

VISVESVARAYA TECHNOLOGICAL UNIVERSITY

“JNANA SANGAMA” BELAGAVI-590 018, KARNATAKA



PROJECT REPORT

ON

“WEB TRAFFIC TIME SERIES FORECASTING USING MACHINE LEARNING ”

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENT
FOR THE AWARD OF THE DEGREE,

BACHELOR OF ENGINEERING
IN
COMPUTER SCIENCE & ENGINEERING

Submitted By

1. ASHWINI

[1CG17CS010]

2. ASSIYA MUSKAN

[1CG17CS011]

3. CHINMAYEE R

[1CG17CS018]

4. NAYANA R S

[1CG17CS062]

Under the guidance of:

Mr Girish L

Asst. Proffessor Dept. of CSE,
CIT, Gubbi, Tumakuru.

HOD:

Dr. Shantala C P, Ph.D

Prof & Head, Dept. of CSE,
CIT, Gubbi, Tumakuru.



Partnering in Academic Excellence

Channabasaveshwara Institute of Technology

(NAAC Accredited & ISO 9001:2015 Certified Institution)

NH 206 (B.H. Road), Gubbi, Tumakuru – 572 216. Karnataka.



(Affiliated to Visvesvaraya Technological University, Belagavi & Recognized by AICTE New Delhi)

2020-21



Partnering in Academic Excellence

Channabasaveshwara Institute of Technology

(NAAC Accredited & ISO 9001:2015 Certified Institution)

NH 206 (B.H. Road), Gubbi, Tumakuru – 572 216. Karnataka.



(Affiliated to Visvesvaraya Technological University, Belagavi & Recognized by AICTE New Delhi)

2020-21

DEPARTMENT OF COMPUTER SCIENCE ENGINEERING

CERTIFICATE

This is to certify that the project work entitled “**WEB TRAFFIC TIME SERIES FORECASTING USING MACHINE LEARNING**” has been successfully carried out by, **ASHWINI[1CG17CS010]**, **ASSIYA MUSKAN[1CG17CS011]**, **CHINMAYEE R [1CG17CS018]** , **NAYANA R S[1CG17CS062]** students of **CHANNABASAVESHWARA INSTITUTE OF TECHNOLOGY, GUBBI, TUMAKURU**, under our supervision and guidance and submitted in partial fulfillment of the requirements for the award of Degree in **Bachelor of Engineering** by **Visvesvaraya Technological University, Belagavi** during the academic year of 2020–21. It is certified that all corrections/suggestions indicated for internal assessment have been incorporated in the report deposited in the departmental library. The project report has been approved as it satisfies the academic requirements for the above said degree.

Guide:

Mr Girish L

Asst. Professor, Dept. of CSE,
CIT, Gubbi, Tumakuru.

H.O.D:

Dr. Shantala C P, Ph.D

Prof & Head, Dept. of CSE,
CIT, Gubbi, Tumakuru.

Principal:

Dr. Suresh D S Ph.D

CIT, Gubbi, Tumakuru.

Examiners:

1.

2.



Partnering in Academic Excellence

Channabasaveshwara Institute of Technology

(NAAC Accredited & ISO 9001:2015 Certified Institution)

NH 206 (B.H. Road), Gubbi, Tumakuru – 572 216. Karnataka.



(Affiliated to Visvesvaraya Technological University, Belagavi & Recognized by AICTE New Delhi)

2020-2021

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

UNDERTAKING

We the students **ASHWINI [1CG17CS010]** ,**ASSIYA MUSKAN [1CG17CS011]** ,
CHINMAYEE R [1CG17CS018] , **NAYANA R S[1CG17CS062]** of **VIII semester B.E. Computer Science and Engineering** of **CHANNABASAVESHWARA INSTITUTE OF TECHNOLOGY, GUBBI, TUMAKURU** declare that Project work entitled **“WEB TRAFFIC TIME SERIES FORECASTING USING MACHINE LEARNING”** has been carried out and submitted in partial fulfillment of the requirements for the award of degree in Bachelor of Engineering in **Computer Science and Engineering** by the Visvesvaraya Technological University during the academic year 2020-2021.

1. ASHWINI

[1CG17CS010]

3. CHINMAYEE R

[1CG17CS018]

2. ASSIYA MUSKAN

[1CG17CS011]

4. NAYANA R S

[1CG17CS062]



Partnering in Academic Excellence

Channabasaveshwara Institute of Technology

(NAAC Accredited & ISO 9001:2015 Certified Institution)

NH 206 (B.H. Road), Gubbi, Tumakuru – 572 216. Karnataka.



(Affiliated to Visvesvaraya Technological University, Belagavi & Recognized by AICTE New Delhi)

2020-21

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that the Project work entitled “**WEB TRAFFIC TIME SERIES FORECASTING USING MACHINE LEARNING**” is a bonafide work of **ASHWINI [1CG17CS010], ASSIYA MUSKAN [1CG17CS011], CHINMAYEE R [1CG17CS018] , NAYANA R S [1CG17CS062]** students of **VIII** semester B.E computer Science and Engineering carried out at **Channabasaveshwara Institute of Technology**, Gubbi, Tumakuru, in partial fulfillment of the requirements of the award of degree in B.E. in **Computer Science and Engineering** of Visvesvaraya Technological University, Belagavi under my supervision and guidance. Certified that to the best of my knowledge the work reported here in does not form part of any other thesis on the basis of which degree or award was conferred on earlier occasion to this or any other candidates.

Guide:

Mr Girish L

Asst. Professor,

Dept. of CSE,

CIT, Gubbi, Tumakuru.

ACKNOWLEDGEMENT

A great deal of time and lot of effort has gone into completing this project report and documenting it. The number of hours spent in getting through various books and other materials related to this topic chosen by us have reaffirmed its power and utility in doing this project.

Several special people have contributed significantly to this effort. First of all, we are grateful to our institution “**Channabasaveshwara Institute of Technology**”, Gubbi which provided us an opportunity in fulfilling our most cherished desire of reaching the goal.

We acknowledge and express our sincere thanks to the beloved Director and Principal **Dr. Suresh D S** for his many valuable suggestions and continued encouragement and support in the academic endeavors.

We wish to express our deep sense of gratitude to our guide **Mr Girish L**, Asst. Professor Department of Computer Science and Engineering for all the guidance and who still remains a constant driving force and motivated through innovative ideas with tireless support and advice during the project to examine and helpful suggestions offered.

This would never been possible without the support and technical supervision by all the faculty members of CITRIS for all their guidance.

We would express our gratitude towards our parents and friends for their kind cooperation and encouragement which helped us in completion of this project.

Finally, we would like to thank all the teaching and non-teaching staff of Dept of CSE, for their cooperation.

Thanking everyone....

Ashwini [1CG17CS010]

Assiya Muskan [1CG17CS011]

Chinmayee R [1CG17CS018]

Nayana R S [1CG17CS062]



INTERNATIONAL JOURNAL OF ADVANCED SCIENTIFIC INNOVATION

Website: www.ijasi.org, Email: journalijasi@gmail.com

CERTIFICATE OF PUBLICATION

CERTIFIES THAT

Nayana R S

Published a Paper Entitled

Web Traffic Time Series Forecasting using Machine Learning

in International Journal of Advanced Scientific Innovation,
Volume 2, Issue 2, July 2021

ISSN: 2582-8436


Editor in Chief, IJASI



INTERNATIONAL JOURNAL OF ADVANCED SCIENTIFIC INNOVATION

Website: www.ijasi.org, Email: journalijasi@gmail.com

CERTIFICATE OF PUBLICATION

CERTIFIES THAT

Ashwini

Published a Paper Entitled

Web Traffic Time Series Forecasting using Machine Learning

in International Journal of Advanced Scientific Innovation,
Volume 2, Issue 2, July 2021

ISSN: 2582-8436


Editor in Chief, IJASI



INTERNATIONAL JOURNAL OF ADVANCED SCIENTIFIC INNOVATION

Website: www.ijasi.org, Email: journalijasi@gmail.com

CERTIFICATE OF PUBLICATION

CERTIFIES THAT

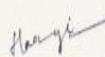
Assiya Muskan

Published a Paper Entitled

Web Traffic Time Series Forecasting using Machine Learning

in International Journal of Advanced Scientific Innovation,
Volume 2, Issue 2, July 2021

ISSN: 2582-8436


Editor in Chief, IJASI



INTERNATIONAL JOURNAL OF ADVANCED SCIENTIFIC INNOVATION

Website: www.ijasi.org, Email: journalijasi@gmail.com

CERTIFICATE OF PUBLICATION

CERTIFIES THAT

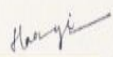
Chinmayee R

Published a Paper Entitled

Web Traffic Time Series Forecasting using Machine Learning

in International Journal of Advanced Scientific Innovation,
Volume 2, Issue 2, July 2021

ISSN: 2582-8436


Editor in Chief, IJASI



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.
- (c) Understanding the user behavior.

