Web Traffic Time Series Forecasting using an Ensemble of Statistical and Deep Learning Models

Vishnu Vardhan Manivannan
Department of Industrial and Systems Engineering
State University of New York at Buffalo
vmanivan@buffalo.edu

***Abstract*—This project aims to develop a robust univariate time series forecasting model tailored for predicting web traffic. Leveraging an ensemble machine learning model that combines AutoRegressive Integrated Moving Average (ARIMA) and Long Short-Term Memory (LSTM), the study intends to analyze and forecast the web traffic patterns of pages on the Wikipedia domain over a period of 18 months. The objective is to enhance the accuracy of web traffic forecasts by harnessing the complementary strengths of ARIMA and LSTM models. The implementation of this advanced ensemble model is anticipated to yield a meaningful enhancement in the prediction of web traffic patterns, enabling more efficient server resource management and contributing to an enhanced user experience on the Wikipedia platform.**

Keywords—ARIMA, LSTM, Time Series Forecasting, Web Traffic, Wikipedia

I. Introduction

Time series forecasting for web traffic involves analyzing historical data to predict future web activity. This data typically includes the number of visitors, page views, and unique sessions over time, often showing patterns like daily, weekly, or seasonal variations. The complexity of forecasting web traffic arises from these patterns, coupled with irregular spikes due to events like marketing campaigns or viral content.

Traditional statistical methods like ARIMA (Auto-Regressive Integrated Moving Average) have long been used for such predictions, focusing on capturing underlying trends and seasonal effects. However, with the advent of big data and machine learning, more sophisticated techniques have emerged. Models like LSTM (Long Short-Term Memory) neural networks excel in handling large datasets with complex, nonlinear relationships, making them particularly well-suited for web traffic forecasting, which often deals with high variability and non-stationary data.

Forecasting web traffic is vital for companies as it directly impacts resource management. Anticipating high traffic periods enables companies to scale server capacities to maintain website performance and avoid downtimes, which are critical for user experience and retention. Similarly, server resources can be scaled down during predicted lower traffic periods, reducing operational costs. Accurate traffic predictions also play a key role in revenue optimization, particularly for sites reliant on advertising or e-commerce. According to authoritative data, Google.com's traffic will be reduced by 20% for every additional 500ms of response time, and Amazon.com's sales will be reduced by 1% for every additional 100ms of response time. By anticipating high-traffic periods, companies can strategically time their marketing and promotional efforts. Additionally, understanding traffic patterns aids in content strategy, aligning content release with peak viewer interest.

II. Related Work

There are several research papers that discuss various approaches to forecasting web traffic ranging from traditional statistical approaches like AR and ARIMA to advanced deep learning based approaches such as LSTMs, transformers and GANs.

[1] explore a distributed approach to training LSTMs for web traffic forecasting where there exists a copy of an LSTM in separate Virtual Machines (VM) for training on different sections of the data. Each of these LSTMs in turn pass on their hyperparameters to the main LSTM residing on a separate "parameter server". The main LSTM then updates the hyperparameters and then sends them back for re-training.

[2] use a Generative Adversarial Network (GAN) with an LSTM as the generator and a deep Multi-Layer Perceptron (MLP) as the discriminator to forecast web traffic. They found that the GAN approach does not produce and remarkable difference in prediction compared to a baseline ensemble model of ARIMA with an ANN.

[3] perform a comparative analysis of 3 different approaches: a neural net ensemble, ARIMA and Holt-Winters. They found that the neural net ensemble outperformed the other 2 approaches in shorter time scales (5 mins, 1 hr) but the Holt-Winters approach was better in a longer time scale of 1 day. The authors concluded that their research would enable more efficient traffic engineering which will result in financial gains from better network resource management.

[4] explore whether forecasts from individual forecasting models can be improved with the use of combination rules. The authors use FARIMA, FARIMA with student-*t* innovations and Artificial Neural Networks as individual forecasting models, since each one of them explains some statistical characteristic of our data, and then combine the forecasts using three different combination rules. Finally, a scheme where the selection of the model is based on the White's Neural Network test for non-linearity is considered and compared with the results from the combination of forecasts.

