
Deep Learning for Traffic Data Mining

By

Abdullah Al Ahad Khan and ID: 012221021

A thesis submitted for the degree of Master of Science in Computer Science and Engineering

November 14, 2024



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
UNITED INTERNATIONAL UNIVERSITY

Abstract

Traditionally, traffic management relies on several methods, including but not limited to signage, traffic signaling, and manual control. Traffic engineers analyze traffic patterns collected through various sensors and use their expertise to manage traffic flow. While traditional traffic management methods have been adequate to some extent, they often struggle to handle the complexity and dynamic nature of traffic in modern urban environments. To address these limitations, advanced technologies in autonomous traffic volume prediction have been developed. Traffic data are inherently complex, and traditional mathematical models, which rely on simplified assumptions, frequently fail to accurately represent and analyze this complexity. With the rise of neural networks and the availability of large-scale datasets, deep learning-based solutions have emerged as a robust approach to managing real-world traffic scenarios. Neural networks' ability to capture complex, non-linear relationships in traffic data makes them well-suited for handling intricate traffic patterns that are challenging for simpler models. In this study, we present a deep learning approach for accurately predicting traffic volume in Porto, Portugal. Our model leverages an extensive dataset encompassing historical traffic volume data from diverse sensors and monitoring systems. The experiment was conducted in two phases: Phase 1, using a 4-month dataset, and Phase 2, using a 15-month dataset. In Phase 1, Recurrent Neural Networks (RNNs) demonstrated superior performance with an R-squared value of 0.9953 and a MAPE of 4.6%, effectively capturing temporal dependencies to generate precise traffic volume forecasts. In Phase 2, the Gated Recurrent Unit (GRU) model outperformed other models, achieving an R-squared value of 0.9919 and a MAPE of 5.3%, showcasing its ability to handle longer time series data with complex patterns and dependencies. By harnessing the power of these deep learning models, our approach surpasses previous methodologies on the same dataset, outperforming traditional time-series models across different evaluation metrics such as MAE and RMSE. These findings demonstrate the practical potential of deep learning-based models in complex urban environments and offer a robust way to utilize sensor data. The proposed models can adapt to changing traffic conditions and evolving patterns with infrequent training, making them suitable for dynamic environments.

Acknowledgements

This work would not have been possible without the input and support of many people. I want to express my gratitude to everyone who contributed to it somehow.

I want to begin by thanking my thesis supervisor, Dr. Dewan Md. Farid has not only been my mentor but also a guiding light throughout my thesis journey. His support and wealth of knowledge have been invaluable, making this journey smoother and more enlightening.

Next, I am very grateful to Dr. Mohammad Nurul Huda, who has supported and guided me throughout my master's program. His encouragement and feedback have kept me going even in the toughest of times. I also want to express my heartfelt appreciation to Dr. Swakkhar Shatabda, whose machine-learning classes were not just lessons but revelations. His teaching style made complex concepts seem simple; I'm truly grateful for that.

I want to express my deepest gratitude to my late father, whose unwavering belief in my potential continues to inspire me. His encouragement and sacrifices have been the cornerstone of my academic achievements. I am also immensely grateful to my mother for her boundless love, unwavering support, and sacrifices throughout my educational journey. Additionally, I extend my heartfelt thanks to my wife, whose unwavering support and understanding have strengthened and motivated me during challenging times.

I offer my sincerest appreciation to all those mentioned above and to countless others who have supported and encouraged me along the way. Your contributions have played an indispensable role in shaping my academic and personal growth; I am profoundly grateful for that.

Publication List

Work-related to the research presented in this thesis has been published by the author in the following peer-reviewed conferences:

Abdullah Al Ahad Khan, Siraj Us Salekin, Mushfika Sharmin Rahman, and Dewan Md. Farid. Deep learning for traffic data mining. In 6th International Conference on Electrical Information and Communication Technology (EICT), pages 1–6, Khulna, Bangladesh, December 2023.

Declaration

I, Abdullah Al Ahad Khan, declare that this thesis titled, “Deep Learning for Traffic Data Mining” and the work presented in it are my own. I confirm that:

- I completed this work mainly during my Master’s program at United International University.
- I have not submitted any part of this thesis for another degree or qualification at this or any other institution, and I have clearly stated this when applicable.
- Whenever I have used published works by others, I have always properly acknowledged them.
- Any quotes from other works are always properly cited. Apart from such quotes, this thesis is entirely the result of my efforts. I have acknowledged all the major help I received.
- Where the thesis includes work that I did with others, I have clearly described what others contributed and what my own contributions were.

Signature _____

Date_____

Abdullah Al Ahad Khan

Certificate

I do hereby declare that the research works embodied in this thesis entitled “Deep learning for Traffic Data Mining” is the outcome of an original work carried out by Abdullah Al Ahad Khan under my supervision. I further certify that the dissertation meets the requirements and the standard for the degree of MSc in Computer Science and Engineering.

Signature _____

Date_____

Dr. Dewan Md. Farid
Professor
Department of Computer Science & Engineering
United International university
Dhaka-1212, Bnagladesh

Table of Contents

Table of Contents	vii
List of Figures	viii
List of Tables	ix
1 Introduction	1
1.1 Motivation	2
1.2 Objective	2
1.3 Contribution	3
1.4 Organization of the thesis	3
2 Related Works	5
2.1 Introduction to Deep Learning in Traffic Prediction	5
2.2 Traditional and Hybrid Models for Traffic Prediction	5
2.2.1 Machine Learning Techniques for Traffic Forecasting	5
2.2.2 Emergence of Hybrid Models	6
2.3 Deep Learning Techniques for Traffic Forecasting	6
2.3.1 Recurrent Neural Networks (RNN) and Long Short-Term Memory (LSTM) Models	6
2.3.2 Convolutional Neural Networks (CNN) and Attention Mechanisms	6
2.4 Graph-Based and Advanced Models for Traffic Prediction	7
2.4.1 Spatial-Temporal Graph Convolutional Networks (STGCN) and Related Models	7
2.4.2 Innovative Hybrid Approaches and Future Trends	7
2.5 Comparative Studies and Performance Analysis	7
2.5.1 Model Comparisons and Evaluation	7
2.6 Challenges, Future Directions, and Conclusion	8
2.6.1 Addressing Challenges in Deep Learning Applications	8
2.6.2 Future Research Directions	8
2.6.3 Conclusion	8

3	Methodology	9
3.1	Data Collection	9
3.2	Selection of Subdataset	10
3.3	Data Visualization	11
3.3.1	Traffic Volume Distribution Visualization	12
3.3.2	Time Series Decomposition	16
3.4	Data Preprocessing	18
4	Deep Learning Model	19
4.1	Overview of Deep Learning Models	20
4.2	Building Deep Learning Models	21
4.3	Activation Function	22
4.4	Evaluation Metrics Overview	23
4.4.1	Mean Absolute Error(MAE)	23
4.4.2	Root Mean Square Error(RMSE)	23
4.4.3	Mean Absolute Percentage Error(MAPE)	24
4.4.4	R-squared(R^2)	24
5	Experimental Results	25
5.1	Experimental Result & analysis	25
5.1.1	Evaluation Metrics & Learning Curve Analysis for Phase 1	26
5.1.2	Evaluation Metrics & Learning Curve Analysis for Phase 2	28
5.1.3	Analysis Summary	31
5.2	Experimental Setup	31
5.3	Result Visualization	32
5.3.1	Overall Performance Analysis	32
5.3.2	Daily Evaluation Metrics Comparison	38
5.3.3	First Day, Best Day & Worst Day Prediction	40
6	Conclusion	43
6.1	Summary	43
6.2	Limitation	43
6.3	Future Work	44
	References	47

List of Figures

3.1	Process for building deep learning models.	9
3.2	Split of training and testing data.(Phase 1)	11
3.3	Split of training and testing data.(Phase 2)	11
3.4	Average traffic volume in different time slots.(Phase 1)	12
3.5	Average traffic volume in different time slots.(Phase 2)	13
3.6	Histogram of Total Traffic Volume	13
3.7	Density Graph of Total Volume (Phase 1)	14
3.8	Density Graph of Total Volume (Phase 2)	14
3.9	Number of vehicles of different Class	15
3.10	Histogram of Average Length	15
3.11	Histogram of Light Vehicle Rate	16
3.12	Total Volume per equipment ID	16
3.13	Time series decomposition of traffic data.(Phase 1)	17
3.14	Time series decomposition of traffic data.(Phase 2)	18
5.1	Leraning Curve for All Models(Phase 1)	27
5.2	Leraning Curve for All Models(Phase 2)	29
5.3	ANN 1 month Prediction(Actual vs Predicted)	33
5.4	LSTM 1 month Prediction(Actual vs Predicted)	33
5.5	Bi-LSTM 1 month Prediction(Actual vs Predicted)	34
5.6	RNN 1 month Prediction(Actual vs Predicted)	34
5.7	GRU 1 month Prediction(Actual vs Predicted)	35
5.8	ANN 3 month Prediction(Actual vs Predicted)	35
5.9	LSTM 3 month Prediction(Actual vs Predicted)	36
5.10	Bi-LSTM 3 month Prediction(Actual vs Predicted)	36
5.11	RNN 3 month Prediction(Actual vs Predicted)	37
5.12	GRU 3 month Prediction(Actual vs Predicted)	37
5.13	Daily Evaluation Matrices Value Comparison for all models(Phase 1)	38
5.14	Daily Evaluation Matrices Value Comparison for all models(Phase 2)	39
5.15	First day, Best day & Worst Day Prediction by RNN(Phase 1)	41
5.16	First day, Best day & Worst Day Prediction by GRU(Phase 2)	42

List of Tables

3.1	Feature description and data type of the dataset.	10
4.1	Hyparparameter and Parameter of Each Model	22
5.1	Evaluation matrix summary of models(Phase 1)	26
5.2	Evaluation matrix summary of models(Phase 2)	29

Chapter 1

Introduction

Traffic congestion is a global issue affecting cities and regions worldwide, with significant economic and environmental consequences. INRIX's 2020 Global Traffic Scorecard estimated that traffic congestion cost the global economy 1.37 trillion USD in 2019, with Europe alone experiencing 165 billion Euro in 2018, averaging 1,215 per capita. The time lost in traffic is alarming, with cities like Bogota, Colombia, Rio de Janeiro, and Bangkok experiencing significant average hours lost per driver per year. Even developed cities like London, New York City, and Sydney face significant traffic congestion. The Global Mobility Report 2017 estimated that road traffic congestion contributed to an additional 3.2 gigatons of CO₂ emissions globally in 2015, with urban traffic congestion being a major contributor to air pollution and public health. Addressing traffic congestion is crucial for improving economic efficiency, reducing travel time, and promoting sustainable and liveable cities [1], [2].

Porto, Portugal, is not exempt from the problem of traffic congestion. A person with a car in Porto spends an annual average of 63 hours on the road owing to traffic congestion. In addition, the individual emits 152 kg more oxygen and spends 125 euros more annually due to congestion in Porto. So, we propose a deep learning-based model to reduce traffic congestion in Porto City, Portugal. Deep learning is a subset of Machine Learning (ML) that employs artificial neural networks to analyse and predict Big Data [3]. Its ability to solve complex problems in regression analysis, computer vision, natural language processing, and speech recognition has acquired popularity. By modulating the neuron weights via backpropagation, deep learning algorithms identify patterns. However, training and deploying deep learning models require large amounts of data and computational resources, and their inner workings can be difficult to comprehend. Ongoing research addresses these obstacles and improves deep learning capabilities [4], [5].

