Web Traffic Prediction of Wikipedia Pages

Navyasree Petluri, Eyhab Al-Masri
School of Engineering and Technology
University of Washington
Tacoma, USA
{navyap08, ealmasri}@uw.edu

***Abstract*—In recent years, more emphasis on how to predict traffic of web pages has increased significantly and prompted the need for exploring various methods on how to effectively forecast future values of multiple times series. In this paper, we apply a forecasting model for the purpose of predicting web traffic. In particular, we use existing Web Traffic Time Series Forecasting dataset by Google to predict future traffic of Wikipedia articles. Predicting web traffic can help web site owners in many ways including: (a) determining an effective strategy for load balancing of web pages residing in the cloud, (b) forecasting future trends based on historical data and (c) understanding the user behavior. To achieve the goals of this research work, we built a time-series model that utilizes RNN seq2seq model. We then investigate the use of symmetric mean absolute percentage error (SMAPE) for measuring the overall performance and accuracy of the developed model. Finally, we compare the outcome of our developed model to existing ones to determine the effectiveness of our proposed method in predicting future traffic of Wikipedia articles.**

I. INTRODUCTION

Analysis and forecasting web traffic has many applications in various areas and has attracted significant number of studies. It is a proactive approach which helps in providing a secure, reliable and qualitative web communication. Web traffic is the amount of data sent and received by visitors to a website [1] which is determined by the number of visitors and the number of pages they visit. Website owners often use web traffic tools to monitor the incoming and outgoing traffic to determine, for example, the popularity of web pages or patterns and trends based on the page views; collected website traffic information can help in structuring the content of websites, dealing with security issues like Denial-of-service (DDos) attacks, building prior knowledge of any lack of resource availability and optimizing cloud resources in real-time. Web traffic is regularly monitored to understand the customer behavior. For example, collecting web page visits can help business site owners to determine popular products and pages with highest selling rates.

Forecasting web traffic is an active area of research and there are many research studies that investigate web traffic for using many different scientific methods. For example, the authors in [2] propose the use of wavelet pattern analysis and neural networks for predicting web traffic. Although this is useful in predictions, it is adequately complex and time consuming particularly for real-time analysis of website traffic. Another research article focused on the use of genetic algorithms and neural networks [3]. Applying genetic algorithms has been observed to contribute to an overall performance compared to the back-propagation algorithm.

Although there have been numerous research efforts in analyzing and forecasting website traffic, minimal research work has focused on the temporal observations that emerge for the analysis and interference to classification and forecast in helping to predict future views of web pages. Hence, in this project, we will explore the use of a wider range of forecasting model features to accomplish the goals of this project. To overcome many of the existing research challenges, we investigate potentially an optimal number of features required to attain a high accuracy rate in terms of predicting traffic for web pages.

The main objective of our research work is to build a consistent forecasting model to predict the future traffic of Wikipedia pages. We employ the Web Traffic Time Series Forecasting dataset (released by Google) [4] to test our prediction model. Through the development of this model, we focus our investigation on the following two key research aspects, (a) the relationship between the magnitude of the features used for prediction model and its overall performance and (b) the optimal number of features required while maintaining high accuracy prediction results.

To address the above questions, we experimented via running the model against page features as well as time series features. We then analyzed obtained results and compare the accuracy of our model under various conditions to determine the importance of each of the used features. Furthermore, we added additional features such as median of specified window length in each time series (e.g. capturing weekly, monthly, quarterly and yearly page popularity) as well as the medians of variable time frame windows using the golden ratio method [5] as in [6][7]. We further provide detailed analysis of our experiments and provide insights on improving the existing model.

II. METHODOLOGY

In this section, we describe the methodology followed in developing our prediction model. The methodology involves (a) data collection and (b) implementation.

