic is. Web traffic is the number of visitors and interactions with a website's page and content. In simple words, web traffic refers to the data exchanged between a user's device and the website's server when someone visits a website [17]. Different types of web traffic can be classified based on their characteristics and sources.

The type of traffic from search engines like Google, Bing, or Yahoo is organic traffic. This type of traffic mostly occurs when users find a website by entering relevant keywords in a search and then clicking on either of the search results. Direct traffic is another type of traffic that occurs when a user provides the website URL directly to the browser or when the particular website is bookmarked [18]. This type of traffic represents people already familiar with the website and using it without a search engine.

Social traffic is a type of traffic that occurs from social media platforms such as Instagram, LinkedIn, and Snapchat. This type of traffic occurs when we visit a link that has been shared through social media platforms or visit the website through social media profiles, and it is considered social traffic [19]. Email traffic is a type of traffic that occurs when the links within an email are sent to subscribers or users. This includes promotional or transactional emails that contain a link to their website. Mobile traffic is traffic caused when users access websites using mobile phones or tablets [20].

Of all the traffic, as mentioned earlier, social and mobile traffic are the largest contributors to web traffic, as the number of users in both mediums is larger. As the number of users increases, web traffic increases significantly, impacting people who access the web page [21].

Different types of traffic contribute to the web traffic dataset. Sites have their own method for monitoring incoming and outgoing traffic to determine which part of the page is most frequently visited [22]. The organizational sector, groups, and individuals are moving toward the internet, the biggest contributor to the web traffic dataset. We may require a combination of tools, techniques, and strategic thinking to harness web traffic data [23].

Digital marketers must understand the different types of web traffic to effectively analyze and evaluate an individual's presence on an online platform. Website traffic is important for online businesses but can cause issues like slow loading, high costs, security risks, and bad user experiences [24]. To solve these issues, website owners use security and monitoring tools and strategies to handle increased traffic smoothly [25].

Accurate web traffic projections allow website administrators and online companies to better manage resources. This involves reserving server capacity, bandwidth, and other infrastructure resources to handle expected user traffic. A good user experience is essential for keeping and attracting website visitors [26]. Traffic projections may help content-driven websites like news portals and e-commerce platforms customize their content strategy. They may prioritize and personalize information to engage people more successfully by anticipating which articles, goods, or services will likely be in demand. Digital marketers use web traffic estimates to plan and optimize their advertising campaigns [27]. Accurate projections assist companies in determining when and where to deploy advertising funds, ensuring that commercials reach the appropriate audience at the appropriate time [28].

Content Delivery Networks (CDNs) are critical to effectively delivering digital content. Traffic projections can help security teams discover unusual trends and possible Distributed Denial of Service (DDoS) attacks. Organizations may secure their websites and online services by recognizing abrupt traffic surges and implementing proactive security measures. Accurate site traffic projections help with budgetary planning [29]. Businesses may use forecasts to estimate income, allocate budgets, and define financial targets. Forecasting web traffic encourages data-driven decision-making. Organizations may make educated decisions regarding content development, marketing tactics, user experience upgrades, and more by relying on historical and real-time data [30].

Forecasting models are important tools to forecast future values based on previous data and certain parameters. These forecasting models use mathematical statistical concepts to generate accurate predictions. Forecasting web traffic is essential because it allows digital marketers to plan effectively and manage risks [31]. Time series models forecast future value by analyzing previous data over a specified period. Various web traffic forecasting applications include content and marketing strategy, e-commerce inventory management, user experience optimization, and disaster preparedness [32].

Time series analysis is a fundamental approach to web traffic forecasting. It involves analyzing historical data to identify trends and patterns. Various forecasting models exist today, but choosing the right model is the biggest task. The model's complexity should align with available data and the forecasting horizon [33]. Overly complex models can be computationally intensive and prone to overfitting, while overly simplistic models may fail to capture essential nuances [34].

The process of making predictions about future events based on historical data is known as forecasting. Web traffic forecasting predicts the future volume and web traffic patterns that a website or application will experience. Accurate forecasting helps businesses and website owners plan for capacity and allocate resources effectively [35]. Web traffic forecasting is a complicated task, and it is important to know the domain completely. We should also know mathematics and its related theories to forecast web traffic.

The main focus of this publication is to develop a time series forecasting model for web traffic using date and views as a parameter to predict web traffic. This project data has been designed so that everyday views are recorded.

2. Objective

- Forecasting web traffic using the prophet model is to predict future web patterns accurately. By analyzing past data, we aim to capture underlying patterns and trends in web applications.
- Forecasting web traffic can help in term planning for websites /web applications. It can help businesses plan for capacity expansion, resource allocation, and strategic decisions related to their digital presence.
- Identify the seasonal pattern in web traffic data, such as daily, weekly, and yearly cycles. Understanding these patterns is crucial for optimizing marketing campaigns, content scheduling, and server resources to accommodate peak traffic.

3. Literature Review

R. Casado-Vara et al. [1] approach the problem of web traffic prediction by designing a distributed and scalable architecture that follows a design pattern from bottom to top. The authors use a scrapper to gather the web page's view data, perform preprocessing, discover the features, and find the patterns in the embedded data. They use the concept of LSTM to meet the problem of forecasting the web traffic of the chosen page. They also use pattern detection techniques to design a distributed architecture with several LSTMs, which were trained asynchronously using the downpour strategy.

Kochetkova et al. [2] propose two models, seasonal ARIMA and Holt winters, for short-term mobile network traffic forecasting. These models are applied to real network traffic data from a mobile operator in Portugal and evaluated based on their accuracy in forecasting download and upload traffic. The authors demonstrate that the SARIMA model is more appropriate for forecasting download traffic, while the Holt-Winters model is better suited for forecasting upload traffic. The models are suitable for fast and precise forecasting of mobile network traffic and can be implemented in cellular operators' solutions without a significant increase in cost.

Ma et al. [3] propose the Enhanced Information Graph Recursive Network (EIGRN), a deep learning-based model for accurate traffic volume predictions. The model comprises a Graph Embedding Adaptive Graph Convolution Network (GE-AGCN), a Modified Gated Recursive Unit (MGRU), and a local information augmentation module. The model accurately captures spatiotemporal correlations in traffic data by combining graph convolutional networks and recursive neural networks.

Shelatkar et al. [4] have proposed a web traffic forecasting model using ARIMA and LSTM RNN. The main motive of this project is to predict future web traffic to make decisions for better web congestion control. It also focuses on improving web traffic load management and business analysis. Using LSTM RNN improves system efficiency by capturing seasonal cycles and long trends successfully.

Kong et al. [5] have proposed a model based on a deep learning-based network traffic prediction model using the transformer architecture. The model has been designed in such a way that it captures long-term time series features and parallelizes the output. This model has a faster convergence speed, making it easier to handle multidimensional data.

Lohrasbinasab et al. [6] have suggested strategical-based strategies for network traffic predictions. The author has proposed the ARMA AND ARIMA model for forecasting web traffic. The author used the two linear statistical models to uncover patterns from existing data and forecast future points in time series based on lagged data.

Tambe et al. [7] have proposed an ARIMA model for web traffic forecasting. The author has trained the ARIMA model using the page name, date visited, and number of visits as parameters. These parameters track page popularity weekly, monthly, and annually. The authors also apply a golden ratio-based median of variable time frame window median to boost the model efficiency.

Prajam et al. [8] proposed a method to forecast network traffic in real-time using ANN and LSTM and statistical methods such as SMA and ARIMA. The proposed method uses a sliding window approach, which means the model will use a fixed window of past data to predict future data. They have proposed that the ANN model has superior performance and accuracy compared to other statistical methods.

Tedjopurnomo et al. [9] proposed a deep-learning neural network method for traffic prediction. The author has proposed the model for convolution neural networks, feed-forward neural networks, and Recurrent neural networks. This publication spans the years from 2014 to 2019. The author has debated 37 cutting-edge deep neural networks for traffic prediction.