To achieve the goals of this research work, we built a time-series model that utilizes LSTM model. We then investigate the use of symmetric mean absolute percentage error 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.

TABLE OF CONENTS

Contents	Page no
1. INTRODUCTION	10-12
1.1 Objective	11
1.2 Problem Statement	11
1.3 Scope of the Project	11-12
2. LITERATURE SURVEY	13-22
2.1 Web Traffic Time Series Dataset	13-14
2.2 Machine Learning	15-16
2.3 Deep Learning	16-17
2.4 Related Work	17-22
3. SYSTEM ANALYSIS	23-25
3.1 Existing System	23
3.2 Proposed System	24
3.3 DataSet	25
4. SYSTEM DESIGN	26-27
Fig 4.1 Forecasting based on ARIMA Model and LSTM model	26
Fig 4.2 Working Flow of ARIMA Model	27
Fig 4.3 Working Flow of LSTM Model	27
5. Implementation	28 - 31
6. Result	32 – 36
Fig 6.1 Predicting website page view	32
Fig 6.2 Predicting another website page view	32-33
Fig 6.3 Plotting page in different language	33

Fig 6.4	Plotting graph on Main_page_enn_wikipedia using ARIMA Model	34
Fig 6.5	Plotting graph on Wikipedia_Haupteseite_ de_wikipedia.org using ARIMA Model	34
Fig 6.6	Plotting graph on Special_search_commons_ Wikipedia.org using ARIMA Model	35
Fig 6.7	Plotting graph on Wikipedia_Aceueil_principal_ fr.wikipedia.org using ARIMA Model	35
Fig 6.8	Plotting graph on Wikipedia_protada_es.wikipedia .org using ARIMA Model	36
7.	CONCLUSION	37
8.	REFERENCE	38

CHAPTER 1

INTRODUCTION

Recently, more and more people are getting access to the internet all over the world, the rise in traffic for almost all websites are inevitable. The increase in traffic for the websites could cause a lot of problems and the company which manages to cope with the traffic changes in the most efficient way is going to succeed. As most of the people may have encountered a crashed site or very slow loading time for a website when there are a lot of people using it, like when various shopping websites may crash just before festivals as more people try to log into the website than it was originally capable of which causes a lot of inconveniences for the users and as a result of that it could decrease the user's ratings of the site and instead use another site, therefore, reducing their business. Therefore, a traffic management technique or plan should be put in place to reduce the risk of such mishaps which could be detrimental to the existence of the company. Until recently, there wasn't a need for such tools as most servers could handle the traffic influx but the smart phone age has increased the demand to such a high level for some websites that companies could not have reacted quickly enough to maintain their regular customer service level.

Evaluating web traffic on a web server is highly critical for web service providers since, without a proper demand forecast, customers could have lengthy waiting times and abandon that website. However, this is a challenging task since it requires making reliable predictions based on the arbitrary nature of human behavior. We introduce an architecture that collects source data and in a supervised way performs the forecasting of the time series of the page views.

The dataset is processed and the features and hidden patterns in data are obtained for later designing an advanced version of a recurrent neural network called Long Short-Term Memory. In addition, the improvement of the accuracy of the model with the distributed training is remarkable. Since the task of predicting web traffic in as precise quantities as possible requires large datasets, we designed a forecasting system to be accurate despite having limited data in the dataset.

Many methods have been proposed for forecasting web traffic. They can be classified broadly into two groups based on the analysed models: nonlinear prediction and linear prediction. The most widely used linear forecast models are:

- i) AR Model
- ii) MA Model.

The forecasting focused on recurring neural networks is commonly used for nonlinear prediction. Discrete wavelet transform (DWT) divides the data into linear and non-linear components that help improve forecast accuracy

1.1 OBJECTIVES

The goal of the project is to minimize difference between actual and predicted values.

There are no particular latency requirements, but we should try that it should don't take hours to predict for a particular date. Up to 20–30 seconds should be acceptable.

1.2 PROBLEM STATEMENT

Time series can come handy in many problems like analysis, classification and most important forecasting, in this case study we will be focusing on analysis and forecasting. This case study focuses on predicting future values for multiple time series problem. Each time series contains daily traffic on Wikipedia page for a total of 803 days from 2015-07-01 to 2017-09-10. We have a total of 145k time series which means we have data for 145k pages, our goal is to analyze this data, build a model on it and predict future traffic on each of the page for 62 days from 2017-09-13 to 2017-11-13.

1.3 SCOPE

Trend → a general systematic linear or (most often) nonlinear component that changes over time and does not repeat.

Seasonality → a general systematic linear or (most often) nonlinear component that changes over time and does repeat.

Moving Average → calculation to analyze data points by creating series of averages of

different subsets of the full data set.

Auto Regression → is a representation of a type of random process; as such, it is used to describe certain time-varying processes in nature, economics, etc.

Seasonal ARIMA → seasonal AR and MA terms predict using data values and errors at times with lags that are multiples of S (the span of the seasonality)

CHAPTER 2

LITERATURE SURVEY

Literature Survey in the field of web traffic time series forecasting and also the data-set that has been used for our prediction model.

2.1 Web Traffic Time Series Dataset

Wikipedia's page view API is the data used for this project. That data contains daily page visits as a time series to any post. Latest data is obtained through this API. The data is returned in JSON format. The fields extracted from this data are the recorded Dates and Visits on that date. It converts this data into a data frame and fits into the predictive model.

Web Traffic

Web traffic is the amount of data sent and received by visitors to a website. This amount necessarily does not include the traffic generated by bots. Since the mid-1990s, web traffic has been the largest portion of Internet traffic.[1] This result is determined by the number of visitors and the number of pages they visit. Sites monitor the incoming and outgoing traffic to see which parts or pages of their site are popular and if there are any apparent trends, such as one specific page being viewed mostly by people in a particular country. There are many ways to monitor this traffic, and the gathered data is used to help structure sites, highlight security problems or indicate a potential lack of bandwidth.

Web traffic is measured to see the popularity of websites and individual pages or sections within a site. This can be done by viewing the traffic statistics found in the web server log file, an automatically generated list of all the pages served. A hit is generated when any file is served. The page itself is considered a file, but images are also files, thus a page with 5 images could generate 6 hits

(the 5 images and the page itself).

A page view is generated when a visitor requests any page within the website – a visitor will always generate at least one page view (the main page) but could generate many more. Tracking applications external to the website can record traffic by inserting a small piece of HTML code in every page of the website.

