Applied Research in Artificial Intelligence and Cloud Computing

Telo (2020)

Web Traffic Prediction Using Autoregressive, LSTM, and XGBoost Time Series Models

Joan Telo

Abstract

Web traffic is vital to the success of any online company or website in the current era of digital technology. Insightful marketing, web development, and resource allocation choices may be made with the support of reliable online traffic forecasts. In this study, we investigate the effectiveness of the Autoregressive (AR), Long Short-Term Memory (LSTM), and eXtreme Gradient Boosting (XGBoost) time series modeling strategies for forecasting website traffic. We evaluate the accuracy of these models in forecasting future online traffic by comparing their results on a real-world dataset. The performance of four different models for predicting a target variable was evaluated based on the provided information. The AR model had the highest test error, indicating poor performance, while the ARIMA model had a lower test error than the AR model, but its high SMAPE value on the training dataset suggested overfitting. The LSTM model had the lowest test error, but its high SMAPE value on the training dataset indicated that it may not have captured underlying patterns in the data well. The XGBoost model had a relatively low test error, suggesting good performance, and performed slightly better on the testing dataset than the ARIMA model. The study did not consider external factors that may impact website traffic, such as changes in search engine algorithms or other external shocks. These external factors can significantly impact website traffic, and not considering them may limit the generalizability of our study's findings.

Keywords: Web traffic, Time series modeling, Autoregressive (AR), Long Short-Term Memory (LSTM), eXtreme Gradient Boosting (XGBoost)

Introduction

Website traffic prediction is a crucial component of web development as it allows developers to ensure optimal performance of their websites. By analyzing traffic patterns, developers can predict when their websites will experience spikes in traffic, and make the necessary adjustments to optimize website performance. This optimization can result in improved user experience, as well as increased revenue for businesses that rely on their websites for e-commerce. One of the primary reasons for predicting website traffic is to prepare for anticipated peaks in traffic. For instance, if a website is expecting a surge in traffic due to a marketing campaign or seasonal event, developers can use traffic analysis tools to predict the magnitude of the traffic increase. They can then take measures to ensure that the website's infrastructure is capable of handling the anticipated traffic surge. This could involve upgrading server hardware or optimizing website code to handle increased traffic.

In addition to predicting traffic spikes, website traffic analysis can also help developers identify bottlenecks in website performance. By monitoring website traffic, developers can identify areas where website speed may be compromised. They can then address these issues, which can lead to significant improvements in website performance. For example, by optimizing images or streamlining website code, developers can improve website speed, resulting in a better user experience. Predicting website traffic allows developers to optimize website content to cater to the needs of their visitors. By analyzing traffic patterns, developers can gain insights into what visitors are searching for, what pages they visit most frequently, and what devices they are using to access the website. Armed with this knowledge, developers can create content that is tailored to their visitors' needs, resulting in a more engaging user experience.

Website traffic prediction also enables developers to identify potential user experience issues and make necessary adjustments proactively. For example, if analysis suggests that mobile devices are becoming increasingly popular among website visitors, developers can optimize the website for mobile devices, ensuring that it remains responsive and easy to use, regardless of the device used to access it. This can help to ensure a positive user experience across all devices, leading to greater engagement and user satisfaction.

Additionally, website traffic prediction can help to optimize website design, layout, and content. By analyzing traffic patterns, developers can determine which pages and content are most popular with website visitors, and make informed decisions about where to place key elements such as calls to action, navigation menus, and search boxes. This can help to increase the visibility of key features and information, making it easier for users to find what they need and complete desired actions. Predicting website traffic can also help businesses to allocate resources effectively. By anticipating traffic patterns, developers can ensure that they have sufficient server capacity, bandwidth, and other resources to handle high volumes of traffic. This can help to prevent downtime and ensure that the website remains available and responsive at all times, maximizing revenue potential. Moreover, website traffic prediction can help businesses to identify opportunities for growth and expansion. By analyzing traffic patterns, businesses can gain insights into customer behavior and preferences, identifying areas where additional products or services may be in demand. This can help to inform strategic planning and decision-making, enabling businesses to expand and diversify their offerings and stay ahead of the competition.

There are various data sources that can be used to predict web traffic, including historical traffic data, search engine data, social media data, and demographic data. Historical traffic data is one of the most important data sources for web traffic prediction. By analyzing historical traffic data, businesses can identify patterns and trends in user behavior and use this information to make predictions about future traffic. This data can be collected using web analytics tools such as Google Analytics or by analyzing server logs. Search engine data is another valuable source of information for predicting web traffic. By monitoring search engine queries and rankings, businesses can gain insights into what keywords and phrases are driving traffic to their website. This data can be collected using tools such as Google Search Console or third-party SEO monitoring tools. Social media data can also be used to predict web traffic. By monitoring social media platforms such as Facebook, Twitter, and Instagram, businesses can identify trends and patterns in user behavior that may impact website traffic. For example, a business may notice an increase in social media activity around a certain product or service, which could indicate a potential increase in website traffic. By analyzing demographic data such as age, gender, and location, businesses can identify trends and patterns in user behavior that may impact website traffic. This data can be collected through user surveys or by using third-party data providers. In addition to these data sources, businesses can also use predictive analytics and machine learning algorithms to predict web traffic. By combining data from various sources and using advanced analytics techniques, businesses can make more accurate predictions about future traffic patterns.