Therefore, we have used deep learning techniques to predict the number of vehicles on a specific road in Porto City, Portugal. The proposed deep learning model requires data preparation, model training, and evaluation to produce accurate and efficient vehicle volume forecasts. Once the collected data is input into the model, its weights and biases are adjusted to optimize the detection performance. After the model has been trained

and evaluated, it is used to predict vehicles using unknown data. We anticipate that the model can assist the Porto city authorities in gaining a thorough understanding of traffic congestion on a specific road on a particular day to take the necessary measures in the future. In addition, they can use the model to predict the real-time volume of vehicles in Porto City, Portugal.

1.1 Motivation

Traffic congestion in Porto City not only affects travel efficiency but also exacerbates environmental pollution and economic costs. The city's average commuter faces considerable delays and increased fuel consumption, leading to higher CO2 emissions and financial burdens. Traditional traffic management solutions have proven insufficient in dynamically adapting to the complex and fluctuating nature of urban traffic. Therefore, there is a pressing need for innovative solutions that can leverage the vast amounts of traffic data generated in real-time to provide accurate predictions and enable proactive traffic management.

Deep learning models have shown remarkable success in various fields due to their ability to learn from high-dimensional data and uncover complex patterns. Applying these models to traffic data can potentially revolutionize traffic management by providing accurate predictions of traffic volumes, enabling city authorities to implement timely interventions and improve traffic flow. This research is motivated by the need to harness the power of deep learning to address the challenges of traffic congestion in Porto City.

1.2 Objective

The main objective of this research is to leverage real-time traffic data from Porto city to construct a reliable deep learning-based model that can accurately predict traffic patterns. Specifically, this study aims to:

1. **Collect and Analyze Real-Time Traffic Data:** Gather and organize extensive traffic data from Porto city to gain a comprehensive understanding of the city's current traffic dynamics, forming the foundation for predictive model development.
2. **Apply Exploratory Data Analysis (EDA):** Utilize EDA techniques to detect trends and patterns within the collected traffic data, offering insights into recurring traffic behaviors and guiding the model-building process.
3. **Review Machine Learning and Deep Learning Techniques:** Conduct an in-depth analysis of various machine learning and deep learning methodologies to identify the most promising approaches for forecasting traffic flow.
4. **Develop and Evaluate Deep Learning Models:** Construct and rigorously test five distinct deep learning architectures—namely, Artificial Neural Networks

(ANN), Long Short-Term Memory (LSTM), Bidirectional LSTM (Bi-LSTM), Recurrent Neural Networks (RNN), and Gated Recurrent Units (GRU)—to evaluate their effectiveness in predicting traffic volumes.

1.3 Contribution

This research makes several important contributions to the field of traffic prediction and intelligent transportation systems:

1. **Integration of Real-Time Traffic Data from Porto City:** This study uniquely incorporates real-time traffic data from Porto, presenting a data-driven approach that reflects the city’s dynamic traffic environment, thus enabling more accurate model training and validation.
2. **Exploratory Data Analysis for Traffic Insights:** Through the application of EDA, the study reveals key patterns and trends within the urban traffic flow, providing foundational knowledge that can inform future research and traffic management strategies.
3. **Comprehensive Evaluation of Deep Learning Models:** By developing and comparing multiple deep learning frameworks, this research identifies the strengths and limitations of models such as ANN, LSTM, Bi-LSTM, RNN, and GRU in predicting traffic, offering a benchmark for selecting the most suitable model based on specific needs.
4. **Practical Application for Urban Mobility Enhancement:** The findings and tools developed in this study aim to assist city planners and traffic authorities in Porto by providing actionable insights into traffic congestion management, ultimately contributing to more efficient urban mobility.
5. **Advancement of Smart City Technologies:** The study’s methodologies and results are designed to be scalable and adaptable, promoting the broader application of data-driven traffic solutions in other urban areas, thereby supporting the development of intelligent transportation systems and smart city initiatives globally.

1.4 Organization of the thesis

The remainder of this thesis is organized as follows:

- **Chapter 2: Related Works**—This chapter provides a comprehensive review of the existing literature on traffic prediction models, focusing on both traditional statistical methods and modern deep learning approaches. It highlights the strengths and limitations of various techniques and sets the stage for the proposed research.

- **Chapter 3:** Methodology - This chapter details the methodology used in the study, including data collection, preprocessing, model development, and evaluation. It outlines the steps taken to ensure the accuracy and reliability of the traffic volume predictions.
- **Chapter 4:** Data Analysis and Model Implementation - This chapter presents the results of the Exploratory Data Analysis (EDA) conducted on the collected traffic data. It also describes the implementation of the five deep learning models and their respective performances in predicting traffic volumes.
- **Chapter 5:** Results and Discussion—This chapter discusses the study’s findings, comparing the performance of the developed models and analyzing their strengths and weaknesses. It provides insights into the practical implications of the results for traffic management in Porto City.
- **Chapter 6:** Conclusion and Future Work—The final chapter summarizes the research’s key findings, highlighting its contributions to traffic prediction. It also outlines potential areas for future research to enhance further the effectiveness of deep learning models in traffic volume forecasting.

By following this structure, the thesis aims to provide a clear and comprehensive account of the research conducted, offering valuable insights and practical solutions for addressing traffic congestion through deep learning techniques.

Chapter 2

Related Works

2.1 Introduction to Deep Learning in Traffic Prediction

Deep learning has shown remarkable success in various fields, including image and video processing, natural language processing, and time-series forecasting. Deep learning-based models can learn from high-dimensional Big Data, identifying complex patterns and making accurate predictions by understanding relationships within the data[6]. In time-series forecasting, particularly in traffic prediction, these models are valuable due to their ability to handle sequential information, benefiting from advances in computational power and architectural improvements. This section explores the evolution of traffic data prediction techniques, emphasizing the transition from traditional models to state-of-the-art deep learning approaches.

2.2 Traditional and Hybrid Models for Traffic Prediction

2.2.1 Machine Learning Techniques for Traffic Forecasting

Earlier approaches to traffic prediction primarily relied on traditional machine learning models and regression techniques. In 2019, Ishteaque et al.[7] employed five regression models to forecast traffic flow in Porto City using data from 23 sensors, producing a monthly sub-dataset with 1 million instances. The M5P tree model emerged as the best performer with a 2.88% error rate. This study highlighted the limitations of regression-based models in handling complex traffic patterns.

Azzedine Boukerche et al.[8] provided a comprehensive analysis of machine learning models for traffic prediction in 2020, focusing on the challenges of using open-source datasets. They concluded that neural network-based models offer the potential to extract high-level features efficiently while maintaining simplicity in their structure.

2.2.2 Emergence of Hybrid Models

The transition from traditional models to hybrid approaches marked a significant improvement in prediction capabilities. Yuanli Gu et al.[9] proposed an Improved Bayesian Combination Model (IBCM) for predicting traffic volume, integrating deep learning techniques with sub-predictors such as the Gated Recurrent Unit Neural Network (GRUNN), Autoregressive Integrated Moving Average (ARIMA), and Radial Basis Function Neural Network (RBFNN). The IBCM framework demonstrated a significant reduction in Mean Absolute Percentage Error (MAPE) and Variance of Absolute Percentage Error (VAPE) by 4.72% and 6.27

2.3 Deep Learning Techniques for Traffic Forecasting

2.3.1 Recurrent Neural Networks (RNN) and Long Short-Term Memory (LSTM) Models

Deep learning models like Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) networks have become the standard for handling sequential data due to their ability to capture temporal dependencies effectively. Anirudh Ameya et al.[10] reviewed several deep learning architectures, including CNN, RNN, LSTM, and Restricted Boltzmann Machines (RBM), discussing their impact on traffic flow prediction. They noted the superiority of LSTM models in scenarios with significant data complexity and dynamic patterns.

Lara-Benitez et al.[11] conducted a detailed comparison of various deep learning architectures, such as MLP, ERNN, LSTM, GRU, and CNN, concluding that LSTM and CNN were the most effective for time-series forecasting. This study emphasized LSTM's capacity to capture intricate temporal dependencies in traffic data, making it a preferred choice for predicting fluctuating traffic conditions.

2.3.2 Convolutional Neural Networks (CNN) and Attention Mechanisms

In 2020, Haifeng Zheng et al.[12] developed a hybrid deep learning model utilizing Convolutional Neural Networks (CNN) and LSTM architectures to automatically extract features from traffic flow data. Their model incorporated attention-based Conv-LSTM and bidirectional LSTM modules to capture both daily and weekly periodic features, significantly improving prediction accuracy over traditional approaches.

Yuankai Wu et al.[13] introduced the Deep Neural Network with Bidirectional Temporal Flow (DNN-BTF) architecture, integrating attention mechanisms that prioritize important data inputs. This model achieved superior performance compared to state-of-the-art methods by effectively determining the relevance of input features in traffic flow forecasting.

2.4 Graph-Based and Advanced Models for Traffic Prediction

2.4.1 Spatial-Temporal Graph Convolutional Networks (STGCN) and Related Models

Graph-based models have become increasingly popular for capturing spatial-temporal dynamics in traffic networks. Zhang et al.[14] proposed the Spatial-Temporal Graph Inception Residual Network (STGI-ResNet), which effectively preserved road network topology while improving prediction accuracy. Their model demonstrated significant enhancements over traditional models in handling complex spatial-temporal interactions.

Yu, Yin, and Zhu[15] introduced the Spatio-Temporal Graph Convolutional Networks (STGCN) framework, designed to integrate spatial and temporal dependencies seamlessly. Their research underscored the importance of capturing these correlations to improve traffic forecasting accuracy, especially in scenarios involving intricate traffic conditions.

2.4.2 Innovative Hybrid Approaches and Future Trends

Agafonov et al.[16] developed a deep learning framework that combines convolution operations on graph data with recurrent neural networks to enhance short-term traffic forecasting. This approach outperformed baseline models by integrating spatial-temporal features, showcasing the growing trend of hybrid methodologies in traffic prediction.

Lim and Zohren[17] reviewed advancements in deep learning models for time-series forecasting, highlighting the effectiveness of hybrid architectures that blend traditional statistical methods with deep learning components. They identified future research directions focused on developing continuous-time models and architectures that consider hierarchical structures to further improve forecasting capabilities.

2.5 Comparative Studies and Performance Analysis

2.5.1 Model Comparisons and Evaluation

Charalampos et al.[18] compared machine learning techniques for traffic forecasting, finding that neural networks outperformed other methods during significant traffic variations, while support vector machines (SVM) excelled under stable conditions. This comparison highlighted the need for model selection based on specific traffic dynamics.

Elsayed et al.[19] analyzed the relative performance of deep learning models against simpler machine learning techniques. They concluded that, while straightforward models could be effective with proper feature engineering, deep learning models generally provided superior accuracy due to their capability to capture non-linear and complex patterns in traffic data.

2.6 Challenges, Future Directions, and Conclusion

2.6.1 Addressing Challenges in Deep Learning Applications

Manibardo, Laña, and Del Ser[20] emphasized the need for careful methodological rigor in the application of deep learning models for short-term traffic forecasting. They pointed out the challenges in data quality, feature selection, and evaluation metrics that need to be addressed to maximize model accuracy.

