Web Traffic Time series forecasting

Nikhil Verma

*Information Technology*

*National Institution Of Technology,Karnataka*Surathkal,India  
nikhilverma.222it026@nitk.edu.in

ManMohan Nayak  
*Information Technology*

*National Institution Of Technology,Karnataka*Surathkal,India  
manmohannayak.222it019@nitk.edu.in

M B Manikanta Reddy  
*Information Technology*

*National Institution Of Technology,Karnataka*Surathkal,India  
mbmreddy.222it020@nitk.edu.in

***Abstract*—Web traffic forecasting is a significant issue today since it might hinder the operation of important websites. A popular area of study has been time-series forecasting. One of the most challenging issues in the field is making predictions about time series values. The timeseries discipline covers a wide range of topics, including classification and forecasting as well as inference and analysis. The most effective way to communicate the information would be to forecast the network traffic and show it in a dashboard that updates in real-time. Creating a dashboard would facilitate tracking and studying real-time data. Now, we rely too much on Google's servers, but if we wanted to host a server for a huge user base, we might have anticipated the number of people from prior years to prevent server failure. Time series forecasting will give satisfying results in many prediction problems.**

***Keywords—AR ,ARIMA, XGBoost, Prophet, ANN,LSTM***

# **I. Introduction**

The increase in traffic to almost all websites is bound to happen as more and more people around the globe gain access to the internet. The business that is able to deal with the changes in traffic in the most effective manner will succeed. The increase in website traffic has the potential to cause a lot of problems. Most users have probably experienced a website that crashes or loads very slowly when a lot of people are using it, such as when several shopping websites crash right before holidays as more people attempt to log in than the website can handle. This causes a lot of inconveniences for the users. As a result, users may give the site less favorable reviews and choose another one, which may damage their business. Thereby, a traffic management strategy or plan should be implemented to reduce the possibility of such accidents, which could be harmful to the company's ability to continue operating. Before recently, there was no need for such tools because most servers were capable of handling the influx of traffic. However, the proliferation of smartphones has driven consumer demand for some websites to such an extreme that businesses were unable to respond quickly enough to keep their normal level of customer service.

Many methods have been introduced by various researchers in order to find these trends by using both time series analysis methods and machine learning algorithms. Some of the methods are hybrid which are a mix of both time series and ML models. The mostly used methods are Recurrent Neural Networks with Encoder decoder architecture, CNN’s, LSTM, support vector regression and many others. We discuss about them in detail like what have they followed in literature survey part.

II.  **Literature Survey**

**[1]** At the 2018 IEEE International Conference on Big Data, Petluri and Al-Masri presented a paper titled "Web Traffic Prediction of Wikipedia Pages." The authors of this paper proposed an approach to predict Wikipedia page web traffic using predictive machine learning techniques. An overview of web traffic prediction and its importance in various uses, including website design, network resource allocation, and advertising is provided in the paper's introduction. They proceed on to address the difficulties in predicting web traffic, including the complex and variable nature of user behavior and the requirement for precise and timely predictions. Then they go over their suggested strategy, which entails analyzing past web traffic data and forecasting future traffic using a variety of machine learning techniques. To train and assess their method, the authors used a dataset of web traffic for Wikipedia articles. The dataset includes statistics on the amount of web traffic that over 2,000 Wikipedia sites received over a number of months. After that, the writers go over their experimental design and evaluation metrics, including mean absolute error (MAE), root mean square error (RMSE), and mean absolute percentage error. (MAPE). They contrast their suggested strategy with a number of standard techniques, such as a naïve strategy, an autoregressive model, and a support vector regression model. The experimental findings demonstrate that, in terms of prediction accuracy, the proposed approach outperforms the baseline techniques, with the best performance being attained when using a random forest model. Additionally, the authors perform a sensitivity analysis to assess the effects of different input features. on the precision of predictions. Finally, the authors discuss the drawbacks of their strategy and the potential directions for future study, such as the demand for larger datasets and the investigation of novel machine learning approaches.

**[2**]The 2018 Eleventh International Conference on Contemporary Computing featured a paper by Madan and Mangipudi titled "Predicting computer network traffic: a time series forecasting approach using DWT, ARIMA and RNN." Discrete Wavelet Transform (DWT), Autoregressive Integrated Moving Average (ARIMA), and Recurrent Neural Network (RNN) models were combined in this paper's authors' proposed strategy for predicting computer network traffic. In the introduction to the paper, the significance of network traffic prediction and its applications in network resource management, traffic engineering, and security are briefly discussed. The difficulties in predicting network traffic are then covered by the authors, including the complexity of network traffic patterns, the significant unpredictability of network traffic, and the requirement for precise and timely predictions. The authors then go on to outline their suggested methodology, which entails utilizing DWT to split the network traffic time series into various frequency bands, followed by ARIMA and RNN models to forecast the traffic in each band. To test and refine their method, the authors examined a dataset of network traffic for a university campus network. The dataset includes traffic information from many network applications, including file transfers, email, and web browsing, spanning a period of several weeks. After that, the authors go over their experimental design and evaluation metrics, such as mean absolute percentage error (MAPE), mean absolute error (MAE), and root mean square error (RMSE). (RMSE). They contrast their suggested strategy with a number of standard techniques, such as a naïve strategy, ARIMA, and RNN models. The experimental results demonstrate that, with respect to of prediction accuracy, the suggested strategy outperforms the baseline methods, with the best performance being attained when DWT and RNN models are combined. In order to assess the effects of various input attributes and parameters for the model on prediction accuracy, the authors also perform a sensitivity analysis. Finally, the authors examine the drawbacks of their strategy and potential directions for future study, such as the demand for larger datasets and the investigation of fresh machine learning and deep learning approaches.