A. Web Traffic Time Series Dataset

The core dataset used for this project is Google's Web Traffic Time Series Forecasting (provided via Kaggle) [4] consisting of approximately 145,000 Wikipedia articles. The dataset includes a field representing the time series or multiple points given in an order of time. For example, each of the time series represents a number of daily views of a different Wikipedia article, starting from July 1st, 2015 up until September 11th, 2017 [4]. Due to the fact that the does not differentiate between values of zero and missing values in the traffic data, this causes some ambiguity in the overall

978-1-5386-5035-6/18/$31.00 ©2018 IEEE

predictions which we will have to account for in our prediction model.

B. Implementation details

We analyzed existing winning model provided to the Kaggle's competition which used RNN seq2seq model as in [8]. This prediction model is built based on: a) number of hits, b) features which are extracted from page URLs, c) day of week - analyzes the weekly seasonality information, d) year-to-year autocorrelation (quarterly and yearly), e) page popularity and, f) lagged pageviews. We rebuilt the existing winning model with the entire dataset as a training data using RNN seq2seq model with the help of Encoder/decoder Architecture. Encoder is cuDNN GRU [8] as it performs task with better speed compare to regular tensors. Decoder is TensorFlow GRUBlockCell [8]. The generated results from decoding are used as inputs to the next step till the end of the sequence for a given batch size.

C. Feature Engineering

We added new approaches/trials to improve the existing model and analyzed the working of the model based on an enhanced set of features. For example, we considered the inclusion of the median which is the one of the most important measures of dataset behavior. Considering median for entire time series length in monotonic trend or time series with high and sudden spikes may result in less sensitivity to the fluctuations in the data. Hence, rolling median is used as measure of median of the successive segments of data either for fixed window length or for variable window length. For time series exhibiting monotonic trend, the median lies exactly in between low and high value as in [9]. When median for entire time series length is considered as feature, as can be seen on Figure 1 (left), it impacts the model behavior as it affects the low values and high values in the series since the difference is huge relative to median. Figure 1 (right) shows the rolling median is considered for a fixed window size length, the extreme values in the period are least effected by the median of the chosen window. Similarly, seasonality component usually occurs periodically (e.g. quarterly, half yearly, etc.) as in [9] will have same influence as presented in Figure 2 (left) for entire time series length whereas in Figure 2 (right) shows the response of the model to the slight changes in the data.

To capture trends, instead of considering the entire time series length as feature window, we added rolling window to forecast and derive weekly, monthly, quarterly and yearly page popularity. We have measured the rolling median taking 7, 30, 90 and 180 days as window size and normalized each with zero mean and unit variance. Unlike considering each forecast window as an independent feature, we have taken the median of medians as a single feature for each sample.

The main idea is to calculate median of each time series for a variable window length which grows exponentially based on golden ratio [5] and then measure median of rolling medians of variable window length to estimate future forecasts as in [6][7]. In more detail, exponential rolling median gives more weightage to the near data points whereas rolling median for fixed length window size gives equal weightage to all the data points as in [10]. We found window size 6 is giving better

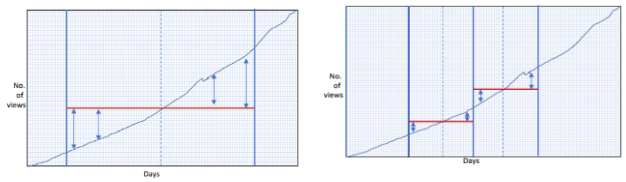


Figure 1: Effect of Rolling Median on Trend

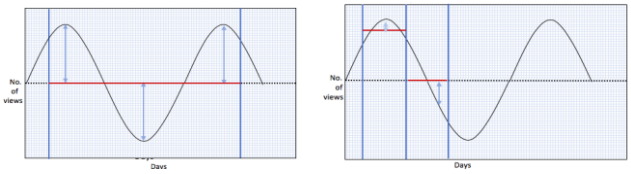


Figure 2: Effect of Rolling Median on Seasonality

scores and hence calculated the median for Fibonacci sequence 6, 12, 18, 30, 48, and so on as taken in [7]. For each time series, we measured the median of medians as feature and results are presented in Table [1]. Final model is evaluated based on the monthly pattern for a time series (Views vs. days) using SMAPE measurement between forecast and actual values. It is a measure of accuracy based on the percentage errors [11]. Equation 1 represents the SMAPE formula, where F_t is forecast value, A_t is the actual value and n is the number of fitted points [11].

