"An Empirical Comparison of LSTM, SARIMAX, and Random Forest Models for Daily Web Traffic Time-Series Forecasting"

Chandana T H *4JN22IS046*
Information Science and Engineering JNN college of Engineering
(VTU Affiliation)

Chinmayi R M *4JN22IS051*
Information Science and Engineering JNN college of Engineering
(VTU Affiliation)

Deepthi G R *4JN22IS058*
Information Science and Engineering JNN college of Engineering
(VTU Affiliation)

Jeevitha A P *4JN23IS409,**Prathima L** Asst. Professor*
Information Science and Engineering JNN college of Engineering
(VTU Affiliation)

Abstract: - The fidelity of forecasting the traffic of the websites is vital to maintain the reliable services, to utilize server resources in the most effective manner, and to concentrate on the seamless user experience. Web traffic has certain trends such as seasonal variations, trend and sudden spikes and makes the traditional predictive tools ineffective. It is proposed in the paper to use the combination of time-series analysis and machine learning and deep learning methods to predict the number of visits to the site per day using the past data. It is a mixture of Random Forest (RF), Seasonal ARIMA and exogenous factors (SARIMAX) and Long Short-Term Memory (LSTM) models. The increase in the quality of prediction is carried out by some of the most significant steps of preprocessing of data including the missing values, the creation of the features of interest, and the normalization of data. Both models produce a prediction in 30 days and are both assessed using the MAE, MAPE, RMSE, accuracy, and precision metrics. A web-based dashboard can be used to visualize results, compare and export them to allow making practical decisions. It is determined that the LSTM and Rand Forrest models are closer to its predictions as compared to the SARIMAX which provides consistent and explainable predictions.

Keywords: Web Traffic Forecasting; Time-series analysis; machine learning; deep learning; LSTM networks; SARIMAX; random forest; model evaluation.

1. INTRODUCTION

The rapid growth of digital platforms has significantly increased the number of websites and online services, resulting in a substantial rise in web traffic across sectors such as e-commerce, education, news, and governmental sites. Web applications should be swift, universally available at all times, and robust enough to manage sudden increases in traffic efficiently. Precise prediction of website traffic is vital for guaranteeing service stability, maximizing server capacity, and maintaining uniform user experiences. Avoiding peak usage times might result in delayed responses, service disruptions, customer complaints, and financial setbacks, particularly for applications heavily dependent on content and income.

Web traffic displays temporal characteristics including trends, seasonality, and sudden shifts, which pose a problem for traditional statistical forecasting techniques like ARIMA and SARIMA. These models frequently overlook intricate, non-linear trends and abrupt changes triggered by marketing strategies, events, or viral phenomena. Utilizing machine learning techniques like Random Forest, SARIMAX, and LSTM models has become prevalent for

forecasting time-series traffic data. These models scrutinize past data to detect patterns, seasonal shifts, and outliers, enhancing the precision and dependability of predictions.

This document presents a Web Traffic Forecasting System that integrates machine learning and deep learning techniques within a single framework. At first, the system removes old visitor information from the website. Subsequently, it delineates crucial attributes. Consequently, it utilizes Random Forest, SARIMAX, and LSTM models to predict daily traffic. An intuitive web application built with Flask and HTML assists users in analyzing predictions, assessing model performance, and downloading forecast outcomes for strategic use. Through meticulous analysis coupled with an intuitive interface, the system facilitates early server monitoring, minimizes the risk of service interruptions, and boosts operational effectiveness. Additionally, the framework can be expanded to manage multiple websites concurrently, incorporate anomaly detection capabilities, and generate automated notifications, providing a scalable solution for enterprise-level traffic control.

2. RELEVANT WORK

To support capacity planning and service reliability, numerous studies have explored time-series and machine learning techniques for website traffic forecasting. The ARIMA model, first introduced by Box and Jenkins [1], remains a foundational statistical approach for time-series prediction and has been widely applied to web traffic data due to its ability to capture trends and temporal dependencies. However, ARIMA assumes linearity and stationarity, which limits its effectiveness when modeling complex and rapidly varying traffic patterns.

To explicitly address seasonality, Hyndman et al. [2] proposed the Seasonal ARIMA (SARIMA) model as an extension of ARIMA that incorporates seasonal components. SARIMA has demonstrated improved accuracy for daily and weekly website traffic forecasting, although it remains sensitive to sudden traffic surges and nonlinear variations.

With the advancement of machine learning, Breiman [3] introduced the Random Forest algorithm, which has been adopted for time-series forecasting due to its robustness and ability to model nonlinear relationships. Several studies have reported that Random Forest models outperform traditional statistical approaches, particularly when handling noisy and irregular web traffic data.