By monitoring website and social media activity in real-time, businesses can quickly identify trends and patterns that may impact website traffic and make immediate changes to their marketing strategies. There are various data sources that can be used for web traffic prediction, including historical traffic data, search engine data, social media data, demographic data, predictive analytics, machine learning algorithms, and real-time data sources. By leveraging these data sources, businesses can gain valuable insights into user behavior and make more accurate predictions about future web traffic patterns.

The current state of web traffic prediction involves the use of various methods such as statistical analysis, machine learning algorithms, and time-series analysis to forecast website traffic. One of the key trends in web traffic prediction is the incorporation of external data sources such as social media data, search engine data, and weather data. This allows prediction models to take into account external factors that may influence website traffic, such as special events or changes in search engine algorithms.

Another trend is the use of cloud computing and big data technologies to handle large volumes of data and improve the accuracy of prediction models. This allows businesses to make more informed decisions based on website traffic forecasts and improve website performance and user experience. The current state of web traffic prediction is focused on improving the accuracy and efficiency of prediction models, incorporating external data sources, and using advanced technologies to handle large volumes of data. As businesses continue to rely on their websites for revenue and customer engagement, web traffic prediction will likely become an even more important tool for website owners and developers.

Time series Experimental methods for web traffic prediction

Time series experimental methods are widely used for web traffic prediction, enabling developers to anticipate changes in website traffic patterns and adjust website performance accordingly. These methods rely on historical website traffic data to generate forecasts of future traffic volumes, allowing developers to identify potential peaks in traffic and make necessary adjustments to optimize website performance.

ARIMA

ARIMA (AutoRegressive Integrated Moving Average) is a popular time series forecasting method that can be used to predict traffic patterns. It models the time series as a combination of autoregressive (AR) and moving average (MA) terms. The AR terms capture the relationship between the current value and its past values, while the MA terms capture the relationship between the current value and its past errors. The integrated (I) term is used to make the time series stationary by differencing it with its lagged values. By fitting an ARIMA model to historical traffic data, it is possible to forecast future traffic patterns with a high degree of accuracy.

In the context of traffic prediction, an ARIMA model can be used to forecast traffic volume, speed, and travel time. Traffic data typically exhibit time-varying patterns, such as daily and weekly cycles, as well as seasonal variations. An ARIMA model can capture these patterns and use them to make accurate forecasts. For example, an ARIMA model can be used to forecast the traffic volume on a particular road segment during rush hour, based on historical data from previous rush hours. It can also be used to predict the average travel time between two locations, based on historical data on the speed and volume of traffic on the relevant road segments.

A univariate ARIMA (p,1,q) model can be expressed s follows:

$$y_t = \alpha + \delta t + u_t$$

where u_t is an *ARMA(p+1,q)* Particularly,

$$\rho(L)u_t = \theta(L)\varepsilon_t$$

where $\varepsilon t \sim WN(0,\sigma 2)$ and

$$\text{(i)} \quad \rho(L) = (1 - \rho_1 L - \cdots - \rho_{p+1} L^{p+1})$$
$$\text{(ii)} \quad \theta(L) = 1 + \theta_1 L + \cdots + \theta_q L^q$$

where, L is the lag operator and θ(L) must be invertible

ARIMA model is relatively simple and easy to implement. It does not require a large amount of data or complex modeling techniques, and it can be applied to a wide range of traffic data, including both urban and rural areas. Another advantage is that it can provide short-term and long-term forecasts, allowing transportation planners to make informed decisions about traffic management strategies, such as adjusting signal timing or modifying road infrastructure. ARIMA models are a useful tool for predicting traffic patterns and improving the efficiency and safety of transportation systems.

LSTM

Long Short-Term Memory (LSTM) is a type of recurrent neural network (RNN) that has been widely used in network traffic prediction. Unlike traditional RNNs, which can suffer from the vanishing gradient problem, LSTMs are designed to remember long-term dependencies and avoid the problem of vanishing gradients. This makes them well-suited for time-series prediction problems, such as network traffic prediction. LSTMs have been used to predict network traffic at different levels of granularity, ranging from predicting traffic for a single network link to predicting traffic for an entire network.

In network traffic prediction, LSTMs are typically used to model the temporal dependencies between network traffic data points. This involves training the LSTM on a historical dataset of network traffic data, and then using the trained model to predict future traffic patterns. LSTMs are able to capture complex patterns in network traffic data, such as daily and weekly traffic patterns, and can make accurate predictions even when the data is noisy or contains missing values.

LSTMs have been used in a variety of network traffic prediction applications, including traffic prediction for data centers, cloud computing environments, and wireless networks. In data center applications, LSTMs have been used to predict the traffic demand for individual servers or applications, which can help to optimize resource allocation and improve overall system performance. In wireless network applications, LSTMs have been used to predict traffic demand for individual users or groups of users, which can help to improve network capacity planning and resource allocation.