**[3]** The proposed method involves using a distributed asynchronous training approach to train the LSTM neural network on large amounts of web traffic data. The authors explain the technical details of this approach and how it allows for faster and more efficient training of the neural network. the research demonstrates the potential of LSTM neural networks with distributed asynchronous training for accurately forecasting web traffic time series data. The method is then evaluated on several real-world datasets of web traffic, and the results show that it outperforms several benchmark models in terms of accuracy and efficiency. The authors also conduct a sensitivity analysis to determine the impact of different hyperparameters on the performance of the method.

**[4]**GACAN Graph Attention Convolution Attention Networks for Traffic Forecasting Based on Multi-granularity Time Series," presented by Zhang, Sikai, and others, will be published in 2021.The introduction of the study emphasizes the significance of traffic forecasting for applications such as urban planning and transportation management. They point out that time series models or graph-based models are typically used in present traffic forecasting techniques, but rarely both. The authors suggest a novel method for forecasting traffic that combines graph attention networks (GATs) and convolutional neural networks (CNNs). The authors then go into similar research in graph-based models and traffic forecasts. They point out that the majority of traffic forecasting techniques now in use either employ statistical models or deep learning models. Due to their capacity to recognize intricate patterns in traffic data, deep learning models have grown in popularity in recent years. However, to capture temporal dependencies in the data, the majority of current deep learning models for traffic forecasting use either recurrent neural networks (RNNs) or convolutional neural networks (CNNs). The authors next go through graph-based traffic forecasting models. They point out that the ability of these models to simulate spatial interdependence in traffic data has given rise to their prominence in recent years. However, the majority of currently used graph-based models depict the geographical dependencies in the data using fixed graphs, which restricts their capacity to accurately capture the dynamic nature of traffic flow. The authors next present their suggested approach, GACAN, which builds a hybrid traffic forecasting model by fusing graph attention networks with convolutional neural networks. They clarify how GACAN captures the dynamic nature of traffic flow by using a dynamic graph structure that evolves over time. The utilization of multi-granularity time series data by GACAN to identify both short-term and long-term trends in the data is also explained. The authors next offer studies to assess GACAN's performance in comparison to other traffic forecasting techniques currently in use. They compare GACAN to three cutting-edge techniques for traffic forecasting using two real-world datasets. On both datasets, they demonstrate that GACAN performs better than these approaches, proving the viability of their suggested strategy.

**[5]** The paper "Web traffic time series forecasting using ARIMA and LSTM RNN" by Tejas Shelatkar and his colleagues presents a comparative study of two different methods for web traffic time series forecasting: the Autoregressive Integrated Moving Average (ARIMA) model and the Long Short-Term Memory Recurrent Neural Network (LSTM RNN) model. The two models are then evaluated on a real-world dataset of web traffic, and the results show that both models are effective in forecasting web traffic, with the LSTM RNN model slightly outperforming the ARIMA model in terms of accuracy. The authors also conduct a sensitivity analysis to determine the impact of different hyperparameters on the performance of the models. Overall, the paper provides a valuable comparison of two different methods for web traffic forecasting and highlights the strengths and weaknesses of each approach. The results suggest that the LSTM RNN model may be more suitable for handling the complex and non-linear nature of web traffic time series data, but the ARIMA model is still a viable option for forecasting web traffic. The study could be useful for web service providers and e-commerce companies in selecting an appropriate method for web traffic forecasting.

**[6]** The paper "A network traffic forecasting method based on SA optimized ARIMA-BP neural network" by Hanyu Yang and his colleagues presents a method for network traffic forecasting using a combination of the Autoregressive Integrated Moving Average (ARIMA) model and Back Propagation (BP) neural network, optimized using Simulated Annealing (SA) algorithm. The proposed method involves using the SA algorithm to optimize the hyperparameters of the ARIMA-BP neural network model, including the ARIMA order, the number of hidden layers, and the number of neurons in each layer. The authors explain the technical details of this optimization process and how it improves the accuracy and efficiency of the model.

The method is then evaluated on a real-world dataset of network traffic from a local area network, and the results show that it outperforms several benchmark models in terms of accuracy and efficiency. The authors also conduct a sensitivity analysis to determine the impact of different hyperparameters on the performance of the method. Overall, the paper demonstrates the potential of using a combination of the ARIMA model and BP neural network, optimized using the SA algorithm, for accurately forecasting network traffic. The method could be useful for network operators and engineers in optimizing network performance and improving the quality of service for users.