In recent years, deep learning methods have gained significant attention for web traffic prediction. Hochreiter and Schmid Huber [4] proposed the Long Short-Term Memory (LSTM) network, which is capable of learning long-term temporal dependencies in sequential data. LSTM-based models have consistently demonstrated superior performance in capturing complex traffic patterns and peak fluctuations compared to conventional statistical methods.

3. PROPOSED MODEL

In this section, we describe the proposed architecture for web traffic forecasting. The proposed Web Traffic Forecasting System is a modular, end-to-end solution for predicting website traffic. It begins with data preprocessing, which cleans historical traffic data by handling missing values, removing duplicates, generating continuous date ranges, and creating features such as lag values, rolling averages, and day-of-week indicators. For sequence-based learning with LSTM, the data is scaled and converted into sliding windows. The forecasting module employs three models: SARIMAX to capture linear trends and seasonality, Random Forest to model non-linear patterns, and LSTM to learn long-term temporal dependencies, each generating 30-day forecasts stored in a structured format to facilitate comparison and enhance reliability. A web interface built with Flask, HTML, CSS, and JavaScript visualizes predictions through interactive graphs and tables. This layered design ensures modularity, scalability, and ease of interpretation, making the system suitable for both academic research and practical deployment.

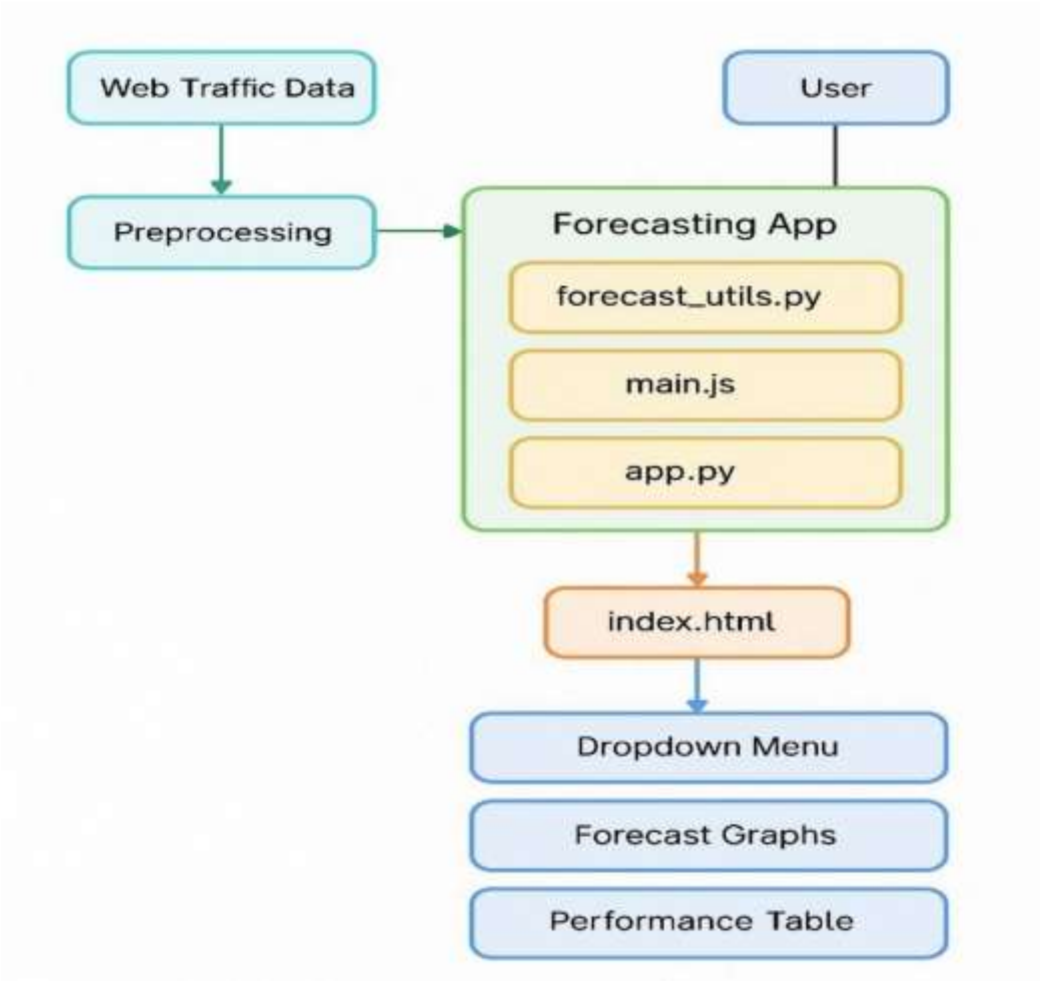

Figure 1. Proposed architecture for web traffic forecasting.