[5] performed a comparative analysis of AR (Auto Regressive), ARIMA, LSTM and XGBoost (eXtreme Gradient Boosting)

models for web traffic time series forecasting. The AR model produced the highest test error, the ARIMA model had lower test error but had high SMAPE value indicating overfitting on the data, the LSTM had lowest test error yet had a high SMAPE value indicating higher bias and the XGBoost had a relatively low test error and better performance over the test data than ARIMA.

[6] utilize a Recurrent Neural Network (RNN) seq2seq model for time series forecasting of Wikipedia web traffic with heavy emphasis on feature engineering. They do not report a significant improvement over traditional or baseline models.

[7] offers a comprehensive review of Transformer models in time series analysis, highlighting their capability to capture long-range dependencies. It examines these models from structural adaptations for time series challenges and their applications in forecasting, anomaly detection, and classification. The paper also conducts empirical analyses on model performance and proposes future research directions, marking a significant step in understanding the application of Transformers in time series data.

[8] provides a comprehensive survey of deep learning techniques in time-series forecasting, covering various domains. It examines the integration of temporal data through common encoder and decoder designs, highlights the emergence of hybrid models blending statistical methods with neural networks, and discusses deep learning's role in decision support. The review also touches on state-of-the-art approaches, including multi-horizon forecasting and interpretability, and acknowledges the breadth of existing literature on both traditional and automated time-series forecasting methods.

III. KNOWLEDGE GAPS

While most traditional as well as deep learning approaches to web traffic time series forecasting are successful in capturing trends in the data, the forecasts from these models usually don't tend to take exogenous factors such as social media virality, changes in search engine algorithms or other external shocks into account. This is the largest research gap in time series forecasting. Furthermore, deep learning models are typically better at capturing complex, non-linear relationships but they come with their own set of challenges: 1. They require much larger amounts of data to perform well. In the context of time series forecasting, this can be a limitation, especially for new or niche domains where sufficient historical data may not be available. 2. Being black-box models, the results of deep learning models are often not interpretable. For a problem domain such as time series forecasting, which has strong roots in interpretable statistical models, deep learning approaches must start to focus on interpretability to gain acceptance in industries which strongly depend on interpretability. 3. Deep learning models are also much more computationally expensive compared to traditional approaches. Computationally efficient deep learning models for time series forecasting are therefore another knowledge gap that needs to be explored.

IV. PROPOSED METHODOLOGY

This paper proposes an ensemble approach combining ARIMA and an LSTM neural network for forecasting web traffic. The complementary strengths of ARIMA (which excels at capturing the linear, statistically significant patterns) and LSTM (which excels at capturing complex, non-linear patterns) will offer a more robust forecasting capability as compared to any single-model approach.

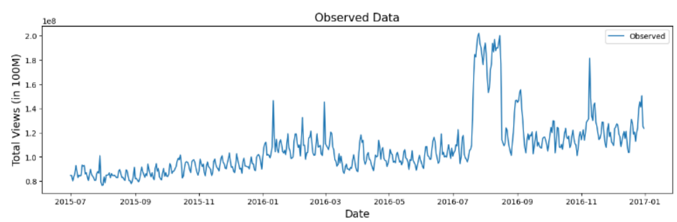
Data Description: The research utilizes web traffic data from pages within the Wikipedia domain, which has been furnished by Google for a research competition hosted on the Kaggle platform. It contains the web traffic data for approximately 145K Wikipedia articles in various languages. For each article, there is the daily number of visits from July of 2015 to December of 2016.