**[7]** explores the use of machine learning ensemble methods for predicting network traffic in optical networks. The paper proposes a hybrid ensemble method that combines the outputs of several individual machine learning models, including Random Forest, Gradient Boosting, and XGBoost, to improve the accuracy and robustness of traffic prediction. The proposed ensemble method is then evaluated on a real-world dataset of network traffic from an optical network. The results show that the hybrid ensemble method outperforms the individual models in terms of accuracy and robustness, especially in situations where the individual models may perform poorly due to data noise or outliers.

**[8]** The paper "Comparative study on the time series forecasting of web traffic based on statistical model and Generative Adversarial model" by Kun Zhou and his colleagues presents a comparative study of two different methods for web traffic time series forecasting: the traditional statistical model (ARIMA) and the Generative Adversarial Network (GAN) model. The authors then describe the ARIMA and GAN models and their strengths and weaknesses in handling time series data. The two models are then evaluated on a real-world dataset of web traffic, and the results show that both models are effective in forecasting web traffic, with the GAN model outperforming the ARIMA model in terms of accuracy. The authors also conduct a sensitivity analysis to determine the impact of different hyperparameters on the performance of the models. Furthermore, the authors propose a novel approach by combining the ARIMA and GAN models in a hybrid approach. The hybrid approach aims to leverage the strengths of both models by using the ARIMA model to capture the linear patterns in the data and the GAN model to capture the non-linear patterns. The hybrid approach is also evaluated on the same real-world dataset of web traffic, and the results show that it outperforms both the ARIMA and GAN models in terms of accuracy.

**III. METHODOLOGY**

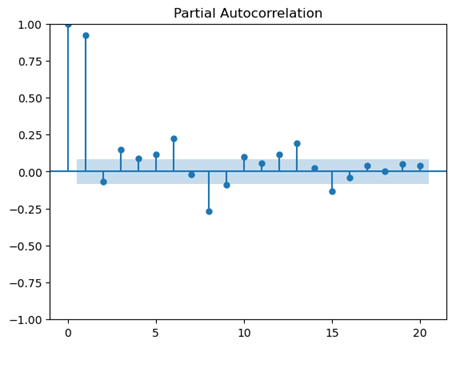
We are taking Wikipedia dataset for performing analysis of our models and analyze what are frequencies in the future for the particular link. Each row represents a link and followed by columns having no of times that particular link is clicked on that day. The training dataset consists of approximately 145k time series. Each of these time series represent a number of daily views of a different Wikipedia article starting from July, 1st, 2015 up until March 1st, 2017. for training the daily web traffic from July,1st,2015 toAugust,10st,2017 is used for training and data of last 31 or 60 days is used for testing the performance of the model.

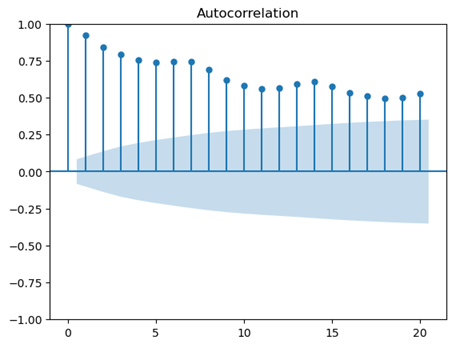
To test for stationarity we have done a dicky fuller test for different language pages.

**Augmented Dicky Fuller Test**

* This test checks if the data is stationary or not.
* It is a hypothesis testing in which the null hypothesis is that the data is non-stationary.
* **p-value > 0.05**: Fail to reject the null hypothesis (H0), the data has a unit root and is non-stationary.
* **p-value <= 0.05**: Reject the null hypothesis (H0), the data does not have a unit root and is stationary.
* The p-value in the augmented Dickey-Fuller (ADF) test is a measure of the strength of evidence against the null hypothesis of a unit root in a time series.
* The test statistic is a t-statistic, and its critical values are determined by the number of observations, the level of significance, and the order of the autoregressive model.

To determine values of p,q,d for ARIMA we are considering ACF and PACF plots.





We’re ARIMA, ARMA, Facebook Prophet, LSTM, XGBoost and Artificial Neural Network to predict values on the dataset we considered.

Each of the model is explained one by one below.

**Auto Regressive(AR)Model**

* It is a type of linear model where the current value of a time series variable is expressed as a linear combination of its past values and a random error term.

Y\_t = c + Σ\_{i=1}^{p} ϕ\_i Y\_{t-i} + ε\_t

where:

* y\_t is the value of the variable being modeled at time t
* c is a constant term (also called the intercept)
* φ\_1, φ\_2, ..., φ\_p are the autoregressive parameters, which represent the coefficients on the past values of y. These parameters are usually estimated from the data.
* ε\_t is the error term, which is assumed to be normally distributed with mean 0 and constant variance.
* The order of the AR model is denoted by p, which represents the number of lagged values of y included in the model. For example, an AR(1) model uses only the lagged value of y at t-1, while an AR(2) model uses both the lagged values of y at t-1 and t-2.