The full process of the Web Traffic Forecasting System is shown in the flow chart in Figure 1, which also illustrates how data moves across different modules before the final output is produced for the user. The system begins with web traffic data, which includes the historical visitor data used for forecasting. The first place this raw data is sent is the Preprocessing module, which performs essential cleaning tasks such handling missing values, generating continuous dates, scaling, and feature creation. The dataset is preprocessed to make it suitable for training the prediction models.
The most important step before training any forecasting model is data preparation. Raw web traffic data often contains noise, erratic values, missing dates, and formatting issues. Feeding this unprocessed data directly into ML/DL models will result in inaccurate predictions. For this reason, the data must be properly cleaned and converted.
The planned Web Traffic Forecasting System utilizes time series-based machine learning and deep learning approaches with historical traffic data to predict future website visitor patterns. The system's architecture is well-thought out, modular, and adaptable, guaranteeing accurate forecasts, ease of use, and versatility for upcoming enhancements. In the overall workflow, each stage is accountable for a specific role in the forecasting process and is related to the others.

3.1 Data Collection, Cleansing, Curation

To start with we got historical statistics of web traffic such as the number of visitors per day, recorded by date, essentially the actual traffic trend on the ground over a substantially long period of time to detect trends, seasonality, and those strange spikes. As our aim was to predict the daily traffic, we maintained it at the day scale and flung into a clean table. In order to ensure that nothing is left out, we constructed a full-fledged date range that contains all the earliest to the latest timestamp, and then we combined it with raw data, which immediately indicated any gaps. Then we sorted out the mess: made vacant values have forward or backward or zero fill, according of the behavior of the traffic, and tore out duplicates or conflicting entries. After the cleaning, we prepared the data to suit the various models by computing lag features, rolling statistics, and calendar variables of ML such as day-of-week. When it comes to deep learning we normalized the series and converted it into sliding windows to ensure that time dependent features remain preserved. Lastly, we divided the curated set into training and test sets in chronological order to prevent any data leakage and ensure that the evaluation is fair.

3.2 Feature Engineering and Pattern Recognition

Therefore, we were requested to study the behavior of traffic and feature engineering was applied and it is really time and pattern oriented. I found myself making lag variables, rolling statistics and calendar flags to identify such fast changes and weekly routines to the ML models. In the deep learning aspect, we cut the time series into windows of constant size such that the network could be directly trained on the timing aspects. Such arrangements allowed the models to identify trends, seasonality, and odd bumps with ease hence making future traffic prediction convenient with very minimal overhead on fancy features.

3.3 Forecasting Models Employed

We combine various forecasting models that consider various time dynamics to have good predictions of the daily traffic of the site. I chose a machine-learning regression, deep-learning sequential model, and a statistical time- series model. All of them are set to pick up things such as seasonality, nonlinear idiosyncrasy, and longer-term cycles. With all of them used simultaneously the predictions are stronger and allow us to compare performance in a methodical manner.

3.3.1 Seasonal ARIMA with Exogenous Variables (SARIMAX)

The SARIMAX (seasonal ARIMA with exogenous variables) model is used when the variation is seasonal and the variation in the trend is thought to be caused by other variables (exogenous variables). SARIMAX is based on ARIMA, but it adds seasonal components and external influences to it, allowing us to model repeating patterns and external effects more directly. In the case of the website traffic, it pegs weekly seasonality and consistent trends that one would normally see in counts of users per day. It combines autoregressive words, differencing, moving-average words and seasonal parameters to train repeating timing patterns. Once the data on the daily traffic are pre-processed and made stationary, SARIMAX crunches out multi-step predictions using the extrapolated learn trend and the seasonal fragments, and this makes it a transparent and understandable baseline model.

3.3.2 Forecasting Model based on Random Forests.

The Random Forest is an ensemble learning algorithm that combines the answers of several decision trees to enhance generalization and decrease overfitting. As it is not a natural time-series approach, I encode the data as a supervised learning representation by generating lagging features, rolling statistics and calendar features. That allows the model-to-model non-linear relationships and irregular traffic patterns that are a result of user behavior or outside influences. Random Forest is noise-tolerant, and its large-scale compute demands are not very great, so it would be a good short-term predictor of web traffic with that non-linear idiosyncrasy.

3.3.3 Long Short-Term Memory (LSTM)-based Forecasting Model.

LSTM is another type of deep learning RNN, which stores long-term contextual knowledge by using memory cells and gating. I would train my LSTM on sliding-window sequences based on normalized daily traffic data in my system, at which point it would learn how patterns change and directly identify delayed effects in history. It is particularly effective when long-term temporal dependencies and complicated traffic dynamics are required, which explains why it is appropriate when the horizon is longer. Nevertheless, it is more complex in computational terms and sensitive to hyperparameters choices and therefore careful tuning is needed.

