Web Traffic Analysis and Forecasting using Deep Learning Time-Series Approach In case of Commercial Bank of Ethiopia

A Thesis Presented

by

Demeke Admasu Bekele

to

The Faculty of Informatics

of

St. Mary's University

In Partial Fulfillment of the Requirements

for the Degree of Master of Science

in

Computer Science

February 2024

ACCEPTANCE

Web Traffic Analysis and Forecasting using Deep Learning Time-Series Approach

By

Demeke Admasu Bekele

Accepted by the Faculty of Informatics, St. Mary's University, in partial fulfillment of the requirements for the degree of Master of Science in Computer Science

Thesis Examination Committee:

______________________________	____________________
Internal Examiner	**Signature and Date**
Million Meshesha (PhD)	Million
External Examiner	**Signature and Date**
Dr. Alembante Mulu Kumlign	____________________
Dean, Faculty of Informatics	**Signature and Date**

February 2024

DECLARATION

I, the undersigned, declare that this thesis work is my original work, has not been presented for a degree in this or any other universities, and all sources of materials used for the thesis work have been duly acknowledged.

Demeke Admasu Bekele

Signature

Addis Ababa
Ethiopia

This thesis has been submitted for examination with my approval as advisor.

Advisor Dr. Tessfu Geteye

Signature

Addis Ababa
Ethiopia

February 2024

Acknowledgment

I would like to begin by expressing my sincere gratitude to the Almighty God, whose favor and direction have been invaluable throughout this journey. His support and divine intervention have provided me with strength, and inspiration to overcome challenges and achieve success. For all of his blessings and for giving me the confidence to embark on this research, I am immensely grateful to God.

I would like to sincerely thank and show my gratitude to all of the people who have provided invaluable support throughout my studies.

First and foremost, I extend my deepest gratitude to my advisor, Dr. Tessfu Geteye. His assistance, expertise, and encouragement have been extremely helpful in shaping the direction of this thesis. His insightful feedback and constructive criticism have been instrumental in helping me achieve success in this paper.

I am also very appreciative of my beloved wife, w/ro Tenagne Bayisa. Her patience, understanding, and constant backing have been a continual source of strength. Her confidence in me and her support during challenging situations have been essential to my success.

Furthermore, I would like to extend my sincere appreciation to my dear friend Ato Yigeremu Yohannes. Your continuous support, insightful discussions, and willingness to lend a helping hand have been invaluable. Your friendship has provided me with the strength and inspiration to overcome obstacles and strive for excellence.

Finally, I extend my gratitude to everyone who has provided support during this journey, including my educators, relatives, peers, and mentors. I deeply appreciate your encouragement, guidance, and attentive ears. I am sincerely thankful for the combined support and motivation I have received from these exceptional individuals. Their belief in me and their contributions have been instrumental in the successful completion of this endeavor.

Table of contents

List Acronym and Abbreviation

2D	Two Dimensional
3D	Three Dimensional
AI	Artificial Intelligence
ANN	Artificial Neural Network
ARIMA	Autoregressive Integrated Moving Average
biGRU	Bidirectional Gated Recurrent Unit
biLSTM	Bidirectional Long Short Term Memory
CBE	Commercial Bank of Ethiopia
CSV	Comma Separated Values
DL	Deep Learning
EDA	Exploratory Data Analysis
GRU	Gated Recurrent Unit
LSTM	Long Short Term Memory
MAE	Mean Absolute Error
ML	Machine Learning
MLP	Multilayer Perceptron
MSE	Mean Squared Error
RNN	Recurrent Neural Network

List of Figures

List of Tables

Abstract

Web traffic forecasting holds immense importance in making data-driven decisions across diverse domains. However, existing studies often rely on Wikipedia datasets that might not fully capture the distinctive aspects of web traffic. Additionally, there is a tendency to prioritize conventional models, overlooking the exploration of potentially superior models. The lack of comprehensive comparisons among different deep learning models hampers our understanding of their relative performance and how datasets impact their effectiveness. These limitations significantly hinder the generalizability of findings, particularly in developing countries. To address these gaps, this paper aims to investigate and compare the effectiveness of six recurrent neural network models by utilizing a unique dataset from local organizations. The objective is to develop a precise web traffic forecasting models by leveraging deep learning techniques and local data, ultimately enhancing decision-making processes. The research process involves stages such as data collection, preprocessing, hyperparameter tuning, model training, prediction, and evaluation. The research paper provides a comprehensive analysis of experiments conducted on web traffic datasets from the Commercial Bank of Ethiopia (CBE) website. The dataset includes visitor counts of web pages, spanning seven years from January 1, 2016, to January 3, 2023, totaling 2560 days of data. To facilitate the analysis, the dataset is divided into training, validation and testing sets.

In this study, deep learning techniques, including Long Short Term Memory (LSTM), bidirectional LSTM, bidirectional LSTM with attention, Gated Recurrent Unit (GRU), bidirectional GRU, and bidirectional GRU with attention, were effectively employed to analyze and predict web traffic patterns. The bidirectional GRU with attention model showed great promise, delivering the most impressive results with the lowest Mean Absolute Error (MAE) of 0.06102, Mean Squared Error (MSE) of 0.00713, and Root Mean squared Error (RMSE) of 0.08266.

The findings contribute to the understanding of web traffic analysis and display the potential of deep learning in this field. Future work can focus on enhancing and expanding this approach through dataset preparation, model architecture exploration, and ensemble methods. Overall, the study highlights the potential of deep learning to optimize resource allocation, improve web service performance, and enable data-driven decision-making in diverse domains.

Keywords: web traffic analysis, web traffic forecasting, deep learning, LSTM, GRU, bidirectional models, attention mechanisms.

Chapter One

1 Introduction

1.1 Background of the Study

Web traffic management is crucial for businesses or organizations that rely on online platforms to conduct their activities and exchange information. It is a process of ensuring that a website performs well and provides a good user experience for its visitors. Web traffic management involves measuring and regulating the number of people who access a website within a given period [1]. There are various methods to improve web traffic management, such as examining how visitors use the website, detecting and resolving any issues that affect the website's performance, and applying the best practices to enhance the website's functionality. Effective web traffic management can help website owners and managers achieve their business objectives, such as attracting more visitors, increasing conversion rates, and improving customer satisfaction [2, 3].

Web traffic forecasting is the process of estimating the future number of visitors or sessions on a website based on past and present data. Web traffic forecasting is important for different reasons, such as balancing the load, delivering ads, ensuring security, and providing content. Hence, it can enhance user experience, lower operational costs, and boost revenue for websites and platforms [3]. Therefore, web traffic forecasting can assist with website optimization, planning, and quality improvement. Petluri and Al-Masri [8], in their paper "Web traffic prediction of Wikipedia pages," highlight the significance of web traffic forecasting for improving website performance and user experience. They suggest a web traffic-forecasting model based on periodicity detection and sequence alignment. However, web traffic analysis and forecasting can be difficult due to the complex, nonlinear, seasonal, and noisy characteristics of web traffic data, as well as the unpredictable behavior of human users and various data related challenges like missing data, outliers, and high dimensionality [1].

Deep learning (DL) approaches have emerged as a powerful tool for time series forecasting problems, as they can discover complex and nonlinear patterns in data. These models have shown to be more accurate than traditional techniques for web traffic forecasting, which is a promising research area for the future [4, 5, 6, 7]. Web traffic forecasting tasks can benefit from using recurrent neural networks, which are a type of deep learning model that can process sequential data and remember past inputs. Deep learning models can learn long-term dependencies and handle sequential data efficiently [16, 21, 35, 38].

Numerous researchers have been drawn to the field of web traffic forecasting, employing various models to predict future web traffic patterns [1, 3, 8, 11, 13, 14, 16, 20, 21]. However, most of these studies have focused solely on conventional models, and even when deep learning models have utilized, they have predominantly relied on a single type of deep learning model, specifically LSTM. This neglects the exploration of other promising models that may yield superior results. Additionally, the majority of these studies have relied on datasets sourced from the Wikipedia website, which may not adequately capture the unique characteristics of web traffic in different domains and regions. Consequently, the models' generalizability and applicability are limited, particularly in developing countries like Ethiopia, where web traffic forecasting holds significant implications. Therefore, further research is necessary to investigate web traffic forecasting using diverse models and datasets, with specific attention to developing countries.

This research aims to enhance web traffic forecasting by developing deep learning models that utilize a locally acquired dataset. Accurate web traffic prediction is vital for data-driven decision-making. Our objective is to investigate the suitability and effectiveness of deep learning models specifically for web traffic forecasting in our unique setting. To accomplish this, we collected a dataset from local organizations and utilized recurrent neural network models, such as LSTM, bidirectional Long Short Term Memory (biLSTM), biLSTM with attention, GRU, bidirectional Gated Recurrent Unit (biGRU), and biGRU with attention.

The study followed a systematic plan with multiple steps to achieve its goal. It began with a literature review to gain a comprehensive understanding of the domain, leading to the formulation of research objectives. The second section involved designing, implementing, and evaluating the model. In conclusion, the study presents the main findings and proposes future research directions to advance web traffic forecasting.

This study rigorously followed a process to achieve accurate web traffic forecasting results through the utilization of deep learning techniques. Its contribution is of significant importance, as it advances the development of precise web traffic forecasting models. Moreover, the study highlights the effectiveness of deep learning techniques in the field of web traffic forecasting.

1.2 Motivation

Web traffic forecasting is a crucial task for data-driven decision making in various domains, such as, publication, internal communication, e-commerce, online advertising, and web analytics [1, 2, and 3].

Web traffic forecasting, despite its significant importance, is not extensively utilized for research purposes in our country. This represents a gap in leveraging local data and context to contribute to advancements in the field. By focusing on local web traffic patterns and characteristics, researchers have the opportunity to develop tailored forecasting models that address specific challenges and factors unique to our country. Furthermore, most previous studies conducted worldwide rely on a dataset obtained from Wikipedia's site, which may not adequately encompass the characteristics and challenges of web traffic in diverse contexts. Consequently, there is a requirement for additional datasets specific to web traffic forecasting, facilitating the evaluation and implementation of existing models.

Additionally, there is a notable absence of comprehensive comparisons that evaluate the performance of different deep learning models on a specific dataset, highlighting the impact of the dataset on the effectiveness of the models. Hence, it is crucial to conduct an investigation and comparative analysis of various deep learning models that can effectively capture complex patterns, thereby enabling accurate and reliable web traffic forecasts.

To address this gap, this paper motivated to investigate the suitability and effectiveness of six recurrent neural network models for web traffic forecasting in our specific setting. The study utilizes a dataset obtained from local organization. The performance of the following models is compared: LSTM, GRU, bidirectional LSTM, bidirectional GRU, bidirectional LSTM with attention, and bidirectional GRU with attention. By conducting this comparative analysis, the paper seeks to provide insights into the performance of these models and their applicability for accurate web traffic forecasting in our context.

1.3 **Statement of the Problem**

Web traffic forecasting is an essential undertaking in domains that rely on data-driven decision-making. However, it encounters numerous challenges due to the complexity and ever-changing nature of web traffic patterns [1, 8]. The currently available deep learning models have not undergone sufficient testing and comparison specifically for web traffic forecasting using local data. It is worth noting that local data may present distinct features and challenges compared to global or regional data. Moreover, there is a limit of publicly available datasets for web traffic forecasting, which limits the reproducibility and generalizability of the existing models. Therefore, there is a need to explore and compare different recurrent neural network models for web traffic forecasting on local data, and to create and share a novel dataset.

1.4 **Research Question**

The specific research questions are:

- How can we effectively analyze the web traffic data?
- How can we build a model that can capture the temporal and spatial patterns of web traffic?
- In terms of accuracy, how do deep learning models perform compared to each other and the methods used in previous studies when it comes to forecasting web traffic?
- Which deep learning model is the suitable in forecasting web traffic using locally collected datasets?

1.5 **Objectives**

1.5.1 General Objective

The general objective of the research is to develop a web traffic analysis and forecasting using deep learning time-series approach.

1.5.2 Specific Objectives

The specific objectives of the study are:

- To examine a relevant research paper pertaining to the analysis and prediction of web traffic.

- To develop a comprehensive dataset that can be used to predict web traffic accurately.
- To develop deep learning models that can accurately forecast web traffic.
- To evaluate the performance of developed models.
- To select the best model for web traffic forecasting through a comparative evaluation.

1.6 Scope and Limitation of the study

1.6.1 Scope of the Study

The scope of this study encompasses the creation of a dataset specifically designed for web traffic forecasting. Additionally, it evaluates and compares the effectiveness of different deep learning models in accurately forecasting web traffic, utilizing the newly developed dataset. The study has focused on the following aspects:

- Data collection: The study has gathered web traffic data from the internal communication portal of the Commercial Bank of Ethiopia.
- Data preprocessing: The study has preprocessed the data to ensure data quality, using methods such as data cleaning, normalization, and data splitting. The study also performs exploratory data analysis to understand the characteristics and patterns of the data.

- Model training and evaluation: The study applied six deep learning models, namely LSTM, GRU, bidirectional LSTM, bidirectional GRU, bidirectional LSTM with attention, and bidirectional GRU with attention, to the web traffic dataset and measure their accuracy and reliability in forecasting web traffic. The study used MSE, MAE and RMSE metrics to assess the performance of the models.

- Performance comparison: The study included a performance comparison to evaluate the effectiveness of the proposed models in relation to each other and with baseline methods used for web traffic forecasting. This comparative analysis allowed for a comprehensive assessment of their respective capabilities. By conducting these analyses, the study provided valuable insights into the relative performance of the models on the forecasting process.

- Analysis and discussion: During the analysis and discussion phase, this study thoroughly examined and discussed the results obtained from training and evaluating the models. Additionally, future research directions in the field identified. As a result of this process, a comprehensive understanding of the research outcomes achieved, leading to meaningful insights and implications.

1.6.2 Limitation of the Study

The study acknowledges several limitations that should take into consideration. To begin with, the data utilized in the study was obtained solely from a single website. As a result, the findings may have limited generalizability to other domains or contexts. The study may not encompass the full range of diversity and dynamics in web traffic patterns and trends across different websites, platforms, domains, regions, and so on. Additionally, the data collected from the company website was not extensive in terms of size, which further challenges the analysis.

Another limitation of this study is that it does not address the various factors involved in web traffic management. The study specifically focuses on analyzing and forecasting web traffic visitor numbers, without considering other aspects related to web traffic management.

Furthermore, it is important to note that the study does not address the ethical, social, and legal implications associated with the use of deep learning techniques for web traffic prediction. While the focus of the study is on the technical aspects of the models and their performance, it does not delve into the broader implications and considerations surrounding the application of these models in real-world settings. Therefore, it is crucial for future research and implementation to carefully consider and address these ethical, social, and legal dimensions to ensure responsible and accountable use of deep learning methods in web traffic prediction.

1.7 **Significance of the study**

This study offers two significant contributions. Firstly, it addresses a notable gap in previous research by creating a novel dataset sourced from websites in our country. The absence of such a dataset had hindered the development of web traffic forecasting models tailored to the local characteristics. The availability of this dataset now opens avenues for the development of models specific to the Ethiopian context.

Secondly, the study conducts a comprehensive comparison of various deep learning models for web traffic analysis and prediction. Deep learning models have demonstrated remarkable performance in time series forecasting across different domains [5, 10], but their application in web traffic prediction has been limited. Therefore, this work advances the field of web traffic prediction by introducing a new dataset and conducting a thorough evaluation of different deep learning models within this specific context.

Web traffic analysis and forecasting can assist website owners and managers in measuring the efficacy of their website strategies and making data-driven decisions to improve website performance, content, marketing, and security. In addition, the web traffic forecasting findings will help website owners in proactive resource allocation, load balancing, and user experience optimization by providing insights of traffic volume. By analyzing and predicting web traffic, the system owners can automate their system and make it more interesting for the users. The system can offer customized suggestions, pertinent information, and a user-friendly interface that suit the users' needs and preferences. The system can also help the users discover the information they need more quickly and easily, increasing their satisfaction and loyalty.

1.8 **Methodology**

Research methodology is a systematic and scientific approach used to collect, analyze, and interpret data in order to investigate research questions. To align with our objectives, it is crucial to study the theories and guiding principles of data analysis techniques. In the context of web traffic analysis and forecasting, the research methodology involves various approaches. These include important processes such as data collection, preprocessing, model selection, training, prediction, and evaluation.

- Define the research problem and objective: The research problem and objective defined in this phase by identifying the problem that needs to address and by outlining the study's objectives. Additionally, the research gaps outlined, along with the rationale for filling it.