Irie et al. [10] have presented a system for sequential modeling, monitoring, and forecasting of streaming online traffic data. The author has proposed a PGSS model, and its limitations in adjusting to structural fractures are then summarized. The author then presented the sa-PGSS model and the associated particle-based algorithm.

Liu et al. [11] present a flexible forecasting framework for hierarchical time series with seasonal patterns using a web traffic data case study. The proposed forecasting framework is based on a hierarchical decomposition of the time series, and it incorporates seasonal patterns and multiple models to capture different temporal characteristics in the data. The framework also allows for incorporating external factors into the forecasting process. Results from the case study show that the proposed forecasting framework can effectively capture the seasonal patterns in web traffic data and provide accurate forecasts of future web traffic.

Nunnagoppula et al. [12] examine the potential of deep learning techniques for forecasting website traffic. It presents a novel approach for predicting website traffic based on a hybrid deep learning model that combines Long Short-Term Memory (LSTM) and Convolutional Neural Network (CNN) models. This proposed approach is evaluated using real-world website traffic data and compared to existing methods. The results indicate that the proposed approach yields better accuracy and performance than existing methods. Additionally, the paper discusses the advantages and challenges of deep learning for website traffic forecasting, as well as potential applications and future research directions.

4. Proposed Methodology

4.1. Prophet

Prophet, a forecasting tool created by Facebook, excels in modeling and predicting time series data with numerous seasonality components and significant seasonal trends. It may anticipate online traffic by tracking seasonality daily, monthly, and annually. In this model, we have used a prophet to forecast the web traffic. Facebook's Core Data Science team created the Prophet model. Sean J. Taylor and Benjamin Letham initially described it in their "Forecasting at Scale" work 2017. Prophet is an open-source forecasting tool for time series data. It has grown in popularity because of its ease of use and efficacy in forecasting a wide range of time series data, including business indicators, financial data, and more. Its capacity to manage data with seasonality, holidays, and missing values while giving interpretable projections makes it frequently utilized [36].

The prophet model has advantages, such as effectively handling seasonality, vacations, and trend variations. Data preparation is minimal and has a simple syntax and an easy-to-use UI. It can capture numerous seasons and is suitable for missing data and outliers in time series [37]. Prophet is programmed to identify and model yearly and weekly seasonality in data. This function is especially useful for datasets with various recurrent patterns since it eliminates the need for manual seasonality detection. Users can select holidays and their effects on time series data in the prophet. It includes built-in visualization functions for plotting forecasts and components, making it easy to communicate forecasting results. It is critical to understand that no single forecasting model is always preferable [38]. The selection of a forecasting model is determined by the unique properties of the data as well as the forecasting purpose.

The prophet model is trained on a bespoke dataset that includes dates and views. We divided the dataset into test and train halves, with 80 percent for training and 20 percent for testing. The columns of the data frame have been renamed to ds, which includes the dates, and y, which contains the views; this step is critical for using the prophet model [39]. The train data is then sent into the prophet model for training, and the trained model is used to produce future forecasts. The plotly plot and plots for forecast components such as trend and seasonality are then plotted. Then, the extracted values from the forecast for the test data are stored in y_pred. The true values are extracted from the test data and stored in y_true [40].

Prophet can discover instances in time data where the trend substantially shifts. This capacity to respond to rapid changes in trends enables more accurate forecasting in dynamic contexts [41]. The model produces confidence intervals around the predicted values. Users may use this feature to evaluate possible outcomes and make educated decisions depending on the uncertainty [42]. Prophet is intended to be efficient and scalable to handle small and large datasets. It can easily handle millions of observations [43]. Prophet model is used in many industries and applications for time series forecasting tasks, and some areas are E-commerce and Retail, Finance and banking, Energy and Utilities, Healthcare, Marketing and Advertising, Supply chain and Logistics, Human Resources, Weather and climate, and Real Estate [44].

Prophet was developed primarily for univariate time series forecasting, which models and forecasts a single time series. If your forecasting work demands multivariate forecasting or numerous correlated time series, you may need to examine alternate models [45]. While prophet simplifies many forecasting parts, hyperparameters such as seasonality prior scales and changepoint prior scales must still be tuned [46]. The selection of hyperparameters can affect forecast accuracy. While prophet supports bespoke holidays and events, it lacks sophisticated tools for adding domain-specific information or external factors, which may be required in some forecasting applications [47].

4.2. Architecture Diagram

This diagram depicts the sequence of activities and steps in the code for time series forecasting using Facebook Prophet. Each box indicates a crucial coding step or procedure. Initially, the algorithm loads and analyses the dataset to learn about its structure and properties. Following data loading, the algorithm visualizes the data, maybe utilizing multiple plots and charts to get insights. The code configures and initializes the Facebook Prophet forecasting model, adjusting hyperparameters and other necessary settings [48]. The algorithm may then decompose the time series data to separate the trend, seasonality, and residual components. The Prophet model is trained using previous data to understand patterns and trends. The trained model creates projections for future dates, which are then generated for the forecast period. The method then evaluates the model's performance by computing measures like Mean Absolute Error (MAE), Mean Squared Error (MSE), and Root Mean Squared Error (RMSE) [49]. The algorithm may also classify binary by adding a threshold to the predictions and assessing precision and other classification parameters. Cross-validation may be used to assess how effectively a model generalizes over multiple time frames or eras [50]. This visual representation offers a clear overview of your code's sequential actions and processes, making comprehending the workflow required in time series forecasting using Facebook Prophet simpler (Fig.1).

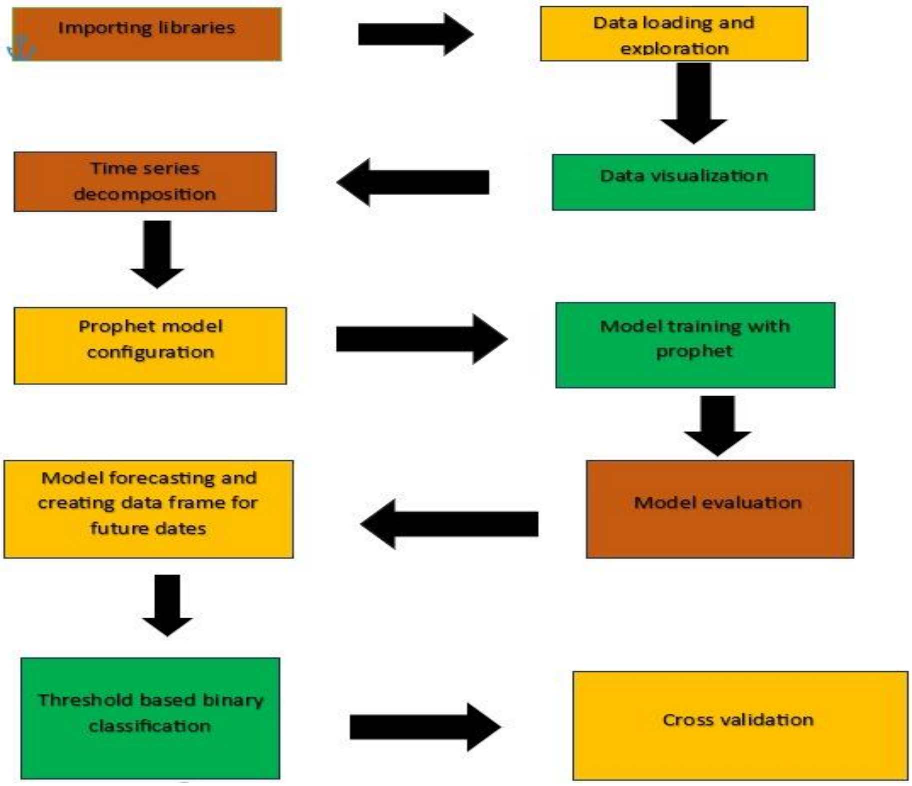

Figure 1: Architecture Diagram for Prophet Model

4.3. Algorithm: Time Series Analysis with Prophet and Evaluation

Step 1: Import Necessary Libraries
- Import pandas, numpy, warnings, seaborn, matplotlib, pyplot, prophet, and relevant functions for metrics.