3.4 Performance Evaluation Metrics

3.4.1 MAE (Mean Absolute Error): MAE measures the average magnitude of errors between predicted and actual values without considering their direction. It shows how much the model is off, on average, in the same units as the target variable—in this case, the number of website visitors. This metric is intuitive and easy to interpret, making it practical for real-world planning and resource allocation, such as server scaling or load balancing. Since MAE treats all errors equally, it provides a clear picture of overall model accuracy but does not penalize large errors more than small ones.

$$MAE = \frac{1}{n}\sum | y_{\text{actual}} - y_{\text{predicted}} |$$

3.4.2 MAPE (Mean Absolute Percentage Error): MAPE expresses prediction error as a percentage of the actual values, which allows for easier comparison across websites or datasets with different traffic scales. It is especially useful when website traffic varies significantly, as it normalizes the error relative to the magnitude of the data. MAPE helps stakeholders understand the relative accuracy of models in business terms—for instance, knowing that predictions are off by 5% on average is more actionable than knowing the absolute visitor count error. One limitation of MAPE is that it can be distorted by very small actual values, so care must be taken when interpreting results for low-traffic pages.

$$MAPE = \frac{1}{n}\sum \left| \frac{y_{\text{actual}} - y_{\text{predicted}}}{y_{\text{actual}}} \right| \times 100$$

3.4.3 RMSE (Root Mean Square Error): RMSE measures the square root of the average squared differences between predicted and actual values. Unlike MAE, it penalizes larger errors, making it sensitive to outliers or sudden traffic spikes. This makes RMSE useful for assessing model stability and reliability, highlighting extreme deviations that could impact operational planning.

$$RMSE = \sqrt{\frac{1}{n}\sum (y_{\text{actual}} - y_{\text{predicted}})^2}$$

5. RESULT AND ANALYSIS

Figure 2. Admin Login Interface

The admin login page acts as a secure gateway to the web traffic forecasting system, ensuring that only authorized users can access sensitive data and forecasting features. It uses username and password authentication to prevent unauthorized access and protect system integrity. The interface is designed to be simple and user-friendly, allowing administrators to efficiently train models, view forecasts, and export results after login. This controlled access ensures data confidentiality, accountability, and stable system operation through effective role-based access control.