- Data collection: To conduct the study on web traffic analysis and prediction, secondary data type obtained from the Commercial Bank of Ethiopia, specifically from their internal portal website. The data consists of numerical statistics representing the number of website traffic. Thirty-two web pages from the portal used for data collection. The data spans a period of five years, from January 1, 2016, to January 3, 2023. For each web page, daily visitor counts recorded, resulting in a dataset comprising 2560 days in total.
- Exploratory Data Analysis: During the exploratory data analysis phase, the gathered data thoroughly examined in order to identify patterns, trends, and anomalies. To accomplish this, we employed a line chart data visualization techniques, which enabled us to extract valuable insights and gain a deeper understanding of the fundamental attributes present within the data. By utilizing these techniques, we able visually represent the data in a manner that facilitated the discovery of meaningful information and provided valuable context for further analysis.
- Data Preprocessing: To guarantee the accuracy and consistency of the data, this approach is an essential part of data analysis. It entails cleaning, normalizing, splitting and reshaping. The main activities performed in data preparation include the following:
  - Data cleaning is a crucial step in the data-preprocessing pipeline, aimed at identifying and rectifying errors, inconsistencies, and inaccuracies within the dataset. In our process, we have specifically focused smoothing noisy data, and handling missing values by imputing them with zero. The first aspect of data cleaning involved handling missing values by imputing them with zeros. This allows us to maintain the integrity of the dataset and ensure that missing values do not disrupt subsequent analyses. By performing these data cleaning procedures, we enhance the quality and reliability of the dataset, enabling more accurate and meaningful analysis and interpretation of the collected data.
  - In order to facilitate analysis and querying by models, we conducted data normalization. By establishing a standardized data format through data normalization, we enhanced data integrity and enabled analysis that is more accurate by models.
  - To cater to the specific requirements of deep learning recurrent models, we have formatted the dataset in a three dimensional structure. This formatting process

ensures that the dataset aligns perfectly with the input expectations of these models. By organizing the data in this manner, we enable the models to leverage their recurrent architecture, which designed to analyze sequential data. This alignment facilitates the models' ability to capture temporal dependencies and extract valuable insights from the dataset, ultimately enhancing their performance and enabling them to uncover intricate patterns and relationships within the data.

  - In addition to formatting the dataset to meet the specific requirements of deep learning recurrent models, we have also taken the crucial step of splitting the dataset into training, validation and testing sets. This division allows us to evaluate the performance and generalization capability of the models more effectively. The training set utilized to train the models, enabling them to learn from the data and optimize their parameters. On the other hand, the testing set serves as an independent evaluation set, enabling us to assess how well the models generalize to unseen data. By employing this division, we can make reliable assessments of the models' performance and ensure that they are capable of accurately predicting and understanding new data beyond the training set.

- Model Selection: At this stage of the research methodology, we focused on selecting an appropriate deep learning model that demonstrates intelligent performance in analyzing and forecasting web traffic. Recurrent Neural network models have shown remarkable capabilities in handling time-series data, making them well suited for this task. By leveraging the power of deep learning algorithms, we aim to develop a model that can effectively analyze and predict web traffic patterns with a high level of accuracy and precision [7, 12, 15, and 16].

- Model Training: The selected models trained using the prepared dataset, where the model learns relationships and patterns from historical web traffic data. Several models are trained using web traffic data in this work, including LSTM, GRU, biLSTM, biGRU, and with attention, and biGRU with attention models.
- Prediction: After training the models using the dataset, we have proceeded to the next crucial step, performing predictions using the trained models. With the models now equipped with learned parameters and insights from the training process, we can leverage their capabilities to make accurate predictions on new, unseen data. By inputting relevant

data into the models, they utilize their internal mechanisms to process the information and generate predictions or outcomes based on the patterns and relationships they have learned. This prediction phase allows us to leverage the power of the trained models to make informed decisions and gain valuable insights.

- Model Evaluation: Once the trained models have made predictions on the target variable, it is crucial to evaluate their performance. This evaluation typically accomplished by employing appropriate evaluation metrics, such as mean absolute error, mean squared error, and root mean squared error. These metrics allowed us to quantitatively assess the accuracy and effectiveness of the models in predicting web traffic patterns.
- Model Comparison: In order to determine the most suitable model for web traffic forecasting tasks, a comparison conducted based on their evaluation results. By comparing the evaluation results, we can identify the model that outperforms the others in terms of accuracy and predictive capability. The model with lower MAE, MSE, and RMSE values indicates a better fit to the data and more precise predictions of web traffic patterns.
- Interpretation and Reporting: The study engages in thorough interpretation and reporting of the collected results, ensuring a comprehensive analysis that yields significant conclusions. By carefully examining the results, the study draws meaningful inferences from the data, contributing to a conclusive understanding of the subject matter. A concise summary provided to encapsulate the research results, accompanied by suggested applications of deep learning for web traffic prediction. Additionally, the study identifies specific areas for future research, aiming to improve the accuracy and efficiency of web traffic forecasts. The research methodology diligently documents and reports the entire process, offering detailed insights into the limitations, and potential future directions of the research. This comprehensive approach ensures transparency and provides a foundation for further advancements in the field.

1.9 **Organization of the thesis**

The paper follows the structure below. Chapter one has an introduction: This chapter provides the background, motivation, problem of statement, research questions, objectives, scope, significance of the study and methodology. It also gives an overview of the thesis organization. Chapter two is literature review: This chapter examines the relevant literature on the research topic. It critically assesses the existing knowledge, and identifies the research gaps. Chapter three is methodology: This chapter explains and justifies the methodology used for the study. It describes data collection method, data processing and analysis techniques. Chapter four is Implementation, Experimental Results and Discussion: This chapter presents and analyzes data; answers research questions, reports findings, and discusses their validity and reliability. Chapter five is Conclusion and Future Works: This chapter summarizes the main points of the thesis, draws conclusions based on the findings and discussion, and suggests recommendations for future research and practice.

Chapter Two

2 Literature Review

2.1 Introduction

This chapter's goal is to present a comprehensive review of the literature on web traffic analysis and prediction. Assessing the different approaches that scholars employ to predict web traffic is the aim of this chapter. An extensive review of deep learning applications in web traffic analysis and prediction given in this chapter. It covers various methods that researchers have used to forecast time series problems and the metrics that have been employed to evaluate these methods. The choice of forecasting method depends on the specific needs and context of the website, and researchers are constantly exploring new approaches to enhance accuracy and effectiveness in this area. Evaluation metrics are covered after an explanation of web traffic analysis and forecasting techniques section of the chapter. By including the web traffic data sets that utilized in their study, this chapter also covers the review of related publications. In the end, the summary of the chapter shown here.

2.2 Web Traffic and Analysis

2.2.1 Overview of Website

A website is a grouping of web pages with the same domain name that are run at least on one web server. A web browser can be used to access a website by typing in its URL or by clicking a link to it on another website. Static and dynamic content are both available on websites; static pages are created and saved as HTML on a web server, whilst dynamic pages are created as needed using content from a database. Depending on the specific webpage a user is visiting, some websites serve a blend of static and dynamic content [2].

A company's website can have a significant impact on its success. Website traffic is a crucial metric that indicates the potential number of clients, business leads, sales, readers, or followers a company can attract depending on the type of website. Maintaining more website visitors is beneficial for a business's growth, income, and client retention. Monitoring a website's traffic, or the amount of visits, is crucial for evaluation purposes. Website traffic can also provide valuable information about the company’s performance and future prospects. Three elements that highlight the significance of website traffic are: promoting higher conversion rates, measuring marketing efficiency, and identifying new opportunities for growth. By driving more focused traffic to their website, a business can boost the likelihood of converting visitors into customers. Monitoring the sources of website traffic can help a company assess the effectiveness of its marketing channels and initiatives, leading to improved marketing strategies. Studying visitor behavior and feedback can also help a company discover new growth opportunities [2] [23].

2.2.2 Web Traffic

Website traffic encompasses various types, including direct, organic search, paid search, and social media. Direct traffic occurs when visitors access a website by directly entering its URL into the address bar, using a bookmark, or following a link from an email. Increasing direct traffic can be facilitated by having a straightforward and memorable URL [24, 25].

Organic search traffic is generated when visitors arrive at a website through search engines like Google. This type of traffic relies on adhering to best practices in search engine optimization (SEO) to improve rankings and visibility in search results.

Paid search traffic, on the other hand, originates from visitors who click on pay-per-click (PPC) advertisements. These ads can be customized and target specific keywords and locations to maximize their effectiveness.

Social media traffic refers to visitors who arrive at a website through links shared on social networks such as Facebook, Twitter, and Instagram. Leveraging a presence on relevant social media platforms and sharing engaging content can drive traffic from these networks to a website.

2.2.3 Web Traffic Analysis and Forecasting

Web analytics is a crucial aspect of understanding website traffic and making data-driven decisions to improve website design, marketing strategies, and product offerings. There are various built-in or third party analytics tools available for measuring website traffic, which require calibration with a website to gather real-time data. Metrics such as session numbers, bounce rate, time on site, conversion rate, and cost per visitor are used to measure website traffic and visitor behavior [26]. Session numbers refer to the number of visitors or sessions, while the best analytics tools measure the duration of each visit and the time spent on each page, as well as distinguishing between first-time and repeat visitors. Bounce rate is a crucial metric that indicates the proportion of visitors who leave the site after viewing only one page, which can affect search engine optimization (SEO). Time on site provides a more detailed picture of the time visitors spend on the website, including the time spent on each page. For e-commerce platforms, conversion rate is a critical metric that measures the percentage of visitors who made a purchase. Additionally, if paid advertising like pay per click (PPC) is used, then cost per visitor becomes an important metric. By using these metrics and analytics tools, businesses can gain insights into their website traffic, identify areas for improvement, and make data-driven decisions to better serve their clients and grow their business [27].

2.3 Approaches for Web Traffic Analysis and Forecasting

As technology grows more pervasive in our daily lives, businesses are spending more and more on learning algorithms to simplify things for customers. Examples of this include time series prediction of future occurrences of things and pattern-recognizing technology. These technologies often related to artificial intelligence, machine learning, deep learning, and neural networks, but these terms sometimes used interchangeably without clarifying the differences between them [9]. Artificial intelligence is a broad term that refers to the ability of machines to perform tasks that normally require human intelligence, such as reasoning, decision-making, and problem solving. Machine learning is a subset of Artificial Intelligence (AI) that involves using data and algorithms to enable machines to learn from data and improve their performance without explicit programming. Deep learning is a type of Machine Learning that uses artificial neural networks to mimic the way the human brain learns from data. Artificial Neural Network (ANNs) are composed of layers of nodes that process and transmit information [6]. Deep Learning uses multiple layers of nodes to extract higher-level features from data and achieve better accuracy and precision in various tasks. Neural networks are a general term that covers different types of ANNs, such as recurrent neural networks (RNNs), convolutional neural networks (CNNs), and transformers. These types differ in their architecture, functionality, and applications [9] [10].

Both deep learning and machine learning are crucial techniques for predicting outcomes from data analysis. Each has benefits and drawbacks of their own. Though less computationally demanding than deep learning, machine learning is less accurate. Small and medium-sized datasets can be handled using machine learning techniques, which can also operate on low-end hardware. Machine learning algorithms are also easier to interpret and explain which can help with debugging and understanding the results. However, the performance of machine learning algorithms tends to plateau after training on big datasets, and hand engineering the features is necessary, which can be laborious and domain-specific [34]. Although deep learning requires more computing power than machine learning, it is more accurate. Large and complicated datasets can be handled by deep learning algorithms, which also have the ability to automatically extract features from the data without the need for human feature engineering. Predictive modeling, missing data, sequential data, organized and unstructured data, and non-linear relationships can all be handled using deep learning algorithms. Deep learning algorithms, however, can be more prone to biases and errors since they need a lot of data and computer power to train, and because they are more difficult to understand and interpret [6, 34].

While deep learning has proven to be effective in handling large datasets, addressing the limitations of small datasets requires specialized techniques. M. Romero et al. [31] explored various approaches to tackle the challenges posed by small datasets in deep learning. Notably, they emphasized the efficacy of dropout as a dependable regularization method, even when confronted with limited data availability. Their study further confirms the value of dropout as a beneficial regularization technique, particularly in situations where data scarcity is a concern.

Depending on the issue and the data, several academics have employed both machine learning and deep learning for web traffic analysis and forecasting. It is impossible to say with certainty which approach is superior because each has advantages and disadvantages of its own. The optimal option is determined by the researcher’s particular objectives, available data, and computing capacity.

2.3.1 Deep Learning Approach

One of the branches of machine learning is “deep learning”, which uses artificial neural networks to learn from data. Artificial neural networks consist of layers of interconnected nodes that can process data and adjust their weights according to the input. Some of the common deep learning algorithms that can be used for time series forecasting are RNN, LSTM, and GRU. Each of these algorithms has its own advantages and disadvantages, and the choice of algorithm depends on the specific problem being solved [16, 21, 35, 38, 39].

Neural networks, or artificial neural networks (ANNs), are made of layers of nodes, which include an input layer, one or more hidden layers, and an output layer. Each node, or artificial neuron, connects to another and has a weight and a threshold. If the output of a node is higher than the threshold, the node is activated and sends data to the next layer. If not, the node does not pass any data to the next layer. The number of layers in a neural network determines how “deep” it is. A neural network with more than three layers, which include the input and output layers, is called a deep learning algorithm or a deep neural network. A neural network with only three layers is just a simple neural network.

Deep learning has been successful in the time series domain because it can learn from large amounts of sequential data and provide accurate predictions, which are important for many applications, such as language modeling, speech recognition, and medical predictions. However, it also faces some challenges, such as the need for innovation in model structure, training methods, reducing training time, online learning, and overcoming adversarial samples [8, 9].

Deep learning methods have been applied to web traffic analysis and forecasting to discover and exploit the hidden patterns and structures in historical web traffic data. They can produce reliable and accurate predictions of future web traffic based on these patterns [4, 5, 6, and 38].

2.3.2 Recurrent Neural Networks

Recurrent Neural Networks are artificial neural networks that process sequential data like natural language or time series. They can remember previous states of the sequence and model temporal dependencies. RNNs have many applications that involve sequential data, such as natural language processing, speech recognition and machine translation [10]. RNNs are different from feedforward neural networks, which treat the inputs and outputs as independent of each other. RNNs can consider the previous state of the sequence while working on the current state, enabling them to model temporal dependencies in data. These neural networks have feedback loops in their structure, which let them store and use information from previous time steps. RNNs can capture the temporal dependencies and patterns in sequential data, such as web traffic [21]. RNNs have a feedback loop that links the output of one step to the input of the next step, forming a memory of the past inputs. This memory can assist RNNs to learn long-term patterns and produce outputs that depend on the context of the sequence [35, 37]. Figure 2.1 shows RNN structure which has an input layer, hidden layer and output layer.

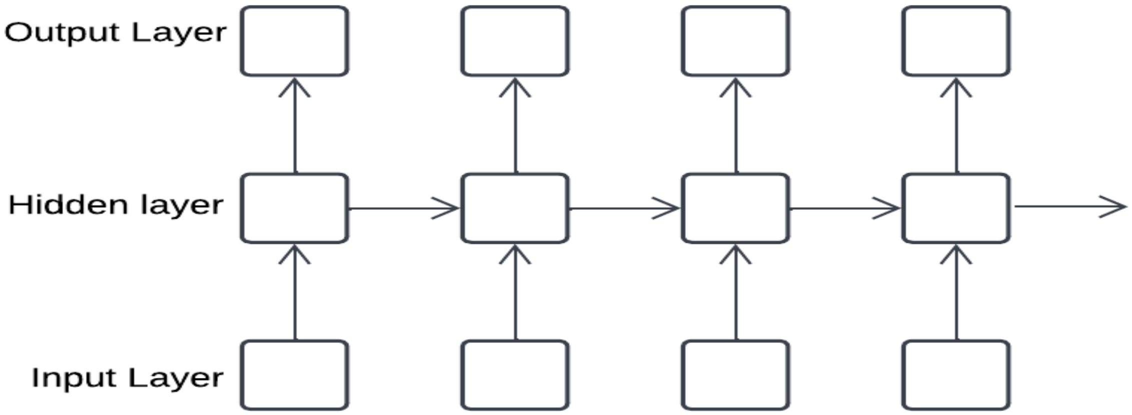

Figure 2.1 Recurrent Neural Network Model

RNNs can be challenging to train for long sequences due to a number of issues, including the vanishing or exploding gradient problem. Numerous RNN architectures, including Attention-based, Long Short-Term Memory networks and Gated Recurrent Unit, have been created to address these issues. Because these networks employ distinct mechanisms to regulate memory and information flow within RNNs, they are better equipped to handle complex and long sequences [19, 37].

2.3.3 Long Short Term Memory

One of the challenges of sequential data analysis is to capture the long-term dependencies and patterns in the data, which are often lost or distorted by conventional Recurrent Neural Networks. RNNs are neural networks that process sequential data by passing information from one time step to the next through loops in their structure. However, the information that flows through the loops can either fade or explode over time, resulting in the problem of vanishing or exploding gradients. This problem prevents the network from updating its weights effectively and learning the long-term patterns in the data [19, 36, 37].

To address the challenge of capturing long-term dependencies in sequential data, such as time series, speech, and text, the concept of Long Short-Term Memory networks is introduced as a specialized type of recurrent neural network. LSTM networks are designed to effectively learn and retain information over extended periods. They consist of a memory cell capable of storing information for an extended duration, along with three gates that regulate the flow of information within the network [16, 21, 28].

These gates, illustrated in Figure 2.2, comprise the input gate, the forget gate, and the output gate. The input gate determines which new information should be stored in the memory cell. The forget gate determines which old information should be discarded from the memory cell. Lastly, the output gate determines which information should be output from the memory cell. By incorporating these gates, LSTM networks possess the ability to selectively retain or discard information as it propagates through the network. This enables the networks to capture intricate patterns and dependencies in sequential data [16, 28, 33, 44, 48].