2.6.2 Future Research Directions

Yuan, Zhou, and Yang[21] proposed that future research should focus on developing models that better integrate heterogeneous data sources, such as weather conditions and road characteristics, to enhance traffic forecasting accuracy. Zheng and Huang[22] suggested exploring hybrid models that combine traditional statistical techniques with neural networks for more robust predictions.

2.6.3 Conclusion

These studies collectively demonstrate the significant advancements in deep learning techniques for traffic prediction. The ability of deep learning models to capture complex patterns and dependencies in traffic data makes them highly suitable for traffic volume forecasting. Leveraging these advanced models enables more accurate and reliable traffic predictions, ultimately enhancing the efficiency of intelligent transportation systems.

Chapter 3

Methodology

This section will review the experiment’s workflow (Fig. 1) step-by-step. We will review every step of the traffic volume prediction process we used for this experiment, from data collection and preprocessing to model selection and training.

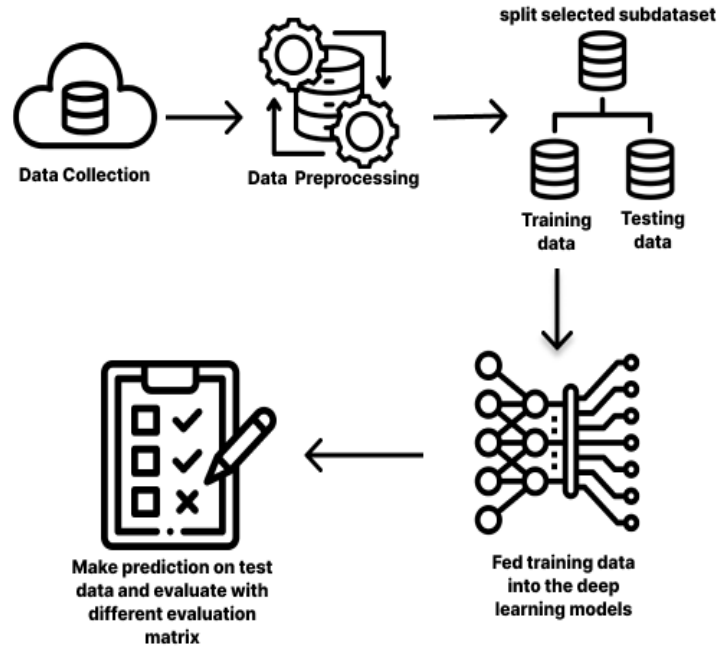


Figure 3.1: Process for building deep learning models.

3.1 Data Collection

The dataset of this study was collected in Porto City, Portugal. With a population of 2.4 million, it is the country’s second-largest metropolis. Porto has many crowded roadways, including a 21-kilometre-long ring-shaped expressway called VCI (Via de Cintura Interna),

designated IC23. This highway sees a tremendous quantity of traffic. The information was gathered using 22 loop sensors typically used to collect road traffic data. These sensors were installed on VCI, and data was stored in a PostgreSQL database. This dataset contains nearly two years of road traffic data from March 2013 to March 2015, with 20 attributes and 25,555,018 instances. Table 4.1 describes the dataset’s columns or features and the data types.

Table 3.1: Feature description and data type of the dataset.

Attribute Name	Description	Data Type
EQUIPMENTID	Sensor ID	int64
LANE_BUNDLE_DIRECTION	Lane Direction	object
TOTAL_VOLUME	Number of vehicles	int64
AVG_SPEED_ARITHMATIC	The average arithmetic speed of vehicles	float64
AVG_SPEED_HARMONIC	The average harmonic speed of vehicles	float64
AVG_LENGTH	The average length of the vehicles	float64
AVG_SPACING	The average spacing between vehicles	float64
OCCUPANCY	Occupancy	float64
LIGHT_VEHICLE_RATE	The % is the ratio of light vehicles	float64
VOLUME_CLASSE_A	No. of vehicles of class A	int64
VOLUME_CLASSE_B	No. of vehicles of class B	int64
VOLUME_CLASSE_C	No. of vehicles of class C	int64
VOLUME_CLASSE_D	No. of vehicles of class D	int64
VOLUME_CLASSE_0	No. of unidentified vehicles	int64
AXLE_CLASS_VOLUMES	No. of vehicles per vehicle axis number	object
AGG_PERIOD_LEN_MINS	Is the time between each measurement the sensors made. It is always 5 mins	int64
AGG_PERIOD_START	Time Interval	object
NR_LANES	No. of lanes in the street of the event	int64
AGGREGATE_BY_LANE_BUNDLEID	Similar to AGGREGATE_BY_LANEID	int64
AGG_ID	Street ID	int64

3.2 Selection of Subdataset

Instead of using the entire dataset, which includes over 25 million instances from March 2013 to March 2015, this study utilizes a portion of the data to make traffic volume predictions. This approach enables more efficient processing and helps avoid computational constraints associated with the large dataset.

In the first phase of the study, we focus on a sample of data from December 2014 through March 2015, totaling around 4.2 million instances. The first three months of this subset (December 2014 to February 2015) are used to train the forecasting models, while the model’s performance is evaluated using the data from the fourth month (March 2015). This initial phase allows us to build and test the models on a smaller, more manageable dataset.

For the second phase of the study, we expand the data collection to cover a longer period. Specifically, we use data from January 2014 to March 2015, providing a total of 15 months of data. In this phase, the first 12 months (January 2014 to December 2014) are used for training the models, and the last three months (January 2015 to March 2015) are used for testing. This extended period allows for a more robust evaluation of the models, considering a larger and more varied dataset.

The strategy of splitting the data into training and testing sets ensures that the models

are trained on a substantial amount of data, which helps capture the underlying traffic volume patterns. The subsequent testing phase allows an unbiased evaluation of the model's performance.

Data Split for Training and Testing

1. Phase 1:

- **Training Data:** December 2014 to February 2015 (3 months)
- **Testing Data:** March 2015 (1 month)

2. Phase 2:

- **Training Data:** January 2014 to December 2014 (12 months)
- **Testing Data:** January 2015 to March 2015 (3 months)

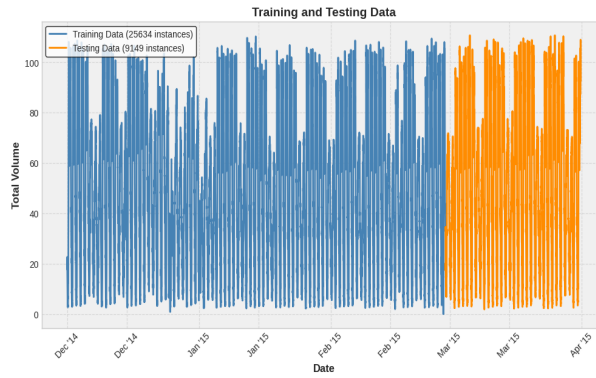


Figure 3.2: Split of training and testing data.(Phase 1)

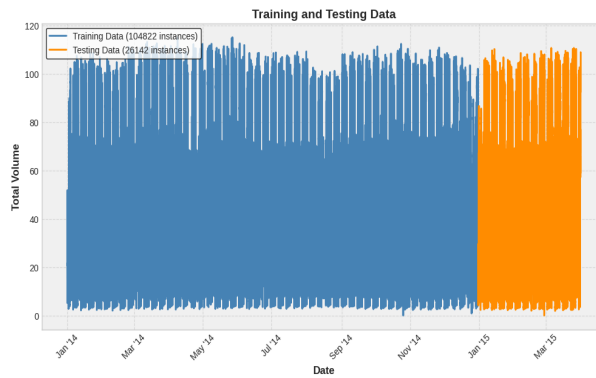


Figure 3.3: Split of training and testing data.(Phase 2)

3.3 Data Visualization

We conducted a comprehensive data visualization analysis to understand the traffic dataset better and extract meaningful insights. This step is crucial for identifying patterns, trends,

and anomalies within the data, which can significantly enhance the accuracy and reliability of our predictive models. By visualizing various aspects of the dataset, we aim to uncover temporal patterns, volume distributions, and other relevant characteristics that inform our approach to traffic volume forecasting. In this section, we present a series of visualizations that highlight different features of the dataset, including traffic volume distribution across different time periods and phases.

3.3.1 Traffic Volume Distribution Visualization

To understand the distribution of traffic volume across different times of the day, we created pie charts for each phase. These visualizations revealed the temporal patterns in the dataset, allowing us to identify peak and off-peak traffic periods.

The pie chart in 3.4 illustrates the distribution of traffic volume across different time slots from December 2014 to March 2015. This visualization highlights that the highest traffic volume occurs between 4 PM and 8 PM (27.8%), followed by equal volumes between 8 AM to 12 PM and 12 PM to 4 PM (both at 22.9%). The lowest traffic volumes are observed between 12 AM to 4 AM (4.3%) and 4 AM to 8 AM (7.1%).

The pie chart in 3.5 shows the distribution of traffic volume for the period from January 2014 to March 2015. Similar to the previous phase, the highest traffic volume is observed between 4 PM and 8 PM (27.9%), with significant volumes between 8 AM to 12 PM (23.1%) and 12 PM to 4 PM (22.9%). The lowest traffic volumes occur between 12 AM to 4 AM (4.2%) and 4 AM to 8 AM (6.6%).

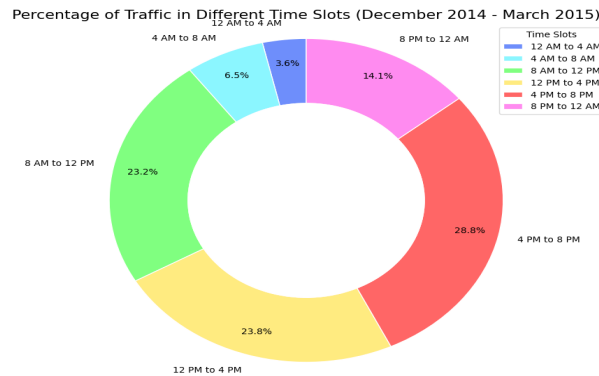


Figure 3.4: Average traffic volume in different time slots.(Phase 1)

The histogram (Fig. 3.6) displays the distribution of total traffic volume over the entire dataset. The frequency of traffic volume is plotted, showing a right-skewed distribution where most data points are concentrated at lower traffic volumes. This indicates that higher traffic volumes are less frequent than lower volumes, typical in urban traffic scenarios where congestion peaks at specific times while remaining low at other times.

The density graph for Phase 1 (Fig. 3.7) covers the period from December 2014 to March 2015. This visualization reveals the distribution of traffic volume, highlighting multiple peaks that correspond to different levels of traffic density throughout the day.