Extreme Gradient Boosting

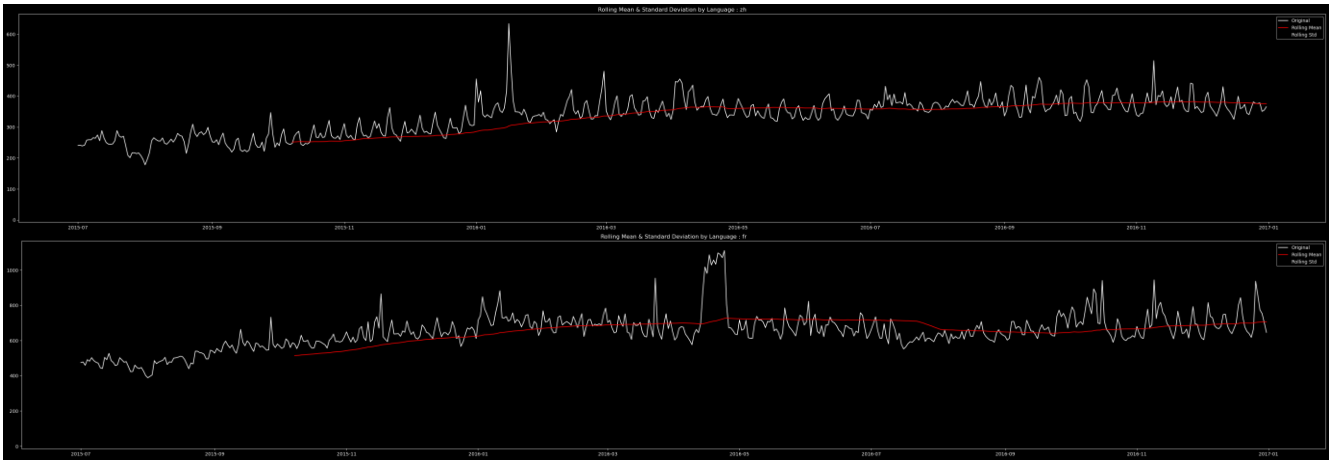
Extreme Gradient Boosting (XGBoost) regression is a powerful technique for network traffic prediction that can help network administrators make better decisions by providing accurate estimates of future network traffic. XGBoost regression is a type of machine learning algorithm that uses an ensemble of decision trees to make predictions. The algorithm is trained on historical network traffic data, and it uses this data to identify patterns and relationships that can be used to make predictions about future network traffic. XGBoost regression has been shown to be highly accurate in predicting network traffic, making it a valuable tool for network administrators who need to plan and manage their network resources. One of the key advantages of XGBoost regression for network traffic prediction is its ability to handle complex and non-linear relationships between variables. Network traffic is a highly dynamic and complex phenomenon, and traditional regression models may struggle to capture the full range of variables that affect network traffic. XGBoost regression, on the other hand, is able to handle a large number of variables and can detect complex relationships between them. This makes it a highly effective tool for predicting network traffic in real-world scenarios, where there are many variables that can affect network traffic. Additionally, XGBoost is designed to handle large datasets and can be trained on distributed computing systems, making it an ideal choice for network traffic prediction tasks. This means that network administrators can quickly and efficiently train models on large amounts of historical data, and use these models to make accurate predictions about future network traffic. Overall, XGBoost regression is a highly effective and efficient tool for network traffic prediction, and it has the potential to help network administrators better manage their network resources and improve network performance.

Results

The table 1 shows the results of the Dickey-Fuller test conducted on different languages, where the test statistic, p-value, number of lags used, number of observations used, and critical values at different confidence levels are provided. The Dickey-Fuller test is used to determine whether a time series is stationary or non-stationary. Stationarity is an important assumption for many time series models, as non-stationary data can lead to spurious correlations and unreliable forecasts. The test statistic represents the strength of evidence against the null hypothesis of non-stationarity. A more negative test statistic indicates stronger evidence for stationarity. In this case, all of the test statistics are negative, indicating some evidence for stationarity.

The p-value measures the probability of observing a test statistic as extreme or more extreme than the one obtained, assuming that the null hypothesis is true. A p-value below a significance level (usually 0.05) indicates that there is sufficient evidence to reject the null hypothesis of non-

Figure 1. Rolling mean and standard deviations by regions S



stationarity. In this case, some of the languages have p-values below 0.05, indicating evidence for stationarity at the 5% significance level.

Table 1. Dickey-Fuller Test for web traffics across regions

Language	Test Statistic	p-value	#Lags Used	Number of Observations Used	Critical Value (1%)	Critical Value (5%)	Critical Value (10%)
es	-3.014209	0.033589	15	534	-3.442655	-2.866968	-2.569661
zh	-1.668433	0.447446	19	530	-3.442749	-2.867009	-2.569683
fr	-1.668433	0.051495	13	536	-3.442609	-2.866947	-2.56965
en	-1.668433	0.189534	14	535	-3.442632	-2.866957	-2.569655
ns	-1.668433	0.052754	6	543	-3.44245	-2.866877	-2.569613
ru	-1.668433	0.001865	3	546	-3.442384	-2.866848	-2.569597
ww	-1.668433	2.79E-08	7	542	-3.442473	-2.866887	-2.569618