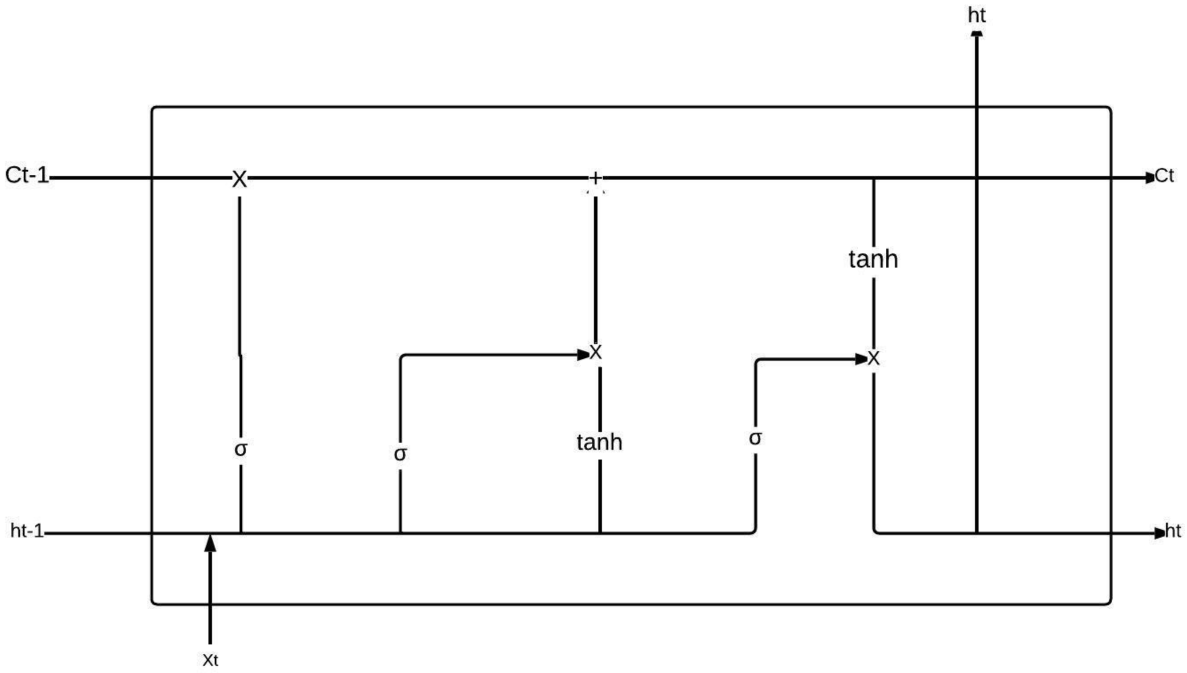

Figure 2.2 LSTM Architecture

2.3.4 Bidirectional Long Short Term Memory

A bidirectional long-short term memory is a special kind of RNN that can process the sequence in both directions: from the beginning to the end, and from the end to the beginning. This allows biLSTM to learn from both the past and the future context of each element in the sequence, which can improve its performance and generalization. biLSTM can handle different kinds of sequential problems, such as univariate or multivariate time series forecasting, and single-step or multi-step forecasting. Nevertheless, long-short term memory is another kind of RNN that can process the sequence in only one direction, either forward or backward. LSTM can also learn long-term dependencies in the sequence, but it cannot access the future context of each element. Therefore, biLSTM can have an edge over LSTM when the future context is relevant for the problem [42, 43, 45, 46].

2.3.5 Bidirectional Long Short Term Memory with Attention

Bidirectional LSTM with attention is a type of neural network model that can process sequential data in both forward and backward directions. This model can learn the context and dependencies of the input sequence from both ends, by using two RNNs that run in opposite directions. It can also use an attention mechanism to assign different weights to different parts of the sequence, based on their relevance for the output. This model is suitable for tasks such as text classification, sentiment analysis, and machine translation [45, 47, 48].

Bidirectional LSTM with attention can be applied to various time series forecasting problems, such as predicting stock prices, electricity demand, weather, etc. The idea is to use multiple input features (such as price, volume, indicators, etc.) to forecast a single output feature (such as future price) at each time step. The network can learn the complex and nonlinear relationships between the input features and the output feature, as well as the temporal patterns and trends in the data. The attention mechanism can help the network to weigh the importance of each input feature and each time step, and to focus on the most relevant information for the prediction [45, 46, 48].

One example of applying bidirectional LSTM with attention to time series forecasting is the paper by Abbasimehr and Paki1 [44], which uses this method to improve the accuracy of forecasting various real-world time series datasets. They compare their method with several benchmarking methods, such as Autoregressive Integrated Moving Average (ARIMA), feedforward neural networks, and LSTM with attention. They show that their method outperforms all the other methods in most datasets in terms of symmetric mean absolute percentage error (SMAPE). They also show that the attention mechanism can provide useful insights into the importance of different input features and time steps for the prediction.

2.3.6 Gated Recurrent Unit

Gated Recurrent Unit is a type of recurrent neural network that can handle sequential data effectively. Chung et al. [30] introduced the GRU as a new type of recurrent unit that can handle varying time scales of dependencies. They demonstrated that the GRU performed better than the LSTM on several natural language processing tasks, while requiring less training time [28]. The GRU has a single hidden state that combines the functions of the cell state and the hidden state in the LSTM. It has only two gates: the update gate and the reset gate. The update gate decides how much information from the previous hidden state and the current input should be kept in the new hidden state. The reset gate decides how much information from the previous hidden state should be used to calculate the new hidden state [28, 29, 39]. The GRU can update its hidden state by selectively retaining or discarding information from the previous hidden state and the new proposed hidden state, which computed by the reset gate. This allows the GRU to capture long-term dependencies effectively. If the update gate values are close to one, the GRU can preserve most of the previous hidden state without changing it or re-computing it.

The reset gate and the update gate play important roles in determining the new hidden state of a neural network as illustrated on Figure 2.3. By multiplying the update gate with the previous hidden state, the network selects which parts of the previous hidden state it will store in its memory while disregarding the rest. Then, it fills in the missing information by using the inverse of the Update gate to filter the proposed new hidden state from the reset gate. This mechanism allows the network to maintain long-term dependencies. If the update vector values are close to one, the Update gate can retain most of the previous memories in the hidden state without changing the entire hidden state or re-computing it.

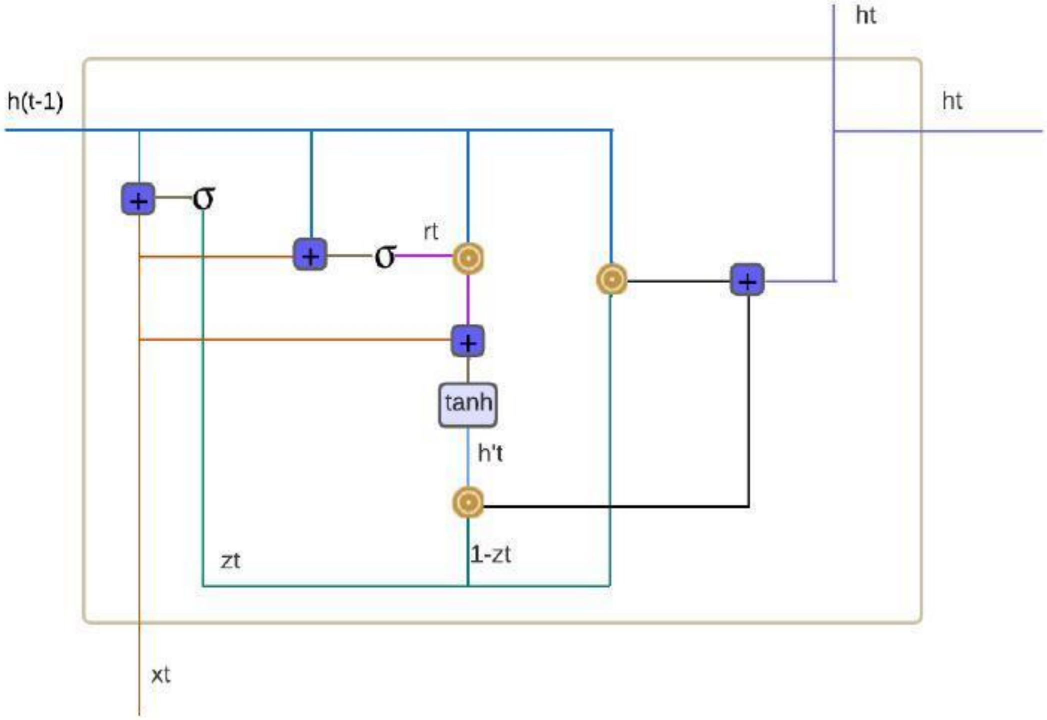

Figure 2.3 GRU Model Architecture

The update gate(zt) is represented by the following equation, which describes how much of the GRU unit is updated:

$zt \ = \sigma(W^{(z)}xt + U^{(z)}ht - 1)$ ……………………………………………………… Eq 2. 1

In this network unit, when the input at time t (*xt*) is fed into the model, it is multiplied by its own weight (W(z)). Similarly, the information from the previous time step (h(t-1)) is multiplied by its own weight (U(z)). These two products are then added together, and the result is passed through a sigmoid activation function to constrain the output between 0 and 1. The resulting value represents the update gate, which determines the amount of past information that should be carried forward to future time steps. This is a powerful feature because the model can decide to retain all the relevant information from the past, thereby reducing the risk of the vanishing gradient problem.

In the GRU model, both the Update and Reset gate vectors are generated using the same formula. However, the weights applied to the input and hidden state are unique to each gate, resulting in different final vectors for each gate. This enables the gates to perform their respective functions effectively.

In the GRU model, the reset gate (rt) at a given time step (t) is responsible for deciding how much of the previous information should be forgotten. To calculate the reset gate, a formula is used that takes into account both the current input and the hidden state from the previous time step. This gate is a crucial component of the GRU model as it allows the network to selectively forget or retain information from the past depending on the task.

$$rt = \sigma(W^{(r)}xt + U^{(r)}ht - 1) \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \text{Eq 2. 2}$$

The reset gate is computed by multiplying the previous hidden state and current input with their respective weights, and then adding them together. This sum is then passed through a sigmoid function, which maps the resulting value to a range between 0 and 1. This allows the gate to filter the less important information and retain the more important information for subsequent steps. During the backpropagation training of the whole network, the weights in the equation are updated so that the vector learns to retain only the useful features.

The new memory gate in the GRU model is created by applying the hyperbolic tangent function to the reset gate. This gate is defined by the following function:

$$ht' = tan\,h\,(Wxt + rt\ \odot Uht - 1) \quad \ldots\ldots\ldots\ldots\ldots\ldots \text{Eq 2. 3}$$

To calculate the new memory gate in the GRU model, the current input (xt) is multiplied by a weight matrix (W). The previous hidden state (ht-1) is multiplied by another weight matrix (U) and then the resulting products are then subjected to an element-wise multiplication (Hadamard product) with the reset gate, resulting in a vector. Then the result is added together. Finally, a non-linear activation function (tanh) is applied to the resulting vector to obtain the new memory gate (ht’). This gate is an important component of the GRU model, as it allows the network to selectively update its internal state based on the current input and the previous hidden state, while retaining the relevant information from the past.

The equation 2.4 shows how the GRU hidden state is computed. It involves selecting information from both the current memory content (ht’) and the previous hidden state (h(t-1)), and combining them using an update gate (zt) and element-wise multiplication. Specifically, the equation first multiplies the update gate and the previous hidden state, and then multiplies the complement of the update gate and the current memory content. Finally, it adds the two results to obtain the new hidden state.

$$ht = zt \odot ht - 1 + (1 - zt) \odot ht' \quad \text{.. Equ 2. 4}$$

2.3.7 Bidirectional Gated Recurrent Unit

A bidirectional GRU is a special kind of recurrent neural network that has the ability to process sequential data in both forward and backward directions. This means that the network can use the information from the past and the future of the sequence to learn the patterns and dependencies in the data. By doing so, the network can capture both the long-term and short-term relationships in the data, which can enhance the performance of forecasting tasks. Bidirectional GRU can be applied to various time series forecasting problems, such as predicting the future values of the stock price, the wind speed, the electricity load, and so on. The basic idea is to feed the historical data (or the imputed data if there are missing values) into the bidirectional GRU network, and then use the output of the network to forecast the future values of the time series [49, 50, 51].

2.3.8 Bidirectional Gated Recurrent Unit with Attention

A bidirectional GRU with attention is a type of neural network that can handle sequential data, such as time series, by processing it from both ends. This means that the network can capture both the past and the future context of each point in the sequence. However, not all points in the sequence are equally important for the prediction task. Some points may be more informative than others. To address this issue, the network uses an attention mechanism, which is a way of learning how to pay attention to the most important parts of the input. The attention mechanism works by computing a score for each hidden state of the bidirectional GRU, which represents how similar it is to the current state of the network. The scores are then used to calculate a weighted average of the hidden states, which is called the context vector. The context vector is a summary of the most relevant information in the input sequence, and it is used to generate the output. By using the attention mechanism, the network can learn to focus on the most important features or words in the input sequence, and to produce a more accurate and meaningful output [52].

2.4 **Evaluation Metrics**

MAE, MSE, and RMSE are commonly used evaluation metrics in data analysis and machine learning to measure the performance of predictive models. These metrics are especially useful for evaluating regression models, where the goal is to predict continuous numerical values. They quantify the accuracy of predictions by measuring the differences between predicted and actual values, allowing for comparison and selection of the best-performing model [18, 33, 53].

Mean Absolute Error represents the average of the absolute differences between the predicted and actual values within a dataset. It provides a measure of the average magnitude of errors, regardless of their direction. A lower MAE signifies better model performance, and a perfect prediction is indicated by a value of zero. MAE is particularly valuable in assessing the disparity between two continuous variables, such as an original time series and a predicted time series generated by a model.

In the context of web traffic analysis, MAE serves as an effective tool for evaluating the accuracy of forecasting techniques. It accomplishes this by comparing predicted values (yp) with the actual observed values (yi). This evaluation aligns with Equation 2.5, commonly used by researchers to gauge the efficacy of deep learning-based models. By incorporating MAE as an evaluation criterion, the researchers can derive valuable insights into the precision of a model and the level of agreement between predicted and actual values. This, in turn, aids in identifying areas for improvement and refining the forecasting approach [21].

$$MAE = \frac{|Yi - Yp|}{n} \quad \text{……………………….………………………….. Eq 2. 5}$$

n is the number of observations.

Mean Squared Error is a metric that measures how far the predicted and actual values are from each other by taking the average of their squared differences. MSE is more sensitive to outliers in the dataset than MAE, as it gives more weight to larger errors. MSE is commonly used because of its mathematical properties, such as being differentiable [33, 53]. Like MAE, lower MSE values mean better model performance, and a value of 0 means a perfect prediction. MSE is a widely used measure of accuracy for regression models. It shows the average squared difference between predicted and actual values in a dataset, Equation 2.6. A lower MSE value implies a higher accuracy of the predicted values, as they match the actual observed values more closely [15]. This shows that the model is better at finding the patterns and trends of web traffic.

$$MSE = \frac{\sum(y_i - y_P)2}{n} \quad \text{…………………….…………..………………………… Eq 2. 6}$$

Root Mean Squared Error is the square root of MSE, providing an interpretable metric in the same units as the target variable. RMSE is often preferred when the scale of the target variable is significant, as it provides a more intuitive understanding of the prediction errors. Similar to MAE and MSE, a lower RMSE indicates better model performance, and a value of 0 represents a perfect prediction [53].

In the context of our study, RMSE can be a valuable tool for understanding the magnitude of the errors in our model's predictions. By identifying any outliers or extreme values that may be affecting the overall performance of the model, RMSE can help to refine our forecasting approach and improve the accuracy of our predictions. Therefore, RMSE can be particularly useful in providing insights into the effectiveness of our model and in guiding our decision-making process [53].

$$RMSE = \sqrt{\frac{\sum(y_i - y_P)2}{n}} \quad \text{Eq 2. 7}$$

In general, by employing MAE, MSE, and RMSE as evaluation metrics, web traffic forecasting models can gain valuable insights into their effectiveness and performance. These metrics can provide a comprehensive assessment of the accuracy and precision of the model's predictions, enabling researchers to identify areas for improvement and refine the forecasting approach.

2.5 Review of Related works

The study conducted by Vrushant et al. [1] that developed an accurate forecasting model for web traffic based on page name, visited date, and number of visits per page over a year. In order to maintain continuous customer service levels in the digital age, the study emphasizes the significance of effective web traffic management strategies and employs the ARIMA model to anticipate future web traffic values. Time series analysis and predictive modeling are the two main theories, concepts, and constructs that support the study question, which is about predicting web traffic. The study also explores the weaknesses of other popular forecasting methods and proposes an unsupervised model to find unobserved patterns in human behavior. The study advances our comprehension of the issue research problem and makes suggestions for possible future directions to improve the predictive precision of web traffic forecasts. The study does, however, have some flaws, such as its reliance on a small dataset and the lack of a comprehensive evaluation framework. Larger and more varied datasets, develop comprehensive evaluation methodologies, and alternative forecasting strategies should all be explored in future studies. In general, the study makes a valuable contribution to web traffic forecasting by proposing an ARIMA based model, identifying the limitations of other techniques, and suggesting potential future research directions [1].

Liu et al. [11] have built a framework known as High-Frequency Autoregressive Integrated Moving Average (HF-ARIMA) or “HF-A” for short, combines various separate forecasting models and uses heuristics to deal with the difficulties of forecasting complicated patterns and meeting inequality restrictions. The HF-ARIMA model is a modification of the ARIMA model that incorporates both spatial and temporal dimensions of data. The ARIMA model is a type of time series forecasting method that uses the past values of the series to predict its future values. It is a combination of autoregressive (AR) and moving average (MA) models. The lack of a flexible forecasting framework that can precisely handle hierarchical time series (HTS) with seasonal trends is the research gap addressed in the paper. The proposed framework addresses the difficulties of forecasting HTS with inequality constraints and seasonal patterns by utilizing the hierarchical relational structure of HTS, incorporating various individual forecasting models, and combining heuristics based on the HTS dataset’s own characterization. The suggested forecasting method for hierarchical time series with seasonal trends evaluated using a real-world web traffic dataset called MEDIA. The data, which spans the dates of January 1, 2013, and March 15, 2015, shows weekly and yearly seasonal patterns along with a website hierarchy. The authors evaluated and compared the performance of the various methods by computing the average mean absolute percentage error (Avg-MAPE) to gauge forecasting accuracy. The experiments demonstrated that the proposed framework outperforms alternative methods in terms of accuracy [11].