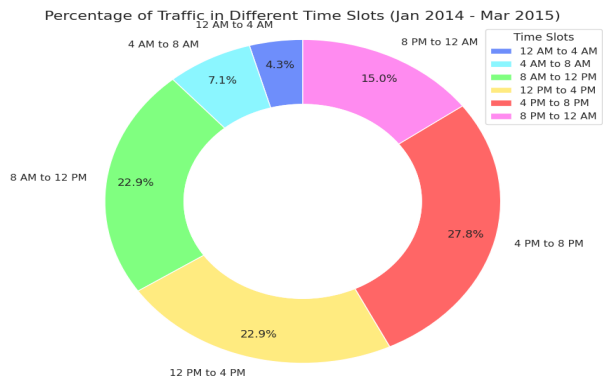


Figure 3.5: Average traffic volume in different time slots.(Phase 2)

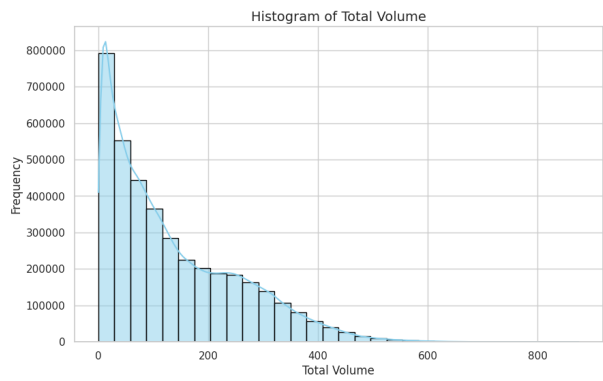


Figure 3.6: Histogram of Total Traffic Volume

Notably, the traffic volume shows significant variability, reflecting the fluctuating nature of traffic flow within this shorter timeframe.

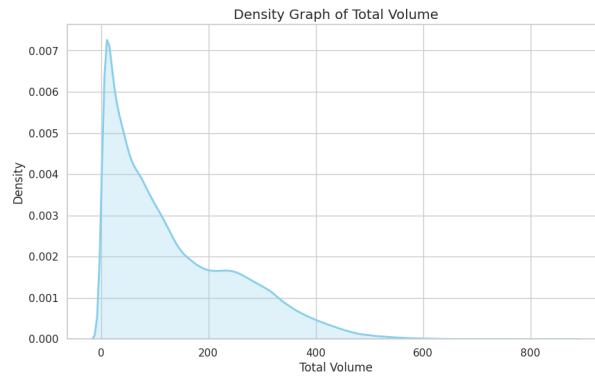


Figure 3.7: Density Graph of Total Volume (Phase 1)

The density graph for Phase 2 (Fig. 3.8) spans from January 2014 to March 2015. This graph also shows the distribution of traffic volume over a longer period, capturing more comprehensive trends and variations. The density graph demonstrates a broader range of traffic volumes with distinct peaks, indicating more pronounced daily and seasonal patterns in traffic flow.

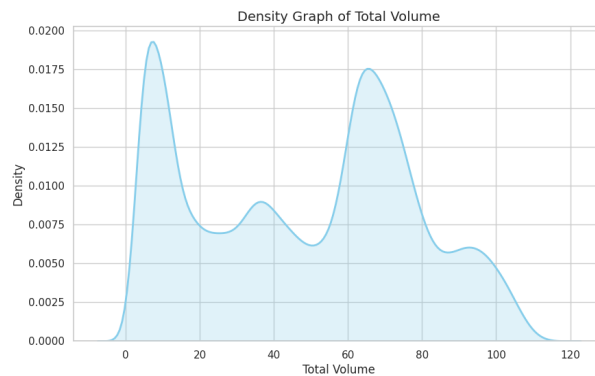


Figure 3.8: Density Graph of Total Volume (Phase 2)

This bar chart in Fig.3.9 displays the total number of vehicles classified into different categories (Class A, Class B, Class C, Class D, and Class 0) along with the total volume of vehicles recorded. The chart shows that Class B vehicles have the highest count, followed by the total volume, indicating that Class B vehicles dominate the traffic flow in the dataset. The visualization helps understand the distribution of vehicle types and their respective counts, providing a clear picture of traffic composition.

The histogram in Fig.3.10 represents the distribution of the average length of vehicles. Most vehicles have an average length concentrated around 500 units, with a sharp peak indicating a high frequency of vehicles within this length range. This histogram helps identify the typical vehicle sizes on the road, which is crucial for traffic management and planning.

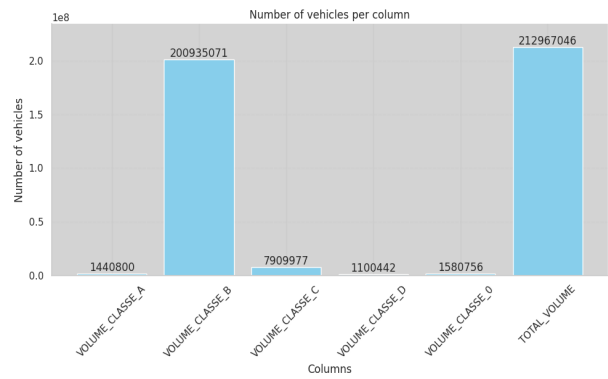


Figure 3.9: Number of vehicles of different Class

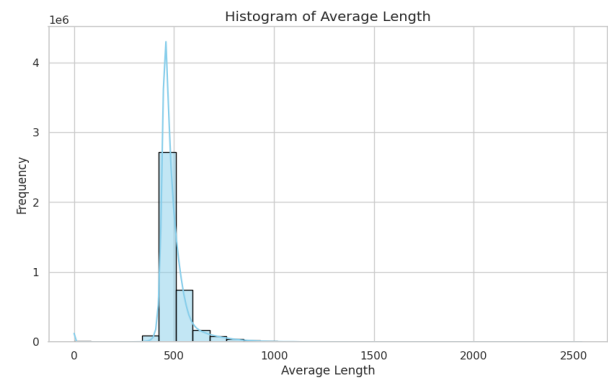


Figure 3.10: Histogram of Average Length

This histogram illustrates the distribution of the light vehicle rate, which is the percentage ratio of light vehicles in the traffic. The majority of the data points are skewed towards higher percentages, indicating that light vehicles constitute a significant portion of the traffic. This visualization is essential for understanding the proportion of light vehicles and planning infrastructure accordingly.

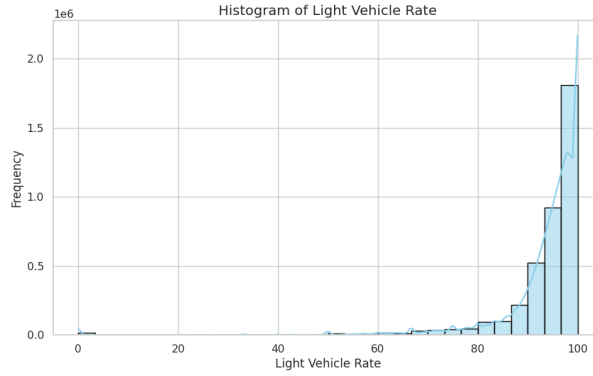


Figure 3.11: Histogram of Light Vehicle Rate

The bar chart in Fig.3.12 shows the total volume of vehicles recorded by different sensor IDs (EQUIPMENTID). This chart highlights the distribution of traffic volumes captured by various sensors, indicating the areas with higher traffic densities. Sensors with IDs around the mid-30s and mid-40s have recorded the highest volumes, suggesting these areas experience heavier traffic.

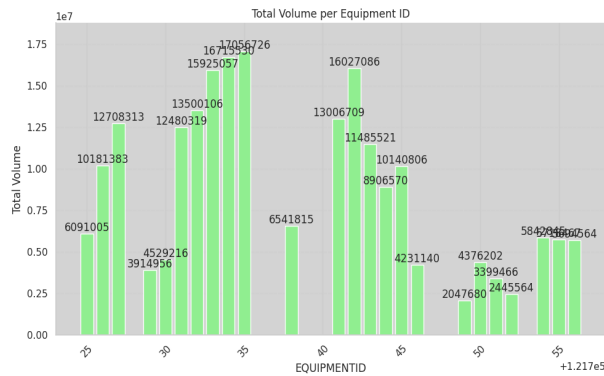


Figure 3.12: Total Volume per equipment ID

By analyzing these visualizations, we can identify consistent peak periods and ensure our models are trained to handle the high variability in traffic volumes during these times. This detailed understanding of traffic patterns is crucial for optimizing traffic management and planning strategies in Porto city.

3.3.2 Time Series Decomposition

Time series decomposition[23] is a crucial analytical method used to break down a complex time series dataset into its core components: trend, seasonality, and residuals. This

method helps to understand and interpret the underlying patterns and behaviours in the data, which is essential for building accurate predictive models. By isolating these components, we can better understand the temporal structures and fluctuations within the data, which enhances our ability to make reliable forecasts.

The trend component reveals the long-term progression of the time series data, indicating whether the data is generally increasing, decreasing, or remaining stable over time. Seasonality captures the repeating patterns or cycles in the data at regular intervals, often influenced by external factors such as time of day, day of the week, or year's season. The residual component represents the irregular fluctuations that cannot be explained by the trend or seasonality, encompassing random noise and anomalies.

In Phase 1 3.13, the time series decomposition of data from December 2014 to March 2015 reveals a moderate upward trend in traffic volume, with consistent daily seasonal patterns. The trend component indicates a gradual increase over time, while the seasonal component highlights regular daily cycles. The residuals show stable irregularities with occasional spikes. This suggests a relatively stable traffic pattern with predictable variations and few anomalies.

In contrast, Phase 2 3.14 covers data from January 2014 to March 2015, exhibiting more complex dynamics in the time series. The trend component demonstrates periods of increase and decrease, reflecting more significant changes in traffic volume over the year. The seasonal component shows consistent daily patterns, though with more pronounced fluctuations compared to Phase 1. The residuals in Phase 2 indicate higher variability and more frequent irregular events affecting traffic volume, suggesting a more dynamic and less predictable traffic pattern. This comparison underscores the importance of considering both stable patterns and irregularities for accurate traffic volume prediction, highlighting the need for robust models that can adapt to varying traffic conditions.

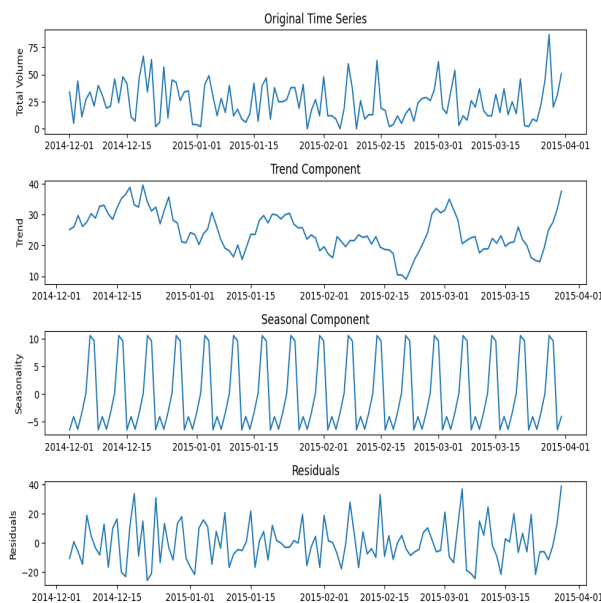


Figure 3.13: Time series decomposition of traffic data.(Phase 1)