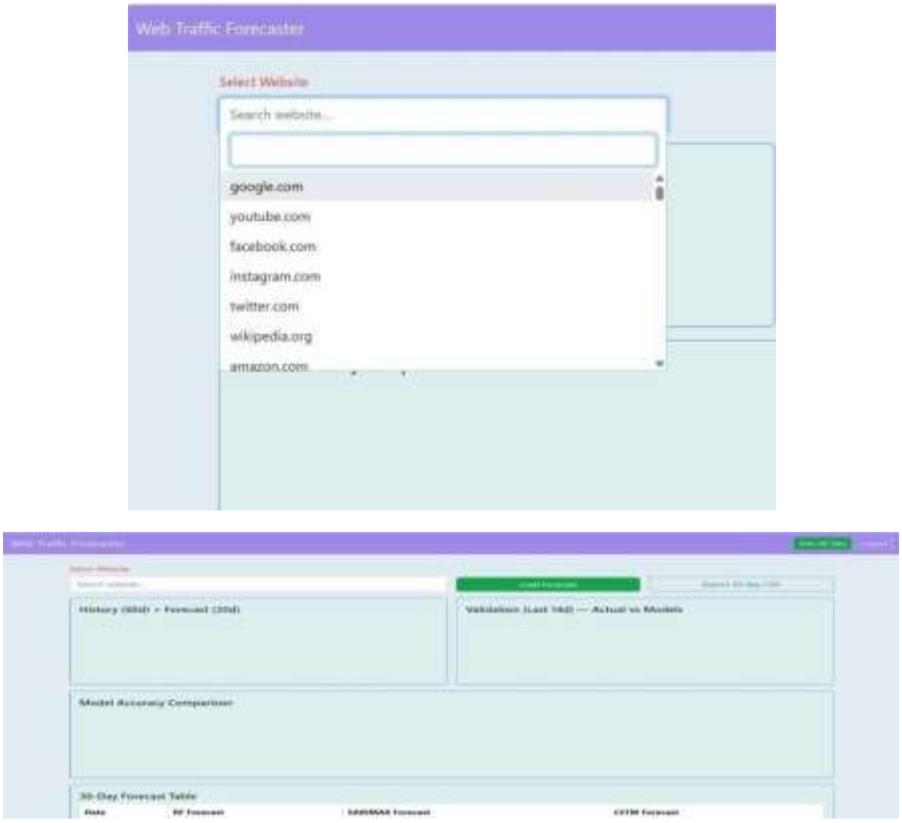

Figure 3. Web Traffic Forecasting System Dashboard Interface

The Select Website interface allows users to choose a specific website domain for which traffic forecasting is required. A search-enabled dropdown makes it easy to locate websites from the dataset. Once a website is selected, the user can trigger the forecasting process using the “Load Forecast” button, which retrieves historical data and displays predictive results. This interface connects the user directly to the backend forecasting engine. Overall, this screen acts as the primary interaction point between the user and the forecasting system. It ensures smooth integration between data selection, model execution, and visualization, making the system intuitive even for non-technical users. The dropdown menu reduces manual input errors by allowing users to select only valid website domains available in the dataset. It also saves time by providing quick search and selection, making the forecasting process faster and more user-friendly for all users.

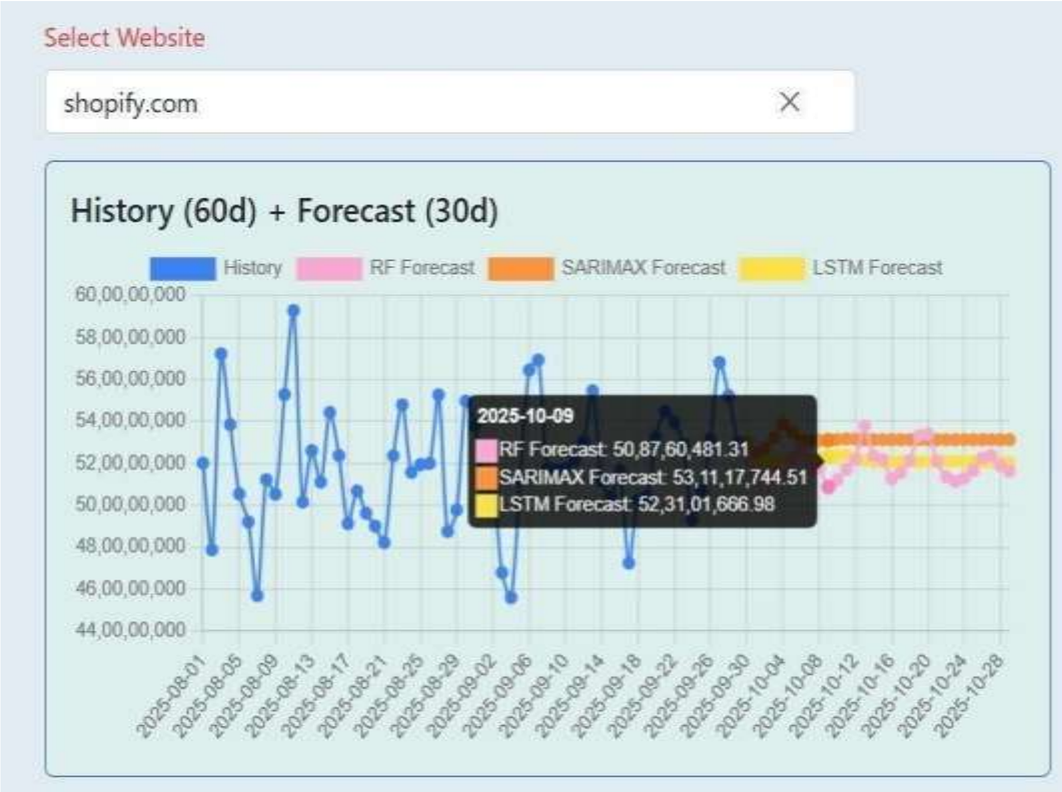

Figure 4. Traffic Forecasting Dashboard with Historical Data

Figure 4.3 illustrates the web traffic forecasting results for the selected website (shopify.com) by combining 60 days of historical data with a 30-day future forecast, where the blue line represents past traffic exhibiting daily fluctuations due to dynamic user behavior influenced by weekdays, promotions, and external events. The historical data serves as the basis for learning trends and seasonality, enabling the models to generate future predictions. The forecasted section compares outputs from Random Forest, SARIMAX, and LSTM models, each producing distinct estimates that reflect their respective learning characteristics, with SARIMAX providing smoother seasonal forecasts, Random Forest capturing short-term variations, and LSTM balancing both long-term dependencies and trends. This visualization supports effective capacity planning by allowing system administrators to anticipate future traffic, allocate resources proactively, and prevent performance bottlenecks, while the side-by-side model comparison helps identify the most reliable forecasting approach, making the system practical and valuable for real-world deployment.