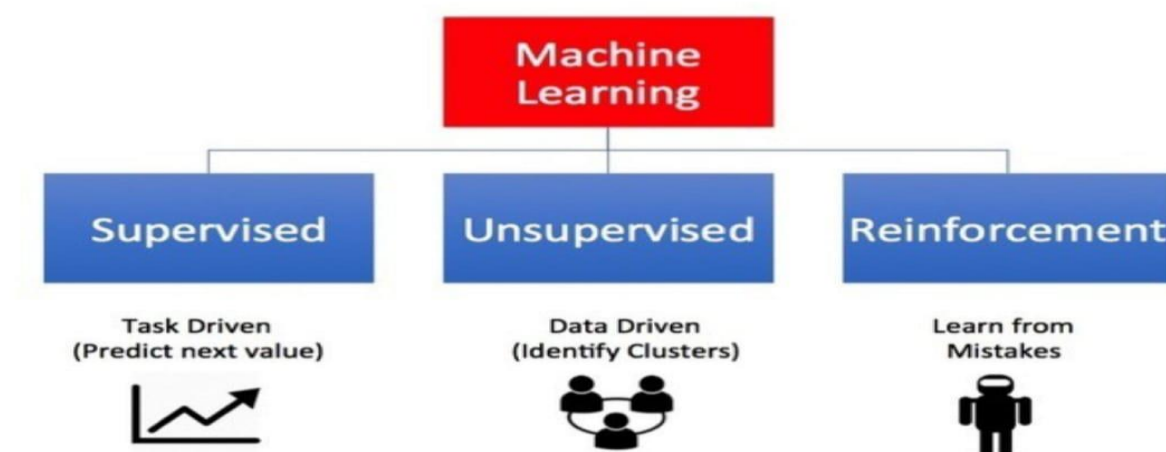
Types of web traffic

1. **Organic traffic:** Organic traffic is something that lands you an website as you search a keyword on a search engine.
2. **Direct traffic:** Direct traffic refers to those who directly type your website in the URL column.
3. **Search traffic:** Search traffic implies answers relating to user searches.
4. **Paid traffic:** Paid traffic refers to the users who enter your websites through your paid advertising campaigns.
5. **Email marketing traffic:** Email marketing traffic is the traffic generated by the links you send marketing emails.
6. **Social media traffic:** Social media traffic brings in more customers through your presence in social media.
7. **Referral Traffic:** Referral traffic means traffic means traffic that occurs on your site.

2.2 MACHINE LEARNING

At a high-level, machine learning is simply the study of teaching a computer program or algorithm how to progressively improve upon a set task that it is given. On the research-side of things, machine learning can be viewed through the lens of theoretical and mathematical modeling of how this process works. However, more practically it is the study of how to build applications that exhibit this iterative improvement. There are many ways to frame this idea, but largely there are three major recognized categories: supervised learning, unsupervised learning, and reinforcement learning.

TYPES OF MACHINE LEARNING



Supervised Learning

Supervised learning is one of the most basic types of machine learning. In this type, the machine learning algorithm is trained on labeled data. Even though the data needs to be labeled accurately for this method to work, supervised learning is extremely powerful when used in the right circumstances.

In supervised learning, the ML algorithm is given a small training dataset to work with. This training dataset is a smaller part of the bigger dataset and serves to give the algorithm a basic idea of the problem, solution, and data points to be dealt with. The training dataset is also very

similar to the final dataset in its characteristics and provides the algorithm with the labeled parameters required for the problem.

Unsupervised Learning

Unsupervised machine learning holds the advantage of being able to work with unlabeled data. This means that human labor is not required to make the dataset machine-readable, allowing much larger datasets to be worked on by the program.

In supervised learning, the labels allow the algorithm to find the exact nature of the relationship between any two data points. However, unsupervised learning does not have labels to work off of, resulting in the creation of hidden structures. Relationships between data points are perceived by the algorithm in an abstract manner, with no input required from human beings.

Reinforcement Learning

Reinforcement learning directly takes inspiration from how human beings learn from data in their lives. It features an algorithm that improves upon itself and learns from new situations using a trial-and-error method. Favorable outputs are encouraged or ‘reinforced’, and non-favorable outputs are discouraged or ‘punished’.

Based on the psychological concept of conditioning, reinforcement learning works by putting the algorithm in a work environment with an interpreter and a reward system. In every-iteration of the algorithm, the output result is given to the interpreter, which decides whether the outcome is favorable or not.

2.3 Deep Learning

Deep learning methods uses lot for time series forecasting, such as the automatic learning of temporal dependence and the automatic handling of temporal structures like trends and seasonality.

Traditionally, time series forecasting has been dominated by linear methods because they are well understood and effective on many simpler forecasting problems.

Deep learning neural networks are able to automatically learn arbitrary complex mappings from inputs to outputs and support multiple inputs and outputs.

Deep Learning Algorithms

You will discover 4 deep learning methods that you can use to develop defensible time series forecasting methods.

MLPs - The classical neural network architecture including how to grid search model hyper parameters.

CNNs - Simple CNN models as well as multi-channel models and advanced multi-headed and multi-output models.

LSTMs - Simple LSTM models, Stacked LSTMs, Bidirectional LSTMs and Encoder-Decoder models for sequence-to-sequence learning.

Hybrids - Hybrids of MLP, CNN and LSTM models such as CNN-LSTMs, Con-LSTMs and more.

2.4 RELATED WORK

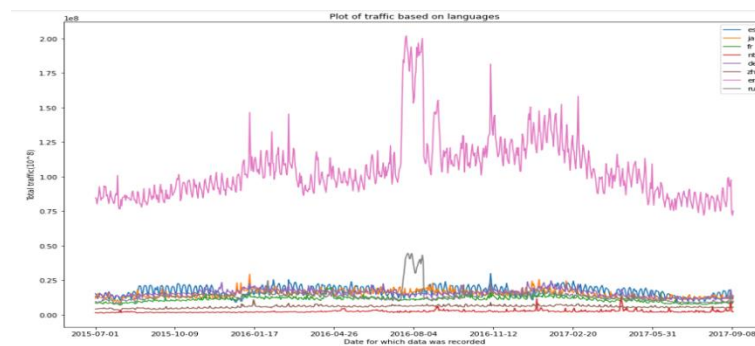
TITLE: Predicting the language of the Wikipedia page(for example, for English en.wikipedia.org, for Spanish es.wikipedia.org,zh,wikipedia.org for Chinese)

PROBLEM:

Two letter words corresponds to different languages:

de-German,en-English,es-Spanish,fr-French,ja-Japanese,ru-Russia,zh-Chinese,nt refers to media pages(Wikimedia)

DESCRIPTION:



As anyone would have expected, English Wikipedia has the largest traffic of all languages but there is a pattern around August 2016 and interestingly Russian Wikipedia has the same pattern during same time. There is also a pattern around January 2016 in English Wikipedia and there some spikes can also be seen in Japanese wiki during the same period.

DATASET:

Train.csv contains about 145k rows each of which represent a different Wikipedia page and it has 804 columns, except the first column each column represent a date and it has daily traffic on that particular Wikipedia page. First column contains the name of the page which includes the language of the Wikipedia page (for example, for English en.wikipedia.org, for Spanish es.wikipedia.org,zh,wikipedia.org for Chinese) +type of access(desktop, all access) .

	Page	2015-07-01	2015-07-02	2015-07-03	2015-07-04	2015-07-05	2015-07-06	2015-07-07	2015-07-08	2015-07-09	2015-07-10	2015-07-11	2015-07-12	2015-07-13	2015-07-14	2015-07-15	2015-07-16	2015-07-17
0	2NE1_zh.wikipedia.org_all-access_spider	18.0	11.0	5.0	13.0	14.0	9.0	9.0	22.0	26.0	24.0	19.0	10.0	14.0	15.0	8.0	16.0	8.0
1	2PM_zh.wikipedia.org_all-access_spider	11.0	14.0	15.0	18.0	11.0	13.0	22.0	11.0	10.0	4.0	41.0	65.0	57.0	38.0	20.0	62.0	44.0

ALGORITHM:

ARIMA MODEL

ARIMA (Auto-Regressive Integrated Moving Average) the model has a huge advantage in univariate time series forecasting. ARIMA model attempts to describe the trends and seasonality in time series as a function of lagged values (Auto Regressive parameter) and Averages changing over time intervals (Moving Averages). The model includes differencing (Integrating) the original time series data. Differencing time-series means forming a new time series by subtracting the previous observation from the current time. The point of this is to remove certain trends, such as seasonality, trends, or inconsistent variance in time series data.

The ARIMA equation has two important components Auto-Regressive (AR) part and the Moving Average (MA) part.

