SURVEY PAPER

WEB TRAFFIC FORECASTING

Rutul Patel
19IT107
IT-Department,

C.S.P.I.T.,Charusat University,
Changa, Anand.
19it107@charusat.edu.in

Shachi Patel
19IT108
IT-Department,

C.S.P.I.T., Charusat University,
Changa, Anand
19it108@charusat.edu.in

Abstract**—**With the introduction of the internet, online usage has expanded, making it important to forecast traffic on web pages in order to control web server loads. One of the most difficult tasks is predicting future traffic on various web sites. Web traffic prediction may be used to support online businesses in a variety of ways. You may prepare for load balancing to be set up on the cloud or server of web sites, understand user behavior, effectively promote items on pages with large visits based on user interests, and spot anomalies, for example. You can also understand user behavior. Forecasting web traffic presents a significant challenge since it might impair the operation of important websites. Research on time-series forecasting has been very active. One of the most challenging issues in the field is predicting time series values in the future. The time series discipline covers a wide range of topics, including inference, analysis, forecasting, and classification. In this paper we explained the two existing models of past which are ARIMA (Autoregressive Integrated Moving Average) and other one is LSTM RNN (Long Short Term Memory). Also, we described boosting algorithm for LSTM RNN named Adaboost.

I. INTRODUCTION

With the rapid development of internet, traffic congestion in the websites has increased. The increase in traffic for the websites could cause a lot of problems such as crashed sites or slow buffering. This congestion could cause a lot inconveniences for the users. As a result of that it could decrease the user's ratings of the site, leading it to the depletion of the business. Therefore, the network performance needs to be monitored so that relevant prediction can be made. Analysis and forecasting web traffic is helpful for several business operations. Moreover, it can solve many problems in different domains ranging from finance, dynamic systems control and marketing. As internet traffic data is similar to time series data, algorithms or models of time series data can be applied on it. In time series prediction domain, many forecasting methods have been proposed, which can be classified into two kinds: linear prediction and non-linear prediction. Linear forecasting models include ARIMA, ARIMA and HoltWinters Algorithm whereas, forecasting focused on recurring neural networks is commonly used for nonlinear prediction. In this survey paper, we are planning to explain a linear-ARIMA and a non-linear-LSTM RNN prediction algorithm. Also, a boosting algorithm-AdaBoost for LSTM will be described.

II. LITERATURE SURVEY

A. Web Traffic

Web traffic is a way to measure the visits of an online page per a particular time session. It is determined from the number of visitors and the number of pages they visit. Web traffic data represents the amount of data sent and received by visitors to a website.

B. Time Series Data

The set of data which is noted in equal time portion is called Time series data. It is also referred as various-time data. Time series forecasting is used to predict future values based on value observed in past.

III. METHODOLOGIES

1. ARIMA Model

The autoregressive integrated moving average (ARIMA) model is a generalization of the straightforward autoregressive moving average (ARMA) model. Future time-series data points are forecasted or predicted using these two models. The strength of a dependent variable in relation to other changing factors is demonstrated through regression analysis in the form of ARIMA.

By concentrating on differences between series values rather than actual values, the model ultimately aims to estimate future time series movement. ARIMA models are employed when there are indications of non-stationarity in the data. In time series analysis, non-stationary data are always transformed into stationary data.

The trend and the seasonal components are the most frequent reasons why time series data are non-stationary. Applying the differencing step is how non-stationary data is converted to stationary. To remove the trend component from the data, one or more times of differencing steps may be used. Similar to this, seasonal differencing could be used to eliminate the seasonal components from data.

We may break down the model into smaller parts as follows based on the name [7]:

AR: an Auto - regressive model that simulates a certain kind of random process. The model's output is linearly related to its own prior value, or the amount of lagged data points or previous observations.

MA: a moving average model, whose results rely linearly on the most recent observation and a number of prior ones of a stochastic term.

I: the term "integrated" refers to the process of differencing to produce stationary time series data, which involves removing the seasonal and trend components.