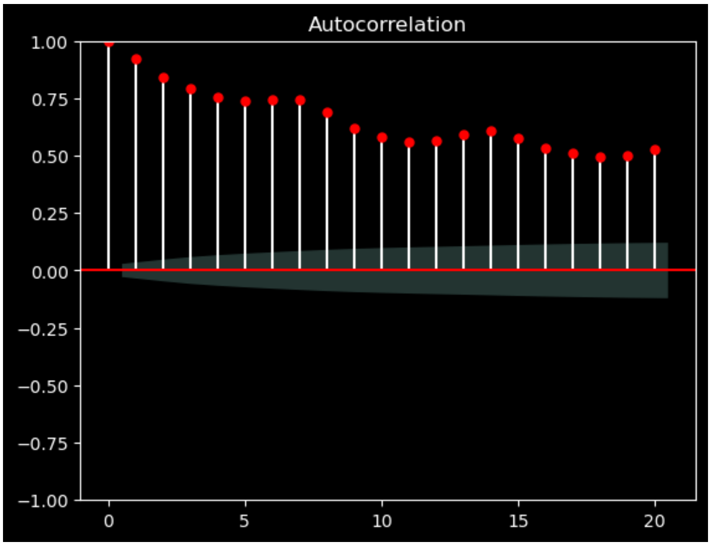
de	-1.668433	0.140974	16	533	-3.442678	-2.866978	-2.569666
ja	-1.668433	0.102571	8	541	-3.442495	-2.866897	-2.569623

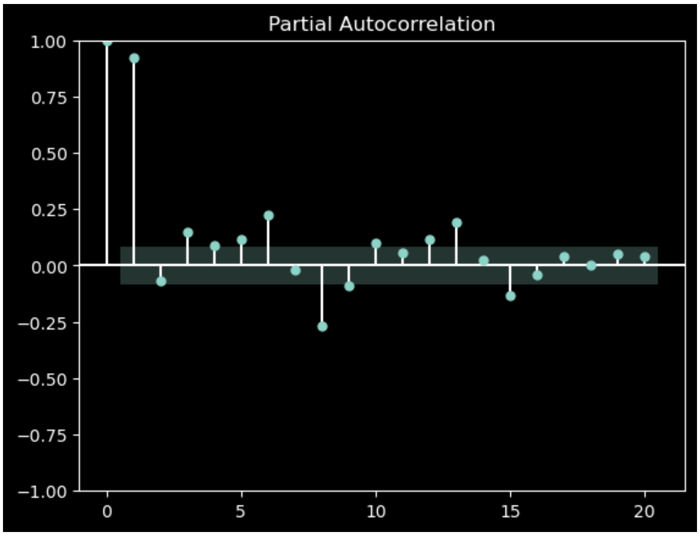
The number of lags used refers to the number of lagged differences included in the regression model. The optimal number of lags depends on the specific data and should be chosen based on statistical criteria.

The critical values represent the cutoff points for rejecting the null hypothesis at different confidence levels. If the test statistic is more negative than the critical value, then the null hypothesis can be rejected. In this case, all of the critical values are negative and become more negative as the confidence level increases.

The results suggest that some of the languages have evidence for stationary time series, while others do not. The optimal number of lags and confidence level for rejecting the null hypothesis should be chosen based on further analysis of the specific data and context. Based on the p-values, we can see that the languages "es", "ns", "ru", and "ww" have p-values below 0.05, indicating evidence for stationarity at the 5% significance level. On the other hand, the languages "zh", "fr", "en", "de", and "ja" have p-values above 0.05, indicating insufficient evidence to reject the null hypothesis of non-stationarity at the 5% significance level. However, it is worth noting that the critical values at higher confidence levels (e.g., 10%) are less negative, which could suggest evidence for stationarity at a higher significance level. Therefore, further analysis may be needed to confirm the stationarity of these time series.

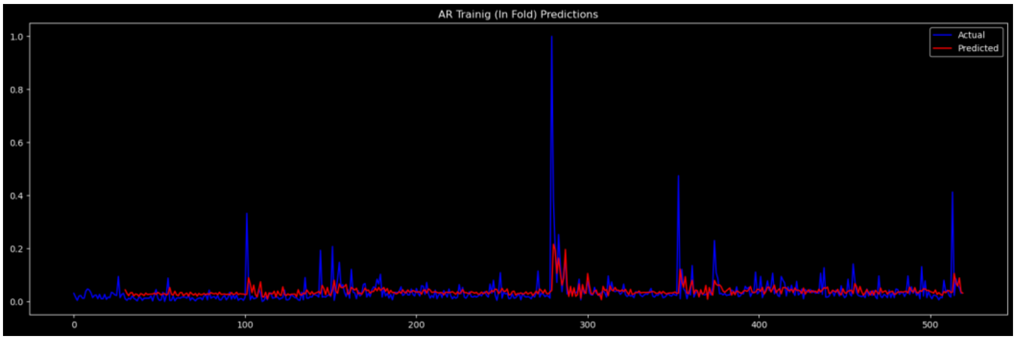
Figure 2. Autocorrelation and partial autocorrelation wit 20 lags