Figure 5. Actual vs Model Comparison graph

International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 09 Issue: 12 | Dec - 2025 SJIF Rating: 8.586 ISSN: 2582-3930

Figure 4.4 compares actual website traffic with one-step-ahead forecasts from Random Forest, SARIMAX, and LSTM models over a 14-day period. The actual traffic shows significant fluctuations due to real-world factors, which are followed reasonably well by the Random Forest model, though with minor errors during sudden changes. SARIMAX produces smoother, more stable forecasts that capture overall trends but respond slowly to abrupt variations. LSTM achieves a balanced performance by closely tracking traffic trends while smoothing extreme spikes, making it well suited for dynamic web traffic forecasting.

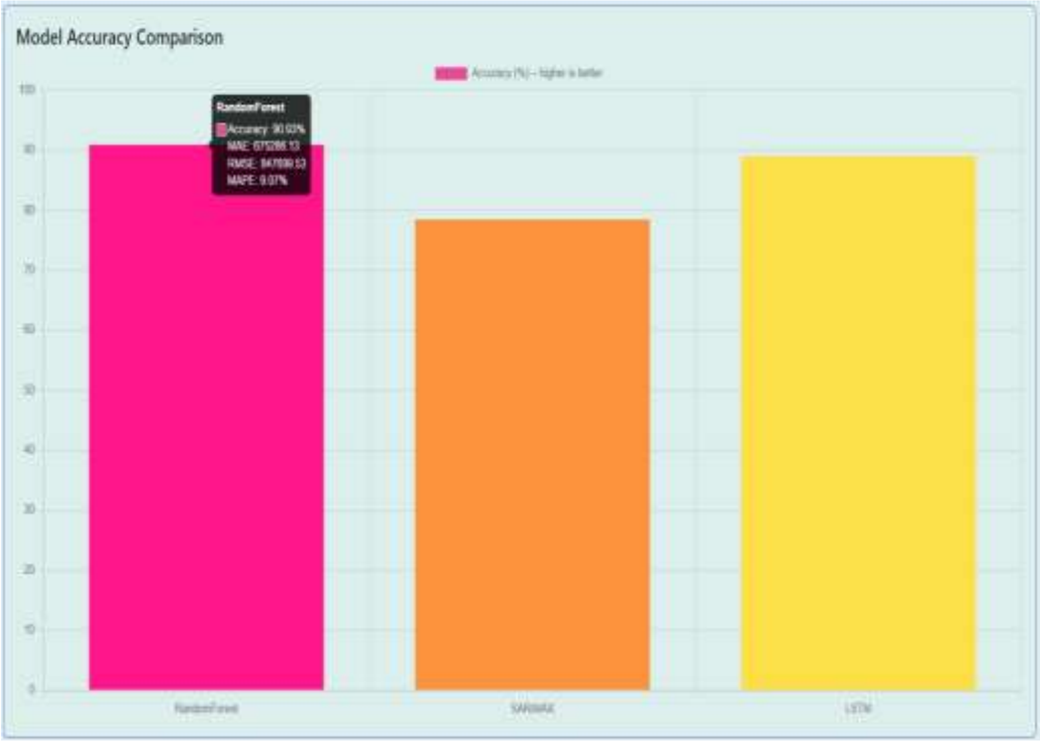

Figure 6. Model Accuracy Comparison

The Model Accuracy Comparison bar chart visually compares the performance of Random Forest, SARIMAX, and LSTM models. Accuracy values are plotted on the y-axis, while the different models are shown on the x-axis. This visualization clearly highlights which model performs best for the given dataset.

From the graph, it is evident that LSTM and Random Forest achieve higher accuracy compared to SARIMAX. This indicates that machine learning and deep learning models are better at capturing complex and non-linear traffic patterns than traditional statistical models. The chart simplifies performance evaluation by presenting results in an easily interpretable format.

This comparison is crucial for model selection, as it helps determine which algorithm should be prioritized for future forecasting tasks. Visual analysis reduces dependency on raw numerical metrics and supports faster decision-making.

30-Day Forecast Table

Show 10 entries Search:

Date	RF Forecast	SARIMAX Forecast	LSTM Forecast
2025-09-30	522198138.63	530138065.65	522898912.07
2025-10-01	514485363.83	525029910.34	521721773.14
2025-10-02	501112218.18	527179959.82	522457871.57
2025-10-03	515566731.91	532238375.95	523052255.37
2025-10-04	514089117.01	538883500.57	523690562.86
2025-10-05	522074890.87	535056277.12	523782996.45
2025-10-06	526947281.5	530653694.13	523546971.32
2025-10-07	517954234.49	531089263.61	523334501.37
2025-10-08	517701536.29	530977926.33	523296498.09
2025-10-09	508760481.31	531117744.51	523951893.6