As per Taylor and Letham study, Prophet is one of a machine learning model for time series forecasting developed by Facebook’s core data science team. It is a powerful tool for time series analysis and forecasting that uses a generalized additive model (GAM) with seasonality, holidays, and trend components to model time series data. Prophet is based on Bayesian modeling techniques and uses a decomposable time series model to capture different time series components and their interactions. It also includes a range of built-in features that make it easy to handle missing data, outliers, and trend changes [12].

S. Srivastava’s article examines website traffic data from an online marketplace for a week, which was preprocessed by aggregating events, filling in missing values, and removing outliers. The dataset consists of three primary variables - user_id, event, and timestamp. The article compares different models such as ARIMA, SARIMA, and Prophet for forecasting web traffic and concludes that Prophet is the most effective model [13].

The paper of Subashini et al. titled “Forecasting website traffic using prophet time series model” presents the Prophet time series model as a practical and accessible approach to forecasting website traffic. The authors suggest that the Prophet model is a good choice for producing quick and accurate forecasts of website traffic, particularly for those with good domain knowledge but lacking technical skills in forecasting models. The paper also explores the concept of “holiday” in forecasting and discusses how it can improve future predictions. The paper does not provide a detailed description of how the dataset was collected and preprocessed. However, it mentions that the training datasets were downloaded from Google Analytics. The paper evaluates the performance of the Prophet model by comparing its predictions with the actual website traffic data. Although the paper claims that the Prophet model is a good choice for producing quick and accurate forecasts of website traffic, it does not provide specific metrics or statistical analysis to support this claim [14]. However, further research is needed to provide a more detailed evaluation and comparison of the Prophet model with other time series models and to incorporate external factors into the forecasting process.

Vinayakumar et al. [15], in their study titled “Applying Deep Learning Approaches for Network Traffic Prediction,” state that Recurrent Neural Networks are a family of techniques that are frequently used for modeling time series data, with the aim of predicting future data based on past information. RNNs include several network architectures, such as Simple RNN, Long Short-Term Memory, Gated Recurrent Unit, and Identity Recurrent Unit, which are capable of learning temporal patterns and long-range dependencies in large sequences of arbitrary length.

According to research by Zhou et al. [3], the Generative Adversarial Network (GAN), a deep learning model, uses two neural networks to produce false new data. The discriminator and generator are the names of the two neural networks. In their study, the authors assess the performance of conventional statistical models against a Generative Adversarial Network (GAN) model for time-series web traffic forecasting that employs LSTM as a generator and Multilayer Perceptron (MLP) as a discriminator. They suggest a hybrid approach that integrates the four conventional approaches, and they discover that the GAN model exhibits promising outcomes in comparison to the conventional statistical techniques. However, the authors note the lack of evaluation metrics for GAN models, making it difficult to judge the goodness of the forecasted time series. The authors also use a novel way of seasonal, trend and cycle pattern decomposing method for the specific time series daily data. They use two datasets for the study: the Kaggle web traffic competition hosted by Google in 2017 and the M4 time series competition. They compare the accuracy of the statistical and GAN models with the prophet forecasting library from Facebook and find that there is no significant difference in accuracy between the hybrid methods and GAN model for this specific time series forecasting. The authors mention that the GAN model has two major limitations, including unstable training processes and lack of metrics, which require further research. In general, the paper offers a novel method for time series web traffic forecasting and emphasizes the need for additional study to address the limitations of GAN models for real-world use.

In their study, Casado-Vara et al [16], created an artificial intelligence-based architecture for web traffic forecasting that leverages LSTM for time series forecasting and focuses on capturing seasonality patterns and long-term trends. They present a training system with a parameter server that improves LSTM training using the Downpour strategy and achieves good quality forecasts in the seven dominant languages of the dataset. The dataset used in the study collected using a web scraper that collects network traffic data from Wikipedia’s API. The data cleaned and analyzed using Python data analysis libraries to extract features and identify hidden patterns. The LSTM model used for forecasting the flow of page views on websites in the short and medium term. The evaluation results show promising results, but the model struggles to predict the largest peaks in the time series. So that the model was not able to predict the largest peaks that occur in the time series since they assume anomalous behavior of the time series. The authors suggest future research to deepen the hidden pattern extraction and investigate the unsupervised model proposed in previous papers.

In our thorough literature review, we found a scarcity of research papers specifically investigating the use of biLSTM, GRU, biGRU, and attention-based models for web traffic forecasting. However, we did come across a comprehensive study that focused on related domains, including network traffic prediction and computer resource usage forecasting. While the study did not directly examine the application of these models in web traffic forecasting, it emphasized their importance in other domains. Consequently, this review provides a valuable starting point for future research in the field, presenting potential avenues to explore for enhancing web traffic forecasting techniques [15, 17, 18, and 19].

Table 2.1 presents a summary of the reviewed papers on web traffic forecasting.

Table 2-1 Web traffic analysis and forecasting review

Author	Title	Method	Dataset	Contribution
[11]	A Flexible Forecasting Framework for Hierarchical Time Series with Seasonal Patterns: A Case Study of Web Traffic	The authors propose a new framework for hierarchical forecasting (HF) that combines the advantages of different methods. They call it HF-ARIMA.	They use a real web traffic dataset from a web portal site called MEDIA, which has daily Page Views for different web pages, which cover the date from 01/01/2013 to 15/03/2015. MEDIA has weekly and yearly patterns	They compare their model with three other methods: SARIMA, GP, and LDS and show that their method has the lowest average absolute percentage error rate (Avg-MAPI).
[1]	Forecast Web Traffic Time Series Using ARIMA Model	They use an ARIMA model.	The data having feature page name, date, and visits are used.	They create a reliable forecasting model for predicting future traffic to Wikipedia pages

[8]	Web Traffic Prediction of Wikipedia Page	The authors used an RNN seq2seq model with Encoder/decoder Architecture to rebuild a winning model from a Kaggle competition.	Google’s Web Traffic dataset, which has daily views of approximately 145,000 Wikipedia articles from July 2015 1st, 2015 up to September 11th, 2017 used.	They rebuilt an existing model and added new features to improve its efficiency. They experimented with different combinations of features, including capturing page popularity on various time scales
[32]	Web Traffic Forecasting Using ARIMA and LSTM	ARIMA and LSTM algorithms are used	Dataset adopted for this project is daily views of Wikipedia articles provided by Kaggle comprising roughly 145,000 records. The dataset involves two fields, date and page.	To achieve improved precision and accuracy, an ensemble approach is employed by combining the two models together.
[21]	Web Traffic Time Series Forecasting using ARIMA And LSTM RNN	They use LSTM and ARIMA models	The authors use Wikipedia’s page view API to get the latest data on daily visits to any post. The data is in JSON format and has Dates and Visits fields.	Their system effectively captures seasonal patterns and long-term trends. They suggest that incorporating additional information such as holidays, day of the week, language, and region could further enhance their model’s ability to

				accurately capture the highs and lows of web traffic.
[16]	Web Traffic Time Series Forecasting Using LSTM Neural Networks with Distributed Asynchronous Training	They use a distributed architecture with LSTM models that are trained asynchronously with Downpour.	They use their own scraper to get1085 dates data from 1 January 2018 to 31 December 2020 for the top 1000 pages in seven languages of Wikipedia.	The authors forecast each page views for Wikipedia pages in different languages. They use MAE to measure the accuracy of their model.
[20]	Time Series Forecasting Using Exponential Smoothing To Predict The Number of Website Visitor of Sebelas Maret University	They used Exponential Smoothing method.	They used data from Google Analytics from January 2008 to June 2014. They split the data into training and testing sets. They use 60 months of data for training and 6 months for testing.	The authors has able to forecast the website visitor of UNS (uns.ac.id). They use MAPE to evaluate the forecasting accuracy.

[3]	Comparative study on the time series forecasting of web traffic based on statistical model and Generative Adversarial model	The authors use a Generative Adversarial Model (GAN) with LSTM as generator and Multi-Layer Perceptron (MLP) as discriminator.	They utilized the Kaggle dataset, which obtained from Wikipedia for a competition hosted by Google in 2017. They focused on a subset of this dataset, specifically consisting of 100,000 time series. This subset provided daily data for multiple web pages, serving as the basis for their analysis and experiments.	They compare their method with traditional statistical methods and find no significant difference in accuracy. They also use a combination of four traditional methods to lower the error rate by 10 to 20 percentage points.

2.6 **Review of works on Web Traffic Forecasting Datasets**

The authors used Google Analytics to collect monthly website traffic data of Sebelas Maret University (uns.ac.id) from January 2008 to June 2014. They split the data into 60 training points (January 2009 to December 2013) and 6 testing points (January 2014 to June 2014). They applied a smoothing method that accounted for the overall, trend, and seasonal components of the data, with optimal parameters of alpha = 0.24, beta = 0.10, and gamma = 0.01. They used the training data to estimate the parameters and the testing data to evaluate the forecasts [20].

Liu et al. [11] used a real-world dataset of web traffic from a popular, MEDIA, web portal site to test their proposed method. The dataset, MEDIA, is a hierarchical dataset that shows the daily page views (PVs) of the “media” homepage and its subpages, such as sports, auto, finance, news, and celebrity. The “sports” subpage also has two subpages for different channels (desktop or mobile). The dataset spans from January 1, 2013, to March 15, 2015, and has weekly and yearly seasonality. The researchers used the data before the last month for training and the data in the last month for testing. They applied their method to the training data to forecast the web traffic for the testing data.

Petluri’s study aimed to build a trustworthy model for forecasting web traffic of Wikipedia pages. They used Google’s Web Traffic Time Series Forecasting dataset, which has about 145,000 Wikipedia articles with daily views from July 1, 2015 to September 11, 2017, to test their model. However, the dataset had some uncertainty because it did not distinguish between zero and missing values in traffic data, which affected the predictions. The study used 740 days of training data to forecast traffic values from July 1 2015 to September 11, 2017 and compared them with the actual data. The study also added a rolling window to capture trends and calculated weekly, monthly, quarterly, and yearly page popularity. They computed the rolling median with window sizes of 7, 30, 90, and 180 days, normalized each with zero mean and unit variance, and took the median of medians as a single feature for each sample instead of treating each forecast window as a separate feature [2].

The authors of “Web Traffic Time Series Forecasting using ARIMA and LSTM RNN” used the Wikipedia page-view API to get the daily page views of any specific article. The API provided the data in JSON format, which the authors parsed to extract the dates and the number of visits. They then converted the data into a data frame to use it for the prediction model [21].

According to Tambe et al. [1], the paper used the Kaggle dataset of web traffic time-series forecasting to build a forecasting model based on the ARIMA (Autoregressive Integrated Moving Average) model. The dataset contained the name, date, and number of visits of Wikipedia pages for a year. The authors used these variables to train their model and predict the future traffic for each page. The ARIMA model produced accurate forecasts of web traffic for each page [1].

Casado-Vara et al. [16] study used an updated Wikipedia web traffic dataset for 2018 up to 2020 to build and test a sophisticated recurrent neural network called Long Short-Term Memory for predicting web page views. The dataset have web pages and page views of the most popular pages in seven languages. The dataset was processed to extract the features and hidden patterns in the data. The LSTM has trained using pattern detection techniques and distributed training, and a new dataset created for this project was used to validate the model. The study used 20% of the dataset for testing or validating, and 80% for training.

In general, several researchers appear to have used Kaggle datasets, based on our observation of their datasets. The current research landscape lacks diversity and inclusion in datasets, particularly in the context of web traffic forecasting. To address this gap and incorporate contextual information from the local region, our intention is to develop a time series web traffic forecasting dataset using data obtained from a local organization. By doing so, we aim to provide a more comprehensive understanding of time series prediction and enable researchers to explore a broader range of applications by incorporating data from a previously unexplored domain. Our contributions will help in the creation of more precise and reliable prediction models as well as widen the scope of research in this area.

2.7 Summary

In this chapter, we conduct a comprehensive survey of the current state-of-the-art in web traffic analysis and prediction. We delve into the diverse range of approaches employed in web traffic analysis and prediction, including deep learning models, datasets used, and evaluation criteria employed. The chapter explores existing studies that have utilized various techniques and methods for web traffic prediction, shedding light on how these studies evaluate their results. Furthermore, we critically examine the literature to identify the limitations present in the available studies, thereby highlighting areas that require further investigation and improvement.

The chapter seeks to assess the existing challenges of the studies that researchers have employed to forecast web traffic. Prediction techniques are the specific procedures and approaches that have been followed to use the deep learning models and methods on the web traffic data, including the data-preprocessing, model training and testing, and the hyperparameter tuning. There are different kinds of deep learning models that have been applied to web traffic forecasting, such as long short-term memory, gated recurrent unit, both models with attention mechanisms.

The chapter also covers various aspects of web traffic analysis and prediction, such as the types of data sets and the evaluation criteria. Data sets refer to the sources and features of the web traffic data used for analysis and prediction, such as the domain, the granularity, the features, the size, etc. Metrics refer to the numerical measures used to compare and assess the accuracy and efficiency of the models and methods, such as mean absolute error, mean square error and root mean square error.

Chapter Three

3 Design and Methodology

3.1 Introduction

This chapter describes the methodology followed to achieve the objectives of this thesis, which include the proposed web traffic forecasting model architecture. The methodology provides a structured framework for conducting the study and outlines the steps taken at each stage to ensure that the research performed in a systematic and rigorous way.

The aim of this study is to develop a web traffic-forecasting model that provides businesses with reliable and accurate forecasts to support informed decision-making. A structured approach with three main sections used to achieve this goal. The first section literature review, which involves reviewing relevant literature to gain a comprehensive understanding of the domain. This leads to research problem formulation and the development of objectives for the study. The second section is model design, implementation and evaluation, which involves building the web traffic-forecasting model. The model is designed to detect patterns and trends in the web traffic data, and to use this information to make accurate forecasts of web traffic trends. The performance of the models is evaluated using appropriate metrics. The metrics are used to measure the effectiveness of the model in forecasting web traffic trends, and to identify areas for improvement. Finally, this study provides a summary of the main findings and suggests future directions for advancing research in web traffic forecasting. It highlights the need for further investigation and improvement in this field to enhance the accuracy and effectiveness of predictions.

3.2 Research Design

In order to build an effective web traffic-forecasting model, a systematic study approach needs to be developed. This study approach consists of a set of principles that guide the research process as illustrated in Figure 3.1. The approach has been broken down into three main sections [16].

The first section of the approach involves identifying the problem domain, which requires a comprehensive understanding of the issue through a review of relevant literature. This process leads to formulating the general and specific objectives of the thesis, which are to improve the accuracy of web traffic forecasts by developing a novel model and incorporating new data sources. The general objective is motivated by the lack of geographical and domain diversity in previous studies, which limits the generalizability and applicability of their results.

Following the initial phase of problem domain identification, our study progresses to the subsequent step, which involves both data collection and model development. We acquire web traffic data from a local organization. The subsequent step after data collection is dedicated to model design and implementation. The model implementation stage involves the training and testing of our model using the data collected in the previous stage. To enhance the accuracy and performance of web traffic forecasting, we employ a range of deep learning algorithms. These algorithms include LSTM, GRU, bidirectional LSTM, bidirectional GRU, as well as bidirectional LSTM with attention and bidirectional GRU with attention. In addition to selecting these algorithms, we carefully tune their hyperparameters to optimize their effectiveness. Hyperparameters are crucial settings that influence the behavior and performance of deep learning models. By fine-tuning these hyperparameters, we aim to achieve the best possible performance and accuracy in our web traffic-forecasting model. Through this approach, we leverage the capabilities of various deep learning algorithms and optimize their hyperparameters to create a robust and accurate model for web traffic forecasting.

Once the model has undergone the training and prediction process, we proceed to evaluate its performance using a separate test dataset that is distinct from the training data. This testing phase allows us to assess the model's ability to generalize and handle new data effectively. In the stage of evaluating the models, we compute various metrics to assess their performance. These metrics encompass mean absolute error, mean squared error, and root mean squared error, which serve as quantitative measures of the model's accuracy and predictive capabilities. They serve as measurements of the disparities between the predicted web traffic values and the actual values within the test dataset. Lower values of these metrics indicate better performance of the model. A lower MAE, MSE, and RMSE signify that the model's predictions are closer to the actual values, indicating higher accuracy and precision in forecasting web traffic.

In the final stage of our study, we thoroughly discuss and draw conclusions based on the evaluation phase. Evaluation metrics play a crucial role in assessing the effectiveness of our web traffic-forecasting model. Through the analysis of metrics such as MAE, MSE, and RMSE, we gain valuable insights into the model's predictive performance. These metrics enable us to make informed judgments about the model's accuracy and reliability in capturing web traffic patterns. The conclusions drawn from this evaluation provide valuable insights and guidance for assessing the model's suitability and effectiveness in our specific use cases. By analyzing the evaluation results and findings, we can make informed decisions about the model's suitability and its potential impact on achieving our desired outcomes.