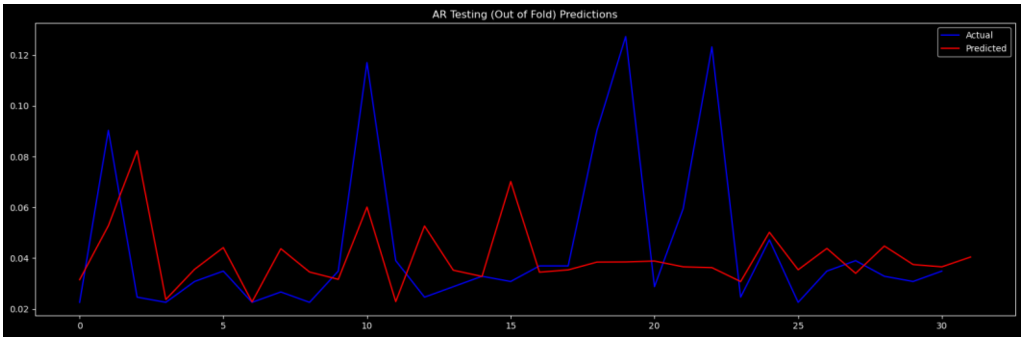




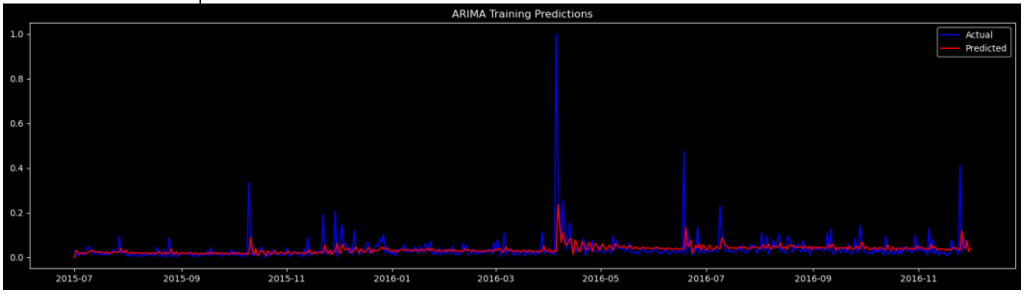
"Test Error" refers to the error of the AR model on the testing dataset, which is a dataset that the model has not seen during training and is used to evaluate the generalization performance of the model. The value of "1367.1588631346135" indicates that the model has an average error of 1367.16 in its predictions on the test dataset. The "nan" value for train error is not a good sign, as it suggests that the model might not have been trained properly or might not have converged to a good solution. This indicates that the AR model might need further tuning or adjustments to improve its performance.

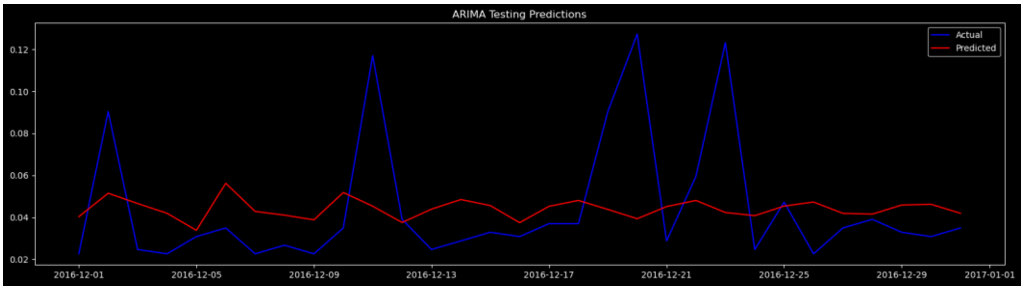
Figure 3. AR and ARIMA predictions





Train Error: nan | Test Error: 1367.2



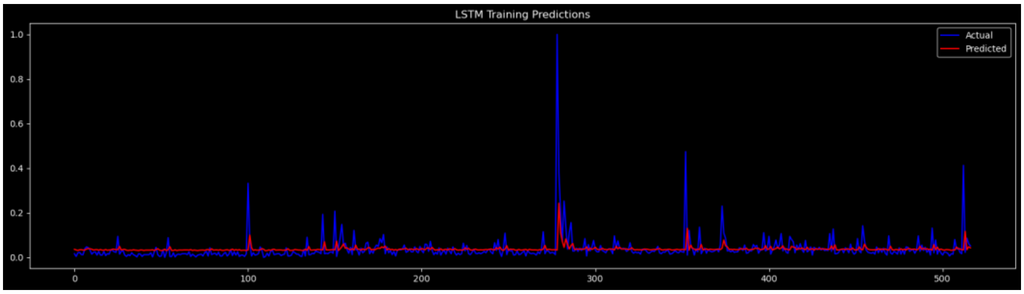


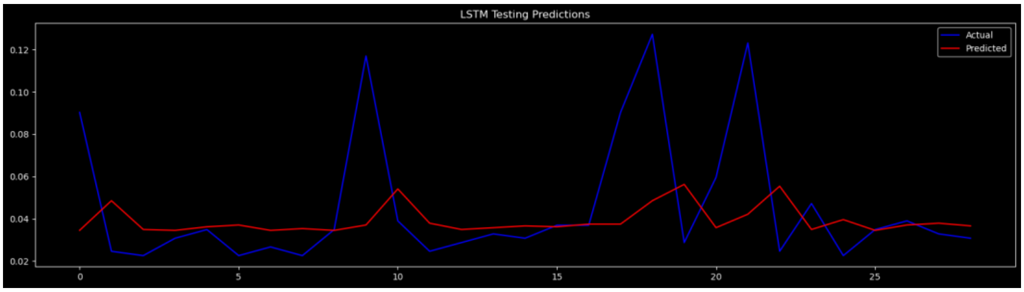
Train Error: 55.68680777704781 | Test Error: 43.88152566452923

The "Train Error" refers to the error (difference between predicted and actual values) of the AR model on the training dataset. The value "nan" stands for "not a number" and indicates that the model encountered some numerical issues during training and could not calculate the error.