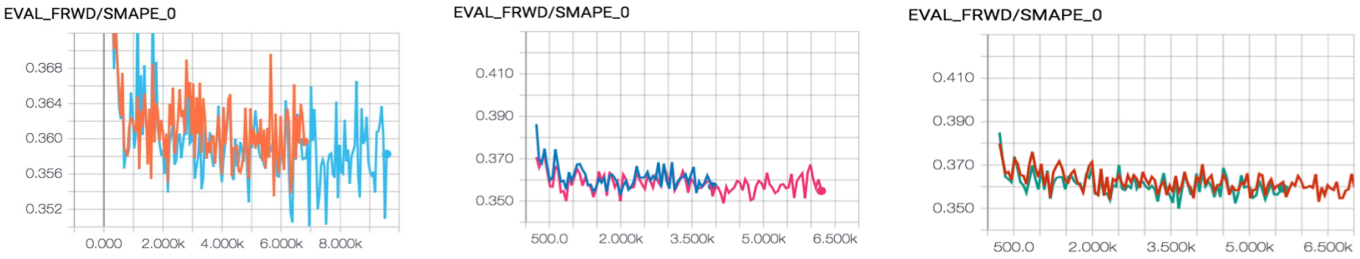
$$\text{SMAPE} = \frac{100\%}{n} \sum_{t=1}^{n} \frac{|F_t - A_t|}{(|A_t| + |F_t|)/2} \quad (1)$$

TABLE I. OBSERVED RESULTS WITH SMAPE METRIC

Model and its features	Training set(days)	SMAPE
Existing model	804	0.351
	740	0.354
Model with only 7 day median and page popularity	804	0.350
Model with 7, 30, 90, 180 days median as independent features	804	0.349
Model with median of medians with Fibonacci sequence of variable window length	804	0.349
Model with all the newly added features	740	0.351

III. EXPERIMENTAL ANALYSIS AND RESULTS

As can be seen on Table I from the observed results, the lower the value of measurements translates into having a better model. Though we obtained all predictions during September 13th 2017 to November 13th 2017, it cannot be compared to the actual web traffic values. In order to visualize the differences between actual data and predicted data, we have used 740 days as training set to predict traffic values from July 11th 2017 to September 11th 2017. We have plotted the number of hits vs. days for the entire time series along with the actual values and predictions during testing period for four different web pages.



(a) Orange-Existing Model; Blue-New Model (b) Blue – Existing Model; Pink – New Model (c) Red-Existing Model; Green-New Model

Figure 3: SMAPE Evaluation for Three Trained Models

In the new model, predictions are comparatively more accurate, as shown in Figure 4. We have run 3 models in parallel on 2 seeds. Using tensor board, we generated the resulting SMAPE graph for each of the three existing and new models to check the variations in the model performance at every step as shown in Figure 3. With minimalistic inputs, RNN performs best in uncovering the features independently as they make use of sequential information where model is run for every element of the sample, with the current output relying on the stored past calculations given in [12]. With simple median as feature, RNN is able to estimate the proportion of web traffic with which it was able to do predictions with stability based on the previous computations which being the reason for slight improvement in case of rolling median and Fibonacci median as features.