**Autoregressive Integrated Moving Average (ARIMA)**

* It is an extension of the AR model that also includes the integration and moving average components.
* Mathematically, an ARIMA(p, d, q) model can be written as:

y’(t) = c + ϕ1\* y′(t−1) +⋯ + ϕp\*y′(t−p) + θ1\*ε(t−1) +⋯ + θq\*ε(t−q) + εt

where:

* y\_t is the value of the variable being modeled at time t
* c is a constant term (also called the intercept)
* φ\_1, φ\_2, ..., φ\_p are the autoregressive parameters, which represent the coefficients on the past values of y. These parameters are usually estimated from the data.
* ε\_t is the error term, which is assumed to be normally distributed with mean 0 and constant variance.
* θ\_1, θ\_2, ..., θ\_q are the moving average parameters, which represent the coefficients on the past values of the error term. These parameters are also estimated from the data.
* d is the order of differencing, which represents the number of times the data needs to be differenced to make it stationary. If the data is already stationary, d=0.
* **Facebook Prophet Model**
* It is a time series forecasting tool developed by Facebook's Core Data Science team. It is an open-source tool that can be used to model and forecast time series data with daily, weekly, and yearly seasonality.

y(t) = g(t) + s(t) + h(t) + ε\_t

y(t) is the observed value at time t, g(t) is a piecewise linear or logistic growth curve for modeling trend, s(t) represents seasonality (daily, weekly, yearly), h(t) represents holiday effects, and ε\_t is the error term.

**Xtreme Gradient Boosting (XGBOOST)**

XGBoost is a machine learning algorithm that can be used for regression, classification, and ranking tasks. It is based on gradient boosting and involves training a series of decision trees to make predictions, which are then combined to improve overall performance. XGBoost includes features such as regularization, gradient-based optimization, automatic handling of missing values, parallel processing, and customizable loss functions to improve its performance.

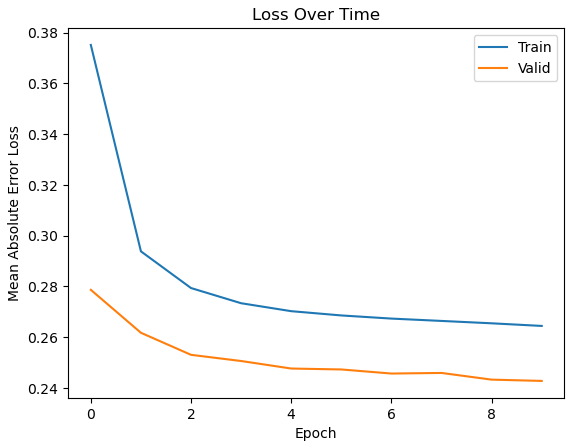
To train an XGBoost model for time series forecasting, the training set should be divided into multiple time periods. The model can then be trained on the historical data from the earlier time periods and evaluated on the data from the later time periods.XGBoost also includes built-in functionality for handling time series data, such as handling missing values and enabling early stopping to prevent overfitting. By using XGBoost for time series modeling, it is possible to achieve high accuracy in forecasting and achieve better performance compared to other traditional time series models.

**Long short term memory (LSTM)**

* It is a type of recurrent neural network (RNN) that is commonly used for sequence modeling and time series forecasting.
* An LSTM network can remember long term dependency and has cell that can store information over multiple steps.