Step 2: Load Data
- Read a CSV file containing time series data into a pandas Data Frame.

Step 3: Data Exploration
- Display basic information about the dataset:
    - Display the first few rows of the data (data. head()).
    - Determine the shape of the data (data. shape).

- Generate descriptive statistics (data. describe ()).
- Display data type information (data.info ()).
- Display the last few rows of the data (data. tail ()).
- Check for missing values (data. isnull(). sum ()).

Step 4: Data Visualization
- Use seaborn and matplotlib to visualize the data:
  - Create a scatterplot of 'Date' vs. 'Views'.
  - Create a boxplot of 'Views'.
  - Create a line plot of 'Date' vs. 'Views'.

Step 5: Prepare Data for Prophet
- Rename Data Frame columns to 'ds' (date) and 'y' (target variable).

Step 6: Split Data into Training and Testing Sets
- Define the training size (e.g., 80% of the data).
- Split the data into training and testing sets.

Step 7: Create and Configure Prophet Model
- Create a Prophet model with specific configuration settings (seasonality, holidays, etc.).
- Define holiday dates if applicable.
- Fit the Prophet model to the training data.

Step 8: Generate Future Dates for Forecasting
- Create a Data Frame with future dates for forecasting (including test data).

Step 9: Make Predictions with the Prophet Model
- Use the trained model to make predictions on future dates.

Step 10: Evaluate Model Performance
- Calculate various evaluation metrics such as Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and Precision.
- Visualize the error metrics over time.

Step 11: Time Series Decomposition (Optional)
- Decompose the time series into trend, seasonal, and residual components using seasonal decompose.
- Visualize each component.

Step 12: Cross-Validation (Optional)
- Perform cross-validation to assess the model's performance over different time horizons.
- Visualize cross-validation metrics, e.g., MAE.

Step 13: End
- End of the algorithm.

4.4. Formulas

Precision is defined as:

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False positives}}$$

- True Positives are the cases correctly predicted as positive.
- False Positives are the cases incorrectly predicted as positive.

Mean Square Error:

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n} \left| Y_i - \hat{y}_i \right|$$

n = total no of observations.

yi represents the actual values in the dataset at time point i.

y^i represents the forecasted values at time point i generated by the forecasting model.

Mean Squared Error:

MSE=$n1\sum i=1n(yi-y\hat{}i)2$

n = total no of observations

yi represents the actual values in the dataset at time point i.

y^i represents the forecasted values at time point i generated by the forecasting model.

Root Mean Squared Error:

RMSE=$\sqrt{\ } n1\sum i = 1n(yi - y\hat{}i)2$

MAE denotes the average magnitude of forecast errors. MAE handles all errors equally, unlike MSE, which squares the discrepancies and penalizes larger errors more harshly. As a result, it offers a measure of the forecasting model's "typical" inaccuracy.

4.5. Seasonal Decomposition

Seasonal decomposition is a time series analysis statistical approach that divides a dataset into separate components, including the trend, seasonal changes, and residual or error components. It breaks down a time series into constituent components, including the trend, seasonal patterns, and residuals. Visualizing the decomposition in the context of the Prophet model, which is built to handle time series data with complicated seasonal patterns, lets users assess the model's performance and provides a great tool for examining and understanding the data. "Trend" is one of the major components into which a time series is divided in seasonal decomposition. The trend component indicates the time series data's long-term, progressive, consistent pattern or direction. The seasonal component indicates the time series data's regular, periodic, and recurrent patterns or variations at specified periods. After eliminating the impacts of the trend and seasonal impacts, the residual component captures the unexplained variability or noise in the time series data. In other words, it comprises data variations that cannot be traced to a long-term trend or regular seasonal patterns.

4.6. Arima Model

ARIMA (Autoregressive Integrated Moving Average) is a popular time series forecasting model in statistics and econometrics. It is intended to detect and forecast patterns in time series data, a succession of observations recorded at regular intervals. The Autoregressive (AR) Component (p) describes the connection between the current observation and its prior values, where "p" specifies the autoregressive component's order. In other words, it considers the impact of earlier time steps on the current value. Integrated (I) Component (d) is responsible for differencing the time series data to make it stationary. Stationarity denotes that the time series' statistical features, such as mean and variance, do not fluctuate over time.

Moving Average (MA) Component (q) allows for the link between the present observation and the model's previous forecast mistakes or residuals. To summarise, an ARIMA model is characterized by three parameters (p, d, q), which control the order of autoregressive, differencing, and moving average components. ARIMA models are sophisticated time series forecasting methods frequently utilized in domains such as economics, finance, epidemiology, etc. When dealing with seasonal data, they can be expanded to seasonal ARIMA (SARIMA) models. The Arima model analyses and preprocesses time series data, determining the proper p, d, and q values based on autocorrelation and partial autocorrelation plots. Prophet was built from the ground up to handle time series data with significant seasonal trends and holiday impacts. It can identify and predict seasonality annually, weekly, and daily, making it ideal for data with complicated seasonal fluctuations. On the other hand, ARIMA may need manual intervention and extra seasonal differencing for such data. Prophet can recognize and manage outliers in data automatically, ensuring that extreme results or anomalies do not impact the forecast. Outlier identification and treatment in ARIMA often need manual intervention.

Prophet gracefully handles missing data points in time series, making it resilient in cases where data completeness is a concern. ARIMA may require imputation or further data preparation to handle missing values appropriately. When compared to ARIMA, Prophet has fewer hyperparameters to tweak. ARIMA models need careful parameter selection, such as the autoregressive order (p), differencing order (d), and moving average order (q), which can be difficult and time-consuming. ARIMA models are unsuitable for detecting complicated seasonal trends in data. While they can deal with some seasonality through differencing, they may struggle with data with significant seasonal components. ARIMA models presume a linear connection between past and future observations. ARIMA may produce inferior outcomes when the underlying connection is nonlinear. In our case, the prophet is better than Arima. We have compared these two models with their mean absolute error, mean square error, and root mean square error.

4.7. Execution

To implement the prophet model for time series forecasting, we can use the prophet library developed by Facebook. First, install the Prophet library.

Pip install prophet

Then, import the necessary libraries to manipulate the data. Then, create a prophet model object and fit your data.

Model=prophet ()

The next step is to create a data frame with dates for which you want to make web traffic forecasts using the ‘make_future_dataframe.’ Then, we use the prediction method to forecast future dates. Then, we plot the graph using matplotlib or seaborn libraries to visualize the forecasted values.

5. Implementation

5.1. Data and Preprocessing

The dataset used in the project consists of 2 parameters: dates and views. The dataset time range is between 2015 and 2016. The major preparation step is renaming the columns of the Data Frame to conform to the prophet's intended format. The dataset is loaded, analyzed for null values, and contains no null values.