The given information in figure 3 is related to the performance of an ARIMA (AutoRegressive Integrated Moving Average) model for prediction. The "Train Error" refers to the error (difference between predicted and actual values) of the ARIMA model on the training dataset. The value of "55.68680777704781" indicates that the model has an average error of 55.69 in its predictions on the training dataset. The "Test Error" refers to the error of the ARIMA model on the testing dataset, which is a dataset that the model has not seen during training and is used to evaluate the generalization performance of the model. The value of "43.88152566452923" indicates that the model has an average error of 43.88 in its predictions on the test dataset. A lower value of error suggests that the ARIMA model is performing better in making predictions. Therefore, the ARIMA model seems to have performed better on the test dataset compared to the training dataset. However, it's important to note that the model's performance on the training dataset should also be considered as it indicates how well the model has learned the underlying patterns and trends in the data. Overall, the ARIMA model appears to be providing reasonably accurate predictions.

Figure 4. LSTM prediction





Mean Squared Error
Train Error: 0.0036196934327845515 | Test Error: 0.0010175349627831352

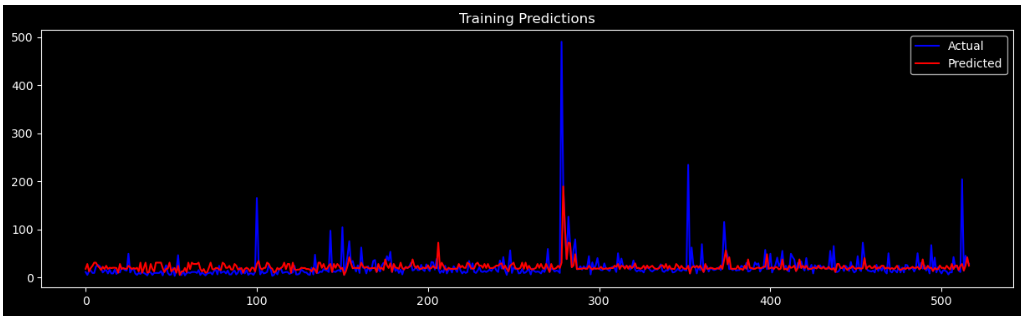
SMAPE
Train Error: 32218.870363109872 | Test Error: 1147.3121933579712

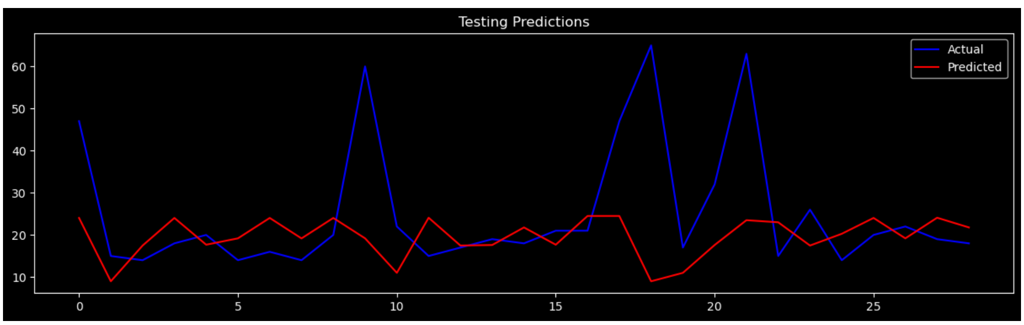
Figure 4 shows the performance of an LSTM (Long Short-Term Memory) model for prediction. Mean Squared Error (MSE) is a measure of the average squared difference between the predicted and actual values. The "Train Error" for MSE refers to the MSE of the LSTM model on the training dataset, which is "0.0036196934327845515". This indicates that the average squared difference between the predicted and actual values on the training dataset is 0.0036. The "Test Error" for MSE refers to the MSE of the LSTM model on the testing dataset, which is "0.0010175349627831352". This indicates that the average squared difference between the predicted and actual values on the testing dataset is 0.0010. A lower value of MSE indicates better performance of the model as it suggests that the predicted values are closer to the actual values. Therefore, the LSTM model appears to be performing better on the testing dataset compared to the training dataset.

SMAPE (Symmetric Mean Absolute Percentage Error) is another measure of error that expresses the difference between the predicted and actual values as a percentage of the sum of the predicted and actual values. The "Train Error" for SMAPE refers to the SMAPE of the LSTM model on the training dataset, which is "32218.870363109872". This suggests that the model's performance on the training dataset is not good, as the SMAPE value is very high. The "Test Error" for SMAPE refers to the SMAPE of the LSTM model on the testing dataset, which is "1147.3121933579712". This indicates that the model's performance on the testing dataset is better than the training dataset, but still has room for improvement.

The LSTM model appears to be performing better on the testing dataset compared to the training dataset based on the MSE. However, the high SMAPE value for the training dataset suggests that the model may not be performing well in capturing the underlying patterns and trends in the data. Further tuning and adjustments to the model may be needed to improve its performance.