Showing 1 to 10 of 30 entries Previous 1 2 3 Next

Figure 7. 30-Day Forecast Table

Figure 8. Export 30 days web traffic data

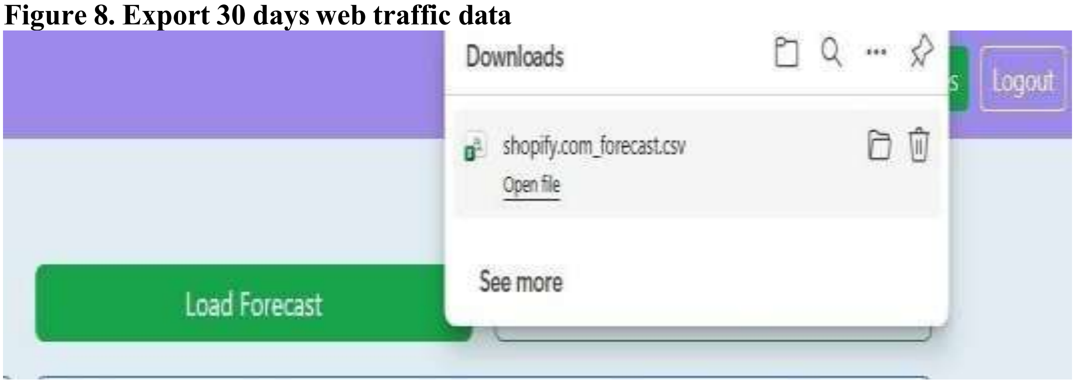

Figure 9. Downloaded forecast CSV file

The 30-Day Forecast Table presents the predicted website traffic values generated by different forecasting models for future dates. Each row corresponds to a specific calendar date, while the columns display the forecasted visitor counts produced by the Random Forest and SARIMAX models. This tabular representation allows users to clearly compare how different models estimate future traffic on the same day. The table supports pagination and sorting, making it easy to navigate through all 30 forecasted days. The numerical values in the table represent expected daily website visits based on historical patterns learned by the models. Small variations between Random Forest and SARIMAX forecasts highlight differences in how each algorithm interprets trends and fluctuations in the data. Random Forest often captures non-linear behavior, while SARIMAX produces smoother forecasts based on seasonal and trend components. This comparison helps in identifying stable versus aggressive prediction patterns.

Additionally, the table plays an important role in decision-making, as it allows administrators to export the data in CSV format for further analysis or reporting. The structured layout ensures transparency and usability, enabling organizations to plan server capacity, marketing campaigns, and maintenance schedules using reliable numerical forecasts. The ability to export the table in CSV format further extends its usefulness beyond the forecasting dashboard. Once exported, the data can be archived for future reference, compared against actual traffic once the forecast period passes, and used to evaluate long-term model performance. Over time, organizations can build a historical record of forecasts versus actual outcomes, enabling continuous improvement of forecasting strategies and model selection. This makes the table not just a reporting tool but also a foundation for performance evaluation.

30-Day Forecast Table

Show 10 entries Search: 2025-10-05

Date	RF Forecast	SARIMAX Forecast	LSTM Forecast
2025-10-05	522074890.87	535056277.12	522366225.47

Showing 1 to 1 of 1 entries (filtered from 30 total entries) Previous 1 Next

Figure 10. Search bar filtering the 30-day forecast table to show traffic predictions for a specific date.

In academic and analytical contexts, the 30-Day Forecast Table also serves as concrete evidence of system functionality. It demonstrates that the forecasting pipeline successfully converts historical data into structured, actionable outputs. The presence of clear dates, numerical values, and model-wise predictions validates the correctness of data preprocessing, model execution, and result integration. This makes the table an important artifact for project evaluation, documentation, and demonstration. The table's structured layout also ensures scalability. While the current implementation focuses on a 30-day horizon, the same format can easily support longer forecasting periods such as 60 or 90 days. Additional models, such as LSTM or hybrid approaches, can be incorporated as new columns without altering the overall design. This flexibility ensures that the forecasting system can evolve over time without disrupting user interaction or data interpretation.

Overall, the extended role of the 30-Day Forecast Table lies in transforming forecasting outputs into practical intelligence. It supports detailed planning, encourages informed comparison between models, enhances transparency, and facilitates collaboration across organizational roles. By presenting future traffic estimates in a clear, structured, and exportable form, the table ensures that forecasting results are not merely theoretical predictions but actionable insights that directly support website stability, performance optimization, and strategic decision-making.