5.2. Data visualization

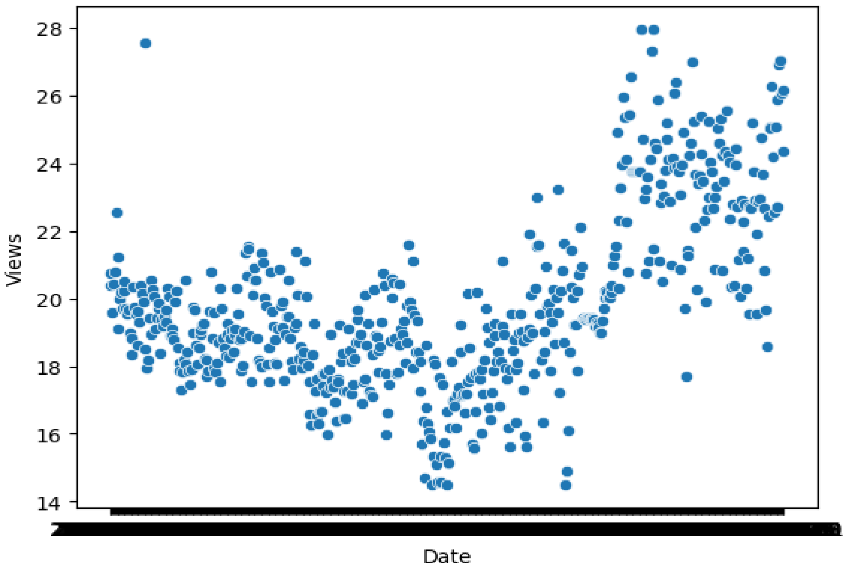

Figure 2: Scatter plot

A scatter plot in Figure 2 is a data visualization technique that shows the connection between two variables. It is frequently used to study the relationship between two variables and to find patterns, trends, clusters, or outliers in data. Each point on the scatter plot corresponds to a data point in the dataset, and the plot aids in visualizing the distribution and any potential patterns or interactions between two variables. Using the seaborn library, a scatter plot is constructed to visualize the relationship between two variables, especially 'Date' and 'Views.' x=data['Date']: The variable plotted on the x-axis is specified here. It reads the data Frame’s 'Date' column, presumably containing dates. y=data['Views']: The variable to be plotted on the y-axis is specified here. It reads the data's 'Views' column.

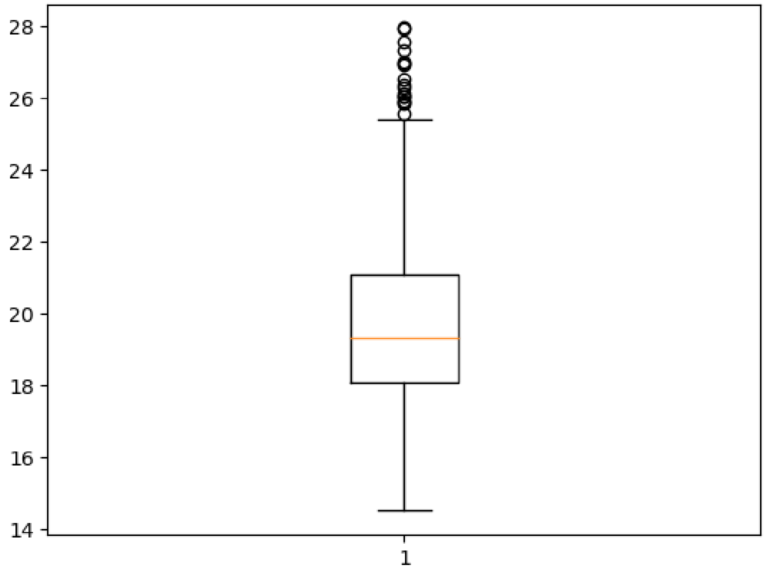

Figure 3: Box plot

A box plot in Figure 3, often known as a box-and-whisker plot, is a graphical depiction of a dataset's distribution and spread. It overviews important statistical parameters such as the median, quartiles, and probable outliers. The plot's box depicts the interquartile range (IQR), encompassing the data's middle 50%. The box's bottom border corresponds to the 25th percentile (Q1), while the box's top edge corresponds to the 75th percentile (Q3). The line within the box reflects the data's median (50th percentile). The whiskers go from the box's edges to the minimum and maximum values within a certain range. These values are often generated using the IQR and can aid in identifying potential outliers.

.

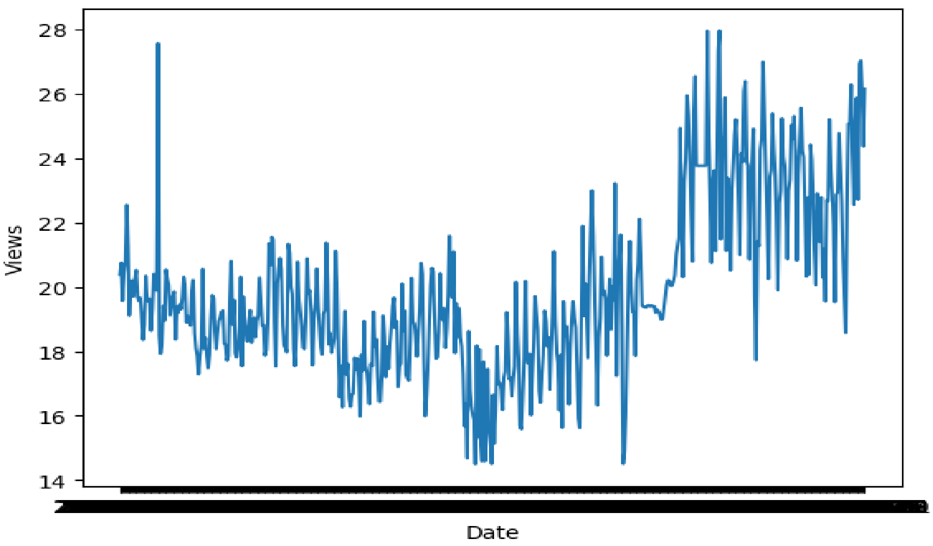

Figure 4: Line plot

A line plot will produce a line plot with the x-axis representing dates or a continuous variable and the y-axis representing the number of views. The line plot in Figure 4 will link the data points with lines, demonstrating how the variable 'Views' varies over time or across the supplied x-values. The plot's line links data points chronologically, demonstrating the trend or pattern in the 'Views' variable over time. The line's slope and direction indicate whether the values are growing, declining, or keeping relatively steady.

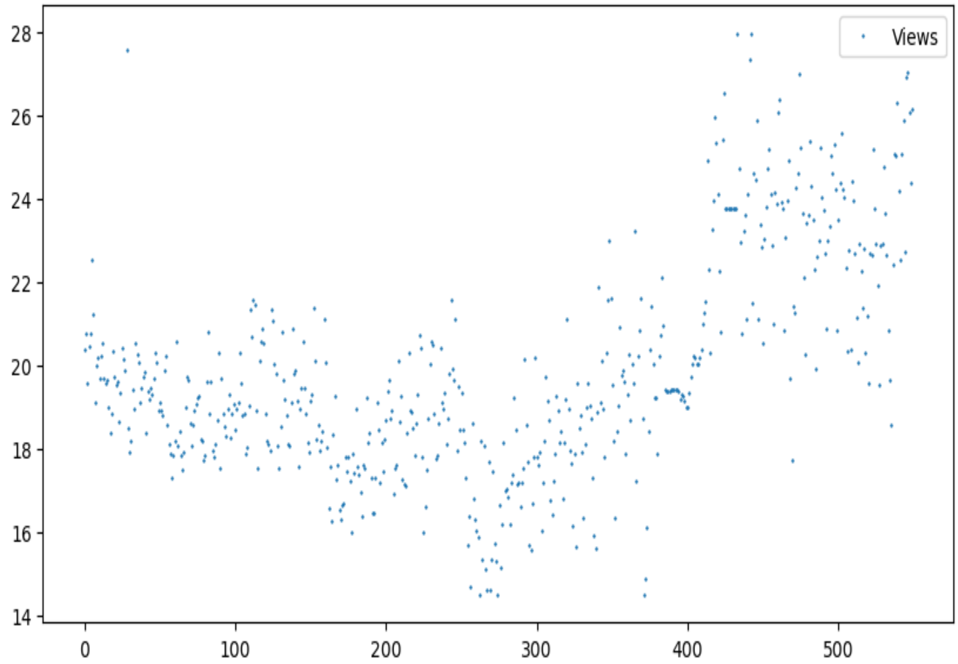

Figure 5: Color Palettes

Color palettes are commonly used to customize the colors of plots or visualizations. It might be used in the prophet to customize the colors of components in plots created by the prophet. Seaborn is a well-known Python data visualization package that includes utilities for producing visually appealing and useful graphs. sns.color_palette() is a Seaborn function that lets you change or customize the color palette used in your Seaborn plots. For instance, you may use sns.color_palette() to define a specific set of colors for a Seaborn plot, such as a line plot or a scatterplot (Figure 5).