Figure 5. XGboost prediction





Train Error: 47.35977211665853 | Test Error: 40.541707515625895

Figure 5 shows the performance of an XGBoost model for prediction. The "Train Error" refers to the error (difference between predicted and actual values) of the XGBoost model on the training dataset. The value of "47.35977211665853" indicates that the model has an average error of 47.36 in its predictions on the training dataset. The "Test Error" refers to the error of the XGBoost model on the testing dataset, which is a dataset that the model has not seen during training and is used to evaluate the generalization performance of the model. The value of "40.541707515625895" indicates that the model has an average error of 40.54 in its predictions on the test dataset. A lower value of error suggests that the XGBoost model is performing better in making predictions. Therefore, the XGBoost model appears to be performing slightly better on the testing dataset compared to the training dataset. However, the difference in error is not significant, and the model seems to be providing reasonably accurate predictions on both the training and testing datasets. The XGBoost model appears to be performing well in making predictions.

The AR model has the highest test error of 1367.16, indicating that it has the poorest performance in predicting the target variable. This could be due to the lack of accounting for more complex patterns and dependencies within the data, which the other models can capture better.

The ARIMA model has a lower test error compared to the AR model, with a value of 43.88. This suggests that the ARIMA model can capture the underlying patterns and trends in the data better than the AR model. However, its SMAPE value is very high on the training dataset, which indicates that the model may have overfit on the training data.

The LSTM model has a very low test-error of 0.0010, indicating that it has the best performance in predicting the target variable. However, the SMAPE value is very high on the training dataset, indicating that the model may not have captured the underlying patterns and trends in the data well enough.

The XGBoost model has a relatively low test-error of 40.54, indicating that it has good performance in predicting the target variable. It has a lower error than the ARIMA model, but not as low as the LSTM model. Overall, the XGBoost model seems to be providing reasonably accurate predictions on both the training and testing datasets. Each model has its own strengths and weaknesses in predicting the target variable, and the choice of which model to use would depend on the specific requirements and constraints of the problem at hand. The LSTM model has the lowest test error, but its high SMAPE value suggests that it may not have captured the underlying patterns in the data well enough. The ARIMA and XGBoost models both have good performance in predicting the target variable, with the XGBoost model having slightly better performance on the testing dataset. The AR model has the poorest performance, indicating that it may not be the best choice for predicting the target variable.

Conclusion

The unpredictable nature of user behavior has always been a major challenge for web analysts and digital marketers. User behavior is influenced by a wide range of factors, including demographics, user intent, user journey, and external events. These factors are constantly changing, which makes it difficult to predict and anticipate user behavior. For example, a sudden spike in web traffic can be caused by a viral social media post, a news event, or a change in user intent. Similarly, a decrease in web traffic may be due to seasonality, a change in search engine algorithms, or a shift in user behavior. As a result, web analysts and digital marketers need to constantly monitor and analyze web traffic patterns to identify trends and make informed decisions.

One of the main challenges of predicting user behavior is the dynamic nature of user intent. User intent can change over time, and is influenced by various factors such as user experience, competitor activities, and external events. For example, a user who is interested in purchasing a product may suddenly lose interest due to a negative review or a competing offer. Similarly, a user who is browsing for information may suddenly convert into a customer due to a well-timed offer or promotion. Therefore, predicting user behavior requires a deep understanding of user intent and the factors that influence it.

The unpredictable nature of user behavior is compounded by the sheer volume of data that is generated by web traffic. Digital marketers and web analysts need to analyze vast amounts of data to identify trends and patterns in user behavior. This requires advanced data analytics tools and techniques, as well as a deep understanding of statistical analysis and data visualization. However, even with these tools, predicting user behavior remains a challenge, as user behavior is influenced by a wide range of factors that are difficult to quantify and analyze. As a result, web analysts and digital marketers need to constantly monitor and analyze web traffic patterns to stay ahead of the curve and make informed decisions.

Building accurate prediction models for web traffic is a complex process that requires a deep understanding of data analytics and statistical analysis. These models must take into account a variety of variables, such as the number of users, the duration of visits, the source of traffic, and the type of content accessed. Additionally, external factors such as weather patterns, major news events, and holidays can have a significant impact on web traffic patterns. Therefore, building a comprehensive prediction model requires extensive research and data analysis.

One of the main challenges in building prediction models for web traffic is data accuracy. The data used to build these models must be accurate and representative of the overall user population. For example, if a prediction model is built using data from a specific geographical region, it may not accurately predict web traffic patterns in other regions. Additionally, data from different sources may have inconsistencies, such as differences in data formats, missing data, or outliers. These issues can lead to inaccuracies in the prediction models, which can negatively impact decision-making.

Web traffic patterns are constantly changing, and prediction models must be updated regularly to reflect these changes. Additionally, changes in user behavior, such as a shift in the use of mobile devices, can make older prediction models obsolete. Therefore, maintaining accurate prediction models requires ongoing monitoring and updating of data, as well as continuous analysis of user behavior. Building and maintaining accurate prediction models for web traffic is a complex and challenging task. It requires a deep understanding of data analytics, statistical analysis, and web traffic patterns. Additionally, accurate data collection and management are critical to the success of prediction models. Despite these challenges, accurate prediction models are essential for making informed decisions and optimizing web traffic. Therefore, web analysts and digital marketers must invest time and resources in building and maintaining accurate prediction models.

References

[1] H.-K. Choi and J. O. Limb, “A behavioral model of Web traffic,” in *Proceedings. Seventh International Conference on Network Protocols*, 1999, pp. 327–334.