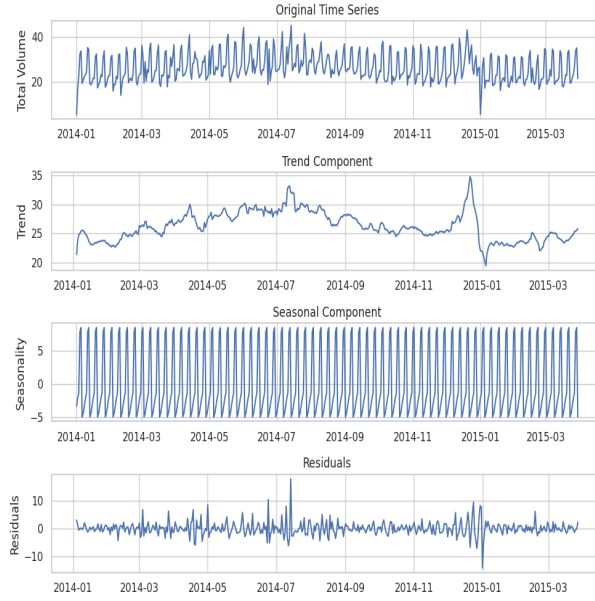


Figure 3.14: Time series decomposition of traffic data.(Phase 2)

3.4 Data Preprocessing

As the primary goal is to predict the traffic volume on a specific date, to improve data analysis and ensure it works with time-based processes, the 'AGG_Period_Start' column in the experiment was changed from an object type to a date-time object type. This change makes it possible to do accurate temporal analysis and use time-related methods. The shift in date-time object type made the data easier to understand and use and ensured that information about time was shown correctly throughout the project. The selected dataset, with 4,092,478 instances, is subjected to a data reduction method to shrink the dataset size and remove redundancy. The particular time intervals are expressed with a 5-minute granularity in the 'AGG.PERIOD.START' column. The dataset is divided into groups based on the 'AGG.PERIOD.START' column because several rows had the same value for this column. The traffic volume for each period is then averaged to get a consolidated picture. This method produced a smaller dataset (35125 instances) with one row for each distinct time interval, making the dataset for analysis more condensed and manageable. Out of the twenty features in the dataset, only two are essential for forecasting: 'AGG.PERIOD.START' and 'TOTAL.VOLUME.' The 'AGG.PERIOD.START' attribute represents time intervals, and turning it into a DateTime object makes it possible to analyze time more accurately. On the other hand, the 'TOTAL.VOLUME' attribute is the target variable that shows the amount of traffic for each time interval. The specific needs of time series forecasting, which focuses on capturing temporal dynamics and predicting future traffic volumes, affect the choice of these two properties.

Chapter 4

Deep Learning Model

This chapter explores the deep-learning models utilized for predicting traffic volume. In contemporary traffic management systems, accurately forecasting traffic volume is paramount for mitigating congestion, optimizing traffic flow, and enhancing the overall efficiency of transportation networks. This chapter will provide a comprehensive overview of various deep learning models that have been applied to this problem, including Artificial Neural Networks (ANN), Long Short-Term Memory (LSTM), Bidirectional Long Short-Term Memory (Bi-LSTM), Recurrent Neural Networks (RNN), and Gated Recurrent Unit (GRU).

Selecting appropriate deep learning models for traffic volume prediction is crucial due to traffic data's inherent complexities and dynamic nature. Traffic patterns are influenced[24] by many factors, including time of day, weather conditions, special events, and roadwork, all of which contribute to the non-linear and highly variable nature of traffic flow. Traditional statistical methods often fail to capture these complexities, making deep learning models, with their ability to learn from large datasets and uncover intricate patterns, an attractive alternative.

Deep learning models excel in handling vast amounts of data and identifying subtle patterns[17] that simpler models might overlook. They can model the data's temporal dependencies and spatial correlations, providing more accurate and robust predictions. The choice of model architecture can significantly impact the prediction accuracy, training time, and computational efficiency. Therefore, this chapter will explore the specific architectures and mechanisms of the aforementioned deep learning models, discuss their strengths and weaknesses, and provide insights into their suitability for traffic volume prediction tasks.

By understanding the intricacies of each model, we can make informed decisions on their application in real-world traffic forecasting scenarios, ultimately contributing to more intelligent and adaptive traffic management systems.

4.1 Overview of Deep Learning Models

Deep learning, a subset of machine learning, involves neural networks with multiple layers that enable the automatic extraction and transformation of features from raw data. This ability to learn and model complex patterns has made deep learning a powerful tool across various domains, including computer vision, natural language processing, and time-series forecasting.

In the context of time-series forecasting, deep learning models are particularly[25] advantageous due to their capacity to handle non-linear relationships and temporal dependencies in data. These models can capture intricate patterns over time, making them well-suited for tasks such as traffic prediction, where data points are sequentially dependent and influenced by numerous external factors.

Traffic prediction is a crucial application of deep learning aimed at forecasting future traffic conditions based on historical data. Accurate traffic predictions[26] enable better traffic management, route optimization, and congestion mitigation, thereby improving urban mobility and reducing environmental impact. Deep learning models have shown significant promise in this field due to their robustness and ability to process vast amounts of traffic data.

This chapter will discuss the following specific deep-learning models used for traffic volume prediction:

Artificial Neural Networks (ANN): These are the foundational deep learning models consisting of interconnected neurons organized in layers. ANNs can learn to map inputs to outputs through training but may struggle with capturing long-term dependencies in sequential data.

Long Short-Term Memory (LSTM): LSTMs are a type of recurrent neural network (RNN) designed to overcome the limitations of traditional RNNs in learning long-term dependencies. They achieve this through a gating mechanism that regulates the flow of information, making them highly effective for time-series forecasting.

Bidirectional Long Short-Term Memory (Bi-LSTM): Bi-LSTMs enhance standard LSTMs by processing data in both forward and backward directions. This bidirectional approach allows the model to capture information from past and future states, providing a more comprehensive understanding of the temporal patterns in the data.

Recurrent Neural Networks (RNN): RNNs are neural networks with loops, enabling information to persist and making them suitable for sequential data. However, they are prone to issues like vanishing and exploding gradients, which can be mitigated by more advanced architectures like LSTMs and GRUs.

Gated Recurrent Unit (GRU): GRUs are a simplified version of LSTMs addressing the vanishing gradient problem. They use gating mechanisms similar to LSTMs but with a reduced number of gates, leading to a simpler and faster model that still captures long-term dependencies effectively.

In the following sections, we will delve into each model's architecture, advantages, and

limitations, illustrating their application and effectiveness in predicting traffic volume.

4.2 Building Deep Learning Models

In this study, we have developed and compared five different deep learning models for time series forecasting: Artificial Neural Network (ANN), Long Short-Term Memory (LSTM), Bidirectional Long Short-Term Memory (Bi-LSTM), Recurrent Neural Network (RNN), and Gated Recurrent Unit (GRU). While varying in their specific architectures, these models share common elements such as the optimizer, loss function, and activation function. We employed the Adam optimiser across all models to ensure consistency and comparability in our experiments. Adam is a popular optimization algorithm in deep learning due to its ability to adjust learning rates dynamically, combining the benefits of AdaGrad and RMSProp. This leads to more efficient and stable training, often resulting in faster convergence and better overall performance.

For the loss function, we utilized the Mean Squared Error (MSE) across all models. MSE is a standard loss function for regression tasks, including time series forecasting, as it measures the average squared difference between predicted and actual values. Minimizing MSE during training helps the models generate predictions that closely align with the true values, thus enhancing forecasting accuracy. We selected the Rectified Linear Unit (ReLU) as the activation function for all models. ReLU introduces non-linearity into the models and helps mitigate the vanishing gradient problem common in deep networks. This activation function effectively captures complex patterns and relationships within the time series data.

The Artificial Neural Network (ANN) model consists of three dense layers. Each layer uses the ReLU activation function to introduce non-linearity and capture complex data patterns. We chose a sequence length of 90, meaning that each input sequence consists of 90 consecutive data points. The ANN model was implemented using the Sequential model from the TensorFlow Keras library, designed to handle and analyze sequential data effectively.

The Long Short-Term Memory (LSTM) model is designed to analyse sequential data. It includes a single LSTM layer with 64 units and uses the ReLU activation function. Similar to the ANN model, a sequence length of 90 was chosen. The LSTM architecture, with its memory cell and gating mechanisms, is well-suited for capturing long-term dependencies in time series data. The Bidirectional Long Short-Term Memory (Bi-LSTM) model extends the LSTM architecture by incorporating a bidirectional LSTM layer with 64 units. This model utilizes the ReLU activation function and captures dependencies in both forward and backward directions within the input sequence. The bidirectional approach enhances the model's ability to understand the temporal dynamics of the time series data more comprehensively.

The Recurrent Neural Network (RNN) model, based on the SimpleRNN architecture, includes a single SimpleRNN layer with 64 units. It also employs the ReLU activation

Paramter/Hyperparameter	ANN	LSTM	BiLSTM	RNN	GRU
Scaling	MinMaxScaler (0, 1)	MinMaxScaler (0, 1)	MinMaxScaler (0, 1)	MinMaxScaler (0, 1)	MinMaxScaler (0, 1)
Sequence Length	90	90	90	90	90
Hidden Layers	2 (64, 32 units)	64 (LSTM units)	64 (BiLSTM units)	64 (RNN units)	64 (GRU units)
Activation Function	Relu	Relu	Relu	Relu	Relu
Output Layer	1 unit	1 unit	1 unit	1 unit	1 unit
Optimizer	Adam	Adam	Adam	Adam	Adam
Loss Function	Mean Squared Error	Mean Squared Error	Mean Squared Error	Mean Squared Error	Mean Squared Error
Early Stopping	N/A	N/A	N/A	Monitor: val_loss, Patience: 3	Monitor: val_loss, Patience: 5

Table 4.1: Hyparparameter and Parameter of Each Model

function. The input sequences are reshaped to fit the SimpleRNN layer’s input shape. The RNN model effectively captures temporal dependencies and patterns within sequential data with its recurrent connections. The Gated Recurrent Unit (GRU) model includes a single GRU layer with 64 units and uses the ReLU activation function. The GRU architecture, equipped with gating mechanisms, selectively updates and resets information. This capability allows the GRU to capture long-term dependencies while addressing the vanishing gradient problem. The GRU model uses a sequence length of 90 like the other models.

We maintained consistency in the choice of hyperparameters across both experimental phases to ensure comparability and reliability of results. The sequence length of 90 was selected based on preliminary experiments showing it provided a good balance between capturing sufficient temporal dependencies and maintaining manageable computational complexity. The use of 64 units in hidden layers was similarly chosen for its effectiveness in balancing model complexity and performance across different architectures.

By adopting these distinct deep learning architectures and specifying their respective parameters, we aim to explore their effectiveness in capturing temporal patterns and forecasting future values in the given time series dataset. These models provide various tools for analyzing and predicting time-dependent phenomena, contributing to the advancement of time series forecasting research.

4.3 Activation Function

Activation functions are a critical component of neural networks, serving as the mechanism that introduces non-linearity into the model. Without activation functions, neural networks would perform linear transformations, limiting their ability to capture and model complex patterns and relationships in the data. Activation functions enable neural networks to learn intricate data distributions, making them capable of solving non-linear problems. This non-linearity allows the networks to stack multiple layers, each transforming the data in a progressively abstract manner, ultimately enabling deep networks to perform tasks such as image recognition, natural language processing, and time-series forecasting with high accuracy.

In our experiment, we employed the Rectified Linear Unit (ReLU) activation function across all models. ReLU is widely used in deep learning due to its simplicity and effectiveness. It introduces non-linearity to the model, allowing it to learn complex patterns while

being computationally efficient. Unlike activation functions like sigmoid or tanh, which can suffer from the vanishing gradient problem and slow down the learning process, ReLU mitigates this issue by setting negative input values to zero, ensuring that gradients remain large enough during backpropagation. This characteristic makes ReLU particularly suitable for deep networks, as it facilitates faster and more reliable convergence during training.