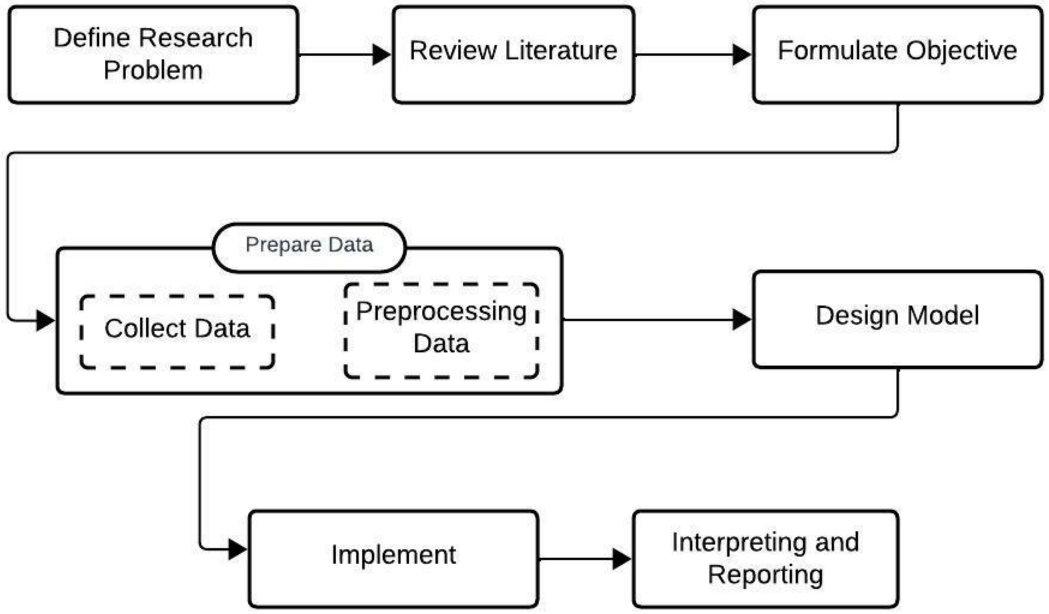

Figure 3.1 Research Flowchart

3.3 Architecture for Web Traffic Analysis and Forecasting Model

This section describes in detail the steps involved in a developing deep learning based model for web traffic forecasting. Figure 3.2 provides a general breakdown of the entire process. The initial phase of the data collection process focuses on extracting the necessary data, which includes relevant features such as date and page session. These features play a crucial role in understanding web traffic patterns and provide valuable insights for analysis. Following data collection, the next step is exploratory data analysis. Exploratory Data Analysis (EDA) involves examining the collected data to gain a deeper understanding of its characteristics, identify any patterns or trends, and establish connections between variables. Through EDA, we can uncover potential data quality concerns or anomalies that need to be addressed during the preprocessing stage.

Once the data has been collected and explored, the preprocessing stage becomes essential. This stage aims to ensure the accuracy and suitability of the data for effective model training. Various preprocessing methods employed to achieve this, including handling missing values, normalizing, reshaping and data splitting into training and testing sets.

By performing these steps, including data collection, exploratory data analysis, and preprocessing, we ensure that the data used for model training is accurate, reliable, and well prepared. This sets the foundation for building a robust and effective model for web traffic analysis and prediction.

In the training phase, the deep learning models, including LSTM, GRU, bidirectional LSTM, bidirectional GRU, bidirectional LSTM with attention, and bidirectional GRU with attention, are constructed and trained using the designated training set. Hyperparameter tuning employed to find the optimal architecture for these models.

After constructing the models with optimal hyperparameters, we move on to the prediction phase. Here, the trained models are utilized to make predictions of web traffic. The objective of this phase is to generate output values or estimates based on the provided input data. By leveraging the models trained with the finest hyperparameters, we can obtain predictions that are anticipated to be more accurate and reliable.

Then in the final steps, the models are tested using a separate testing set, and their performance evaluated using metrics like MAE, MSE, and RMSE. Ultimately, the best-performing model selected based on the evaluation metrics. The selection of the best model is crucial as it determines the model’s suitability for real-world applications and its potential for accurate web traffic forecasting.

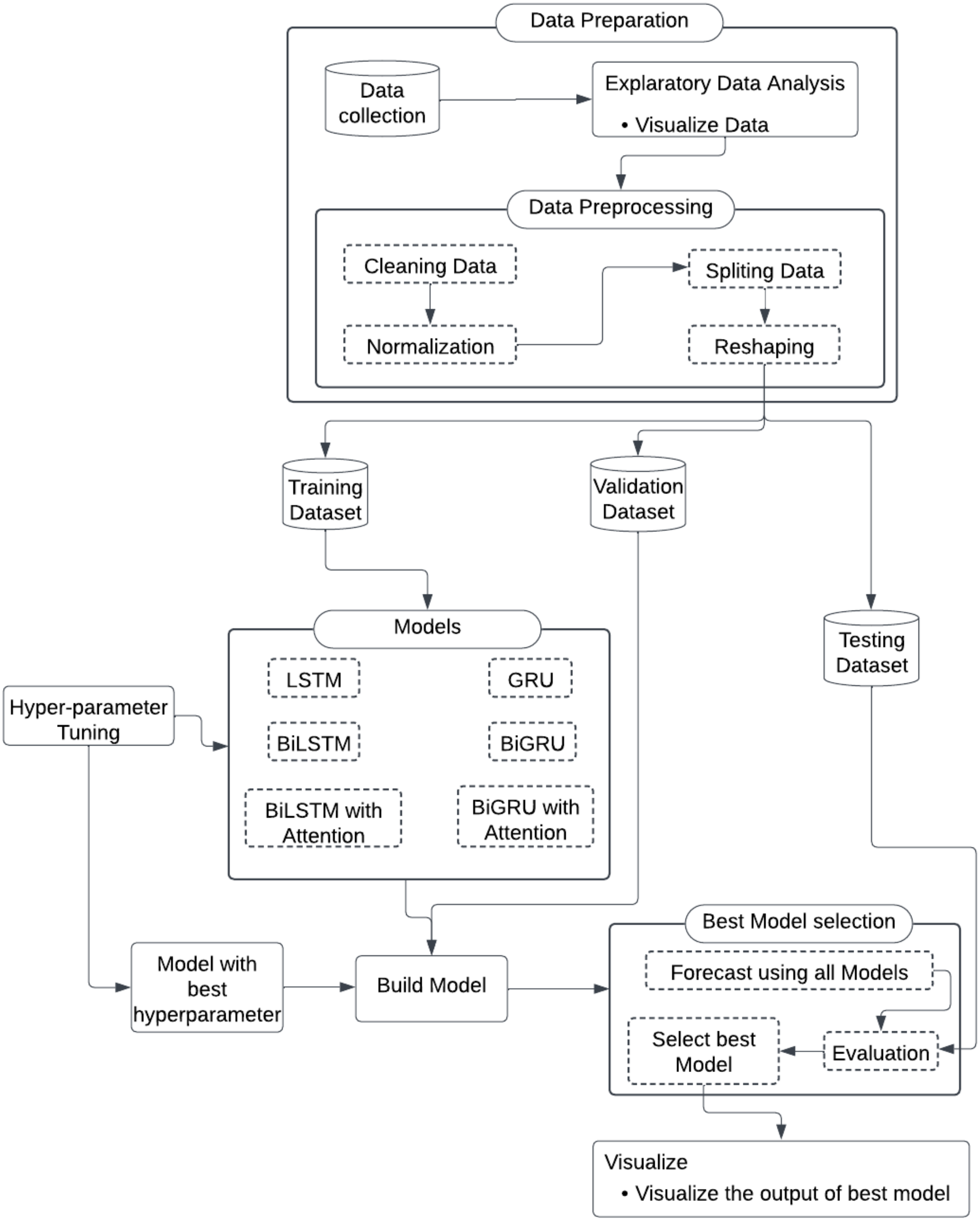

Figure 3.2 Proposed methodology for web traffic analysis and forecasting using deep learning

3.3.1 Data Collection

In this analysis, we carefully selected relevant features for web traffic forecasting based on previous studies [1, 8, 11, and 21]. The dataset used in our study consists of 32 web pages, which collected from the internal communication website of the Commercial Bank of Ethiopia. We utilized secondary data type for our analysis, specifically data collected by the bank using XT visitor counter analytics tools. The collected information, stored in CSV format, includes details such as dates, visitor numbers (session), and page IDs. The dataset comprises numerical statistics that represent website traffic patterns.

For data collection, we used the recorded daily visitor counts for each of the 32 web pages over a period of seven years, from January 1, 2016, to January 3, 2023. This resulted in a dataset containing a total of 2,560 days. With 32 web pages considered, our dataset consists of 81,920 data points (2,560 days * 32 features) available for analysis and examination.

3.3.2 Exploratory Data Analysis

The objective of exploratory data analysis is to uncover hidden information, patterns, and irregularities in the data, as well as extract relevant information. EDA employs techniques of data visualization to facilitate comprehension of the data. Prior to preprocessing, data exploration aids in examining the characteristics of the data, identifying patterns, handling missing values, and selecting potential features for modeling. This stage is crucial for conducting data preprocessing and determining the most suitable strategies for preprocessing the data [16].

When conducting exploratory data analysis, it is common to visualize web traffic data as a time series. In this particular study, a time-series data refers to data collected over a specific time interval, which in this case is daily. By visualizing web traffic data as a time series, we can recognize patterns, trends, and seasonality within the data. Seasonality pertains to regular patterns that occur at specific time intervals, such as daily, weekly, or monthly occurrences. By identifying these patterns, we can gain insights into the behavior of website visitors and comprehend how they engage with the website.

3.3.3 Data Preprocessing

Next, after the collected data explored, the raw data undergoes preprocessing to prepare it for further analysis. By performing these preprocessing tasks, the raw data cleaned and transformed into a consistent and suitable format for subsequent analysis. This ensures that the data used in deep learning projects is accurate, complete, and ready for effective training and forecasting. The following tasks performed as essential steps in deep learning analysis.

- Data Cleaning: In the data cleaning process, various problems such as missing data identified and addressed. The missing values replaced with zeros, indicating that the page not visited due to some issue or reason. Handling missing values is crucial to ensure the completeness and reliability of the data used for model training.

  By addressing missing values, the data cleaned and prepared for further analysis. These steps help to eliminate potential issues caused by faulty data, ensuring the accuracy and integrity of the results obtained from the subsequent modeling and analysis stages.
- Data Normalization: In order to ensure that features are on a comparable scale and to enhance the stability and performance of deep learning models, a normalization technique employed. This technique involves rescaling the feature values to a predefined range, typically between zero and one.

  The purpose of normalization is to reduce the complexity of the problem and facilitate the gradient descent step during model training. By rescaling the feature values, we can prevent certain features with larger numerical ranges from dominating the model's learning process.
- Data splitting:
  - Divide into input sequences and a target variable:

Input sequences are subsets of consecutive data points, typically seven days, used as input for prediction models. The target variable is the subsequent data point following the input sequence and represents the data to be predicted. By dividing the dataset in this way, a relationship established between past data and future data, allowing the prediction models to learn patterns and make predictions about future values of the target variable.

  - Divide into Training and Testing:

The dataset is divided into three parts: a training set, a validation set, and a testing set. The training set comprises 80% of the data. Within the training set, 20% is further allocated as a validation dataset. The remaining 20% of the data is designated as the testing set. The purpose of this division is to effectively train the model on a portion of the data and then evaluate its performance on unseen data.

The testing set used to assess the model's performance on unseen data. By evaluating the model's predictions against the actual values in the testing set, we can measure its ability to generalize and accurately predict web traffic patterns.

This separation of data into training and testing sets helps to mitigate the risk of overfitting, which occurs when a model becomes overly specialized to the training data and performs poorly on new, unseen data. By utilizing separate subsets for training and evaluation, we can gauge the model's effectiveness in predicting web traffic patterns accurately and assess its generalization ability.

- Data Reshaping: As a part of the preprocessing phase in our study, we perform data reshaping. This involves transforming the data from a two-dimensional format to a three-dimensional format to meet the requirements of a specific model. The reshaping process organizes the data into a structure of (samples, time steps, features). This ensures that the data aligned properly for input into the chosen model.

3.3.4 Hyper parameter Tuning

In our analysis, we have conducted hyperparameter tuning to identify the optimal hyperparameters for our deep learning models. Hyperparameters are parameters that are predefined before the training process and have a significant impact on the model's performance, as they control various aspects of its behavior [54].

During the hyperparameter tuning process, we focused on selecting the best hyperparameters for our models. These included important values such as the learning rate, batch size, and number of epochs, number of hidden layers, and number of neurons in each hidden layer, activation functions, regularization parameters, and dropout rate.

To systematically explore the hyperparameter space, we employed the Random Search approach, which involves sampling different values for each hyperparameter. This enabled us to automate the search for optimal hyperparameters, saving us valuable time and effort compared to manual selection or trial-and-error methods. It is worth noting that hyperparameter tuning is an iterative process, and we performed multiple iterations to find the best combination of hyperparameters for our web traffic forecasting analysis.

By performing hyperparameter tuning, we ensured that our deep learning models were optimized and capable of achieving the best possible performance for our web traffic forecasting analysis.

3.3.5 Model Training

After tuning processing, the next important stage in the data analysis process is model training. Choosing an appropriate model is crucial for accurately representing the data and generating reliable predictions [4]. In this study, six different models from the recurrent neural network family employed to forecast web traffic, which is a type of sequential data.

Web traffic data presents certain challenges for modeling due to characteristics like long-term dependencies and non-linearity. Long-term dependencies imply that the current observation influenced not only by immediate observations but also by previous and future ones. Non-linearity means that the relationship between input and output variables is not a simple linear function but rather a complex and dynamic one that can change over time [5, 7].

To address these challenges, the study utilizes various models from the RNN family, including LSTM, GRU, Bidirectional LSTM, Bidirectional GRU, and Bidirectional LSTM with attention, and Bidirectional GRU with attention. RNNs are a class of neural networks designed to process sequential data. Unlike other neural networks, RNNs possess recurrent connections, enabling the model to retain and update information from previous time steps and capture long-term dependencies in the data. Additionally, RNNs employ activation functions that introduce non-linearity to the model's output, enabling it to learn intricate and dynamic relationships [4, 7, 10, 15, 16].

Some of the models used in this study incorporate additional features to enhance their performance. Bidirectional connections enable the model to learn from both past and future information by processing the input sequence in both forward and backward directions and combining the outputs. Attention mechanisms enable the model to focus on the most relevant parts of the input sequence by assigning different weights to different time steps and utilizing these weights to compute the output [42, 45, 46, 51, 52].

These models are expected to perform well in forecasting web traffic as they can learn from data patterns and trends, adapt to changes and uncertainties in the data, and effectively capture long-term dependencies and non-linear relationships.

3.3.6 Prediction

Once the training phase concludes, we proceed to the prediction process. In this phase, we utilize the trained models that have learned patterns and relationships from the training data to generate predictions.

During the prediction phase, we make use of all the trained models to generate forecasts. Each model has learned patterns and relationships from the training data, enabling it to estimate future values. By employing multiple models, we obtain a range of predictions, allowing us to evaluate their performance and their ability to capture underlying patterns in the data.

3.3.7 Evaluation

In this section, we provide an evaluation of the deep learning models proposed for web traffic forecasting. We assess the individual performance of each model by comparing the predictions they generate. This evaluation enables us to measure the prediction capabilities of each model and determine their effectiveness in extracting the underlying information from the data.

The accuracy of the generated predictions by the models is evaluated using three metrics. Those are mean absolute error, mean squared error, and root mean squared error. These metrics provide a quantitative measure of the average error between the predicted and actual web traffic values. A lower value for these error metrics indicates better performance of the model. By evaluating the models on the testing data, we have assessed their ability of forecasting web traffic. The MAE measures the average absolute difference between the predicted and actual values, providing an overall estimation of the model's accuracy. The MSE calculates the average squared difference between the predicted and actual values, giving more weight to larger errors. The RMSE is the square root of the MSE and provides a measure of the average magnitude of the errors. Through the utilization of these evaluation metrics, we can compare the performance of various deep learning models and determine the model that exhibits the lowest error rates. The aim is to select the model with the lowest MAE, MSE, and RMSE values, indicating its ability to provide precise predictions for web traffic values.

Chapter Four

4 Implementation, Experimental Results and Discussion

4.1 Implementation Tools

In our experiment on web traffic analysis and forecasting, we utilized the Google Cloud Platform, specifically Google Colab, which provides access to a GPU with 12 GB RAM. This GPU-enabled environment allows for faster computation and efficient training of deep learning models.

To evaluate our model, we use Python with TensorFlow and Keras. TensorFlow offers powerful tools for neural networks, and Keras simplifies the process with a user-friendly interface. Our research leverages these tools to analyze and forecast web traffic patterns efficiently. TensorFlow and Keras have been instrumental in creating and training models that accurately predict web traffic behavior. These frameworks offer extensive functionality and ease of use, enabling us to develop efficient and robust models for forecasting.

Scikit-learn, a popular machine-learning library, has also played a significant role in our analysis. It provides tools for data preprocessing, evaluation, and visualization. By combining Scikit-learn with TensorFlow and Keras, we have a powerful set of tools for building and evaluating our web traffic-forecasting model.

Pandas, Matplotlib, and NumPy has used for data manipulation, analysis, and visualization. Pandas provides tools for data cleaning, normalization, and exploration. Matplotlib offers various plotting functions that work well with Pandas and NumPy, facilitating the creation of visualizations for web traffic data and model evaluation.

4.2 Data Description

Our proposed method assessed using a real-world web traffic dataset obtained from the website of a Commercial Bank of Ethiopia. The raw dataset is in CSV format. The recorded data spans from January 1, 2016, to January 3, 2023, covering a total of 2,560 days. The dataset contains traffic records for 32 pages on a daily basis. There are 2,560 observations, representing the daily visitor count for each page. This results in a total of 81,920 data points (32 pages multiplied by 2,560 observations). The dataset includes features such as the date and PageIDs to identify the specific page.