**Artificial Neural Network(ANN)**

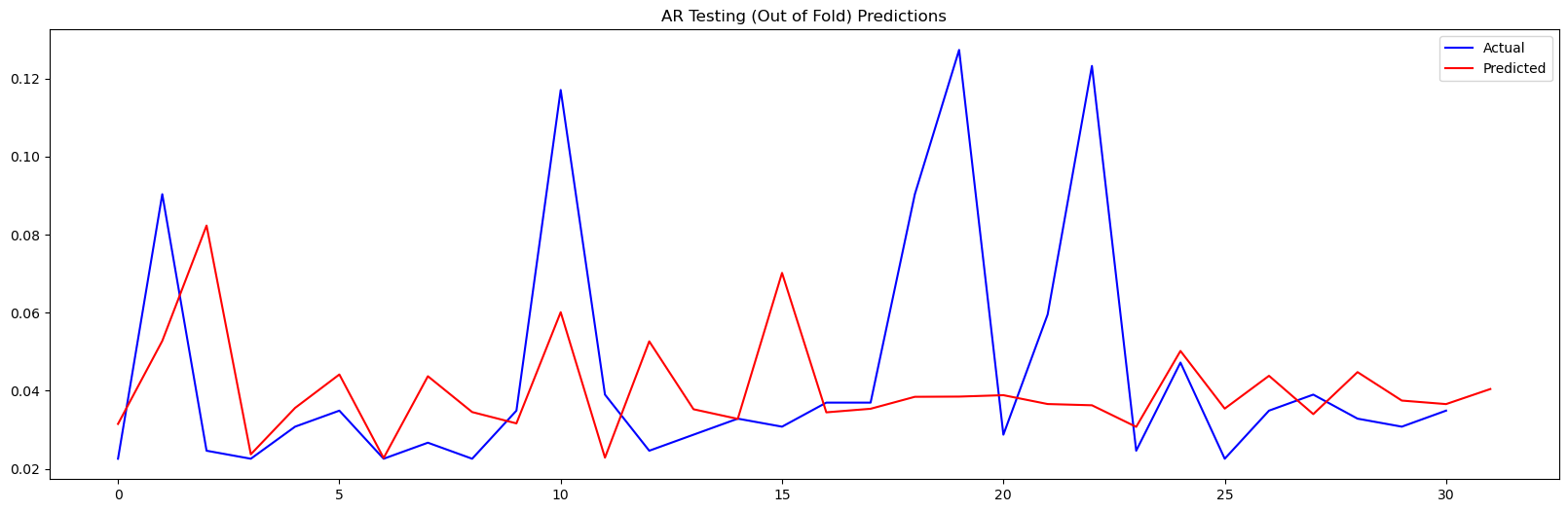
* A deep neural network is a very powerful tool to learn time series modeling. As there are large numbers of layers and trainable parameters, a deep neural network can model complex dependency in time series.
* The model has in total 48657 trainable parameters.
* In this validation set we are using the same time range as the training set, but the validation set is shifted forward for 60 days in time.
* We are using mean absolute error as a loss function.
* For 10 epochs the deep learning model is giving validation loss of .2448 which is very low compared to other models.
* the plot of number of epochs vs training and testing error is as follows -