5.3. Training

The dataset was split into training and testing data, with 80% and 20% of each. The Prophet machine learning model, deployed by Facebook, was used to train the model. For training, the model jupyter notebook has been used. Python modules such as pandas, numpy, matplotlib, and seaborn are used in the experimental analysis. Jupyter Notebook provides an interactive computing environment where you can simultaneously write and execute code in one cell. This interactivity allows easy exploration and experimentation with data and code. The Prophet model has several configuration options, such as seasonality_mode, seasonality_prior_scale, holidays_prior_scale, changepoint_prior_scale, and n_changepoints.

Seasonality is modeled using the seasonality_mode option. This example is set to 'additive,' indicating that seasonal components are added to the trend component to represent the data. The seasonality_prior_scale controls the seasonality component's intensity. A greater value (10.0) makes seasonality more flexible and impactful. The holidays_prior_scale parameter controls the effect of holidays on model predictions. A larger number (for example, 15.0) emphasizes holidays in the model. The flexibility of changepoint detection in the data is influenced by changepoint_prior_scale. A lower number (for example, 0.05) makes changepoints more flexible. The number of potential changepoints to examine in the data is given by n_changepoints. This code specifies 25 change points.

5.4. Evaluation

Test samples were used to test the trained models, and their performance was assessed using measures such as mean absolute error, mean squared error, root mean squared error and precision. The precision value is 1.0, which means that the predictions made by the model are accurate. Depending on the specific aims of your investigation, it gives different visualizations of error metrics and includes an optional estimate of accuracy for binary classification. The mean absolute and root mean squared errors are 1.32 and 1.75, respectively. The evaluation section of the code allows you to evaluate the Prophet forecasting model's performance by measuring the differences between projected and actual values using MAE, MSE, and RMSE.

6. Result and Discussion

We chose Python and seasonality decomposition to develop our prophet model in this experiment. The proposed model was run and evaluated on Windows 11 with Intel Core i5 12450H, 16GB RAM, and GTX 1650 GPU and re-evaluated on the Google platform.

The dataset has been trained and tested using the Prophet model. The dataset is split into 80% and 20% for training and testing. Metrics like precision, mean absolute error, mean squared error, and Root mean squared error are used to evaluate the effectiveness of the prophet model.

The model was evaluated using the metrics MSE, MAS, and RMSE to measure the accuracy of the model. The values are mentioned in Table 1.

Table 1: Metrics for prophet model

Metrics	Value
Mean absolute error	1.32
Mean squared error	3.08
Root mean squared error	1.75

The same dataset was trained with the ARIMA model, and the same metrics as the Prophet model were evaluated for the ARIMA model. The results are:

Table 2: Comparison of Prophet and ARIMA Model

Models	Mean absolute error	Mean squared error	Root mean squared error
Prophet model	1.32	3.08	1.75
ARIMA model	4.83	28.72	5.35

Table 2 clearly illustrates the difference in metrics for the prophet and ARIMA models. The same dataset was trained, but there is a huge difference between the metrics, which clearly shows that the prophet model is best suited and makes accurate predictions for the loaded dataset.

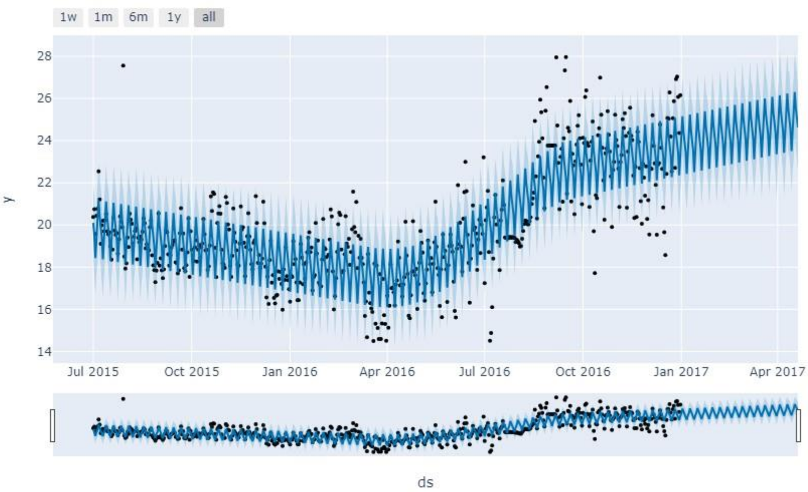

Figure 6: Interactive chart for dates and views

By using the plotly library to plot the graph for forecasted values, Figure 6. This gives an interactive chart where we can alter and see the graph for one week, one month,6months, and one year, and we can view all the graph values as a whole. The graph is plotted between the date and the number of views on that particular day.

Figure 7: Component visualization

The visualization plot, Figure 7, is for the model with forecasted components. This graph is plotted using the 'plot components' method in the Facebook Prophet library to visualize the individual components of a time series decomposition generated by the prophet forecasting model, like trends, seasonality, and holidays. The holiday value is 0 because no dates are mentioned as holidays or special events on this dataset. So, the holiday component does not have any impact on the graph. The trend component represents the underlying long-term movement or direction in the data. The model captures the trend trajectory and projects it to the future. The weekly aspect of the prophet model is a feature given by the prophet model that exhibits weekly patterns, and the model is designed to handle it effectively.

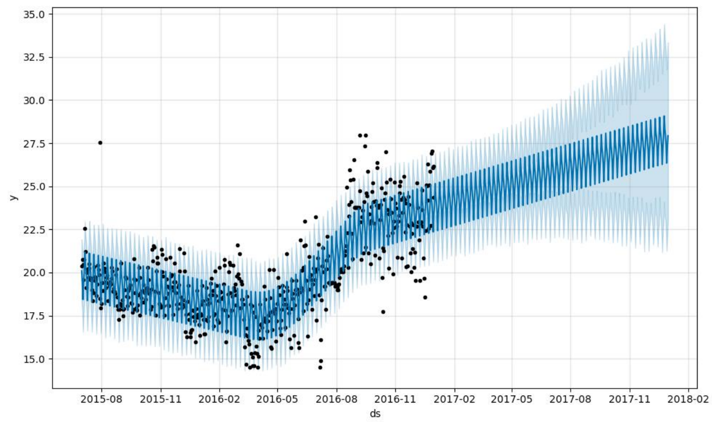

Figure 8: Combination of all components of the forecasted model

Figure 8 combines all three components: trends, seasonality, and noise. The graph is plotted for the date and the forecasted values that the model gave. The x-axis is named ds, and the y-axis is named y. The x-axis marks dates, and the y-axis mentions views

Using the seasonal decomposition library, I tried to break down a time series into its constituent components to understand the underlying patterns and trends predicted by the prophet model. The below graphs represent the trend, seasonal, and residual components separately.

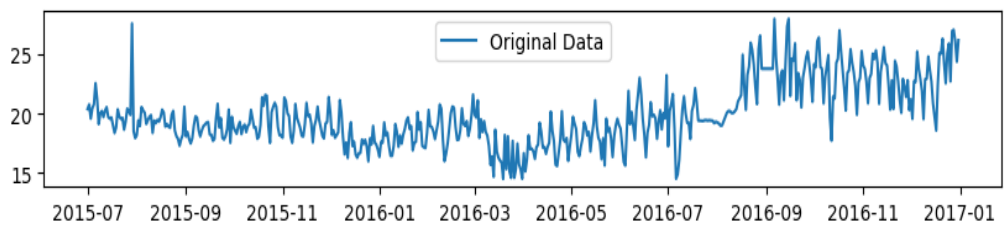

Figure 9: Trend component graph

Figure 9 represents the trend component graph, a visualization plot used in time series analysis to understand the dataset's underlying long-term trend or pattern. It is necessary to understand the trend component as it gives out the overall direction or movement in data and makes informed forecasts and predictions. The above graph illustrates the direction of data moving over time.