4.3 **Data Analysis**

Figure 4.1 depicts the number of website visitors over the course of the year. This graph provides insights into the data pattern for all pages, which is crucial for forecasting purposes. Upon observing Figure 4.1, it is evident that the sessions exhibit an incremental pattern, indicating a substantial increase in the number of visitors. The graph reveals that from 2019 to the middle of 2020, there was a noticeable decline in the number of visitors. When considering the data pattern over the years, it becomes apparent that website visitor statistics follow an upward trend. This trend suggests that the number of visitors to the website has been growing over time.

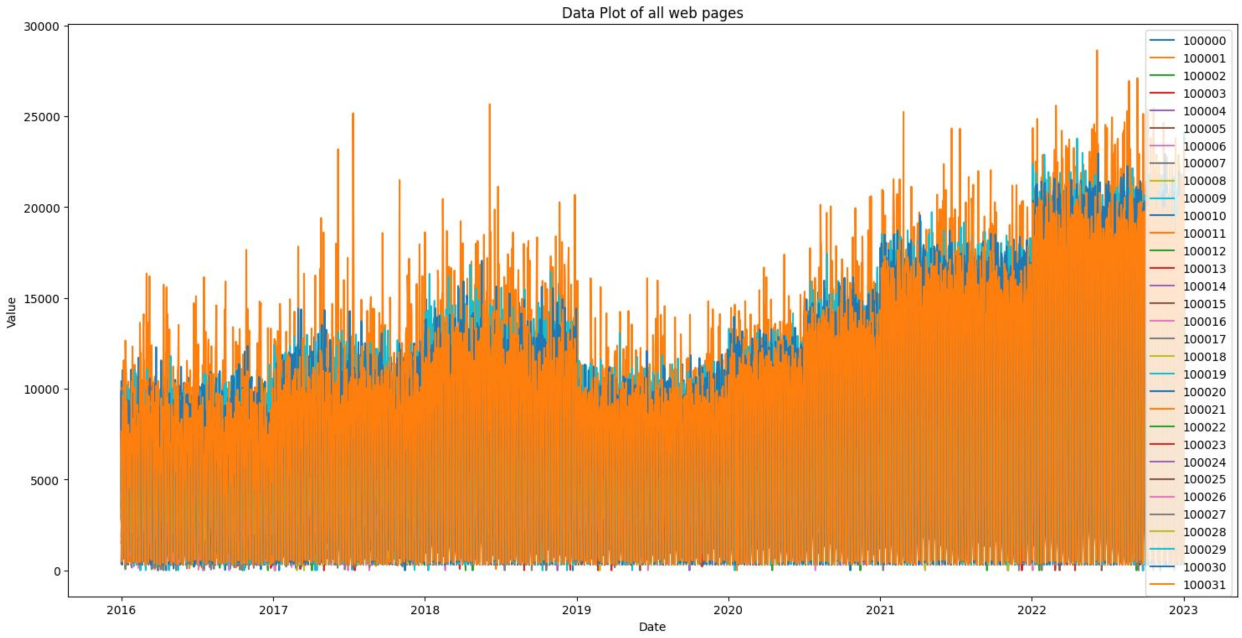

Figure 4.1 CBE website Visitor statistics from January 1, 2016 up to January 3, 2023

Figure 4.2 represents the statistical data for the first 30 dates of the website. It provides valuable insights into the behavior of website visitors during this period. One notable observation from the figure is that on weekends, specifically on Sundays, the number of visitors tends to be consistently around zero. This suggests a distinct pattern in visitor activity, where weekends see a decrease in website traffic compared to weekdays.

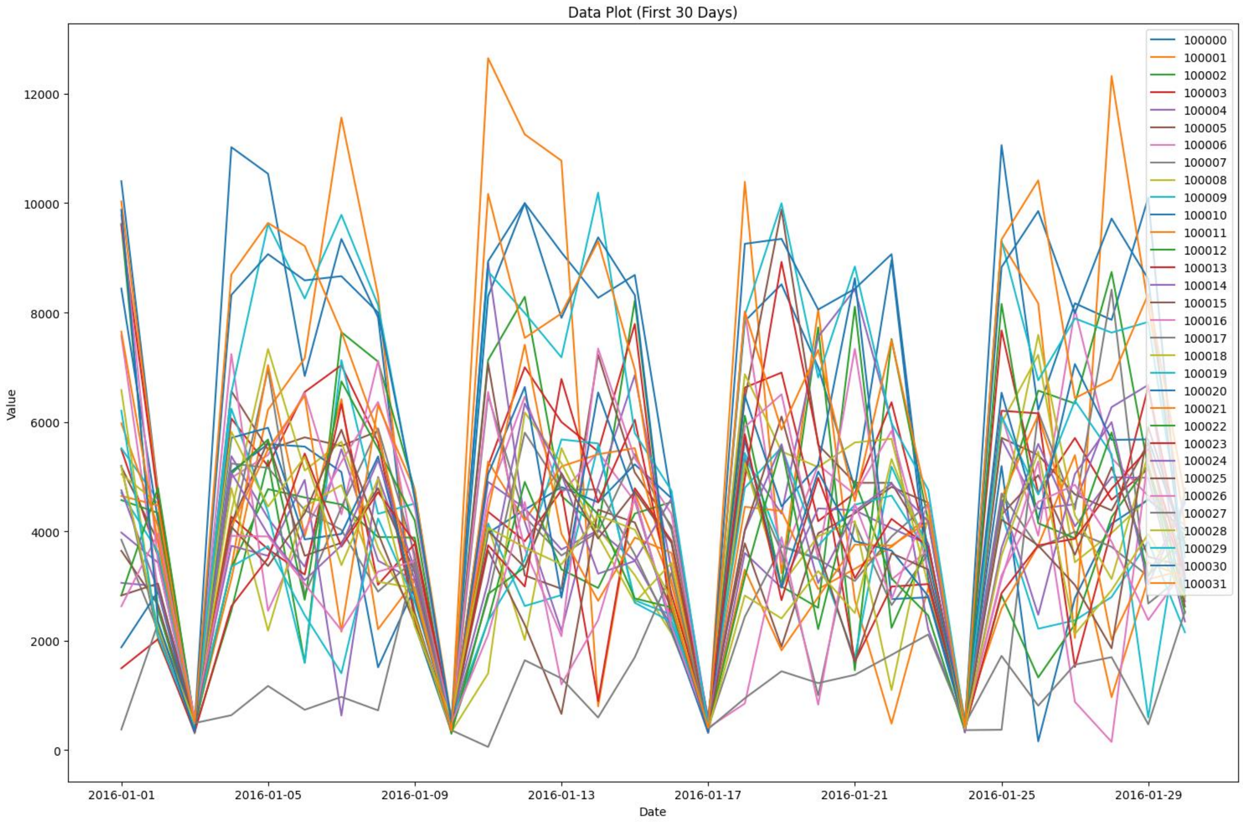

Figure 4.2 First month of CBE website statistics

In addition to the patterns observed in the graph, another valuable insight is that not all pages have the same number of visitors. There is a significant variation in visitor counts across different pages on the website. This discrepancy suggests that certain pages attract a high volume of visitors, while others may have relatively lower visitor numbers. Understanding this discrepancy is crucial for optimizing website performance and identifying areas that may require improvement. By analyzing the visitor distribution across pages, businesses can gain insights into the popularity and effectiveness of different website sections or content. This information can help prioritize resources and efforts towards enhancing the visitor experience on pages with lower engagement or improving high-traffic pages. Furthermore, this insight can also inform content strategy and marketing efforts. By identifying which pages attract the most visitors, businesses can focus their content creation and promotional activities accordingly, ensuring maximum visibility and engagement with their target audience.

Furthermore, upon analyzing the data pattern, it becomes apparent that there are two key components: a trend and a seasonal component. The trend component indicates an increase in general direction or tendency in the data over time. The presence of a trend suggests that there is a long-term changing pattern in the number of visitors to the website. Additionally, the seasonal component in the data pattern indicates recurring patterns or cycles that repeat over fixed intervals. In this case, the data shows a clear seasonal pattern where the number of visitors tends to follow a consistent trend over weekends.

By recognizing and understanding the trend and seasonal components in the data, it becomes possible to make informed decisions, such as adjusting internal communication strategies, or allocating resources, to accommodate the observed patterns and optimize website performance.

4.4 **Preprocessing**

The data underwent several preprocessing tasks to prepare it for model training. Initially, the dataset had seventy missing values, which were filled with zeros to ensure completeness. Next, the dataset was normalized using the ‘MinMaxScaler’ from scikit-learn, scaling the values between 0 and 1, which helps to improve model performance and training efficiency.

In order to generate input sequences and corresponding target values for each feature, the dataset was divided or split. The ‘time_steps’ variable was set to seven, considering seven consecutive time steps as input. The ‘X’ variable stored the input sequences, where each sequence contained ‘time_steps’ consecutive values from the dataset. The ‘y’ variable held the corresponding target values, which were the values following the input sequences. The data was shaped by three-dimensional format, with dimensions representing the number of sequences, elements in each sequence (loopback), and feature length. Both ‘X’ and ‘y’ were converted to NumPy arrays for further processing.

The dataset is split into three sets while maintaining an 80:20 ratio. The training set consists of 1633 days, the validation set consists of 409 days, and the test set consists of 511 days. Sequential data splitting performed to ensure that the temporal order of the time series problem preserved as depicted in Table 4.1.

The training data was stored in ‘X_train’ and ‘y_train’, representing a portion of input sequences and their corresponding target values. On the other hand, the testing data was stored in ‘X_test’ and ‘y_test’, containing the remaining portion of input sequences and target values. The ‘test_size’ parameter was set to 0.2, indicating that 20% of the data was allocated for testing purposes. Additionally, the ‘shuffle’ parameter was set to ‘False’ to ensure that the order of the data was maintained during the split.

The training set had 1633 samples, each with a sequence length of 7 time steps and a feature vector size of 32. ‘X_train’ had a shape of (1633, 7, 32), indicating 1633 training samples, each with a sequence of 7 time steps and a feature vector of size 32. ‘X_test’ had a shape of (511, 7, 32), indicating 511 test samples, each with a sequence of 7 time steps and a feature vector of size 32.

The label vectors in the training set, ‘y_train’, had a shape of (1633, 32), indicating 1633 training samples, each with a corresponding label vector of size 32. Similarly, the label vectors in the test set, ‘y_test’, had a shape of (511, 32), indicating 511 test samples, each with a corresponding label vector of size 32.

The shapes of the data indicate that we are dealing with a multivariate time series-forecasting problem. In this scenario, the goal is to predict multiple outputs (in this case, 32) based on the input sequence.

After performing necessary preprocessing steps such as handling missing values, creating input sequences, scaling, reshaping, and splitting the data, it is now prepared for model training.

Table 4-1 Prepared dataset

Total dataset	Feature	Time steps	Training set	Validation set	Testing set
2560	32	7	1633	409	511

4.5 **Baseline**

In our analysis, we employed the Auto Regressive Integrated Moving Average (ARIMA) model as the baseline model for comparison. The ARIMA model is a commonly used time series forecasting technique that takes into account the autoregressive (AR), integrated (I), and moving average (MA) components of the data.

To use ARIMA as the baseline model, we created input sequences of length seven, meaning each sequence contained seven elements. These elements represent daily data points, depending on the time scale of the analysis. Once the input sequence defined, we used the ARIMA model to make predictions based on this sequence. The model takes into account the historical values in the sequence to forecast the next value. The predicted value then compared with the actual target value to evaluate the accuracy of the model.

By comparing the predicted values with the actual values, we could assess the performance of the ARIMA model in predicting web traffic behavior. This comparison allowed us to gauge the effectiveness of the ARIMA model as a baseline and provided a benchmark for evaluating the performance of more advanced deep learning models.

4.6 **Hyper parameter Tuning**

The optimal hyper parameters for our models were determined through meticulous hyper parameter tuning using the Keras Tuner library. This involved systematically searching different parameter combinations using our training and validation data. By creating a comprehensive grid list of hyper parameter values, as outlined in Table 4.2, we thoroughly explored the entire parameter space to identify the best configuration for our models.

We employed the Keras Tuner library, utilizing the RandomSearch tuner, to perform the hyper parameter search. A function was implemented to create the model architecture using the hyper parameters configuration grid. The tuner has configured with essential parameters, such as the model building function, the validation loss as the objective, two executions per trial, and a maximum of fifty trials.

Table 4-2 Hyper parameter configuration used with Randomsearch method

Parameter Name	Values
“model_type”	LSTM, biLSTM, biLSTM_attention, GRU, biGRU, biGRU_attention
“hidden_ layer”	2, 3, 4, 5
“neurons”	32, 64, 128, 512
“dropout”	0.1, 0.2, 0.3, 0.4, 0.5
“optimizer”	adam, adamax
“activation”	relu, elu, tanh, ReLU, softmax
“learning_rate”	0.0001, .01, 0.1
“epochs”	10, 20, 30, 40
“batch_size”	4, 8, 16, 32, 64

After conducting an extensive search and evaluating various hyperparameter configurations with the training and validation data, the tuner identified the best model architecture. The optimal model architecture was obtained using tuner.get_best_models(num_models=1)[0], while the corresponding hyperparameters were extracted using tuner.get_best_hyperparameters()[0].values.

Following the completion of the hyperparameter tuning phase, we successfully determined the most effective values for each model, which are documented in Table 4.3. These optimized hyperparameters played a crucial role in achieving exceptional performance and accuracy in the subsequent analysis of web traffic forecasting, which followed the hyperparameter tuning process.

Table 4-3 Optimal Hyper parameters value list of Models

Parameter	Value					
	LSTM	biLSTM	biLSTM with attention	GRU	biGRU	biGRU with attention
hidden_layer	3	4	5	3	4	5
neurons	64	32	32	64	32	32
dropout	0.2	0.3	0.3	0.1	0.2	0.1
activation	Elu	elu	tanh	relu	relu	Elu
learning_rate	0.001	0.001	0.01	0.01	0.01	0.001
optimizer	Adam	adam	adamax	adam	adamax	Adam
epoch	30	30	30	30	20	20
batch_size	16	16	16	32	32	32

4.7 Model Building and Prediction

Once we have obtained the optimal hyperparameters through the process of hyperparameter tuning, we proceed to build the models using these carefully selected parameter values. The architectures of the models are then constructed based on these recommended hyperparameters, ensuring that the models are configured with the most suitable settings for optimal performance.

To start the model building task, we begin by loading the ‘best_hyperparameters’ dictionary from a file using the ‘pickle’ module. This dictionary holds the optimal hyperparameters for various types of models that we will be constructing.

Following that, we initialize empty lists and dictionaries to serve as containers for various results that we will collect during the model building process. These include predictions, training and validation losses, as well as evaluation metrics. By initializing these containers upfront, we can conveniently store and organize the relevant information as we progress with the model building and evaluation steps.

We have built a code to iterate over each model type and its corresponding best hyper parameters. It builds a model using the best hyper parameters and trains it on the entire training dataset. Early stopping used as a callback with a patience of three, allowing training to stop if the validation loss does not improve. During the training process, we maintain separate lists to record the training and validation losses. The graph, labeled as Figure 4.3, provides a visual representation of the results, utilizing different line styles and colors to distinguish between model types.

The graph depicts the training and validation losses over the course of training. Each model represented by a specific line style and color, allowing for a clear comparison between their performances.

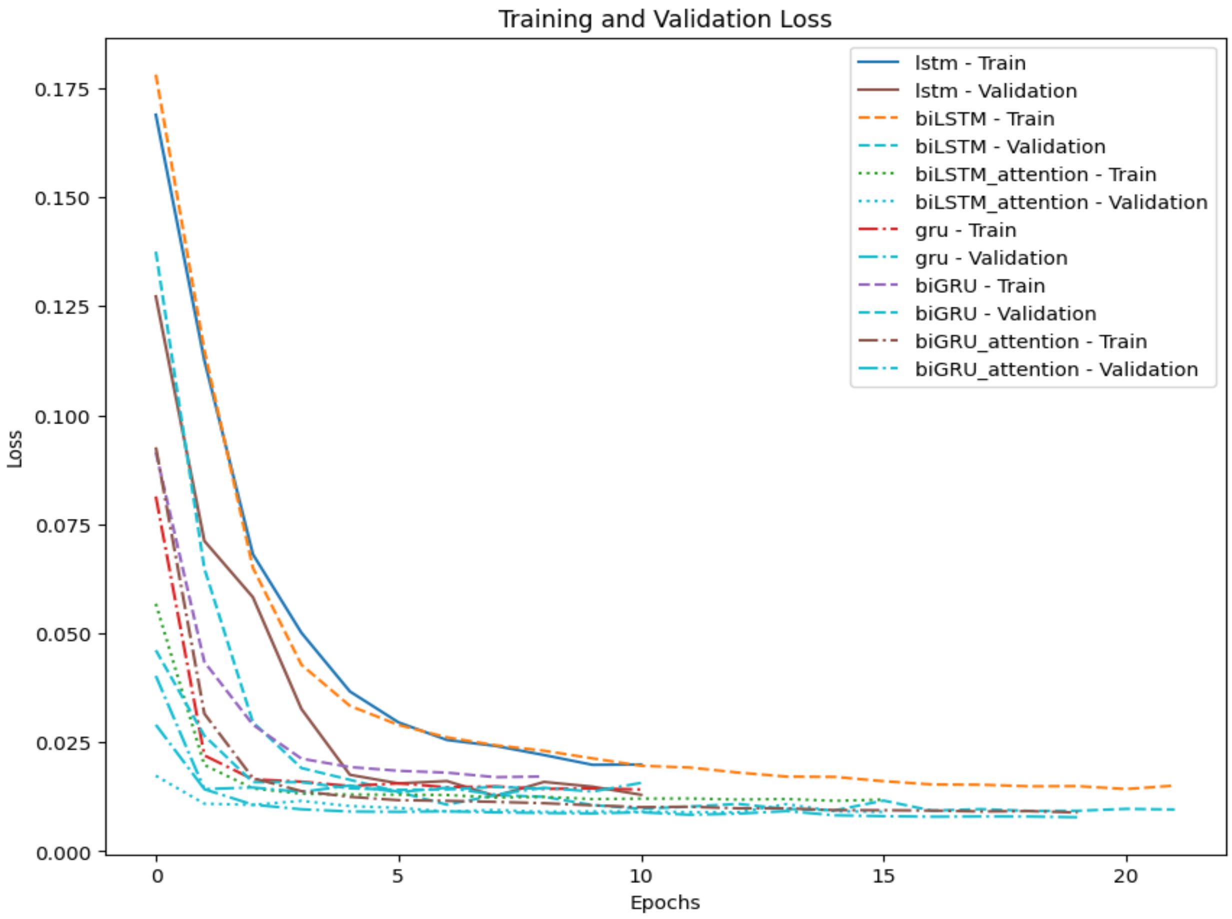

Figure 4.3 Training and Validation losses over epoch

As observed from the graph, both the training and validation losses consistently decrease throughout the training process, indicating that the models are effectively learning from the training data. This trend suggests that the models are adapting well to the provided dataset and improving their understanding of the underlying patterns.