- MODEL

A statistical analysis method called ARIMA (Auto regressive Integrated Moving Average model) uses time series data to more fully comprehend or predict future trends. The strength of one dependent variable in relation to other fluctuating variables is assessed using a regression analysis technique called the autoregressive integrated moving average model [9]. The model looks at the differences between values in a series rather than actual values in order to predict future movements in securities or financial markets.The full model can be written as,

$$\hat{y}_t = \mu + \phi_1 y_{t-1} + \ldots + \phi_p y_{t-p} - \theta_1 e_{t-1} - \ldots - \theta_q e_{t-q}$$

Where, yt' is the differences series (it may have been differences more than once).

The "predictors" on the right-hand side include both lagged values of yt and lagged errors.

Standard notation for the sort of ARIMA model employed would be ARIMA with p, d, and q, where integer values are used in place of the parameters. [9].

The parameters can be defined as:

1) p - order of the auto regressive part
2) d- degree of first difference involved
3) q - order of the moving average part

2. RNN

We start by introducing RNNs because Long Short-Term Memory Networks (LSTMs) are a specific type of RNN. Recurrent neural networks (RNNs) get their name from the way connections between units form cycles. The results of earlier calculations may also have an impact. The standard RNN's architecture is depicted in the figure below.

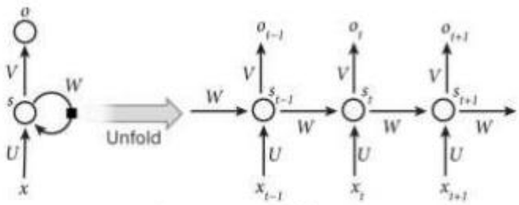

Figure 1: RNN Architecture

The formulae seen in Figure 1 are listed below [6]: At time step t, ot stands for the input, hidden state, and output, respectively. st is calculated asst = f(Uxt + Wst-1), where f is a nonlinear function like tanh. As ot = soft(Vst), ot is calculated.

A series of recurring neural network modules makes up every recurrent neural network. We provide Figure 2, which shows the repeating module of typical RNNs below 12, to assist you in intuitively comprehending LSTMs. [6]. It demonstrates how basic the repeating module's structure is.

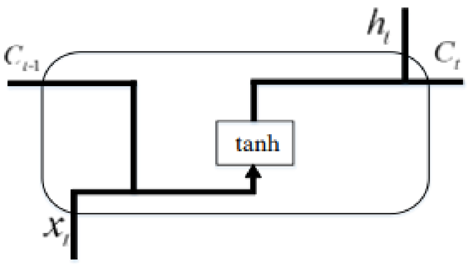

Figure 2: The repeating module of RNNs.

3. LSTM RNN

Following a quick overview of RNNs, the idea of LSTMs is explained [6]. Sepp Hochreiter and Jürgen Schmidhuber first proposed the Long Short-Term Memory (LSTM), which Felix Gers et al. enhanced in 2000. It performs very well in a number of different fields, including classification issues and even time series prediction issues.

Standard Recurrent Neural Networks (RNNs) have issues with gradient points that vanish and explode. To solve these issues, LSTMs were created. Input and forget gates, which provide users more control over the gradient flow and improved long-range dependency preservation, are novel gates that LSTMs use to address these issues.

The repeating module in LSTMs is analogous to that in RNNs. The repeating module, however, is distinct from RNNs. The repeating module's construction is intricate. In contrast to a single straightforward tanh layer, LSTMs have four neural network layers: input, gate, cell, and output. Figure 3 depicts the repeating module as follows

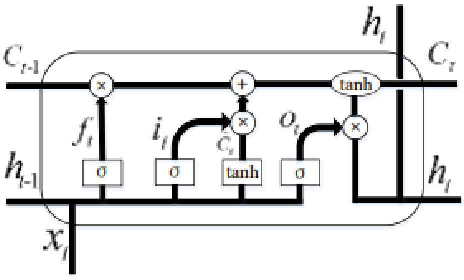

Figure 3. The repeating module of LSTMs.