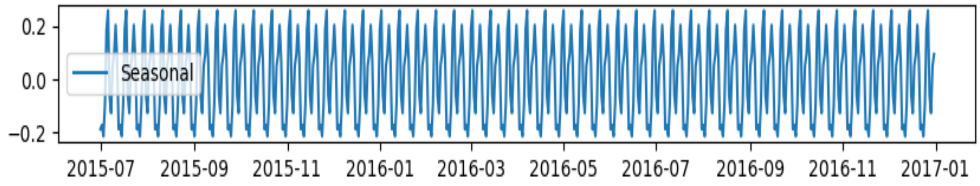

Figure 10: Seasonal component graph

Figure 10 illustrates a seasonal component graph representing the recurring periodic patterns in the data over a fixed interval. This graph helps to understand weekly or yearly patterns. This component graph helps visualize the fluctuations within a time series data set.

This component graph helps understand the patterns and is important for forecasting and understanding underlying trends.

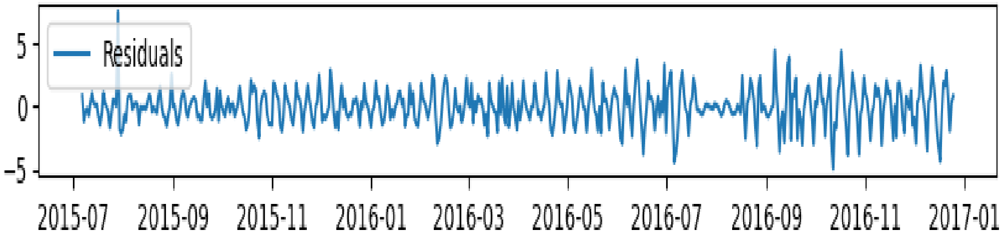

Figure 11: Residual component graph

The residual plots, as represented in Figure 11, are the model's residuals. Residuals are the difference between the observed values and the values predicted by the model. This graph helps assess the model's performance and identify its errors. This plot helps assess the model's validity over time and explains the variability in the data.

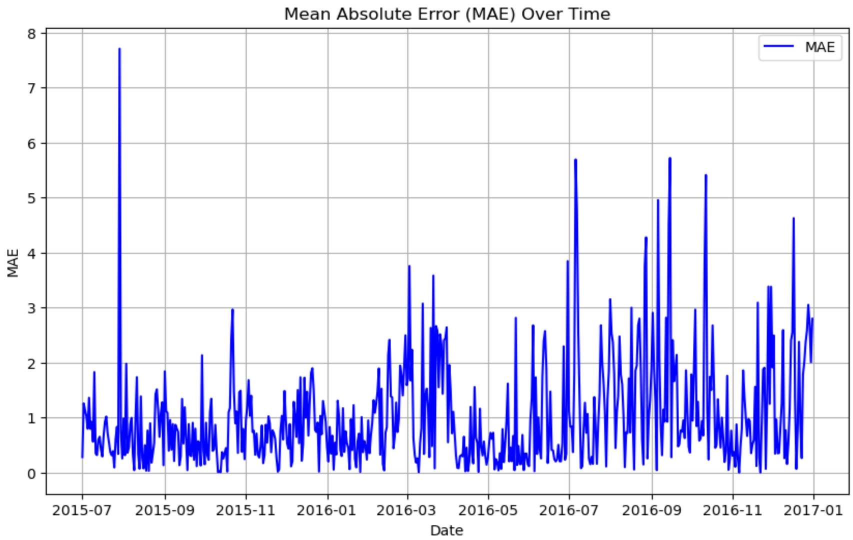

Figure 12: Mean absolute error

The model is evaluated based on its metrics. The mean absolute error over time, as illustrated in Figure 12 metric, is used to evaluate the accuracy of forecasts or predictions made by a time series forecasting model. MAE measures the average magnitude of errors between the predicted values. The MAE overtime analysis shows how well the model predicts its true value as time progresses. The below graph is the plot for MAPE over time. The mean absolute error comes out to be 1.32.

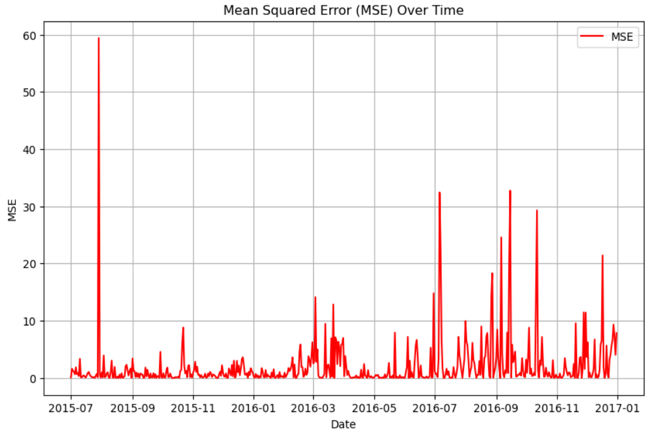

Figure 13: Mean squared error

Figure 13 illustrates the mean squared error calculated over some time. MSE metrics access MSE metrics access the accuracy of predictions by the model as time progresses. By considering the magnitude of errors, calculating MSE over time will help evaluate whether the model's predictions match the true values. The mean squared error comes out to be 3.08.

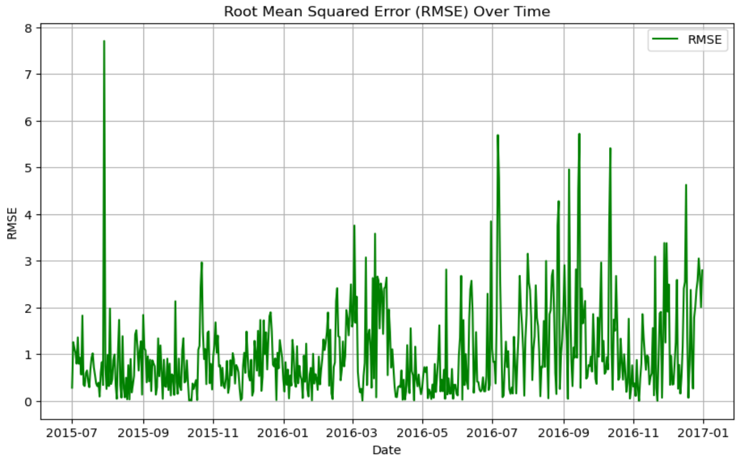

Figure 14: Root mean squared error graph

Figure 14 illustrates the root mean squared error graph, which has been used to evaluate model accuracy. This metric is used to assess the accuracy of the prediction made by the model as time progresses. RMSE is a metric that takes the square root of MSE to measure the error values in the same unit as the original data. The below graph represents the RMSE over time. The root mean squared error comes out to be 1.75.

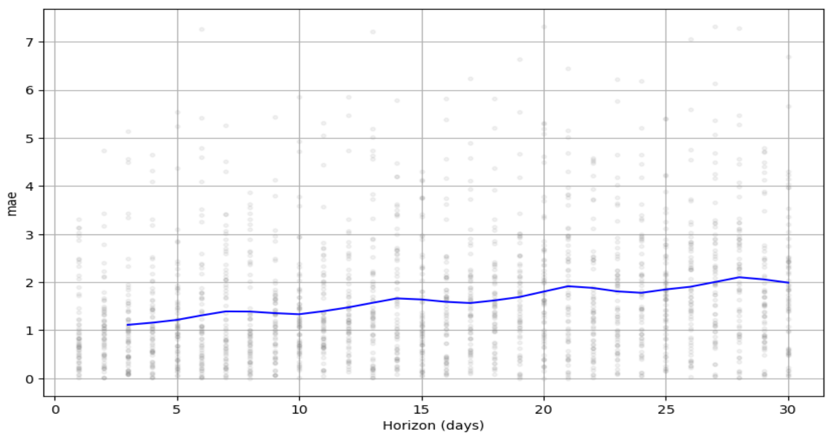

Figure 15: Cross-validation metric graph

Figure 15 represents the cross-validation metrics, a resampling technique that helps us assess the model's performance on unseen data by splitting the dataset into multiple subsets, training the model on some subsets, and evaluating it on others. This metric clearly shows how well the model performs when new data is fed.