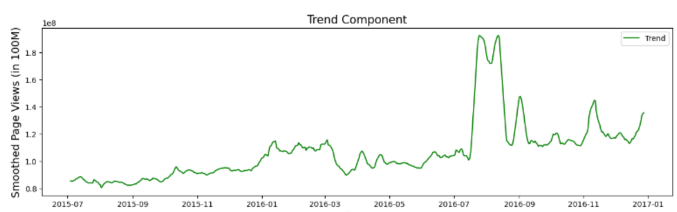
Data Preprocessing: The dataset contains pages of in several different languages such as English, Mandarin, Spanish, French, etc. This project restricts itself to forecasting the combined total views of all English pages. Therefore, the English pages are filtered and the views of all the English pages are added up. This gives us the total views of all English pages per day between 1st July 2015 to 31st Dec 2016. Missing values are imputed with 0 as the presence of missing values indicates that the page did not exist on that date.

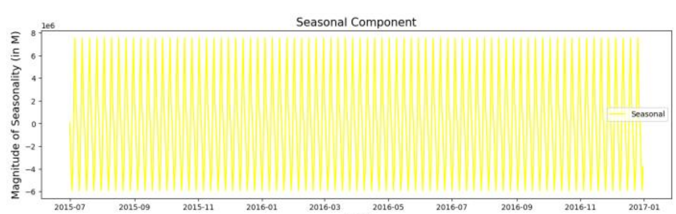
The dates are resampled with a daily sampling. Although the data already consisted of only daily samples, resampling it assigns a frequency to it. This is done because frequency ambiguity could lead to the SARIMAX model auto assigning the frequency which needs to be avoided.

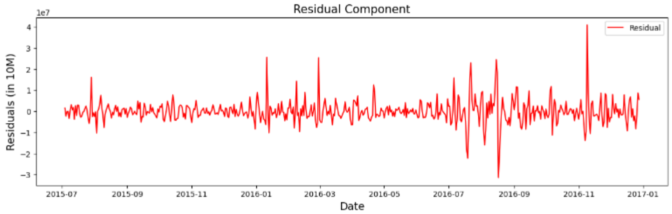
For the LSTM model, the data is first scaled and then passed to the 'TimeseriesGenerator' function in the Keras library with a defined window size to generate batches of time series data.

Exploratory Data Analysis: The data is decomposed into its trend, seasonal and the residual components and these components are then plotted.









It is evident that there is seasonality present in the data. There are also some spikes in the trend graph. These were likely introduced by exogenous factors such as news, social media virality, etc.

Model Assumptions: The ARIMA model assumes that the time series is stationary, which means:

1. Constant Mean: The mean of the series should not be a function of time. Instead, it should be constant over intervals.
2. Constant Variance: The variance of the series should not a be function of time. This property is known as homoscedasticity.
3. Constant Autocovariance: The covariance of the ith term and the (i+m)th term should not be a function of time.

Stationarity is important because the properties of a stationary time series do not change over time, which allows for consistent prediction rules. If a time series has a trend or seasonality (components that typically violate stationarity), the relationships between lagged observations of the series change over time, which makes the model's predictions unreliable.

Data stationarity was tested using two tests: The Augmented Dickey Fuller (ADF) Test and the Kwiatkowski-Phillips-Schmidt-Shin (KPSS) tests. The ADF test tests for a unit root in the time series. If the test statistic is less than the critical value, we reject the null hypothesis of a unit root, and the series is considered stationary. The KPSS test tests for the presence of a trend-stationarity. If the test statistic is greater than the critical value, we reject the null hypothesis that the series is trend-stationary, implying that the series is non-stationary. In both tests the test statistic was greater than the critical value, implying that the time series was not stationary.

Stationarity is removed by differencing of the time series data, but it was not explicitly performed because the auto-ARIMA model can automatically determine the differencing order required to make the data stationary.

The LSTM model does not require the time series data to be stationary.

Data Split: The time series data is 18 months long. The first 15 months (83.33%) were used as training set and the last 3 months (16.67%) were used as the testing set.