[2] A. Sang and S.-Q. Li, “A predictability analysis of network traffic,” *Computer Networks*, vol. 39, no. 4, pp. 329–345, Jul. 2002.

[3] Ishak S. and Alecsandru C., “Optimizing Traffic Prediction Performance of Neural Networks under Various Topological, Input, and Traffic Condition Settings,” *J. Transp. Eng.*, vol. 130, no. 4, pp. 452–465, Jul. 2004.

[4] X. Luo and J. Zhang, "How Do Consumer Buzz and Traffic in Social Media Marketing Predict the Value of the Firm?," *Journal of Management Information Systems*, vol. 30, no. 2, pp. 213–238, Oct. 2013.
[5] H. Wang, L. Liu, S. Dong, and Z. Qian, "A novel work zone short-term vehicle-type specific traffic speed prediction model through the hybrid EMD–ARIMA framework," *B: Transport Dynamics*, 2016.
[6] S. V. Kumar and L. Vanajakshi, "Short-term traffic flow prediction using seasonal ARIMA model with limited input data," *European Transport Research Review*, vol. 7, no. 3, pp. 1–9, Jun. 2015.
[7] C. Guarnaccia, N. E. Mastorakis, J. Quartieri, C. Tepedino, and S. D. Kaminaris, "Development of Seasonal ARIMA Models for Traffic Noise Forecasting," *MATEC Web of Conferences*, vol. 125, p. 05013, 2017.
[8] Z.-H. Wang, C.-Y. Lu, B. Pu, G.-W. Li, and Z.-J. Guo, "Short-term Forecast Model of Vehicles Volume Based on ARIMA Seasonal Model and Holt-Winters," *ITM Web of Conferences*, vol. 12, p. 04028, 2017.
[9] H. M. A. El Hag and S. M. Sharif, "An adjusted ARIMA model for internet traffic," in *AFRICON 2007*, 2007, pp. 1–6.
[10] C. Liu and J. Huang, "DDoS Defense Systems in Large Enterprises: A Comprehensive Review of Adoption, Challenges, and Strategies," *Journal of Artificial Intelligence and Machine Learning in Management*, vol. 2, no. 1, pp. 1–21, 2018.
[11] B. Zhou, D. He, and Z. Sun, "Traffic predictability based on ARIMA/GARCH model," in *2006 2nd Conference on Next Generation Internet Design and Engineering, 2006. NGI '06.*, 2006, p. 8 pp. – 207.
[12] A. Amin, A. Colman, and L. Grunske, "An Approach to Forecasting QoS Attributes of Web Services Based on ARIMA and GARCH Models," in *2012 IEEE 19th International Conference on Web Services*, 2012, pp. 74–81.
[13] G. Rutka, "Network Traffic Prediction using ARIMA and Neural Networks Models," *ELEKTRON ELEKTROTECH*, vol. 84, no. 4, pp. 53–58, Apr. 2008.
[14] A. H. Yaacob, I. K. T. Tan, S. F. Chien, and H. K. Tan, "ARIMA Based Network Anomaly Detection," in *2010 Second International Conference on Communication Software and Networks*, 2010, pp. 205–209.
[15] D. Zeng, J. Xu, J. Gu, L. Liu, and G. Xu, "Short Term Traffic Flow Prediction Using Hybrid ARIMA and ANN Models," in *2008 Workshop on Power Electronics and Intelligent Transportation System*, 2008, pp. 621–625.
[16] L. Horváth, W. Pouliot, and S. Wang, "Detecting at-most-m changes in linear regression models," *J. Time Ser. Anal.*, vol. 38, no. 4, pp. 552–590, Jul. 2017.
[17] B. Rogers, "The dynamics of web traffic," in *The Internet Trap*, Princeton University Press, 2018, pp. 82–101.
[18] W. XingFen, Y. Xiangbin, and M. Yangchun, "Research on User Consumption Behavior Prediction Based on Improved XGBoost Algorithm," in *2018 IEEE International Conference on Big Data (Big Data)*, 2018, pp. 4169–4175.
[19] X. Shi, Q. Li, Y. Qi, T. Huang, and J. Li, "An accident prediction approach based on XGBoost," in *2017 12th International Conference on Intelligent Systems and Knowledge Engineering (ISKE)*, 2017, pp. 1–7.
[20] B. Pan, "Application of XGBoost algorithm in hourly PM2.5 concentration prediction," *IOP Conf. Ser.: Earth Environ. Sci.*, vol. 113, no. 1, p. 012127, Feb. 2018.
[21] J. Zhong, Y. Sun, W. Peng, M. Xie, J. Yang, and X. Tang, "XGBFEMF: An XGBoost-Based Framework for Essential Protein Prediction," *IEEE Trans. Nanobioscience*, vol. 17, no. 3, pp. 243–250, Jul. 2018.

[22] S. Ramraj, N. Uzir, R. Sunil, and S. Banerjee, "Experimenting XGBoost algorithm for prediction and classification of different datasets," *International Journal of Control Theory and Applications*, vol. 9, no. 40, pp. 651–662, 2016.
[23] M. Valipour, "Long-term runoff study using SARIMA and ARIMA models in the United States," *Meteorol. Appl.*, vol. 22, no. 3, pp. 592–598, Jul. 2015.