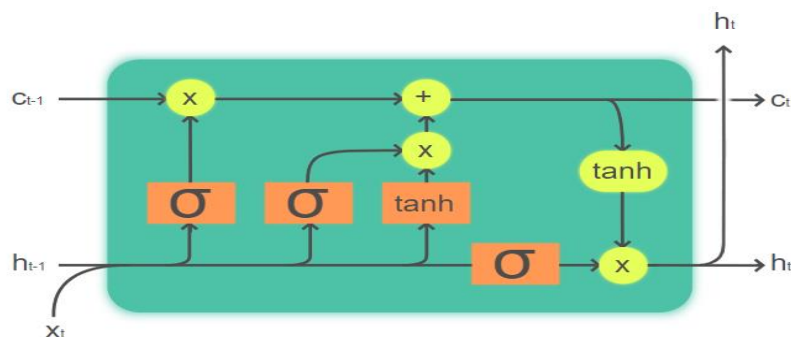
LSTM MODEL

LSTM (Long Short-Term Memory) is a Recurrent Neural Network (RNN) based architecture that is widely used in natural language processing and time series forecasting. Offers a great, intuitive introduction.

The LSTM rectifies a huge issue that recurrent neural networks suffer from: short-memory. Using a series of ‘gates,’ each with its own RNN, the LSTM manages to keep, forget or ignore data points based on a probabilistic model.

LSTMs also help solve exploding and vanishing gradient problems. In simple terms, these problems are a result of repeated weight adjustments as a neural network trains. With repeated epochs, gradients become larger or smaller, and with each adjustment, it becomes easier for the network’s gradients to compound in either direction. This compounding either makes the gradients way too large or way too small. While exploding and vanishing gradients are huge downsides of using traditional RNN’s, LSTM architecture severely mitigates these issues.

After a prediction is made, it is fed back into the model to predict the next value in the sequence. With each prediction, some error is introduced into the model. To avoid exploding gradients, values are ‘squashed’ via (typically) sigmoid & tan activation functions prior to gate entrance & output. Below is a diagram of LSTM architecture



LSTM RNN:

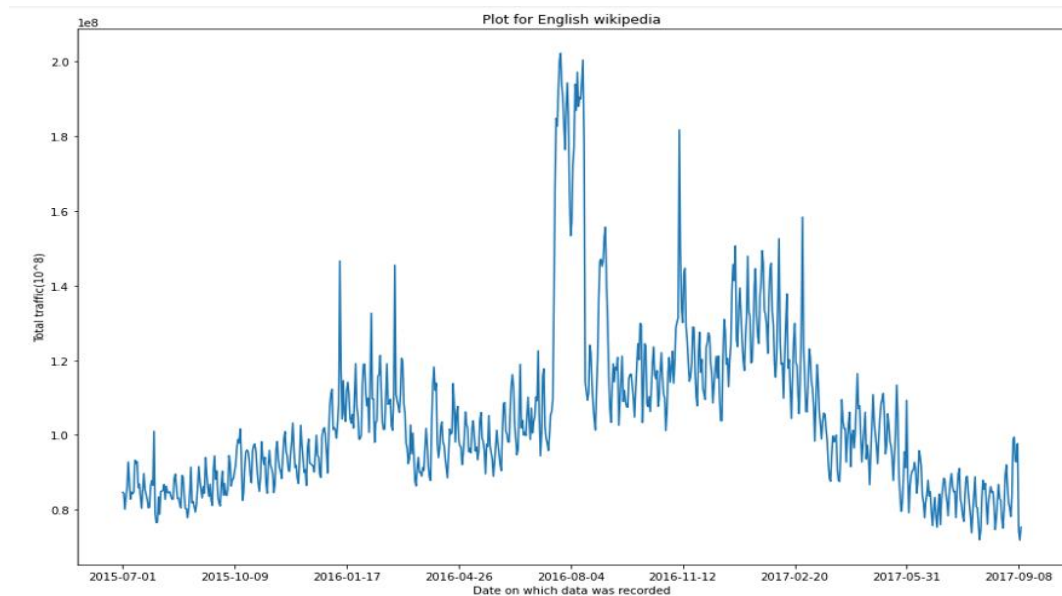
Our proposed procedure utilizes Long Transient Memory (LSTM) RNN. To add a piece of new data to RNN, it totally changes the current data by adding a capacity. Accordingly, the entire data is refreshed, for example there is no regard for 'significant' data and 'not all that significant' data by and large. Both RNNs have the intermittent layer of criticism circles. It permits them to keep data and information in 'memory' over the long haul. Regardless, it very well might be hard to prepare standard RNNs to tackle issues requiring long haul transient conditions to comprehend. LSTM networks are a kind of RNN that utilizes other than standard units, extraordinary units. LSTM frameworks incorporate a 'memory cell' which can hold information in memory for extensive stretches of time. This design assists them with seeing longer-term conditions. GRU's are like LSTMs yet are primarily improved. They likewise utilize a progression of doors to control data stream, yet don't utilize distinctive memory cells, and utilize less entryways. We use LSTM RNN for this impact to have more memory than traditional RNN.

CONCLUSION:

There is difference in traffic based on language of the data.

Now, we will plot individual plot for every language to understand them in deep.

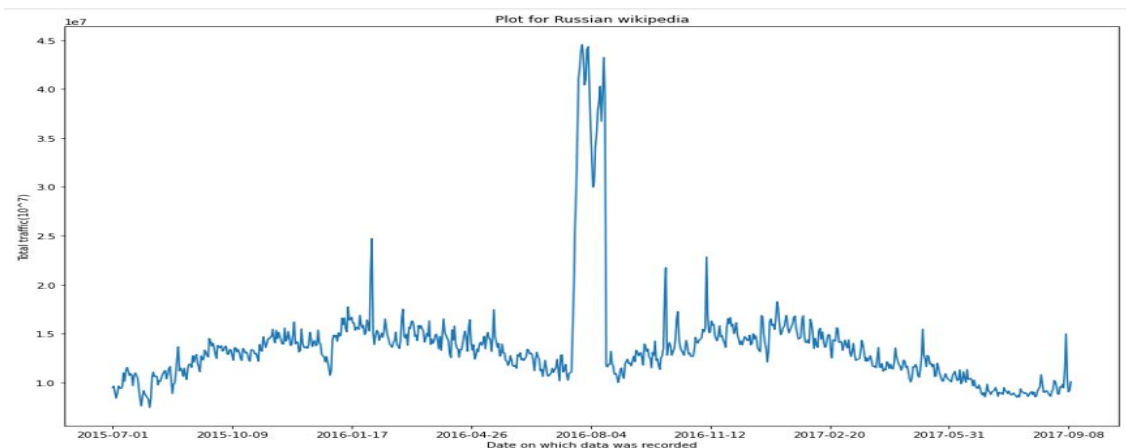
Plot for English Wikipedia-



We can see the nice weekly pattern in the data, regular spikes are there after 7 days and one interesting observation from this plot is that traffic usually goes down during late Q3 or early Q4 and in November 2016, it has a very large spike.

Conclusion- Data has weekly seasonality

Plot for Russian Wikipedia-



Russian Wikipedia does not show a large upward or downward trend but there is a very large spike during Q3 2016. Other than that it has few spikes here and there but not as much as other languages.

Conclusion- This shows that people in Russia doesn't care about weekends and there rate of accessing the Wikipedia pages remains uniform.

CHAPTER 3

SYSTEM ANALYSIS

3.1 EXISTING SYSTEM

Time Series Forecasting is one of the least explored areas and various models are evaluated to improve the accuracy of the forecast. The main focus of the proposal is to predict future web traffic to make decisions for better congestion control. Past Values are considered to predict future values.

ARIMA: Autoregressive Integrated Moving Average

LSTM: Long Short-Term Memory

Factor in Time Series Forecasting

When analyzing a time series, this form of data analysis involves identifying at least three aspects of the data. These factors are autocorrelation, seasonality, and stationarity.

Autocorrelation

In a time series, autocorrelation is the tendency of data observations and patterns to repeat themselves. If these observations and patterns repeat themselves at regular intervals, the result may also be known as seasonality.