IV. RESULT & COMPARISON

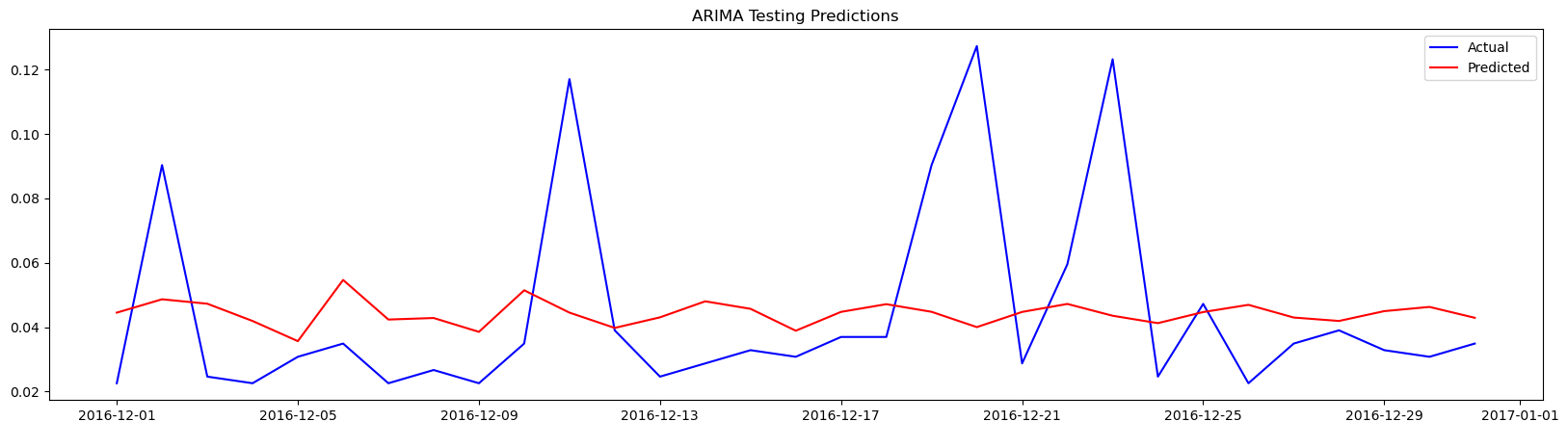
1. Auto Regressive Model (AR) –

SMAPE - 1367.15



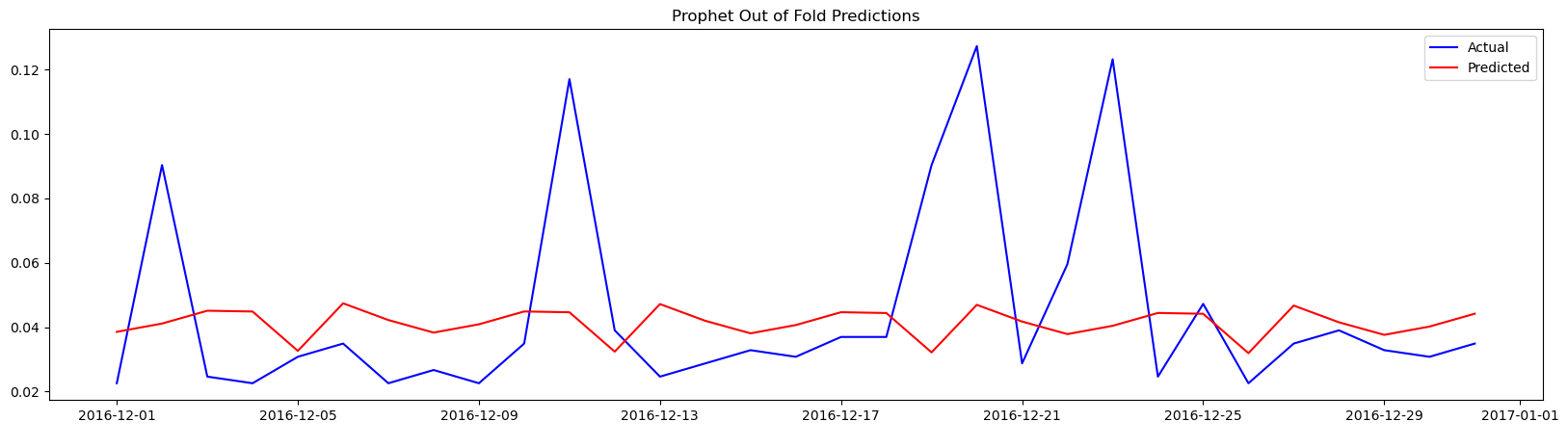
1. Autoregressive Integrated Moving Average (ARIMA) –