The ReLU activation function is mathematically defined as:

$$f(x) = \max(0, x) \quad (4.1)$$

This simple yet powerful function introduces the necessary non-linearity in neural networks, enabling them to learn complex and high-dimensional data patterns effectively. By transforming only the positive parts of the input while nullifying the negatives, ReLU maintains computational efficiency and supports the learning of robust feature representations in deep networks.

4.4 Evaluation Metrics Overview

To comprehensively assess the performance of the deep learning models used for traffic volume prediction, we employed four key evaluation metrics: Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), Mean Absolute Percentage Error (MAPE), and R-squared (R^2). Each of these metrics provides unique insights into the accuracy and reliability of the models, helping us understand their strengths and weaknesses in different scenarios.

4.4.1 Mean Absolute Error(MAE)

MAE measures the average magnitude of the errors in a set of predictions without considering their direction. It is calculated as the average absolute difference between predicted and actual values. The formula for MAE is:

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i|, \quad (4.2)$$

Where y_i represents the actual values, $\hat{y}_i$ represents the predicted values, and n is the number of observations. MAE provides a straightforward interpretation of the prediction error, making it easy to understand how much the predictions deviate from the actual values on average.

4.4.2 Root Mean Square Error(RMSE)

RMSE is a quadratic scoring rule that also measures the average magnitude of the error. However, it gives a higher weight to large errors by squaring the differences before averaging them, making it more sensitive to outliers. The formula for RMSE is:

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2}. \quad (4.3)$$

By penalizing larger errors more heavily, RMSE provides a more comprehensive measure of model performance, particularly in scenarios where large errors are undesirable.

4.4.3 Mean Absolute Percentage Error(MAPE)

MAPE expresses the prediction accuracy as a percentage by comparing the absolute error to the actual values. It is calculated as:

$$MAPE = \frac{100\%}{n} \sum_{i=1}^n \left| \frac{y_i - \hat{y}_i}{y_i} \right|. \quad (4.4)$$

MAPE is particularly useful for understanding the relative error in predictions, making it easier to compare performance across different datasets and scales. However, it can be sensitive to very small actual values, leading to disproportionately large percentage errors.

4.4.4 R-squared(R^2)

R-squared, or the coefficient of determination, measures the proportion of the variance in the dependent variable that is predictable from the independent variables. It is calculated as:

$$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2}, \quad (4.5)$$

Where $\bar{y}$ is the mean of the actual values. R-squared ranges from 0 to 1, with higher values indicating better model performance. It provides an overall measure of how well the model captures the variability in the data, making it a valuable metric for model evaluation.

Using these evaluation metrics, we comprehensively understand the model's performance in predicting traffic volume, enabling us to identify the most accurate and reliable models for our traffic forecasting tasks.

Chapter 5

Experimental Results

This chapter presents and analyzes the results obtained from the application of various deep learning models for traffic volume prediction. This analysis aims to evaluate the performance of each model in forecasting traffic patterns, thereby contributing to the development of more efficient and reliable traffic management systems.

In this study, we have employed five different deep learning models: Artificial Neural Network (ANN), Long Short-Term Memory (LSTM), Bidirectional Long Short-Term Memory (Bi-LSTM), Recurrent Neural Network (RNN), and Gated Recurrent Unit (GRU). To assess the performance of these models, we utilized four key evaluation metrics: Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), Mean Absolute Percentage Error (MAPE), and R-squared (R^2). These metrics provide a comprehensive understanding of the accuracy and effectiveness of each model in predicting traffic volume.

The experiments were conducted in two phases. The first phase involved a 4-month dataset, where three months of data were used for training the models and one month for testing. The second phase extended the dataset to 15 months, with twelve months allocated for training and three months for testing. By comparing the results from these two experimental phases, we aim to understand the impact of extended training periods on model performance and the robustness of each model over different time frames.

This chapter will systematically present the comparative results of the models, including detailed visualizations and analyses of their performance metrics. Additionally, we will explore various aspects such as the actual vs. predicted traffic volumes, weekly MAE analysis, and specific day performance analysis to provide a comprehensive evaluation of the models' predictive capabilities.

5.1 Experimental Result & analysis

The results obtained from the five deep learning models provide comprehensive insights into their performance for time series forecasting across two experimental phases: the 4-month and 15-month datasets. The evaluation metrics, including Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), Mean Absolute Percentage Error (MAPE),

and R-squared (R^2), are crucial in assessing each model's accuracy and predictive capabilities. Additionally, the learning curves for each model offer further understanding of their training and generalization performance.

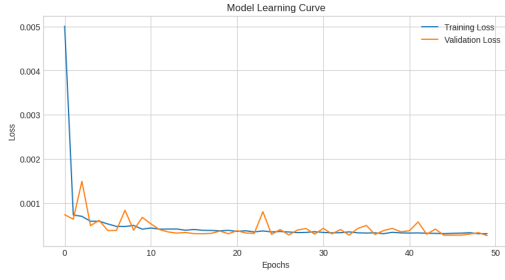
5.1.1 Evaluation Metrics & Learning Curve Analysis for Phase 1

Analyzing the results from Table 5.1, we observe that all models achieved relatively low MAE values, indicating their effectiveness in estimating the forecasting errors. The RNN model performed the best with an MAE of 1.52, closely followed by the GRU model with an MAE of 1.87. The ANN, LSTM, and Bi-LSTM models achieved slightly higher MAE values of 1.98, 2.27, and 2.84, respectively. Similarly, considering the RMSE metric, the RNN model achieved the lowest value of 2.06, indicating its ability to provide accurate predictions. The ANN model is closely followed with an RMSE of 2.41. The LSTM, Bi-LSTM, and GRU models achieved slightly higher RMSE values of 3.47, 3.73, and 2.70, respectively.

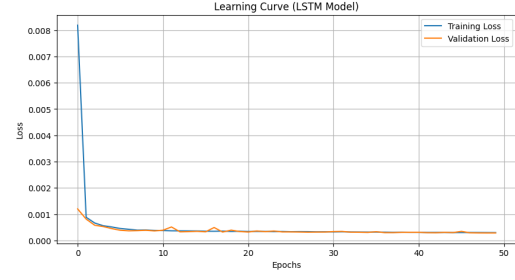
When evaluating the R-squared (R^2) metric, which provides insights into the models' ability to explain the variance in the target variable, the RNN model demonstrated the highest performance with an R^2 value of 0.9953. This indicates a strong correlation between the predicted and actual values. The ANN, LSTM, Bi-LSTM, and GRU models achieved R^2 values of 0.9935, 0.9867, 0.9846, and 0.9919, respectively. Based on the evaluation metrics, the RNN model outperformed the other models in terms of MAE, RMSE, and R^2 , demonstrating its superior performance in forecasting time series data. Furthermore, the result of the deep learning model outperformed the traditional method by 2.36% in MAE and other metrics on this dataset. Moreover, the traditional method used one month of data, whereas we experimented using four months of data[7]. The RNN model's ability to capture temporal dependencies and patterns within the sequential data contributes to its accurate predictions and overall effectiveness. It is important to note that the performance of each model may vary depending on the specific dataset and problem at hand. Therefore, it is recommended to consider the dataset's characteristics and the forecasting task's specific requirements when selecting the most suitable model. Overall, the findings from this study highlight the effectiveness of deep learning models, particularly the RNN model, for time series forecasting. These models offer valuable tools for capturing complex patterns and relationships in sequential data, enabling accurate predictions and facilitating informed decision-making in various domains.

Table 5.1: Evaluation matrix summary of models(Phase 1)

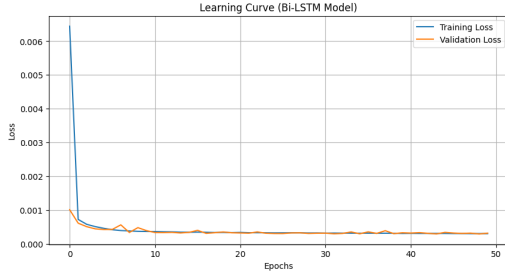
Model	MAE	RMSE	MAPE	R-square (R^2)
ANN	1.98	2.41	0.096	0.9935
LSTM	2.27	3.47	0.062	0.9867
Bi-LSTM	2.84	3.73	0.113	0.9846
RNN	1.52	2.06	0.046	0.9953
GRU	1.87	2.70	0.053	0.9919



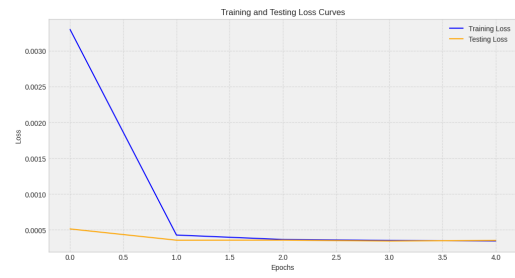
(a) ANN Learning Curve



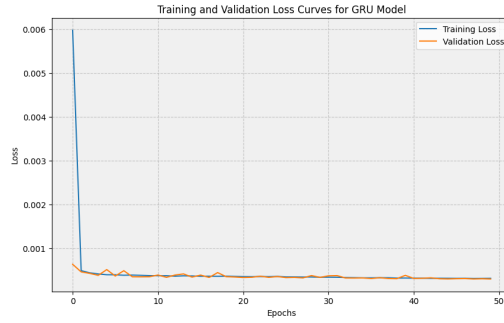
(b) LSTM Learning Curve



(c) BiLSTM Learning Curve



(d) RNN Learning Curve



(e) GRU Learning Curve

Figure 5.1: Learning Curve for All Models(Phase 1)

Analyzing the training and testing loss curves is crucial for evaluating a machine learning model's performance and generalization abilities. In the provided learning curves for different models (ANN, LSTM, Bi-LSTM, RNN, and GRU), we observe a consistent pattern of training loss and validation loss behavior. The training loss, depicted by the training curve, generally exhibits a consistent downward trend across all models, suggesting that each model effectively captures the inherent patterns in the dataset. For the ANN model (Fig. 1), the training and validation loss curves show convergence, indicating that the model has learned the dataset well without overfitting. The validation loss stabilizes after an initial decline, reflecting good generalization. In the LSTM model's learning curve (Fig. 2), a similar pattern is observed where both the training and validation loss curves converge and remain stable after initial fluctuations. This behavior indicates that the LSTM model effectively learns long-term dependencies in the time series data. The Bi-LSTM model's learning curve (Fig. 3) also demonstrates effective learning, with both

training and validation loss curves converging and stabilizing. The Bi-LSTM's bidirectional nature helps capture complex temporal patterns, which is reflected in the stable loss curves. For the GRU model (Fig. 4), the learning curves show a smooth convergence, with the validation loss closely following the training loss. This indicates that the GRU model effectively captures temporal dependencies without significant overfitting. The RNN model (Fig. 5) exhibits a consistent downward trend in the training loss, with the validation loss curve stabilizing, indicating effective learning and good generalization.

The evaluation metrics provide further evidence to support these observations. Specifically, the metrics indicate low Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), Mean Absolute Percentage Error (MAPE), and high R-squared (R2) values across different models. For instance, the RNN model achieved an MAE of 1.52, RMSE of 2.06, MAPE of 0.046, and R-squared of 0.9953 for the 4-month experiment, indicating superior performance. The ANN and GRU models also performed well with relatively low error metrics and high R-squared values.