Seasonality

As touched on above, seasonality is when observations and patterns repeat themselves at regular intervals. The best example of seasonality would be a graph of temperatures across multiple years. During the summer, temperatures are high; during the winter, temperatures are low.

Stationarity

Stationarity is a measure of how little a time series' mean and variance changes over time. For example, if the temperatures measured across a period of ten years are of similar magnitude and variance — after accounting for the seasonality of the dataset — then the time series would be said to have high stationarity.

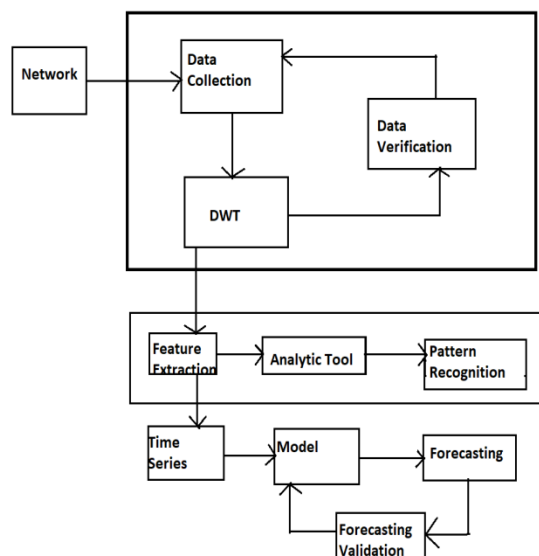
As a more illustrative example of stationarity, consider the effect of global warming on temperatures measured every month. Although the dataset may continue to show signs of autocorrelation and seasonality, the stationarity of the dataset would decrease due to a higher mean temperature and greater variance in temperatures (due to lower and higher extremes).

3.2 PROPOSED SYSTEM

The proposed system for web traffic forecasting. This architecture is modularly designed, distributed and scalable such that with minimal modifications needed it can be easily adapted and used for website traffic predictions, regardless of whether it is a closed computer network or a website. This architecture follows the design pattern from bottom to top, which represent the predication of website.

It is mainly divided into three major layers:

- Data extraction, its transformation by extracting its features and data loading as a time series in the deep learning layer to make predictions. In this study, we focused on the prediction of the Wikipedia web traffic stream. For designing and testing our data analysis and forecasting model of the website traffic page views, we used the dataset 'Wikipedia web traffic'. Even though the Wikipedia web traffic helps us in the design of our model, we created a new version of this dataset since we developed our own wikipedia scrapper in order to build our own Wikipedia's top 1000 page views dataset from all the language available.



3.3 DATASET:

Train.csv contains about 145k rows each of which represent a different Wikipedia page and it has 804 columns, except the first column each column represent a date and it has daily traffic on that particular Wikipedia page. First column contains the name of the page which includes the language of the Wikipedia page(for example, for English en.wikipedia.org, for Spanish es.wikipedia.org,zh.wikipedia.org for Chinese) +type of access(desktop, all access) + agent(spider, actual traffic). For example one name is- 'AKB48_zh.wikipedia.org_all-access_spider'.

Second file is 'key.csv' which has number of rows equal to the number of predictions we have to make. There are two columns in this file- 'page name' and 'id'. For each page name that is present in 'train.csv' file, we have 62 rows present in key.csv file, these 62 rows corresponds to 62 days of predictions for each page.

	Page	2015-07-01	2015-07-02	2015-07-03	2015-07-04	2015-07-05	2015-07-06	2015-07-07	2015-07-08	2015-07-09	2015-07-10	2015-07-11	2015-07-12	2015-07-13	2015-07-14	2015-07-15	2015-07-16	2015-07-17
0	2NE1_zh.wikipedia.org_all-access_spider	18.0	11.0	5.0	13.0	14.0	9.0	9.0	22.0	26.0	24.0	19.0	10.0	14.0	15.0	8.0	16.0	8.0
1	2PM_zh.wikipedia.org_all-access_spider	11.0	14.0	15.0	18.0	11.0	13.0	22.0	11.0	10.0	4.0	41.0	65.0	57.0	38.0	20.0	62.0	44.0

CHAPTER 4

SYSTEM DESIGN

The design of the system deals with how the system is developed. It explains the flow functionalities in brief. The section contains system data flow diagram, Flowchart and Sequence Diagram described below.