SMAPE - 44.52



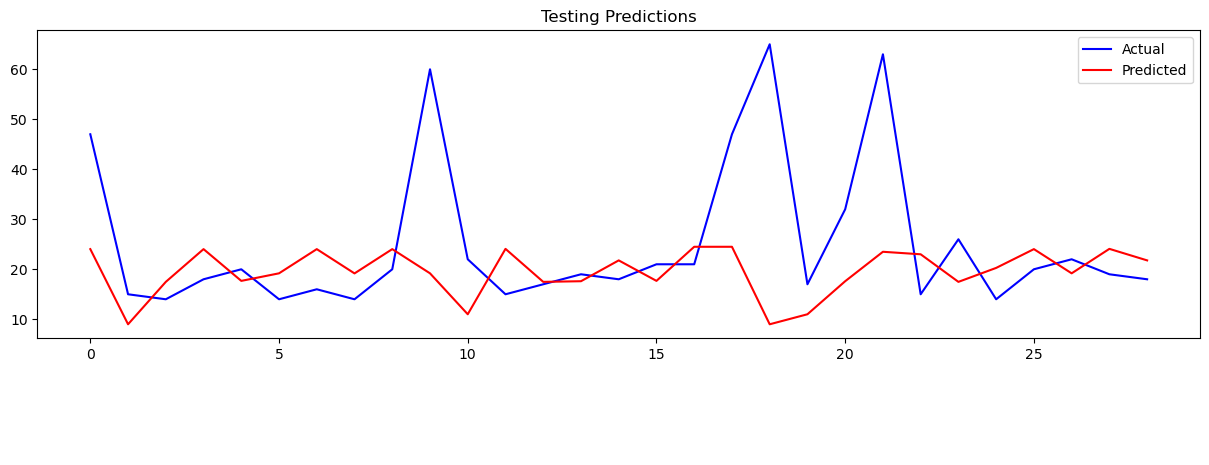
1. Facebook Prophet –

SMAPE - 42.62



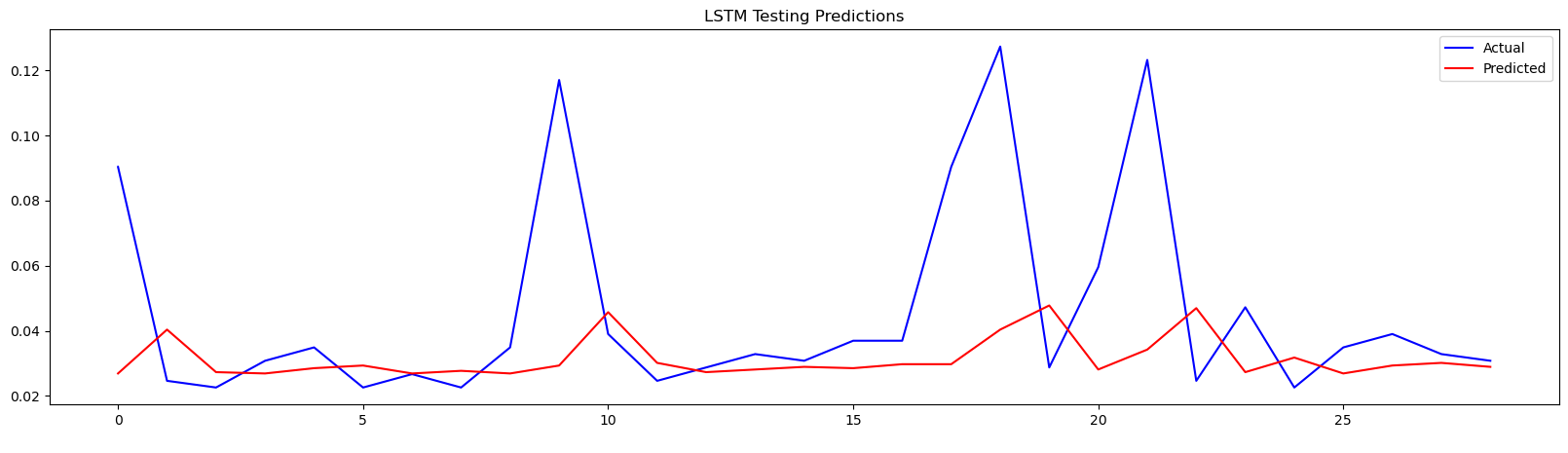
1. Xtreme gradient boosting (XGBOOST) –

SMAPE – 40.54



1. Long short term memory (lstm) –

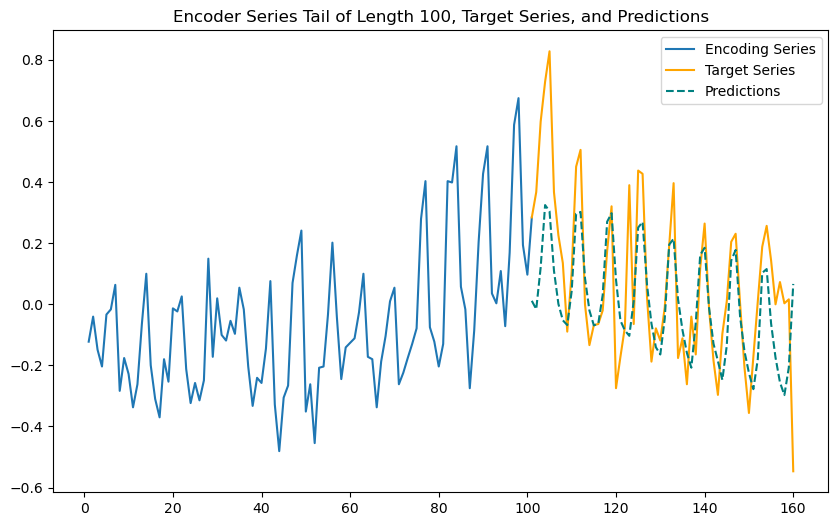
SMAPE – 1118.15



1. ARTIFICIAL NEURAL NETWORK (ANN)-

mean absolute error – 0.27

the plot of next 60 days –



COMPARISON WITH OTHER RESEARCH –

1. Comparing the results of ANN model with "Web traffic prediction of wikipedia pages." by Petluri, Navyasree, and Eyhab Al-Masri.

SMAPE of model in paper – 0.351

SMAPE of proposed ANN model – 0.15

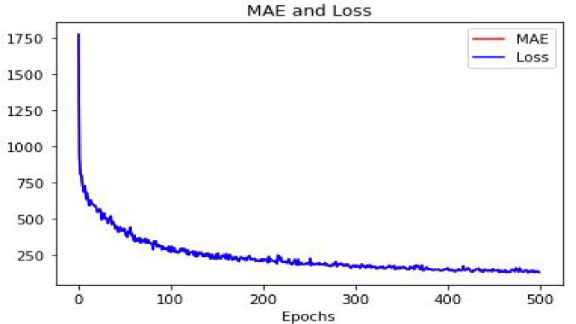
from Symmetric mean absolute percentage error(SMAPE) it is clear that out proposed ANN model has better performance than the model proposed in paper.

1. Comparing the result of ANN modle with "Web

traffic time series forecasting using LSTM neural networks with distributed asynchronous training." by Casado-Vara, Roberto.

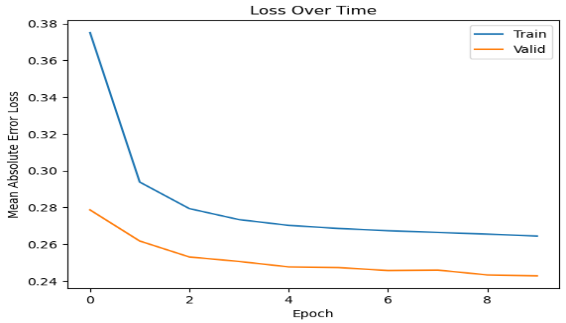
plot of MAE loss and number of epochs in LSTM neural network.

MAE achieved – 132.26



plot of MAE loss and number of epochs in proposed ANN model.