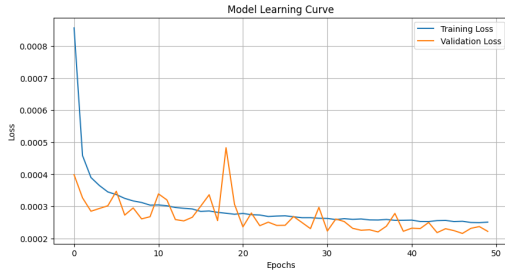
5.1.2 Evaluation Metrics & Learning Curve Analysis for Phase 2

The evaluation metrics for the models applied in Phase One provide insightful information regarding their performance in forecasting traffic volume. The models' Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), Mean Absolute Percentage Error (MAPE), and R-squared (R2) values are as follows:

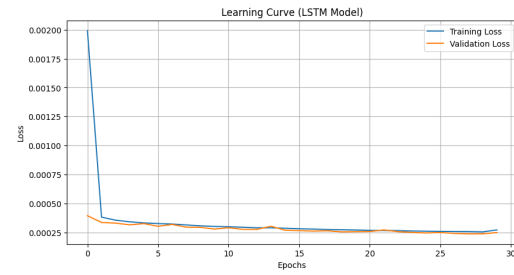
The ANN model achieved an MAE of 1.3707, RMSE of 1.954, MAPE of 0.046, and R2 of 0.9934. These values indicate that the ANN model performed reasonably well, with relatively low error rates and a high R-squared value, showing good predictive accuracy. The LSTM model outperformed the ANN model slightly, with an MAE of 1.3185, RMSE of 1.815, MAPE of 0.045, and R2 of 0.9963. The lower error metrics and higher R-squared value suggest that LSTM captured temporal dependencies in the data more accurately. The Bi-LSTM model also performed well, with an MAE of 1.4099, RMSE of 1.987, MAPE of 0.044, and R2 of 0.9955. Although its MAE and RMSE were slightly higher than LSTM, its MAPE was the lowest among the models, indicating its efficiency in handling percentage errors. The RNN model showed an MAE of 1.4881, RMSE of 2.067, MAPE of 0.051, and R2 of 0.9952. While its performance was commendable, it had higher error rates compared to ANN, LSTM, and Bi-LSTM models, suggesting it was less effective in capturing complex temporal patterns. The GRU model achieved the best overall performance with an MAE of 1.3096, RMSE of 1.829, MAPE of 0.041, and R2 of 0.9962. The GRU's lower error metrics and high R-squared value demonstrate its robustness in effectively learning and predicting traffic volume data.

Table 5.2: Evaluation matrix summary of models(Phase 2)

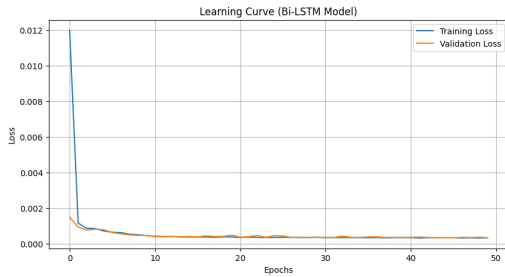
Model	MAE	RMSE	MAPE	R-square (R2)
ANN	1.3707	1.954	0.046	0.9934
LSTM	1.3185	1.815	0.045	0.9963
Bi-LSTM	1.4099	1.987	0.044	0.9955
RNN	1.4881	2.067	0.051	0.9952
GRU	1.3096	1.829	0.041	0.9962



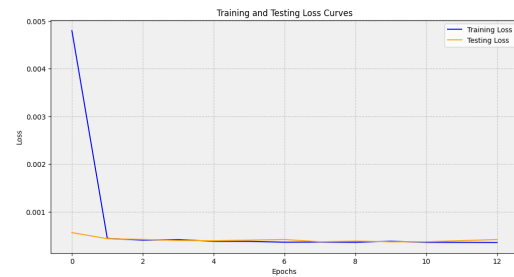
(a) ANN Learning Curve



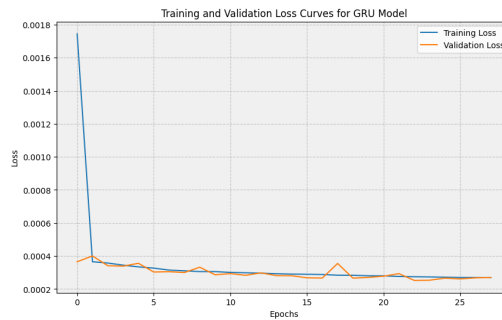
(b) LSTM Learning Curve



(c) BiLSTM Learning Curve



(d) RNN Learning Curve



(e) GRU Learning Curve

Figure 5.2: Learning Curve for All Models(Phase 2)

The learning curve for the ANN model (Figure 1) shows a consistent downward trend in both training and validation losses, indicating effective learning and generalization without overfitting. However, the validation loss exhibits noticeable fluctuations throughout the epochs, suggesting instability in the model's performance on the validation set. The LSTM model's learning curve (Figure 2) also exhibits a steady decrease in loss values, with minimal divergence between training and validation losses, suggesting good generalization.

to unseen data. The Bi-LSTM model’s learning curve (Figure 3) reflects a similar pattern, with training and validation losses decreasing and stabilizing over epochs. The close alignment of these curves indicates that the model learned effectively without overfitting. The RNN model’s learning curve (Figure 4) shows a consistent decrease in training and validation losses, although with slightly more fluctuation than the other models. This fluctuation might indicate some instability during training but overall suggests good learning. The GRU model’s learning curve (Figure 5) demonstrates a rapid decrease in both training and validation losses, stabilizing at low values. This curve indicates efficient learning and excellent generalization to the validation data.

Based on the evaluation metrics and learning curves, the GRU model stands out as the best-performing model for Phase One. It achieved the lowest MAE and RMSE, indicating high accuracy in predicting traffic volumes. Additionally, its learning curve shows rapid convergence and stability, suggesting effective learning and generalization. The LSTM model also performed well, with slightly higher error metrics but excellent generalization, as evidenced by its learning curve. The Bi-LSTM model’s low MAPE highlights its efficiency in percentage error handling, making it a strong candidate as well. The ANN model performed well but outperformed the recurrent architectures, especially GRU and LSTM. While effective, the RNN model showed higher error rates and some instability during training, making it less robust than the other models. Overall, the GRU model’s low error rates and stable learning balance makes it the most reliable choice for traffic volume forecasting in this study.

While all the methods used in this research achieved very high R-squared (R^2) values, it is crucial to recognize that R^2 alone may not always be sufficient to determine the best-performing model, especially when the scores are close. In such cases, relying solely on R^2 can be misleading as it does not always account for other factors like the magnitude of prediction errors. To ensure a more comprehensive evaluation, additional metrics such as Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and Mean Absolute Percentage Error (MAPE) were used to assess the models’ accuracy and reliability. These metrics provide deeper insights into the models’ prediction performance, highlighting the magnitude of errors and their impact on real-world applications.

Factors beyond accuracy metrics were also considered when selecting the most suitable model, such as the model’s computational efficiency, training time, and ability to adapt to varying traffic conditions. For instance, although the Recurrent Neural Network (RNN) model showed the highest R^2 value and lowest error rates in Phase 1, the Gated Recurrent Unit (GRU) model was preferred in Phase 2 for its efficiency in handling longer time series data with complex patterns.

5.1.3 Analysis Summary

In summary, the performance of deep learning models varied across the two phases due to differences in the dataset lengths and temporal dependencies. For shorter-term forecasting (Phase 1), the RNN model was the most effective, likely due to its ability to capture short-term temporal patterns efficiently. For longer-term forecasting (Phase 2), the GRU model outperformed others by effectively capturing long-term dependencies and maintaining stability in learning, as reflected in the lower error metrics and stable learning curves. The GRU model's gating mechanisms gave it an advantage in handling the complexities of a more extended dataset, making it the most reliable model for the 15-month data period. Therefore, the choice of model is not solely dependent on R2 values but also on a holistic assessment of the model's robustness, scalability, and overall suitability for the given data and context. This comprehensive analysis helps ensure that the selected model provides high accuracy and practical utility in real-world traffic forecasting scenarios.

5.2 Experimental Setup

In this study, the experimental setup was designed to leverage the capabilities of modern computational resources and robust programming tools to build and evaluate deep learning models for traffic volume prediction. The setup can be summarized as follows:

Platform:

Google Colab, a free cloud-based service, was used as the primary platform for running the experiments. Colab provides access to powerful GPU resources, essential for efficiently training deep learning models. The collaborative environment of Google Colab also facilitated easy sharing and execution of code, which was beneficial for iterative model development and testing.

Programming Language

Python, a versatile and widely used programming language in the data science community, was employed for all aspects of the experiment. Its extensive libraries and frameworks for machine learning and data processing made it the ideal choice for this study.

Libraries and Framework

Several Python libraries were utilized to support various stages of the experiment, including data preprocessing, model building, training, and evaluation:

- **NumPy and Pandas:** Used for data manipulation and analysis.
- **Matplotlib and Seaborn:** Employed for data visualization to gain insights into the dataset.
- **Scikit-learn:** Utilized for scaling the data and evaluating model performance.
- **TensorFlow and Keras:** Core libraries for building, training, and evaluating the deep learning models.

By leveraging Google Colab for its computational resources, Python for its robust programming capabilities, and specialized libraries for deep learning and data handling, this experimental setup ensured efficient model development, training, and evaluation. This approach facilitated the exploration of different deep learning architectures and their effectiveness in traffic volume prediction.

5.3 Result Visualization

The purpose of this section is to visually present the performance of our developed deep-learning models by comparing the actual and predicted values on the test set. Visualizing these comparisons is crucial for several reasons. Firstly, it allows for an intuitive assessment of how closely the models' predictions align with the actual data, providing a clear, immediate understanding of their accuracy. Secondly, visual representations can reveal patterns, trends, and anomalies not apparent from numerical evaluation metrics alone. Such insights can help identify the strengths and weaknesses of each model, guiding further improvements and refinements. By examining these visualizations, we can gain a deeper appreciation of the models' capabilities and their potential application in real-world forecasting scenarios.

5.3.1 Overall Performance Analysis

Phase 1(4 Month Data)

From figure 5.3 to 5.7, the figures present the actual versus predicted values for all models over one month in Phase 1. The models included are ANN, LSTM, BiLSTM, RNN, and GRU. These visualizations illustrate how well each model captures the patterns in the data, comparing the actual traffic volume with the predicted values generated by each model. The density of the data points in these graphs makes it challenging to assess the performance of each model visually.

Phase 2(15 Month Data)

Similarly, From figure 5.8 to 5.12 displays the actual versus predicted values for all models over three months in Phase 2. The figures represent ANN, LSTM, BiLSTM, RNN, and GRU predictions. These graphs serve the same purpose as those in Phase 1, allowing for a comparison of model performance over a longer period. Again, the high density of data points complicates a straightforward visual evaluation of the models' effectiveness.

Due to the dense nature of the data, these figures may be difficult to interpret directly. Therefore, to provide a clearer understanding of model performance, the upcoming section will focus on visualizing the predictions for the first day, the best day, and the worst day based on daily MAE values. This approach offers more detailed insights into the models' capabilities and performance on specific days.