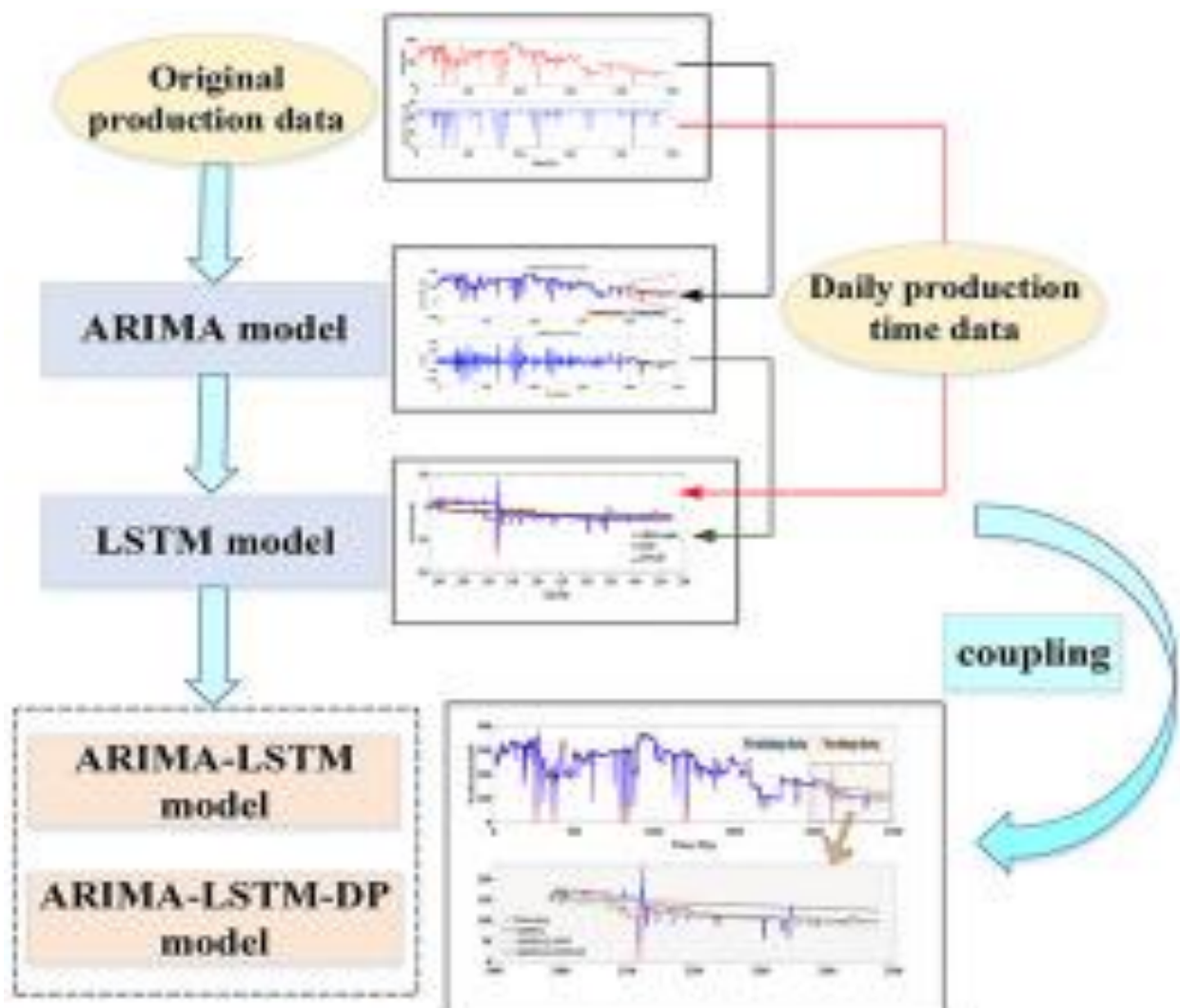


Fig 4.1: Forecasting based on ARIMA and LSTM model

WORKING FLOW DIAGRAM

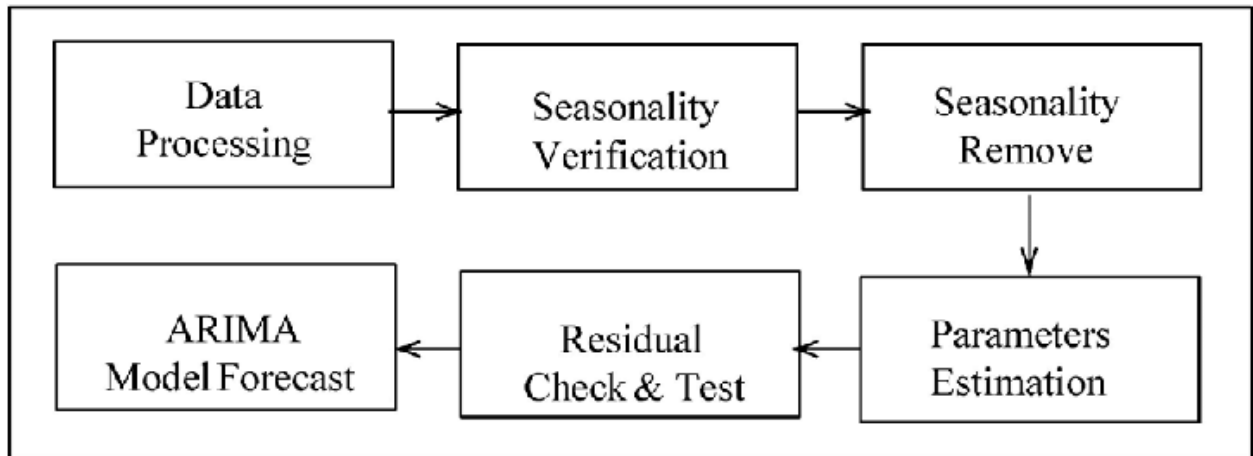


Fig 4.2 : Working flow of ARIMA model

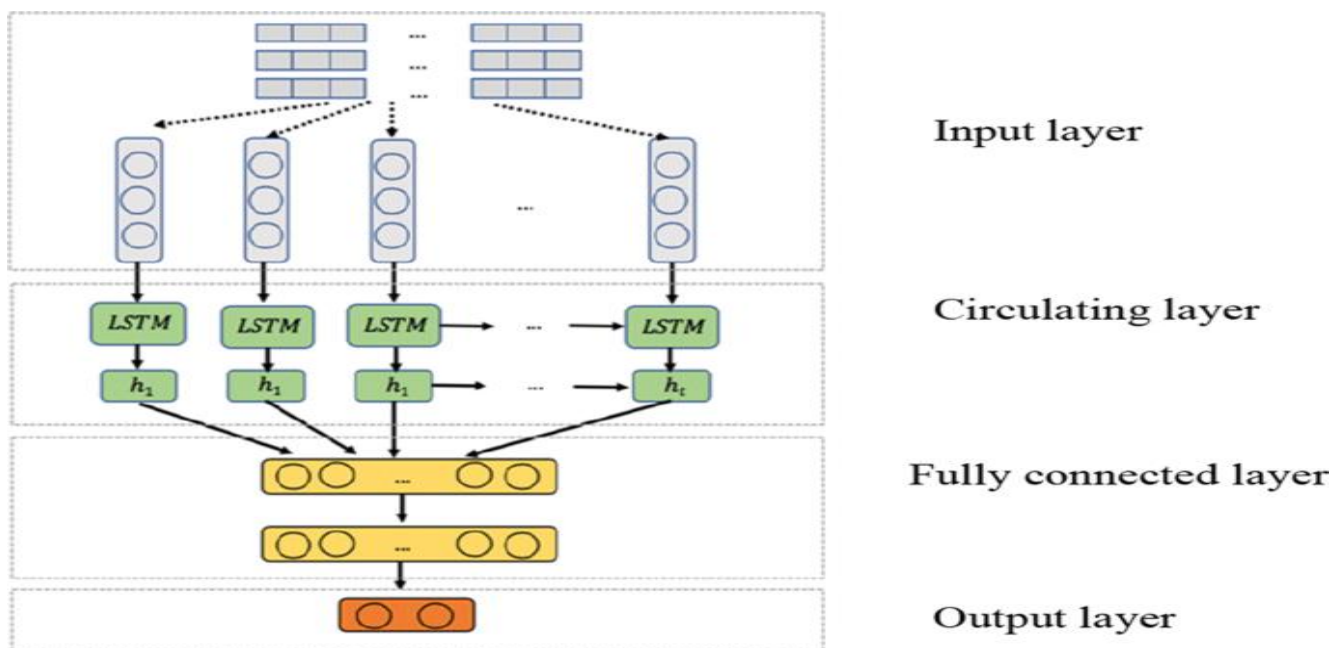


Fig 4.3: Working flow of LSTM model

CHAPTER 5

IMPLEMENTATION

The dataset was divided into training and testing sets. For the time series, we plotted the number of days from start vs. web views along with the real values and forecasts for the ‘web view forecasting’ during the testing period.

The x – axis represent the number of days from start and the y – axis represents the web page views in powers of 100.

❖ Importing Libraries

Numpy as np – It provides support for large multidimensional array objects and various tools to work with them. Various other libraries like Pandas , Matplotlib etc.

Pandas as pd – Pandas is one of the tools in ML which is used for data cleaning and analysis.

Matplotlib as plt – Matplotlib used to plotting libraries in machine learning.

```
In [2]: #Importing Libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

❖ Importing Datasets

Train.csv contains about 145k rows each of which represent a different Wikipedia page and it has 804 columns, except the first column each column represent a date and it has daily traffic on that particular Wikipedia page. First column contains the name of the page which includes the language of the Wikipedia page (for example, for English en.wikipedia.org, for Spanish es.wikipedia.org,zh.wikipedia.org for

Chinese) +type of access(desktop, all access) .

```
In [5]: #Importing dataset
train=pd.read_csv("train_1.csv.zip").fillna(0)
page = train['Page']
train.head()

Out[5]:
```

	Page	2015-07-01	2015-07-02	2015-07-03	2015-07-04	2015-07-05	2015-07-06	2015-07-07	2015-07-08	2015-07-09	...	2016-12-22	2016-12-23	2016-12-24	2016-12-25	2016-12-26	2016-12-27	2016-12-28	2016-12-29
0	2NE1_zh.wikipedia.org_all-access_spider	18.0	11.0	5.0	13.0	14.0	9.0	9.0	22.0	26.0	...	32.0	63.0	15.0	26.0	14.0	20.0	22.0	19.0
1	2PM_zh.wikipedia.org_all-access_spider	11.0	14.0	15.0	18.0	11.0	13.0	22.0	11.0	10.0	...	17.0	42.0	28.0	15.0	9.0	30.0	52.0	45.0
2	3C_zh.wikipedia.org_all-access_spider	1.0	0.0	1.0	1.0	0.0	4.0	0.0	3.0	4.0	...	3.0	1.0	1.0	7.0	4.0	4.0	6.0	3.0
3	4minute_zh.wikipedia.org_all-access_spider	35.0	13.0	10.0	94.0	4.0	26.0	14.0	9.0	11.0	...	32.0	10.0	26.0	27.0	16.0	11.0	17.0	19.0
4	52_Hz_I_Love_You_zh.wikipedia.org_all-access_s...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	48.0	9.0	25.0	13.0	3.0	11.0	27.0	13.0

5 rows × 551 columns

❖ Dropping Page Column

```
In [6]: #Dropping Page Column
train = train.drop('Page',axis = 1)

In [7]: train.iloc[90000,:]

Out[7]:
```

	2015-07-01	2015-07-02	2015-07-03	2015-07-04	2015-07-05	...	2016-12-27	2016-12-28	2016-12-29	2016-12-30	2016-12-31
0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0

Name: 90000, Length: 550, dtype: float64

❖ Using Data From Random Row for Training and Testing

Use the training data to fit the model and testing data to test it. The models generated are to predict the result unknown which is named as test set.

The dataset is divided into train and test set in order to check accuracies, precision by training and testing it on it.