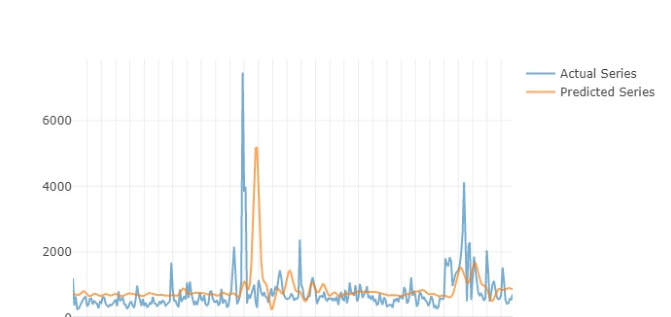
MAE achieved – 0.24



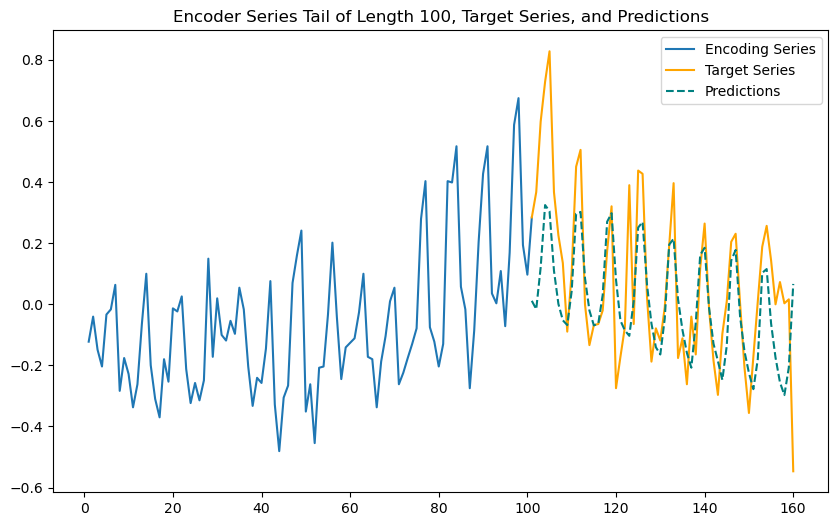
from the performance matrix and error vs epochs graph it is clear that the propose ANN model is giving better performance.

1. Comparing the result of ANN modle with Web traffic time series forecasting using ARIMA and LSTM RNN." by Tejas Shelatkar, Stephen Tondale, Swaraj Yadav ,and Sheetal Ahir.

plot of web traffic in the next 15 days-



plot of next 60 days of web traffic –



from the two graphs of actual and predicted values it is clear that graph generated by our proposed model is more similar with the graph of actual values.

|  |  |  |
| --- | --- | --- |
| **Base Paper Title** | **Base Paper Metrics** | Proposed ANN model Metrics |
| Web traffic prediction of wikipedia pages | SMAPE = 0.351 | SMAPE = 0.15 |
| Web  traffic time series forecasting using LSTM neural networks with distributed asynchronous training | MAE = 132.26 | MAE = 0.24 |

V. CONCLUSION AND FUTURE WORK

Web traffic time series prediction can be carried out using a Deep Artificial Neural Network model more efficiently and effectively. to the model performance will improve as more data is used for training. we have trained the model for predicting web traffic of Wikipedia pages but the proposed model can be used to predict the traffic of any web page by training the model with time series data of the web page. Despite the limitations of our research such as a dataset with relatively limited data and the unpredictable nature of human behavior, we experimentally verified that the forecasting results of our AI model are pretty accurate and close to the real values. in future to more accurately model the time series we can use optimization technique in deep neural network. for predicting Wikipedia traffic as the Wikipedia has pages for different languages and by training the ANN model with data of different language we can more accurately predict the future traffic load.

VI. REFERENCES

1. Petluri, Navyasree, and Eyhab Al-Masri. "Web traffic prediction of wikipedia pages." 2018 IEEE International Conference on Big Data (Big Data). IEEE, 2018.
2. Madan, Rishabh, and Partha Sarathi Mangipudi. "Predicting computer network traffic: a time series forecasting approach using DWT, ARIMA and RNN." 2018 Eleventh International Conference on Contemporary Computing (IC3). IEEE, 2018.
3. Casado-Vara, Roberto, et al. "Web traffic time series forecasting using LSTM neural networks with distributed asynchronous training." Mathematics 9.4 (2021): 421.
4. Zhang, Sikai, et al. "GACAN: Graph Attention-Convolution-Attention Networks for Traffic Forecasting Based on Multi-granularity Time Series." arXiv preprint arXiv:2110.14331 (2021).
5. Shelatkar, Tejas, et al. "Web traffic time series forecasting using ARIMA and LSTM RNN." ITM Web of Conferences. Vol. 32. EDP Sciences, 2020.
6. Yang, Hanyu, et al. "A network traffic forecasting method based on SA optimized ARIMA–BP neural network." Computer Networks 193 (2021): 108102.
7. Szostak, Daniel. "Machine learning ensemble methods for optical network traffic prediction." 14th International Conference on Computational Intelligence in Security for Information Systems and 12th International Conference on European Transnational Educational (CISIS 2021 and ICEUTE 2021) 14. Springer International Publishing, 2022.
8. Zhou, Kun, et al. "Comparative study on the time series forecasting of web traffic based on statistical model and Generative Adversarial model." Knowledge-Based Systems 213 (2021): 106467.