Model Development:

1. ARIMA Model: The auto-ARIMA model from the ‘pmdarima’ library is trained on the training set. This model will focus on capturing linear relationships and seasonality in the web traffic data. The model runs a grid search to find the optimal parameters (p, d, q, P, D, Q) using the AIC (Akaike Information Criterion).

   The Seasonal ARIMA model parameters:

   - p: the number of lag observations in the model; also known as the lag order.
   - d: the number of times that the raw observations are differenced; also known as the degree of differencing.
   - q: the size of the moving average window; also known as the order of the moving average.
   - P: Seasonal autoregressive order.
   - D: Seasonal difference order.
   - Q: Seasonal moving average order.
   - m: The number of time steps for a single seasonal period.

   The final model with the best hyperparameters is used to make the forecast.

2. LSTM: A deep learning model with LSTM and dense layers is developed using the tensorflow library and is trained on the training set. This model will focus on capturing complex, non-linear, long-term dependencies in the data. The model architecture is defined below:

```
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 lstm_3 (LSTM)               (None, 7, 100)            40800

 lstm_4 (LSTM)               (None, 7, 50)             30200

 lstm_5 (LSTM)               (None, 10)                2440

 dense_3 (Dense)             (None, 64)                704

 dense_4 (Dense)             (None, 32)                2080

 dense_5 (Dense)             (None, 1)                 33

=================================================================
Total params: 76257 (297.88 KB)
Trainable params: 76257 (297.88 KB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

In the dense layers, the ReLU activation function is used. The model uses the Mean Squared Error as its loss function (or estimator). It uses the ‘Adam’ optimizer. The Adam optimizer is an advanced stochastic optimization method that adaptively adjusts learning rates for each parameter. It leverages the benefits of both AdaGrad and RMSProp optimizers, employing moment estimates to efficiently handle sparse gradients and non-stationary objectives.

The model is trained over 100 epochs and with different window sizes (representing multiples of the time steps for one season) and the model outputs are pre-processed to extract the forecast.

3. Ensemble Approach: The final forecasts will be generated by averaging the ARIMA and LSTM forecasts.

Model Evaluation: The ensemble model will be evaluated using metrics such as Root Mean Squared Error (RMSE) and Symmetric Mean Absolute Percentage Error (SMAPE).

Results: The performance metrics of the individual and ensemble models are as follows:

Model	RMSE	SMAPE
Seasonal ARIMA	13344923.64	8%
LSTM (best window size = 7)	13767506.34	8%
Ensemble	11521797.42	6%

Figure below shows the forecasts of the seasonal ARIMA and the LSTM over the test data.

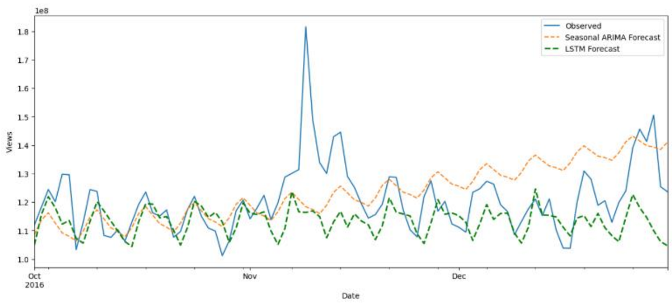
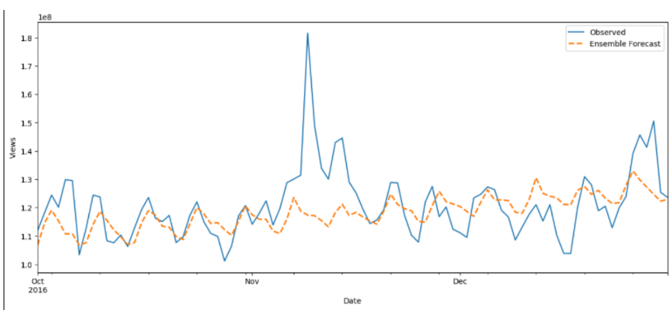


Figure below shows the forecast of the ensemble model over the test data:



V. INFERENCE

It is clear from the results of the individual and ensemble models that ensemble model has outperformed the individual models.

For the seasonal ARIMA model, we observe an RMSE value of around 13.3 million which is not too high considering that the ground truth was in the scale of 100s of millions. This is confirmed by the SMAPE of 8%. The LSTM results are similar with an RMSE of 13.7 million and a SMAPE of 8%.