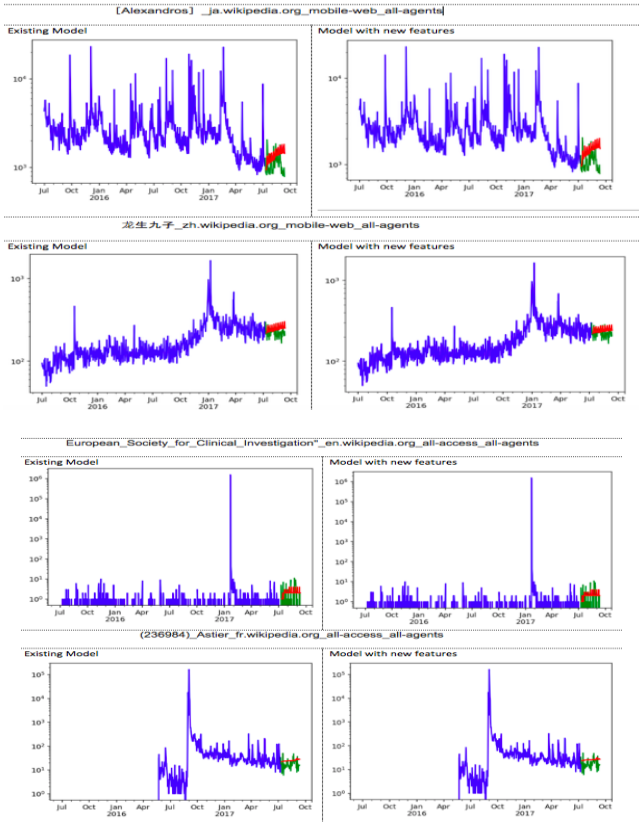


Figure 4: Predictions of Existing Model vs. New Model for 740 days as training set (blue – trained values; green – expected values; red – actual values)

IV. CONCLUSION

In the process of building prediction model, we have successfully rebuilt the existing model and added new features to observe improvements in efficiency of model. Applied new features in different combinations 1) median of specified window length in each time series as an independent features for capturing weekly, monthly, quarterly and yearly page popularity 2) median of medians of variable time frame windows based on golden ratio as in [6][7]. We analyzed the obtained result and compared the accuracies in different cases, to know the importance of each feature. As a next step, we will try to work on how to tune parameters in existing model to observe better results.

V. REFERENCES

[1] https://en.wikipedia.org/wiki/Web_traffic (Last accessed March 3, 2018)

[2] Shuping Yao, Changzhen Hu and Mingqian Sun, "Prediction of Web Traffic Based on Wavelet and Neural Network," *2006 6th World Congress on Intelligent Control and Automation*, Dalian, 2006, pp. 4026-4028.

[3] Meimei Chen, "Short-term forecasting model of web traffic based on genetic algorithm and neural network," 2011 2nd International Conference on Artificial Intelligence, Management Science and Electronic Commerce (AIMSEC), Deng Leng, 2011, pp. 623-626.

[4] https://www.kaggle.com/c/web-traffic-time-series- forecasting (Last accessed November 17, 2018)

[5] https://en.wikipedia.org/wiki/Golden_ratio(Last accessed November 17, 2018)

[6] H. Li, D. Xiao and X. Zhao, "Trend extraction based on variable time window length median filter," 2010 Conference on Control and Fault-Tolerant Systems (SysTol), Nice, 2010, pp. 63-69.

[7] https://www.kaggle.com/safavieh/median-estimation-by-fibonacci-et-al-lb-44-9(Last accessed November 3rd, 2018)

[8] https://www.kaggle.com/c/web-traffic-time-series-forecasting/discussion/39367(Last accessed October 5, 2018)

[9] A. Sagoolmuang and K. Sinapiromsaran, "Median-difference window subseries score for contextual anomaly on time series," 2017 8th International Conference of Information and Communication Technology for Embedded Systems, Chonburi, 2017, pp. 1-6.

[10] https://www.investopedia.com/ask/answers/122314/what-exponential-moving-average-ema-formula-and-how-ema-calculated.asp (Last accessed November 17, 2018)

[11] https://en.wikipedia.org/wiki/Symmetric_mean_absolute_percentage_error (Last accessed November 17, 2018)

[12] http://www.wildml.com/2015/09/recurrent-neural-networks-tutorial-part-1-introduction-to-rnns/ (Last accessed November 3rd, 2018)