An LSTM network computes an output sequence h = (h1, h2,..., hT) for a given input sequence x = (x1, x2,..., xT). The following equations may be used to produce the current output ht and the current cell state, which will then be utilised in the subsequent iteration. We can use the previous output ht1, the last cell state Ct1, and the current input xt as inputs. The following equations are used to repeatedly calculate the output sequence h = (h1, h2,..., hT) from t = 1 to T. [6].

$$i_t=\sigma(W_i \cdot [h_{t-1},x_t]+b_i) \quad (1)$$

$$f_t=\sigma(W_f \cdot [h_{t-1},x_t] +b_f) \quad (2)$$

$$\tilde{C}_t=tanh(W_c \cdot [h_{t-1},x_t]+ b_c) \quad (3)$$

$$C_t=f_t * C_{t-1}+ i_t * \tilde{C}_t \quad (4)$$

$$o_t=\sigma(W_o[h_{t-1},x_t]+b_o) \quad (5)$$

$$h_t=o_t * tanh(C_t) \quad (6)$$

The W terms in this context stand for weight matrices; for instance, Wi denotes the matrix of weights from the input gate to the input; is a sigmoid function; and the b terms stand for bias vectors. The input gate is represented by I the forget gate by f, the output gate by o, the cell state by C, and the output by ht.

IV. Comparative Study

The dataset was analyzed and the sample data used was 'India' [2]. The dataset was additionally split into training and testing sets. For the time series, we plotted the number of hits per day for the article "India" together with actual data and projections. The Time Interval is represented on the x-axis, while the Page Visits are represented on the y-axis in powers of 10 [2]. According to the suggested methodologies, the monthly forecast results for languages are as follows:

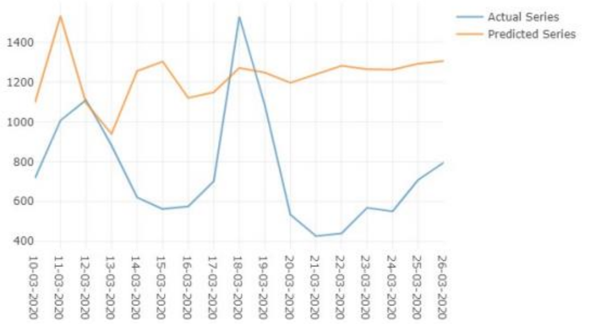

Figure 4: Forecast result for ARIMA

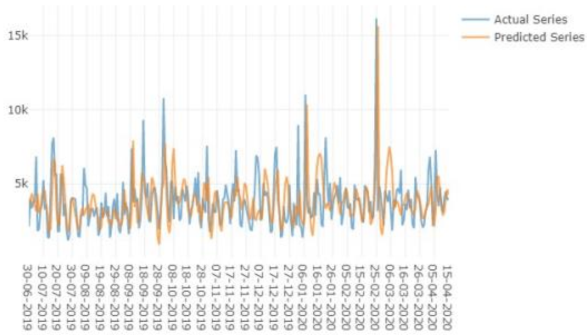

Figure 5: Forecast result for LSTM RNN

The forecast outcomes for the ARIMA model are shown in Figure 4. Over the time span, there is a noticeable tendency, and the forecast confirms this. Figure 5 displays the RNN forecast results and a pattern that successfully recognizes spikes.

V. Boosted LSTM

- AdaBoost

Although there are various boosting algorithms, AdaBoost is the one that is most frequently employed. AdaBoost is a technique that turns a large number of weak learners into a powerful classifier. The combined output of weak learners is used to calculate the final output. The performance of many additional learning algorithms can be enhanced by combining them into a single model using the AdaBoost method [6]. When every weak learner agrees that the final model performs better than random guessing, it is considered a strong model. Next, the AdaBoost algorithm's specifics will be covered.

- Modified AdaBoost

Prediction problems are typically more challenging to resolve using boosting techniques than classification ones. While predictors' outputs are real value rather than classes, classifiers' outputs are classes. The original AdaBoost method is ineffective for problems requiring prediction [6]. AdaBoost algorithm modifications were presented in order to address time series forecasting issues. Figure 6 depicts the improved AdaBoost technique we employ for prediction in this paper.