7. Conclusion

In this publication, we deployed the prophet forecasting model to analyze and predict web traffic to provide accurate insights for a given data. Our model exhibits strong performance in terms of forecasting accuracy by producing a low mean absolute error, mean squared error, and root mean squared error values, which indicates its efficiency in predicting complex web traffic patterns. The proposed model obtained a mean absolute error of 1.32, a mean squared error of 3.08, and a root mean squared error of 1.75. The lower the metric score, the better the efficiency. Accurate web traffic forecasting is difficult, but forecasting web traffic close to its true value can have a significant impact on web optimization, resource allocation, and user experience. There are also some limitations associated with our study. While the prophet model offers robust performance, it may not account for extreme outliers or sudden unexpected events that affect web traffic. Looking ahead, we suggest future research in web traffic, which includes investigating external factors in the forecasting model, which enhances the model's adaptability for changing web environments. Our research emphasized the significance of accurate forecasting of web traffic in today's digital world so that it may help web administrators, e-commerce platforms, and many others. We hope this publication serves as a valuable contribution to the field and encourages others to further advancements in web traffic forecasting methods.

Acknowledgment: The support of all my co-authors is highly appreciated.

Data Availability Statement: The research contains data related to web traffic and associated metrics. The data consists of views and dates as parameters.

Funding Statement: No funding has been obtained to help prepare this manuscript and research work.

Conflicts of Interest Statement: No conflicts of interest have been declared by the author(s). Citations and references are mentioned as per the used information.

Ethics and Consent Statement: The consent was obtained from the organization and individual participants during data collection, and ethical approval and participant consent were received.

References

1. R. Casado-Vara, A. Martin del Rey, D. Pérez-Palau, L. de-la-Fuente-Valentín, and J. M. Corchado, "Web Traffic Time Series Forecasting Using LSTM Neural Networks with Distributed Asynchronous Training," Mathematics, vol. 9, no. 4, p. 421, 2021, doi: 10.3390/math9040421.
2. I. Kochetkova, A. Kushchazli, S. Burtseva, and A. Gorshenin, "Short-Term Mobile Network Traffic Forecasting Using Seasonal ARIMA and Holt-Winters Models," Future Internet, vol. 15, no. 9, p. 290, Aug. 2023.
3. C. Ma, K. Sun, L. Chang, and Z. Qu, "Enhanced Information Graph Recursive Network for Traffic Forecasting," Electronics, vol. 12, no. 11, p. 2519, 2023, doi: 10.3390/electronics12112519.
4. T. Shelatkar, S. Tondale, S. Yadav, and S. Ahir, "Web traffic time series forecasting using ARIMA and LSTM RNN," ITM Web Conf., vol. 32, p. 03017, 2020.
5. Q. Kong et al., "Network traffic prediction: Apply the transformer to time series forecasting," Math. Probl. Eng., vol. 2022, pp. 1–6, 2022.
6. I. Lohrasbinasab, A. Shahraki, A. Taherkordi, and A. Delia Jurcut, "From statistical- to machine learning-based network traffic prediction," Trans. Emerg. Telecommun. Technol., vol. 33, no. 4, 2022.
7. V. Tambe, A. Golait, S. Pardeshi, R. Javheri, and P. G. Arsalwad, "Web traffic time series forecasting using ARIMA model," Int. J. Res. Appl. Sci. Eng. Technol., vol. 10, no. 5, pp. 2447–2453, 2022.
8. S. Prajam, C. Wechtaisong, and A. A. Khan, "Applying machine learning approaches for network traffic forecasting," Indian J. Comput. Sci. Eng., vol. 13, no. 2, pp. 324–335, 2022.
9. D. A. Tedjopurnomo, Z. Bao, B. Zheng, F. Choudhury, and A. K. Qin, "A survey on modern deep neural network for traffic prediction: Trends, methods and challenges," IEEE Trans. Knowl. Data Eng., pp. 1–1, 2020.
10. K. Irie, C. Glynn, and T. Aktekin, "Sequential modeling, monitoring, and forecasting of streaming web traffic data," Ann. Appl. Stat., vol. 16, no. 1, pp.300-325, 2022.
11. Z. Liu, Y. Yan, and M. Hauskrecht, "A flexible forecasting framework for hierarchical time series with seasonal patterns: A case study of web traffic," in The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, 2018.

12. H. Nunnagoppula, K. Katragadda and M. Ramesh, "Website Traffic Forecasting Using Deep Learning Techniques," 2023 International Conference on Artificial Intelligence and Smart Communication (AISC), Greater Noida, India, pp. 531-536, 2023, doi: 10.1109/AISC56616.2023.10085005.
13. A. A. Alfaifi and S. G. Khan, “Utilizing data from Twitter to explore the UX of ‘Madrasati’ as a Saudi e-learning platform compelled by the pandemic,” Arab Gulf Journal of Scientific Research, pp. 200–208, 2022.
14. A. Bhardwaj, J. Pattnayak, D. Prasad Gangodkar, A. Rana, N. Shilpa and P. Tiwari, "An Integration of Wireless Communications and Artificial Intelligence for Autonomous Vehicles for the Successful Communication to Achieve the Destination," 2023 3rd International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE), Greater Noida, India, pp. 748-752, 2023.
15. A. Bhardwaj, R. Raman, J. Singh, K. Pant, N. Yamsani and R. Yadav, "Deep Learning-Based MIMO and NOMA Energy Conservation and Sum Data Rate Management System," 2023 3rd International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE), Greater Noida, India, pp. 866-871, 2023.
16. A. Bhardwaj, S. Rebelli, A. Gehlot, K. Pant, J. L. A. Gonzáles and F. A., "Machine learning integration in Communication system for efficient selection of signals," 2023 3rd International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE), Greater Noida, India, pp. 1529-1533, 2023.
17. A. Chaturvedi, A. Bhardwaj, D. Singh, B. Pant, J. L. A. Gonzáles and F. A., "Integration of DL on Multi-Carrier Non-Orthogonal Multiple Access System with Simultaneous Wireless Information and Power Transfer," 2022 11th International Conference on System Modeling & Advancement in Research Trends (SMART), Moradabad, India, pp. 640-643, 2022.
18. A. Rashi and R. Madamala, “Minimum relevant features to obtain ai explainable system for predicting breast cancer in WDBC,” Int J Health Sci (Qassim), Sep. 2022, doi: 10.53730/ijhs.v6nS9.12538.
19. A. Uthiramoorthy, A. Bhardwaj, J. Singh, K. Pant, M. Tiwari and J. L. A. Gonzáles, "A Comprehensive review on Data Mining Techniques in managing the Medical Data cloud and its security constraints with the maintained of the communication networks," 2023 International Conference on Artificial Intelligence and Smart Communication (AISC), Greater Noida, India, pp. 618-623, 2023, doi: 10.1109/AISC56616.2023.10085161.
20. D. Bhuva and S. Kumar, “Securing space cognitive communication with blockchain,” in 2023 IEEE Cognitive Communications for Aerospace Applications Workshop (CCAAW), 2023.
21. D. R. Bhuva and S. Kumar, “A novel continuous authentication method using biometrics for IOT devices,” Internet of Things, vol. 24, no. 100927, p. 100927, 2023.
22. D. S. Das, D. Gangodkar, R. Singh, P. Vijay, A. Bhardwaj and A. Semwal, "Comparative Analysis of Skin Cancer Prediction using Neural Networks and Transfer Learning," 2022 5th International Conference on Contemporary Computing and Informatics (IC3I), Uttar Pradesh, India, pp. 367-371, 2022.
23. D. Uike, S. Agarwalla, V. Bansal, M. K. Chakravarthi, R. Singh and P. Singh, "Investigating the Role of Block Chain to Secure Identity in IoT for Industrial Automation," 2022 11th International Conference on System Modeling & Advancement in Research Trends (SMART), Moradabad, India, pp. 837-841, 2022.
24. G. Gupta et al., “DDPM: A dengue disease prediction and diagnosis model using sentiment analysis and machine learning algorithms,” Diagnostics (Basel), vol. 13, no. 6, 2023.
25. G. Tutberidze, T. Khoshtaria, and A. Matin, “The impact of social media engagement on consumers’ trust and purchase intention,” Int. J. Technol. Mark., vol. 14, no. 3, p. 1, 2020.
26. M. Awais, A. Bhuva, D. Bhuva, S. Fatima, and T. Sadiq, “Optimized DEC: An effective cough detection framework using optimal weighted Features-aided deep Ensemble classifier for COVID-19,” Biomed. Signal Process. Control, p. 105026, 2023.
27. M. Fazil, S. Khan, B. M. Albahlal, R. M. Alotaibi, T. Siddiqui, and M. A. Shah, “Attentional Multi-Channel Convolution With Bidirectional LSTM Cell Toward Hate Speech Prediction,” IEEE Access, vol. 11, pp. 16801–16811, 2023.
28. O. Fabela, S. Patil, S. Chintamani, and B. H. Dennis, “Estimation of effective thermal conductivity of porous media utilizing inverse heat transfer analysis on cylindrical configuration,” in Volume 8: Heat Transfer and Thermal Engineering, 2017.
29. R. A. A. Agarwal, “Decision Support System designed to detect yellow mosaic in Pigeon pea using Computer Vision,” Design Engineering (Toronto), vol. 8, pp. 832–844, 2021.
30. R. Agarwal and N. Rao, “ML-based classifier for Sloan Digital Sky spectral objects,” Neuroquantology, vol. 20, no. 6, pp. 8329–8358, 2022, doi: 10.14704/nq.2022.20.6.NQ22824.
31. R. Agarwal, S. Hariharan, M. Nagabhushana Rao, and A. Agarwal, “Weed Identification using K-Means Clustering with Color Spaces Features in Multi-Spectral Images Taken by UAV,” in 2021 IEEE International Geoscience and Remote Sensing Symposium IGARSS, pp. 7047–7050, 2021, doi: 10.1109/IGARSS47720.2021.9554097.
32. R. Oak, M. Du, D. Yan, H. Takawale, and I. Amit, “Malware detection on highly imbalanced data through sequence modeling,” in Proceedings of the 12th ACM Workshop on Artificial Intelligence and Security - AISec’19, 2019.