```
In [8]: #Using Data From Random Row for Training and Testing

row = train.iloc[90000,:].values
x = row[0:549]
y = row[1:550]

# Splitting the dataset into the Training set and Test set
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.3, random_state = 0)

print(X_train.shape, y_train.shape)

# Feature Scaling
from sklearn.preprocessing import MinMaxScaler
sc = MinMaxScaler()
X_train = sc.fit_transform(X_train)
y_train = sc.fit_transform(y_train)

print(X_train.shape, y_train.shape)

X_train = sc.fit_transform(X_train)
y_train = sc.fit_transform(y_train)

(384,) (384,)
(384, 1) (384, 1)
```

❖ Importing LSTM Model

LSTM Model (Long Short - Term Memory) networks are a type of Recurrent neural network capable of learning order dependence in sequence prediction problems.

LSTM Model most important in the time series prediction.

Here , installing RNN useful in time series prediction only because of the feature to remember previous inputs as well.

```
In [9]: #Training LSTM

#Reshaping Array
X_train = np.reshape(X_train, (384,1,1))

# Importing the Keras libraries and packages for LSTM
from keras.models import Sequential
from keras.layers import Dense
from keras.layers import LSTM

# Initialising the RNN
model = Sequential()

# Adding the input layer and the LSTM layer
model.add(LSTM(units = 8, activation = 'relu', input_shape = (None, 1)))

# Adding the output layer
model.add(Dense(units = 1))

model.summary()

# Compiling the RNN
model.compile(optimizer = 'adam', loss = 'mean_squared_error')

# Fitting the RNN to the Training set
model.fit(X_train, y_train, batch_size = 10, epochs = 100)

39/39 [=====] - 0s 1ms/step - loss: 0.0022
Epoch 11/100
39/39 [=====] - 0s 1ms/step - loss: 0.0085
Epoch 12/100
39/39 [=====] - 0s 1ms/step - loss: 0.0052
Epoch 13/100
39/39 [=====] - 0s 1ms/step - loss: 0.0058
Epoch 14/100
39/39 [=====] - 0s 1ms/step - loss: 0.0091
Epoch 15/100
39/39 [=====] - 0s 1ms/step - loss: 0.0049
```

❖ Predicted Web View

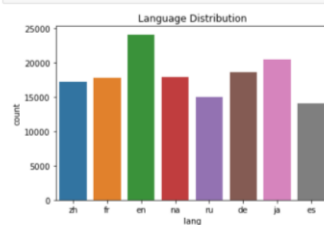
Using LSTM Model predicting the website page view.

[illegible]

❖ Using Matplotlib pyplot libraries to plotting the graph.

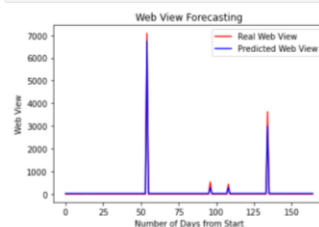
```
In [9]: # FUNCTION FOR CREATING ANOTHER COLUMN IN TRAIN DATASET FOR THE PAGE LANGUAGE
def find_lang(page):
    res= re.search("[a-z]{a-z}.wikipedia.org", page)
    if res:
        return res[0][0:2]
    return("na")
train_1["lang"]= train_1.Page.map(find_lang)

In [10]: sns.countplot(train_1["lang"])
plt.title("Language Distribution")
plt.show()
```



❖ IMPORTING ARIMA MODEL TO FUNCTION FOR CREATING ANOTHER COLUMN IN TRAIN DATASET FOR THE PAGE LANGUAGE