To prevent overfitting and ensure model generalization, early stopping implemented. This technique halts the training process when the validation loss ceases to improve, preventing the models from becoming overly specialized to the training data. By employing early stopping, we can enhance the models' ability to perform well on new, unseen data.

Upon analyzing the graph, it is evident that the models reached the stopping condition before completing 20 epochs. This suggests that the models converged relatively quickly and were able to achieve satisfactory performance within a limited number of training iterations. Notably, the bidirectional GRU with attention demonstrated the best performance, as indicated by its lower validation loss compared to the other models.

An interesting observation from the graph is that the validation losses are higher than the training losses throughout the training process. This discrepancy is a common occurrence, suggesting a slight degree of overfitting to the training data. Nonetheless, the models still exhibit good generalization capabilities, as evidenced by their decreasing validation losses.

Once the model-training phase is completed, we move on to the prediction phase. In this phase, we utilize the trained model to make predictions on the input sequence dataset, denoted as 'X'. By applying the trained model, we allow it to leverage the knowledge it has acquired through the training process, including the learned parameters and its specific architecture.

The prediction process benefits from the model's ability to capture and comprehend the underlying patterns and dependencies within the input dataset. By applying its learned parameters and architecture, the model can effectively analyze the input data and produce predictions that align with the patterns it has learned during training.

4.8 **Evaluation**

To evaluate the model's accuracy and performance, we compare the predicted values against the actual values in the test dataset. We use three commonly used evaluation metrics: Mean Absolute Error, Mean Squared Error, and Root Mean Squared Error. These metrics provide a quantitative measure of the model's effectiveness and suitability for web traffic forecasting.

Table 4.4 presents the evaluation results, providing a comprehensive view of the proposed model's performance. These metrics offer valuable insights into the model's effectiveness and suitability for web traffic forecasting tasks. By analyzing these results, we can make informed assessments about the model's performance and its applicability in accurately forecasting web traffic.

Table 4-4 Deep learning Models with Matrices result of web traffic

Model	**MAE**	**MSE**	**RMSE**
LSTM	0.08223	0.01187	0.1068
biLSTM	0.07096	0.00895	0.09169
biLSTM with attention	0.06666	0.00841	0.08857
GRU	0.08611	0.01519	0.10967
biGRU	0.07879	0.01400	0.10205
biGRU with attention	0.06102	0.00713	0. 08266

Based on the analysis results presented in Table 4.4, the LSTM model achieved an MAE of 0. 08223. The biLSTM model performed slightly better, with an MAE of 0.07096. Interestingly, incorporating attention mechanisms improved the performance further. The biLSTM with attention model achieved an MAE of 0.06666.

Moving on to the GRU model, it exhibited an MAE of 0.08611, indicating a relatively higher prediction error compared to the other models. The biGRU model showed comparable results, with an MAE of 0.07879. However, the biGRU with attention model outperformed all others, with an MAE of 0.06102.

The metrics, including MAE, MSE, and RMSE, offer valuable insights into the accuracy and precision of each model. A lower value for these metrics signifies better performance, indicating smaller prediction errors and a closer alignment with the actual values. In this analysis, the biGRU with attention model consistently exhibited the best overall performance across all metrics, demonstrating its capability to effectively capture and leverage important patterns within the data. Therefore, the biGRU with attention model can be considered the most reliable and effective option for the given task, surpassing the performance of the other models listed in the table. The trends observed in terms of MSE and RMSE align with the MAE results, further reinforcing the conclusion that the biGRU with attention model outperformed the other models across multiple evaluation metrics.

In addition to the evaluation results presented in Table 4.4, the study also includes Figure 4.4, which visually represents the performance of the six deep learning models in terms of MAE over epochs. This graph offers a comprehensive and intuitive understanding of how the models perform throughout the training process. By observing the trend of MAE values over epochs, it becomes possible to assess the effectiveness and improvement of each model as training progresses.

The Figure 4.4 serves as a visual aid to complement the analysis conducted on the models' performance. It offers a clear overview of the relative performance of each model, facilitating easy identification of trends and comparisons. By combining the information from both the table and the graph, readers can gain a more comprehensive understanding of the models' performance and make informed assessments based on both quantitative and visual representations.

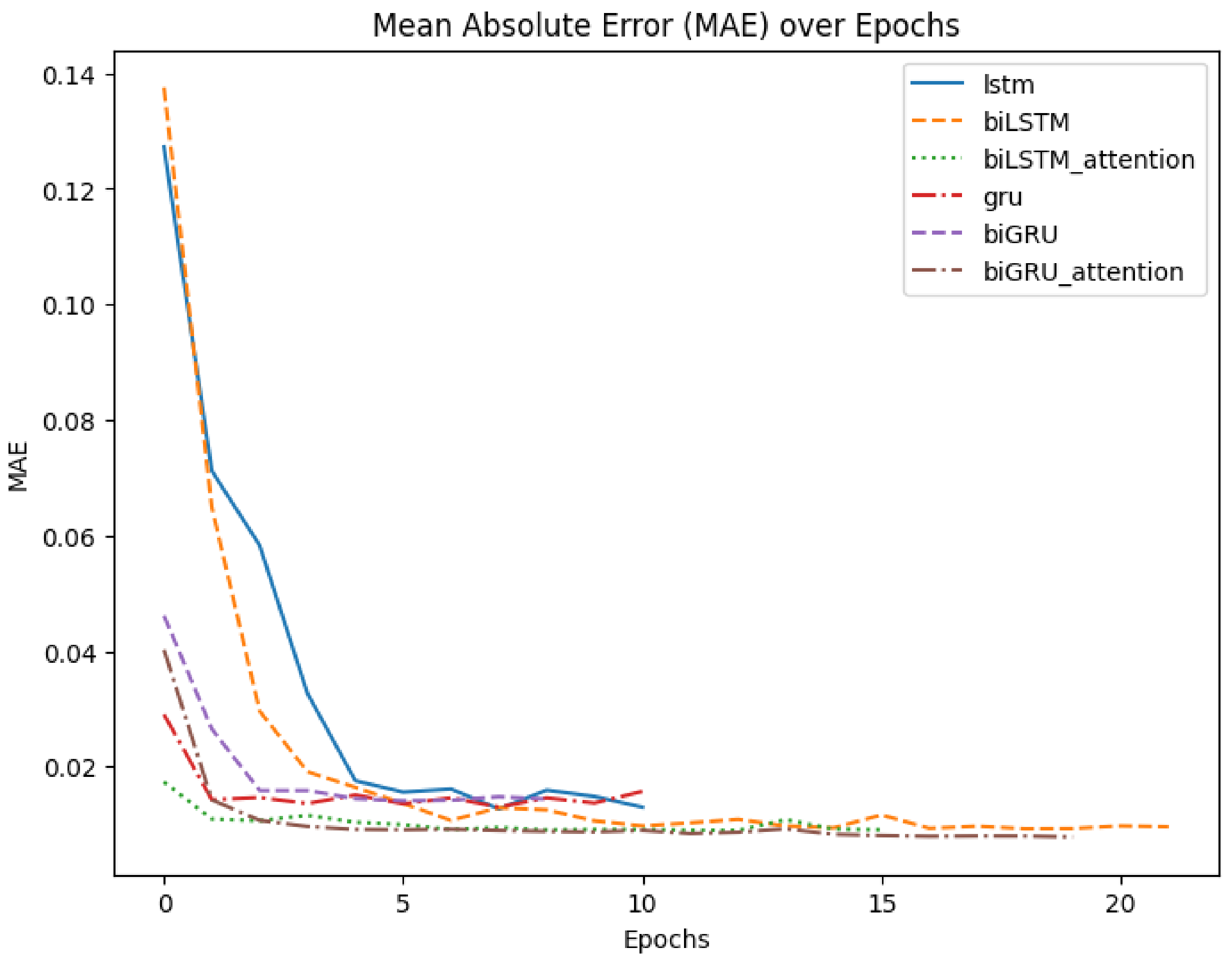

Figure 4.4 Mean Absolute Error over Epoch

In Figures 4.5, we can see a visual representation of the forecasting capabilities of the bidirectional GRU with attention model for the web traffic of our dataset. The graph provides a comparison between the actual web traffic values and the corresponding predicted values generated by the model. The x-axis of the graph represents the different time-periods or timestamps, allowing us to observe the patterns at various scales such as daily, weekly, and monthly. Each subgraph within the figure illustrates the data for a specific time step, providing a comprehensive view of the patterns over time. On the y-axis, indicating the magnitude of the actual and predicted web traffic. By examining the graph, we can observe that the bidirectional GRU with attention model performs well in forecasting web traffic. The predicted values closely align with the actual values, indicating a high level of accuracy in the model's predictions. This visual representation provides valuable insights into the model's ability to capture and forecast the patterns and fluctuations in web traffic.

The closer alignment between the predicted and actual values demonstrates the model's effectiveness in accurately predicting web traffic trends

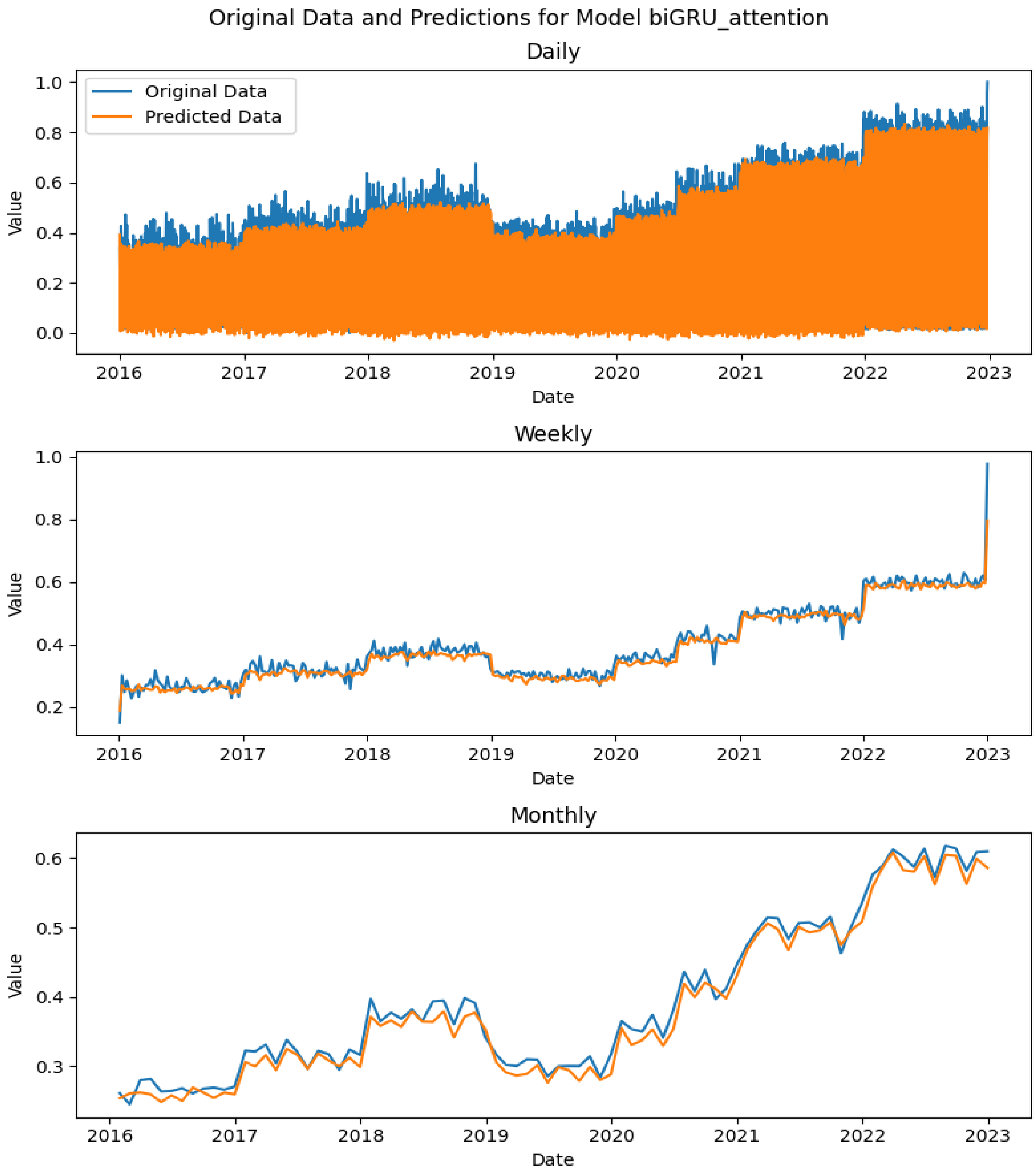

Figure 4.5 Actual and predicted web traffic for the Home page using biGRU with attention

Table 4.5 provides an evaluation of our proposed models, assessing their performance using Mean Absolute Error. The evaluation compares these models to the baseline model, which is ARIMA model, and two previous studies: one conducted by A.K. Mariappan et al. [32] and the other by R. Casado-Vara et al. [16].

A.K. Mariappan et al.'s [32] study employed a hybrid approach, combining Autoregressive Integrated Moving Average and Long Short-Term Memory models for their analysis. On the other hand, R. Casado-Vara et al [16]. developed an architecture for web traffic forecasting using LSTM for time series forecasting.

Table 4-5 Proposed Model comparison with baseline and previous work

Matrices	LSTM	biLSTM	biLSTM with attention	GRU	biGRU	biGRU with attention	ARIMA	ARIMA _LSTM [32]	LSTM [16]
MAE	0.08223	0.07096	0.06666	0.08611	0.07879	0.06102	0.27267	0.08114	132.3

The evaluation results presented in Table 4.5 provide valuable insights into the performance of our proposed models compared to the baseline and previous studies. By examining the Mean Absolute Error (MAE) values, we can thoroughly assess the accuracy and effectiveness of our models in predicting web traffic patterns. The analysis presented in the table compellingly demonstrates the exceptional accuracy of our model in capturing and predicting web traffic patterns. Our model not only outperforms the baseline model (ARIMA) but also surpasses the performance of the models developed by A.K. Mariappan et al. [32] and Casado-Vara et al. [16]. This clear superiority of our model in this domain confirms its effectiveness and demonstrates its superiority over existing approaches.

The evaluation of various models is essential, and our table emphasizes the exceptional performance of the bidirectional GRU with attention model, which achieved a remarkably low MAE of 0.06102. This model demonstrates notably superior performance compared to the other models assessed in the study, providing strong evidence of its effectiveness in web traffic prediction.

The exceptional performance of the deep learning models presented in the table serves as compelling evidence of our model’s precision in delivering accurate web traffic forecasts. Consequently, we can confidently conclude that our model surpasses the baseline, further substantiating its effectiveness and instilling trust in its forecasting capabilities.

Chapter Five

5 Conclusion and Future Works

5.1 Conclusion

Web traffic forecasting is crucial for data-driven decision making in various domains. Despite, there is a lack of research utilizing local data, and many studies rely on Wikipedia datasets that may not adequately represent the distinct features of web traffic. Furthermore, there is a tendency to prioritize conventional models, even in cases where deep learning models are employed, with LSTM being the favored choice. This limits exploration of other potentially superior models. Additionally, there is a lack of comprehensive comparisons of different deep learning models, which hinders understanding of their relative performance and the impact of datasets on model effectiveness. This lack of diversity in models and datasets hampers generalizability, particularly in developing countries like Ethiopia. Further research is needed to explore diverse models, datasets, and conduct comprehensive comparisons to improve web traffic forecasting effectiveness.

To address these gaps, this paper aims to explore and compare the effectiveness of six recurrent neural network models for web traffic forecasting in our specific setting. We utilize a novel dataset from local organizations and evaluate models such as LSTM, GRU, bidirectional LSTM, bidirectional GRU, bidirectional LSTM with attention, and bidirectional GRU with attention. Our objective is to develop accurate web traffic prediction models using deep learning techniques, leveraging local data, and improving decision-making processes.