Given: $(x_i, y_i) \quad x_i \in X, X = \{x_i, i = 1,\ldots,n\}, y_i \in \{-1,+1\}$
Initialize: $D_1(i) = 1/n$
Iterate: For k=1,...,K:

1. Train weak learner using distribution D_k
2. Get hypothesis, h_k, with the error function ε_k, with respect to D_k
$$\varepsilon_k = \Pr_{i \sim D_k}[h_k(x_i) \neq y_i]$$
3. Choose $a_k = \frac{1}{2}\ln(\frac{1-\varepsilon_k}{\varepsilon_k})$
4. Reweight $D_{k+1}(i)$
$$D_{k+1}(i) = \frac{D_k(i)\exp(-a_k y_i h_k(x_i))}{Z_k}$$
Where Z_k is a normalization factor.
$$Z_k = \sum_{i=1}^{n} D_k(i)\exp(-a_k y_i h_k(x_i))$$

Output: The final hypothesis
$$H(x) = sign(\sum_{k=1}^{K}(a_k h_k(x))$$

Figure 6: The AdaBoost method

Given: $(x_i, y_i) \quad x_i \in X, X = \{x_i, i = 1,\ldots,n\}; y_i \in Y, Y = \{y_i, i = 1,\ldots,n\}$
Initialize: $D_1(i) = 1/n$
Iterate: For k=1,...,K:

1. Train weak learner using distribution D_k
2. Get hypothesis, h_k, with the error function ε_k, in regard to D_k .Error function ε_k can be the difference between the actual and predicted values or other error functions which is then weighted by D_k
3. Choose $a_k = \frac{1}{2}\ln(\frac{1-\varepsilon_k}{\varepsilon_k})$
4. Reweight $D_{k+1}(i)$
$$D_{k+1}(i) = \frac{D_k(i)\exp(a_k \varepsilon_k(i)/\varepsilon_k)}{Z_k}$$
Where Z_k is a normalization factor.
$$Z_k = \sum_{i=1}^{n} D_k(i)\exp(a_k \varepsilon_k(i)/\varepsilon_k)$$

Output: The final hypothesis
$$H(x) = \sum_{k=1}^{K}(a_k h_k(x)) / \sum_{k=1}^{K} a_k$$

Figure 7: The modified AdaBoost method

VI. CONCLUSION

Web traffic time series may be predicted more accurately and successfully using a Long Short Term Memory Recurrent Neural Network with Autoregressive Integrated Moving Average. How many people will visit the website in the future may be predicted. Since our system uses LSTM RNN, it is more efficient.

VII. REFERNCES

[1] Petluri, Navyasree & Al-Masri, Eyhab. (2018). Web Traffic Prediction of Wikipedia Pages. 5427-5429. 10.1109/BigData.2018.8622207.

[2] Web Traffic Time Series Forecasting using ARIMA and LSTM RNN Tejas Shelatkar, Stephen Tondale, Swaraj Yadav, Sheetal Ahir ITM Web Conf. 320 3017 (2020) DOI: 10.1051/itmconf/20203203017

[3] Oliveira, Tiago & Barbar, Jamil & Soares, Alexsandro. (2016). Computer network traffic prediction: A comparison between traditional and deep learning neural networks. International Journal of Big Data Intelligence. 3. 28. 10.1504/IJBDI.2016.073903.

[4] Casado-Vara R, Martin del Rey A, Pérez-Palau D, de-la-Fuente-Valentín L, Corchado JM. Web Traffic Time Series Forecasting Using LSTM Neural Networks with Distributed Asynchronous Training. *Mathematics*. 2021; 9(4):421.

[5] S. Mehrmolaei and M. R. Keyvanpour, "Time series forecasting using improved ARIMA," 2016 Artificial Intelligence and Robotics (IRANOPEN), 2016, pp. 92-97, doi: 10.1109/RIOS.2016.7529496.

[6] Bian, Guangkuo & Liu, Jun & Lin, Wenhui. (2018). Internet Traffic Forecasting using Boosting LSTM Method. DEStech Transactions on Computer Science and Engineering. 10.12783/dtcse/csae2017/17517.

[7] Time Series Analysis using Arima Model - Analytics Vidhya

[8] https://machinelearningmastery.com/encoder-decoder-long-short-term-memory-networks/

[9] https://www.ijraset.com/best-journal/web-traffic-time-series-forecasting-using-arima-model