On the other hand, we observe that the ensemble model has an RMSE of 11.5 million and a SMAPE of 6%.

These results coincide with the hypothesized outcome that the combined forecasts of ARIMA (excels at capturing linear, statistically significant patterns) and LSTM (excels at capturing complex, non-linear patterns) will be better than the individual model forecasts.

VI. SIGNIFICANCE AND CONTRIBUTIONS

The success of this ensemble approach can prove to be valuable for improved web traffic forecasting accuracy, not just for Wikipedia but for other web-service based companies as well. Enhanced forecasting accuracy allows businesses to make more informed decisions regarding content scheduling, marketing strategies, server capacity planning, and customer engagement initiatives. This could lead to better allocation of resources and increased revenue. Furthermore, websites can ensure optimal performance even during peak times, which enhances the user experience. This can lead to higher user retention rates and improved brand reputation.

VII. FUTURE WORK

It is observed that the main source of the inaccuracy of the individual model are the spikes caused by exogenous factors. If a methodology to monitor and quantify the exogenous factors can be developed, then it can be used to significantly improve the forecasting capability of the individual models and in-turn the ensemble model. This will allow companies to fine tune their server capacity planning as well as content scheduling which will lead to much higher cost savings/revenue.

REFERENCES

[1] R. Casado-Vara, A. Martin del Rey, D. Pérez-Palau, L. de-la-Fuente-Valentín, and J. M. Corchado, “Web Traffic Time Series Forecasting Using LSTM Neural Networks with Distributed Asynchronous Training,” *Mathematics*, vol. 9, no. 4, p. 421, Feb. 2021, doi: 10.3390/math9040421.

[2] Kun Zhou, Wenyong Wang, Lisheng Huang, Baoyang Liu, “Comparative study on the time series forecasting of web traffic based on statistical model and Generative Adversarial model”, Knowledge-Based Systems, Volume 213, 2021, 106467, ISSN 0950-7051, https://doi.org/10.1016/j.knosys.2020.106467.

[3] Cortez, P., Rio, M., Rocha, M. and Sousa, P. (2012), Multi-scale Internet traffic forecasting using neural networks and time series methods. Expert Systems, 29: 143-155. https://doi.org/10.1111/j.1468-0394.2010.00568.x.

[4] Katris, C., Daskalaki, S. (2015). Combining Time Series Forecasting Methods for Internet Traffic. In: Steland, A., Rafajłowicz, E., Szajowski, K. (eds) Stochastic Models, Statistics and Their Applications. Springer Proceedings in Mathematics & Statistics, vol 122. Springer, Cham. https://doi.org/10.1007/978-3-319-13881-7_34.

[5] J. . Telo, “Web Traffic Prediction Using Autoregressive, LSTM, and XGBoost Time Series Models”, *ARAIC*, vol. 3, no. 1, pp. 1–15, Jan. 2020.

[6] N. Petluri and E. Al-Masri, "Web Traffic Prediction of Wikipedia Pages," 2018 IEEE International Conference on Big Data (Big Data), Seattle, WA, USA, 2018, pp. 5427-5429, doi: 10.1109/BigData.2018.8622207.

[7] Wen, Q., Zhou, T., Zhang, C., Chen, W., Ma, Z., Yan, J., & Sun, L. (2022). Transformers in time series: A survey. *arXiv preprint arXiv:2202.07125*.

[8] Lim Bryan and Zohren Stefan (2021), “Time-series forecasting with deep learning: a survey”, *Phil. Trans. R. Soc. A.***379**2020020920200209, http://doi.org/10.1098/rsta.2020.0209