In this study, we conduct a comprehensive analysis of web traffic and employ deep learning techniques to achieve accurate forecasting. The research process involves multiple stages, including data collection, preprocessing, hyperparameter tuning, model training, prediction, and evaluation. By following a well-structured approach, we effectively analyze and predict web traffic dynamics.

The data collection phase involved gathering historical web traffic data from commercial Bank of Ethiopia, covering a period from January 1, 2016, to January 3, 2023. The dataset consisted of 2,560 days of traffic records for 32 pages, providing a comprehensive sample of daily visitor counts. This provided us with a robust dataset comprising daily visitor counts for multiple web pages, allowing for a comprehensive analysis.

After the data collection phase, we performed preprocessing tasks to ensure the data was suitable for analysis and modeling. Data cleaning, normalization, splitting, and reshaping were applied to transform the raw data into a consistent and appropriate format. These preprocessing techniques guaranteed the accuracy, completeness, and readiness of the data for subsequent deep learning analysis.

Once the data was preprocessed, we proceeded with hyperparameter tuning to identify the optimal parameter values for our models. This involved experimenting with various deep learning architectures, including LSTM networks, bidirectional LSTM, bidirectional LSTM with attention, GRU, bidirectional GRU, and bidirectional GRU with attention. Through meticulous evaluation and exploration of different parameter combinations, we determined the best configuration for our models. Hyperparameter tuning played a crucial role in optimizing the performance of our models.

With the optimal hyperparameters established, we constructed the models based on the recommended architectures. By leveraging recurrent neural network models, which are well-suited for sequential data like web traffic, we aimed to accurately forecast future patterns. The training phase involved iterative optimization to minimize the loss function and improve the models' performance.

After training the models, we used them to predict future web traffic. Each model learned patterns and relationships from the training data, allowing them to make estimations of future values. We generated predictions from the six models and evaluated their performance in capturing the underlying patterns in the data.

Our analysis demonstrates the effectiveness of deep learning models in capturing the intricate temporal dynamics of web traffic. These models exhibit strong predictive capabilities by accurately forecasting web traffic patterns. Evaluation metrics such as MAE, MSE, and RMSE validate the precision of the models' forecasts when compared to the actual web traffic values. Among the tested models, the bidirectional GRU with attention model demonstrated the best performance, achieving the lowest MAE of 0.06102, MSE of 0.00713, and RMSE of 0.08266, indicating its superior accuracy. Following that, the bidirectional LSTM with attention model achieved an MAE of 0.06666, MSE of 0.00841, and RMSE of 0.08857. The biLSTM model ranked third, with an MAE of 0.07096, MSE of 0.00895, and RMSE of 0.09169.

In summary, our analysis indicates that the bidirectional GRU with attention model outperforms the other models in web traffic forecasting. However, the remaining models still demonstrate comparable performance, providing viable alternatives for this task. Our findings, based on a locally collected dataset, affirm the effectiveness of deep learning approaches in accurately predicting web traffic patterns. This comparative analysis of deep learning models underscores their capability and effectiveness in capturing the complexities of web traffic patterns and provides valuable insights into their individual strengths.

These findings highlight the importance of exploring different architectural variations and incorporating attention mechanisms when developing models for forecasting tasks. By experimenting with various model architectures and attention mechanisms, the researcher can gain valuable insights into the approaches that yield the greatest improvements in forecasting accuracy. This iterative process of refining and exploring different model variations contributes to the advancement of the forecasting field, facilitating the development of more robust and accurate models.

5.2 **Future Work**

While this study achieved promising results in web traffic forecasting using deep learning, there are several avenues for future work to further enhance and expand the capabilities of this approach:

- Dataset Preparation: One of the significant challenges encountered was the limited size of the dataset. To address this challenge, future work should focus on gathering a larger and more diverse dataset from various domain to provide the deep learning models with a richer representation of web traffic patterns.
- Model Architecture Exploration: To optimize the model architecture for web traffic forecasting, it is crucial to explore and evaluate additional deep learning architectures beyond those examined in this study. Nevertheless, due to limited computational resources, this study was unable to exhaustively analyze all parameters using grid search optimization. Thus, it is recommended to undertake further investigations involving diverse layer configurations, activation functions, regularization techniques, and explore the potential of transformer models. These endeavors will contribute to optimizing the model architecture and improving the accuracy of web traffic forecasting.

- Ensemble Methods: Investigate the potential of ensemble methods in deep learning-based web traffic forecasting. Combine multiple deep learning models, leveraging techniques such as model averaging or stacking, to harness the complementary strengths of different models and improve overall prediction performance.

By pursuing these avenues of future work, it can advance the accuracy, robustness, and interpretability of web traffic analysis and forecasting using deep learning. This will empower organizations to make informed decisions, optimize resource allocation, and improve overall website performance.

References

[1] A. G. S. P. R. J. G. A. Vrushant. Tambe, “Forecast web traffic time series using ARIMA model,” International Journal of Advanced Research in Computer and Communication Engineering, vol. 11, no. 5, May 2022.

[2] T. HaBO, “Why is website traffic important to my business,” 30 September 2022. [Online]. Available: https://blog.halfabubbleout.com/blog/why-is-website-traffic-important-to-my-business. [Accessed 22 June 2023].

[3] W. W. L. H. a. B. L. K. Zhou, “Comparative study on the time series forecasting of web traffic based on statistical model and Generative Adversarial model,” Knowledge-Based Systems, vol. 213, p. 106467, Feb 2021.

[4] F. E. M. R. M. M. Y. L. P. K. D. P. M. A. H. A. M. M. G. M. R. F. S. K. M. Gheisari, “Deep learning: Applications, architectures, models, tools, and frameworks: A comprehensive survey,” CAAI Transactions on Intelligence Technology, Jan 2023.

[5] P. W. a. K. A. Shi Dong, “A survey on deep learning and its applications,” Elsevier, Feb 2021.

[6] X. Z. Y. a. P. F. Wang, “Recent advances in deep learning,” International Journal of Machine Learning and Cybernetics, vol. 11, pp. 747-750, Feb 2020.

[7] X. Z. Y. L. a. W. L. X. Liang, “A Deep Learning Approach for Web Traffic Forecasting,” in IEEE Access, vol. 7, pp. 133265-133274, 2019.

[8] E. A.-M. N. Petluri, “Web traffic prediction of Wikipedia pages,” in 2018 IEEE International Conference on Big Data (Big Data), Seattle, WA, USA, Dec 2018.

[9] P. Z. a. K. H. C. Janiesch, “Machine learning and deep learning,” Electronic Markets, vol. 31, no. 4, pp. 685-695, 2021.

[10] I. H. Sarker, “Deep Learning: A Comprehensive Overview on Techniques, Taxonomy, Applications and Research Directions,” SN Computer Science, vol. 2, no. 9, pp. 1-32, 2021.

[11] Y. Y. M. H. Zitao Liu, “A Flexible Forecasting Framework for Hierarchical Time Series with Seasonal Paterns: A Case Study of Web Traffic,” in 41st International ACM SIGIR Conference on Research and Development in Information Retrieval, Arbor, Michigan, USA, Jul 2018.

[12] S. J. a. L. B. Taylor, “Forecasting at scale,” The American Statistician, vol. 72, no. 1, pp. 37-44, Aug 2017.

[13] S. Srivastava, “Forecasting with web traffic data, Towards Data Science, Aug. 09, 2020,” 09 Aug 2020. [Online]. Available: https://towardsdatascience.com/forecasting-with-web-traffic-data-6681ff148df0 [Accessed 22 Jan 2023].

[14] K. S. S. a. H. Subashini.A, “Forecasting website traffic using prophet time series model,” International Research Journal of Multidisciplinary Technovation, pp. 56-63, Jan 2019.

[15] R. S. K. P. a. P. P. Vinayakumar, “Applying Deep Learning Approaches for Network Traffic Prediction,” in International Conference on Advances in Computing, Communications and

Informatics (ICACCI), 2017.

[16] R. Casado-Vara, A. Martin del Rey, D. Perez-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, 2021

[17] Z. a. Z. Li. Xiaohua, “Short-term Power Load Forecasting Based on Gated Recurrent Unit Network and Cloud Computing Platform,” in Proceedings of the 4th International Conference on Computer Science and Application Engineering, 2020.

[18] E. P. C. L. a. C. Lens Shiang, “Gated recurrent unit network-based cellular Trafile Prediction,” International Conference on Information Networking (ICOIN), 2020.

[19] A. Z. M. a. P. Troia.S, “Deep learning-based traffic prediction for network optimization,” in 2018 20th International Conference on Transparent Optical Networks (ICTON), 2018.

[20] G. R. A. a. S. W. S. R. Anggrainingsih, “Time series forecasting using exponential smoothing to predict the number of website visitors of Sebelas Maret University,” in in 2015 2nd International Conference on Information Technology, Computer, and Electrical Engineering (ICITACEE), 2015.

[21] S. T. S. Y. a. S. A. T. Shelatkar, “Web traffic time series forecasting using ARIMA and LSTM RNN,” in ITM Web of Conferences, 2020.

[22] A. Kharwal, “Website Traffic Forecasting using Python, The Clever Programmer,” 28 Jun 2022. [Online]. Available: https://thecleverprogrammer.com/2022/06/28/website-traffic-forecasting-using-python/. [Accessed 22 Jan 2023].

[23] E. Inventions, “The importance of website traffic,” Digital Tech, 14 September 2021. [Online]. Available: https://digitaltechhome.com/technology/the-importance-of-website-traffic/ [Accessed 22 June 2023].

[24] B. Rai, “6 different types of website traffic,” Acme Themes, 2 May 2020. [Online]. Available: https://www.acmethemes.com/blog/types-of-website-traffic/. [Accessed 24 June 2023].

[25] S. Gonzalez, “Types of web traffic: Sources and explanations,” Cyber click Digital Marketing Agency, 28 April 2020. [Online]. Available: https://www.cyberclick.net/numericalblogen/types-of-web-traffic-sources-and-explanations. [Accessed 27 June 2023].

[26] A. Weiner, “11 best web analytics tools to boost your business, 8 Oct 2020,” Wix Blog, 8 October 2020. [Online]. Available: https://www.wix.com/blog/best-website-analytics-tools. [Accessed 20 June 2023].

[27] Graphy, “How to measure your website traffic and performance in 2022,”Graphy Blog, 8 May 2020. [Online]. Available: https://graphy.com/blog/how-to-and-measure-your-website-traffic/. [Accessed 25 June 2023].

[28] K. K. D. V. M. S. K. C. K. M. a. B. T. M. ArunKumar, “Comparative analysis of gated recurrent units (GRU), long short-term memory (LSTM) cells, autoregressive integrated moving average (ARIMA), seasonal autoregressive integrated moving average (SARIMA) for forecasting COVID-19 trends,” Alexandria Engineering Journal, vol. 61(10), pp. 7585-

7603, 2022.

[29] W. L. W. M. J. a. D. G. Wu, “Using gated recurrent unit network to forecast short-term load considering impact of electricity price,” Energy Procedia, pp. 20-25, 2018.

[30] S. Kostadinov, “Understanding GRU networks,” Medium, 16 December 2017. [Online]. Available: https://towardsdatascience.com/understanding-gru-networks-2ef37df6c9be. [Accessed 26 07 2023].

[31] M. Romero, Y. Interian, T. Solberg, and G. Valdes, “Training Deep Learning models with small datasets,” *arXiv preprint arXiv:1912.06761*, Dec. 2019

[32] Dr.A.K.Mariappan, Anjana Sriram R.Nivedha Nandhini, S.V Nageswari, “Web Traffic Forecasting Using ARIMA and LSTM”, Journal of University of Shanghai for Science and Technology, Volume 23, Issue 10, October, 2021

[33] M. A. R. Shuvo, M. Zubair, A. T. Purnota, M. I. Hossain, “Traffic Forecasting using Time-Series Analysis,” Sixth International Conference on Inventive Computation Technologies, 2021.

[34] Sayed A. Sayed , Yasser Abdel-Hamid and Hesham Ahmed Hefny, “Artificial intelligence-based traffic flow prediction: a comprehensive review”, Journal of Electrical Systems and Information Technology, 2023

[35] S. A. Akram, S. M. Islam, and D. N. K. Jayakody, “Recurrent neural networks for forecasting time series with multiple seasonality: A comparative study,” IEEE Access, vol. 8, pp. 21463-21477, 2020, doi: 10.1109/ACCESS.2020.2969924

[36] K. Greff, R. K. Srivastava, J. Koutník, B. R. Steunebrink and J. Schmidhuber, “LSTM: A search space odyssey,” IEEE transactions on neural networks and learning systems, 2017, vol. 28, no. 10, pp. 2222-2232

[37] Z. C. Lipton, J. Berkowitz and C. Elkan, “A critical review of recurrent neural networks for sequence learning,” arXiv preprint arXiv:1506.00019, 20153

[38] A. K. Jain, A. K. Singh and A. K. Sharma, “Deep Learning-based Hybrid Technique for Forecasting Web Traffic,” 2023 Fifth International Conference on Electrical, Computer and Communication Technologies (ICECCT), Coimbatore, India, 2023, pp. 1-6, doi: 10.1109/ICECCT56650.2023.10179797

[39] S. K. Singh, A. K. Singh and A. K. Sharma, “Gated Recurrent Unit (GRU) in RNN for traffic forecasting based on time-series data,” 2022 2nd International Conference on Innovative Sustainable Computational Technologies (CISCT), Coimbatore, India, 2022, pp. 1-6, doi: 10.1109/CISCT55310.2022.10046484

[40] L. Brigato and L. Iocchi, “A Close Look at Deep Learning with Small Data,” 2020 25th International Conference on Pattern Recognition (ICPR), Milan, Italy, 2021, pp. 1-8, doi: 10.1109/ICPR48806.2021.9412492.

[41] Thuwarakesh, “How To Train Deep Learning Models With Small Datasets,” 06/27/2023. [Online]. Available: https://www.the-analytics.club/deep-learning-with-small-data/. [Accessed Nov 18, 2023].

[42] M.Ahame. “LSTM and Bidirectional LSTM for Regression.” Towards Data Science. [Online] https://towardsdatascience.com/lstm-and-bidirectional-lstm-for-regression-4fddf910c655 [Accessed: Nov. 19, 2023]

[43] Z. Huang, W. Xu and K. Yu, “Bidirectional LSTM-CRF Models for Sequence Tagging,” [Online]. Available: https://arxiv.org/abs/1508.01991. Accessed: Nov. 20, 2023

[44] H. Abbasimehr and R. Paki, “Improving time series forecasting using LSTM and attention models,” J. Ambient Intell. Humaniz. Comput, vol. 13, no. 2, pp. 673–691, Feb. 2022, doi: 10.1007/s12652-020-02761-x.

[45] Cheng, Q., Chen, Y., Xiao, Y. et al. “A dual-stage attention-based Bi-LSTM network for multivariate time series prediction,” J Supercomput 78, 16214–16235 (2022). https://doi.org/10.1007/s11227-022-04506-3

[46] J, A.K., Trueman, T.E. and Cambria, “A Convolutional Stacked Bidirectional LSTM with a Multiplicative Attention Mechanism for Aspect Category and Sentiment Detection,” Cogn Comput 13, 1423–1432 (2021). https://doi.org/10.1007/s12559-021-09948-0

[47] Mishra, S., Kshirsagar, V., Dwivedula, R., Hota, “Attention-Based Bi-LSTM for Anomaly Detection on Time-Series Data. In: Farkas,” Artificial Neural Networks and Machine Learning – ICANN 2021. ICANN 2021. Lecture Notes in Computer Science, vol 12891. Springer, Cham, 2021, https://doi.org/10.1007/978-3-030-86362-3_11

[48] Abbasimehr, H., Paki, R. “Improving time series forecasting using LSTM and attention models,” J Ambient Intell Human Comput 13, 673–691, 2022, https://doi.org/10.1007/s12652-020-02761-x

[49] Eskandari, H., Imani, M. and Parsa Moghaddam, M. “Best-tree wavelet packet transform bidirectional GRU for short-term load forecasting,” J Supercomput 79, 13545–13577 , 2023, https://doi.org/10.1007/s11227-023-05193-4

[50] F. Liu, Q. Tao, D. Yang and D. Sidorov, “Bidirectional Gated Recurrent Unit-Based Lower Upper Bound Estimation Method for Wind Power Interval Prediction,” in IEEE Transactions on Artificial Intelligence, vol. 3, no. 3, pp. 461-469, June 2022, doi: 10.1109/TAI.2021.3123928.

[51] Chauhan, N.S., Kumar, N. “Traffic Flow Forecasting Using Attention Enabled Bi-LSTM and GRU Hybrid Model” Communications in Computer and Information Science, vol 1794. Springer, Singapore, 2023, https://doi.org/10.1007/978-981-99-1648-1_42

[52] Q. Zhou, C. Zhou and X. Wang, “Stock prediction based on bidirectional gated recurrent unit with convolutional neural network and feature selection,” PLOS ONE, vol. 17, no. 2, Feb. 2022

[53] Berta Etefa, “NARX-Based Locally Distributed Web-Servers Load Prediction,” Addis Ababa University, Feb, 2020

[54] 1S. Singh, “10 Hyperparameters to Keep an Eye on for Your LSTM Model and Other Tips,” Medium, Aug. 30, 2021. [Online]. Available: 10 Hyperparameters to keep an eye on for your LSTM model — and other tips | by Kuldeep Chowdhury | Geek Culture | Medium. [Accessed: Dec. 5, 2023]