33. R. Regin, A. A. Khanna, V. Krishnan, M. Gupta, S. Rubin Bose, and S. S. Rajest, “Information design and unifying approach for secured data sharing using attribute-based access control mechanisms,” in Recent Developments in Machine and Human Intelligence, IGI Global, pp. 256–276, 2023.
34. R. Regin, Shynu, S. R. George, M. Bhattacharya, D. Datta, and S. S. Priscila, “Development of predictive model of diabetic using supervised machine learning classification algorithm of ensemble voting,” Int. J. Bioinform. Res. Appl., vol. 19, no. 3, 2023.
35. R. Yousef, S. Khan, G. Gupta, B. M. Albahlal, S. A. Alajlan, and A. Ali, “Bridged-U-Net-ASPP-EVO and deep learning optimization for brain tumor segmentation,” Diagnostics (Basel), vol. 13, no. 16, 2023.
36. S. Khan et al., “Manufacturing industry based on dynamic soft sensors in integrated with feature representation and classification using fuzzy logic and deep learning architecture,” Int. J. Adv. Manuf. Technol., pp. 1–13, 2023.
37. S. Patil, S. Chintamani, B. H. Dennis, and R. Kumar, “Real time prediction of internal temperature of heat generating bodies using neural network,” Therm. Sci. Eng. Prog., vol. 23, no. 100910, p. 100910, 2021.
38. S. Patil, S. Chintamani, J. Grisham, R. Kumar, and B. H. Dennis, “Inverse determination of temperature distribution in partially cooled heat generating cylinder,” in Volume 8B: Heat Transfer and Thermal Engineering, 2015.
39. S. R. Sandeep, S. Ahamad, D. Saxena, K. Srivastava, S. Jaiswal, and A. Bora, ‘To understand the relationship between Machine learning and Artificial intelligence in large and diversified business organisations’, Materials Today: Proceedings, vol. 56, pp. 2082–2086, 2022.
40. S. R. Vadyala, S. N. Betgeri, and N. P. Betgeri, “Physics-informed neural network method for solving one-dimensional advection equation using PyTorch.” Array, vol. 13, p. 100110, 2022, doi: 10.1016/j.array.2021.100110.
41. S. Sengupta, D. Datta, S. S. Rajest, P. Paramasivan, T. Shynu, and R. Regin, “Development of rough-TOPSIS algorithm as hybrid MCDM and its implementation to predict diabetes,” International Journal of Bioinformatics Research and Applications, vol. 19, no. 4, pp. 252–279, 2023.
42. S. Sonnad, M. Sathe, D. K. Basha, V. Bansal, R. Singh and D. P. Singh, "The Integration of Connectivity and System Integrity Approaches using Internet of Things (IoT) for Enhancing Network Security," 2022 5th International Conference on Contemporary Computing and Informatics (IC3I), Uttar Pradesh, India, 2022, pp. 362-366, 2022.
43. S. Venkatesan, S. Bhatnagar, and J. Luis Tinajero León, “A Recommender System Based on Matrix Factorization Techniques Using Collaborative Filtering Algorithm,” neuroquantology, vol. 21, no. 5, pp. 864–872, 2023.
44. S. Venkatesan, S. Bhatnagar, I. M. H. Cajo, and X. L. G. Cervantes, “Efficient Public Key Cryptosystem for wireless Network,” Neuroquantology, vol. 21, no. 5, pp. 600–606, 2023.
45. T. Khoshtaria and A. Matin, “Qualitative investigation into consumer motivations and attitudes towards research shopping in the Georgian market”,” Administration and Management, vol. 48, pp. 41–52, 2019.
46. T. Matin, “The Impact of Social Media Influencers on Brand Awareness, Image and Trust in their Sponsored Content: An Empirical Study from Georgian Social Media Users,” International Journal of Marketing, Communication and New Media, vol. 10, no. 18, 2022.
47. V. Bansal, A. Bhardwaj, J. Singh, D. Verma, M. Tiwari and S. Siddi, "Using Artificial Intelligence to Integrate Machine Learning, Fuzzy Logic, and The IOT as A Cybersecurity System," 2023 3rd International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE), Greater Noida, India, pp. 762-769, 2023, doi: 10.1109/ICACITE57410.2023.10182967.
48. V. Bansal, S. Pandey, S. K. Shukla, D. Singh, S. A. Rathod and J. L. A. Gonzáles, "A Frame Work of Security Attacks, Issues Classifications and Configuration Strategy for IoT Networks for the Successful Implementation," 2022 5th International Conference on Contemporary Computing and Informatics (IC3I), Uttar Pradesh, India, pp. 1336-1339, 2022, doi: 10.1109/IC3I56241.2022.10073429.
49. V. Gunturu, V. Bansal, M. Sathe, A. Kumar, A. Gehlot and B. Pant, "Wireless Communications Implementation Using Blockchain as Well as Distributed Type of IOT," 2023 International Conference on Artificial Intelligence and Smart Communication (AISC), Greater Noida, India, pp. 979-982, 2023, doi: 10.1109/AISC56616.2023.10085249.
50. V. Suthar, V. Bansal, C. S. Reddy, J. L. A. Gonzáles, D. Singh and D. P. Singh, "Machine Learning Adoption in Blockchain-Based Smart Applications," 2022 5th International Conference on Contemporary Computing and Informatics (IC3I), Uttar Pradesh, India, pp. 372-378, 2022, doi: 10.1109/IC3I56241.2022.10072980.