```
In [12]: #Visualising Result
plt.figure
plt.plot(y_test, color = 'red', label = 'Real Web View')
plt.plot(y_pred, color = 'blue', label = 'Predicted Web View')
plt.title('Web View Forecasting')
plt.xlabel('Number of Days from Start')
plt.ylabel('Web View')
plt.legend()
plt.show()
```



CHAPTER 6

RESULTS

❖ Using LSTM MODEL

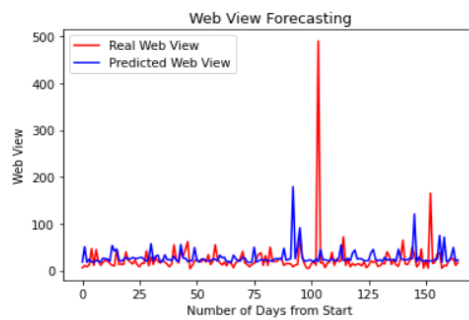
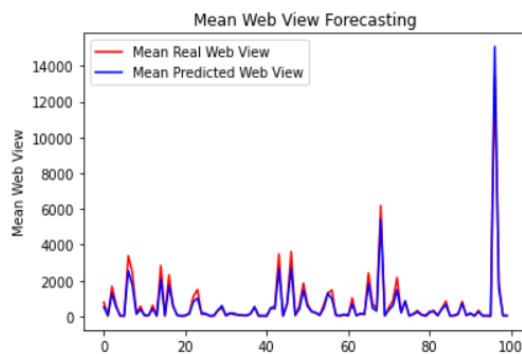


Fig 6.1: Predicting Website Page View



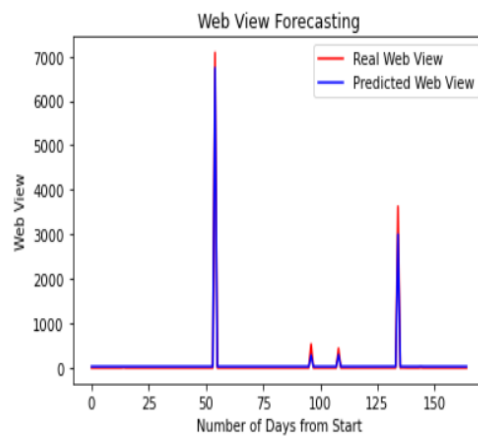


Fig 6.2: Predicting another web page view

❖ **Using ARMIA MODEL**

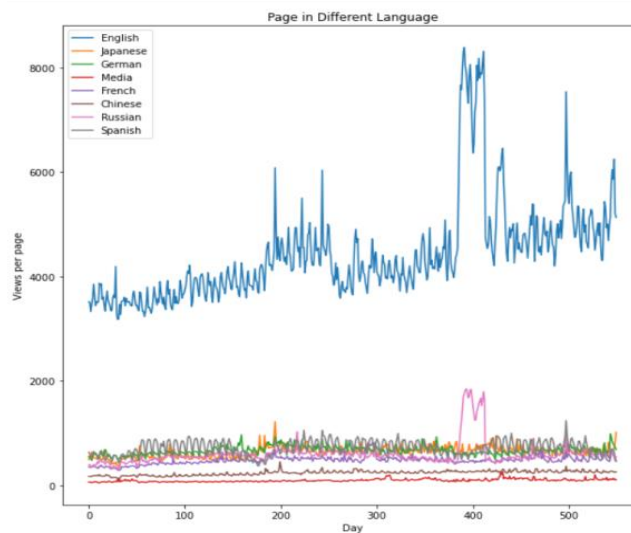


Fig 6.3: Plotting page in different language

Stats models also includes things like ARMA and ARIMA models that can be used to make predictions from time series. This data is not necessarily very stationary and often has strong periodic effects, so these may not necessarily work very well. I'll look at ARIMA predictions for the same set of very high view count pages.

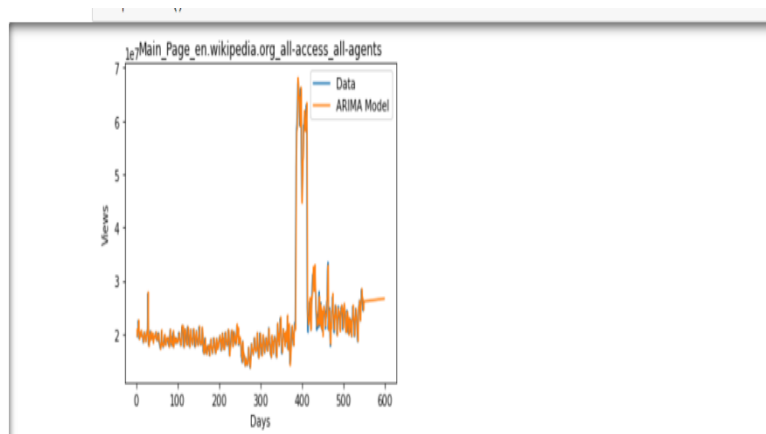


Fig 6.4: Plotting graph on Main_Page_en_wikipedia using ARIMA Model

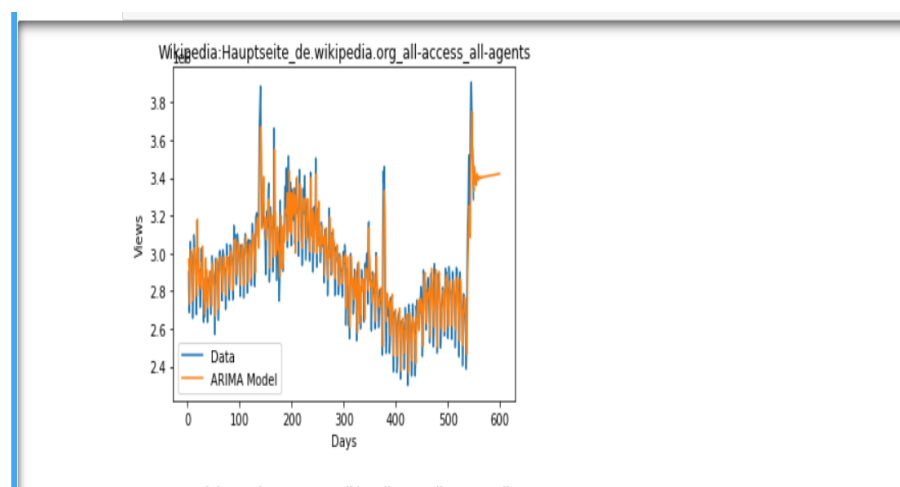


Fig 6.5: Plotting graph onWikipedia_Hauptseite_de.wikipedia.org using ARIMA Model

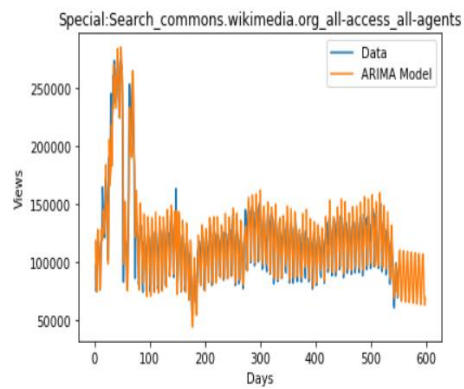


Fig 6.6: Plotting graph on Special_search_commons.wikipedia.org using ARIMA Model

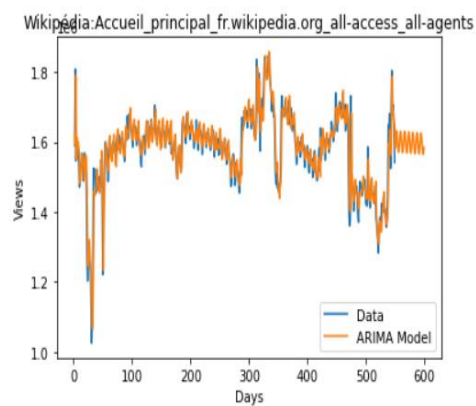


Fig 6.7: Plotting graph on Wikipedia_Accueil_principal_fr.wikipedia.org using ARIMA Model

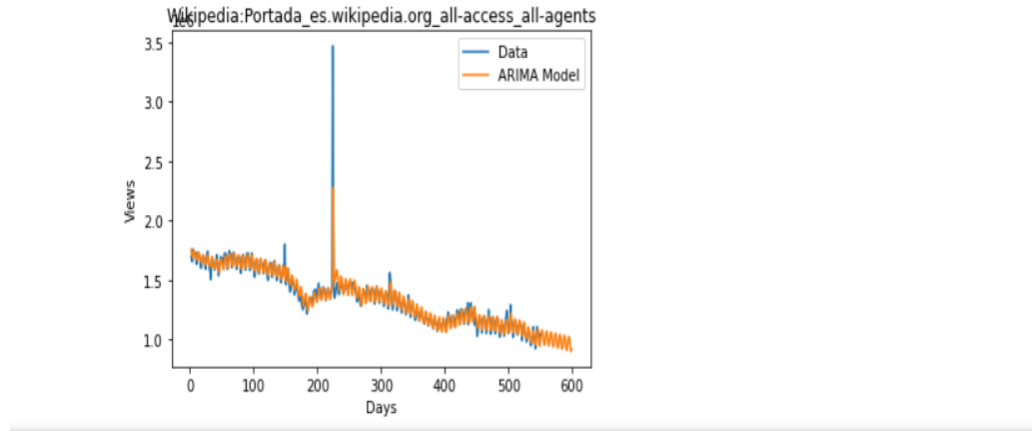


Fig 6.8: Plotting graph on Wikipedia_portada_es.wikipedia.org using ARIMA Model

That the ARIMA model in some cases is able to predict the weekly substructure of the signal, which is good. In other cases it seems to just give a linear fit. This is potentially very useful.

However, if we just blindly apply the ARIMA model to the whole dataset, the results are not nearly as good as just using a basic median model. It still seems to have some interesting properties, so maybe we can combine this with another model to get better results. Or, maybe we can find some subset of data where we expect ARIMA to work better than our other models.

CHAPTER 7

CONCULSION

Web traffic Time series prediction it can be achieved using Long Short Term Memory Recurrent Neural Network and Autoregressive integrated moving average more efficiently and accurately.

How many number of users that will access the website/link in the future is may be predicted. The model that was proposed will keep on updating as many user data is feed.

Our model can be played around all websites because of improving their web traffic load management and business analysis. More efficiency to our system can we get by LSTM RNN. Our system effectively captures seasonal patterns and long-term trends including information about holidays, day of week, language, and region might help our model to capture more correctly the highs and lows.

Time Series Forecasting is one of the least explored areas and various models are evaluated to improve the accuracy of the forecast. The main aim of our system is to predict future web traffic to make decisions for better congestion control. Previous data are considered to predict future values. We will also seems to explore various time series and provide a guidance for modulate the decision-making process in real-time.

In future works, the aim is to deepen in hidden pattern extraction for improving the efficiency of the LSTM and to study how human behavior affects the web traffic. To improve the performance of our model.

REFERENCE

1. "Predicting Computer Network Traffic: A Time Series Forecasting Approach using DWT, ARIMA and RNN" by Rishabh Madan, 2018.
2. "Web Traffic Prediction of Wikipedia Pages" by Navyasree Petluri, Eyhab Al-Masri, 2019.
3. "Time series forecasting using improved ARIMA" by Soheila Mehrmolaei, 2016.
4. Chen, D.; Gao, M.; Liu, A.; Chen, M.; Zhang, Z.; Feng, Y. A Recurrent Neural Network Based Approach for Web Service QoS Prediction. In Proceedings of the 2019 2nd International Conference on Artificial Intelligence and Big Data (ICAIBD), Chengdu,
5. Zhou, K.; Wang, W.; Huang, L.; Liu, B. Comparative study on the time series forecasting web traffic based on statistical model and Generative Adversarial model Knowl.-Based System. 2020, 213, 106467.
6. Bojer, C.S.; Meldgaard, J.P. Kaggle forecasting competitions: An overlooked learning opportunity. Int. J. Forecast. 2020.
7. Montero-Manso, P.; Athanasopoulos, G.; Hyndman, R.J.; Talagala, T.S. Fforma: Featurebased forecast model averaging. Int. J. Forecast. 2020, 36, 86–92