7. CONCLUSION

7.1 Conclusion

The Web-Based Web Traffic Forecasting System demonstrates that a lightweight web application can effectively analyze and visualize website traffic data using CSV files. By combining Flask for backend processing, Jinja templates for data rendering, and interactive JavaScript charts for visualization, the system presents key web metrics such as visits, page views, clicks, impressions, bounce rate, and session duration in a clear and user-friendly manner. The structured data flow ensures accuracy and ease of interpretation without unnecessary complexity. In the context of forecasting, the SARIMAX model provides a reliable statistical foundation that complements modern learning-based approaches. Web traffic data often contains trends and seasonal patterns driven by user behavior, and SARIMAX is well suited to capture these characteristics through explicit modeling of trend, autoregressive, moving average, and seasonal components. This structured and interpretable approach enables stable short- to medium-term forecasts and helps distinguish regular seasonal fluctuations from random noise. Due to its transparency and predictable behavior, it serves as a valuable benchmark for validating and comparing the results of more complex machine learning and deep learning models.

7.2 Future Scope

In the future, this web traffic forecasting system can be enhanced by using finer- grained data such as hourly or minute-level traffic instead of only daily counts, enabling more accurate and real-time predictions. Advanced deep learning models like CNN- LSTM and Transformer-based architectures can be integrated to capture complex temporal patterns and improve long-term forecasting performance. The system can also be deployed on cloud platforms to support intelligent server auto-scaling based on predicted traffic loads, helping websites manage resources efficiently. Additionally, the dashboard can be extended to handle multiple websites simultaneously, making it suitable for organizations managing large web infrastructures. Incorporating external factors such as marketing campaigns, events, and trend data can further improve prediction reliability, transforming the system into a powerful decision-support tool for proactive website performance management.

8. REFERENCES

[1] Shelatkar, Tejas, Stephen Tondale, Swaraj Yadav, and Sheetal Ahir. "Web traffic time series forecasting using ARIMA and LSTM RNN." In ITM Web of Conferences, vol. 32, p. 03017. EDP Sciences, 2020.

[2] J. Zheng and M. Huang, "Traffic Flow Forecast Through Time Series Analysis Based on Deep Learning," in IEEE Access, vol. 8, pp. 82562- 82570, 2020, doi: 10.1109/ACCESS.2020.2990738.

[3] Casado-Vara, Roberto, Angel Martin del Rey, Daniel Pérez-Palau, Luis de- laFuenteValentín, and Juan M. Corchado. "Web traffic time series forecasting using LSTM neural networks with distributed asynchronous training." Mathematics 9, no. 4 (2021): 421.

[4] Wan, Xianbin, Hui Liu, Hao Xu, and Xinchang Zhang. "Network Traffic Prediction Based on LSTM and Transfer Learning." IEEE Access 10 (2022): 86181-86190.

[5] Nihale, Shyam, Shantanu Sharma, Lokesh Parashar, and Upendra Singh. "Network traffic prediction using long short-term memory." In 2020 International Conference on Electronics and Sustainable Communication Systems (ICESC), pp. 338-343. IEEE, 2020.

[6] Jianhu Zheng and Mingfang Huang, "Traffic Flow Forecasting Using Deep Learning and Time Series Analysis," IEEE Access, 2020. P Montero-Manso.

[7] Mohammad Asifur Rahman Shuvo, Muhtadi Zubair, Afsara Tahsin Purnota, Sarowar Hossain, and Muhammad Iqbal Hossain, "Traffic Forecasting Using Time- Series Analysis," 6th International Conference on Inventive Computation Technologies, 2021. (ICICT).

[8] Wunnava, Venkata Praveen. "Exploration of Wikipedia traffic data to analyze the relationship between multiple pages." (2020).

[9] Zhou, K.; Wang, W.; Huang, L.; Liu, B. Comparative study on the time series forecasting of web traffic based on statistical model and Generative Adversarial model. Knowl.-Based Syst. 2020, 213, 106467.

[10] Wunnava, Venkata Praveen. "Exploration of Wikipedia traffic data to analyze the relationship between multiple pages." (2020).

[11] Miyaguchi, A.Chakrabarti, S.; Garcia, N. Forecasting Wikipedia Page Views with GraphEmbeddings.2019.http://cs229.stanford.edu/proj2019aut/data/assignment_3 08832_raw/26647399.pdf (accessed on 30 Nov 2020).

[12] Hyndman, R.J., and Athanasopoulos, G., “Forecasting principles and practice using ARIMA and SARIMAX models”, In proceedings of *IEEE International Conference on Data Analytics*, 2018,pp. 45 – 52.

[13] Box, G.E.P., Jenkins, G.M., and Reinsel, G.C.,“Time series analysis: forecasting and control with seasonal models”,In proceedings of *International Conference on Statistical Computing*, 2015,pp. 210 – 218.

[14] Breiman, L., “Random Forests for regression and time series prediction”, In proceedings of *International Conference on Machine Learning*, 2001, pp. 5 – 32.

[15] Kumar, A., Singh, Y., and Kaur, S.,“Web traffic forecasting using Random Forest and machine learning techniques”, In proceedings of *International Conference on Advances in Computing and Communication Engineering*, 2019,pp. 389 – 394.