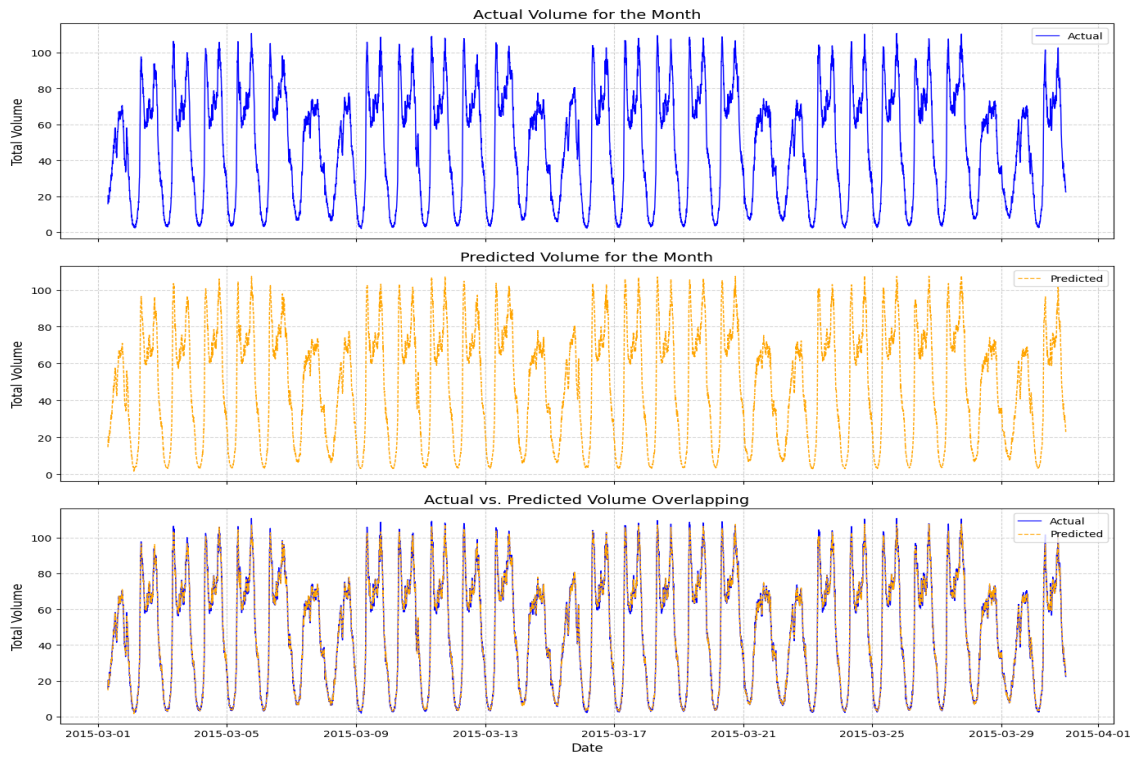


Figure 5.3: ANN 1 month Prediction (Actual vs Predicted)

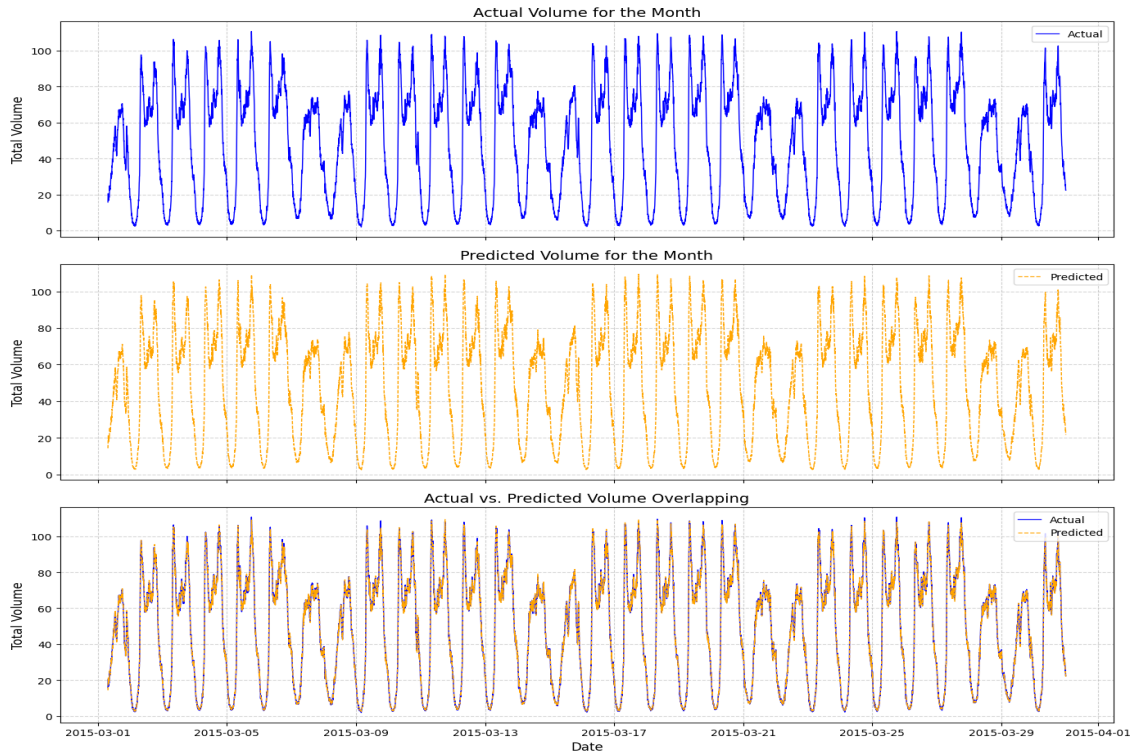


Figure 5.4: LSTM 1 month Prediction (Actual vs Predicted)

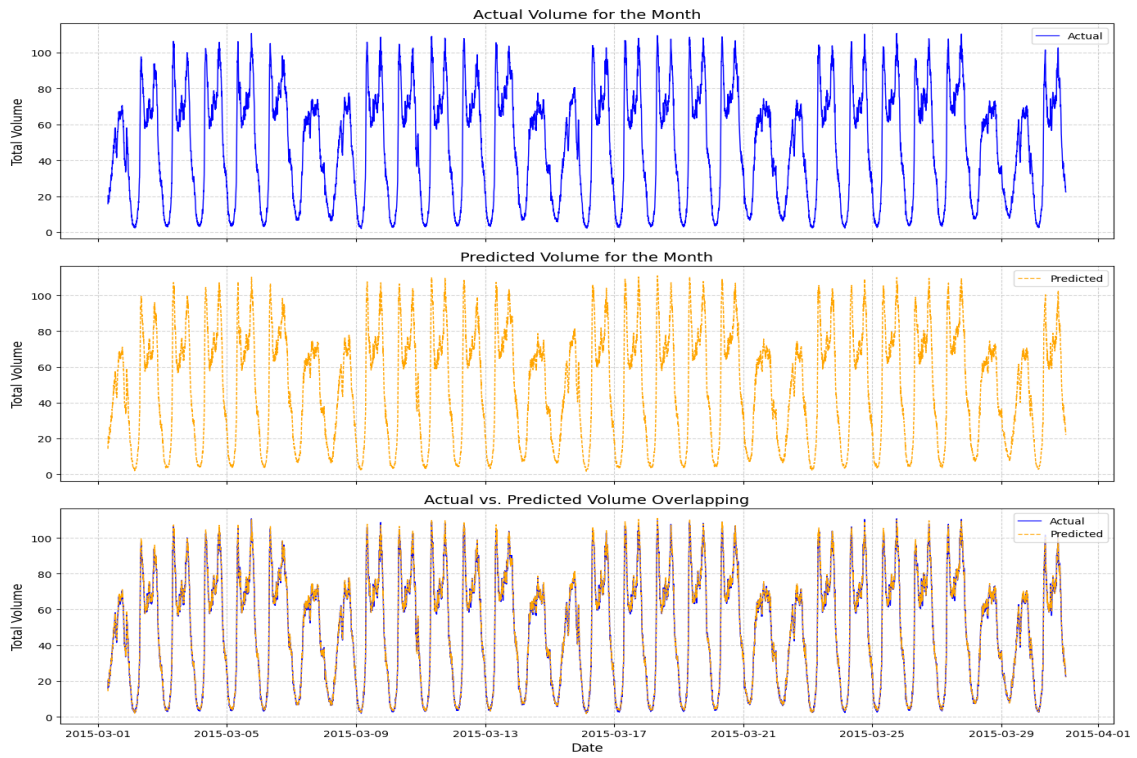


Figure 5.5: Bi-LSTM 1 month Prediction(Actual vs Predicted)

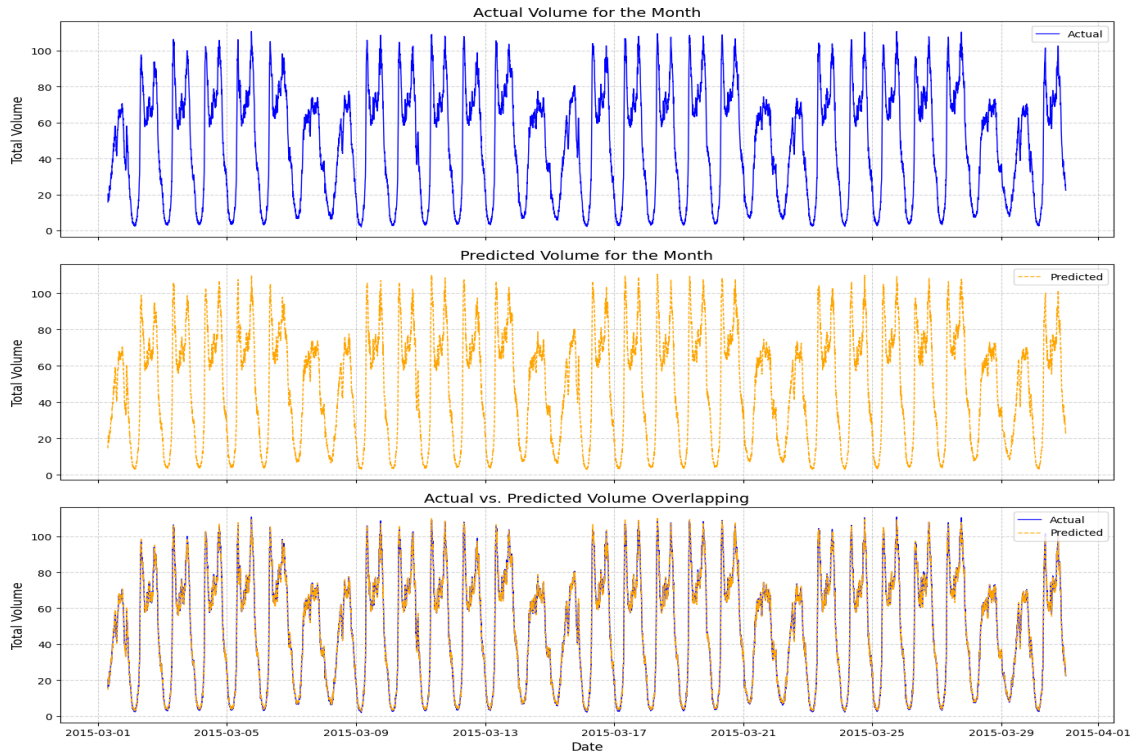


Figure 5.6: RNN 1 month Prediction(Actual vs Predicted)

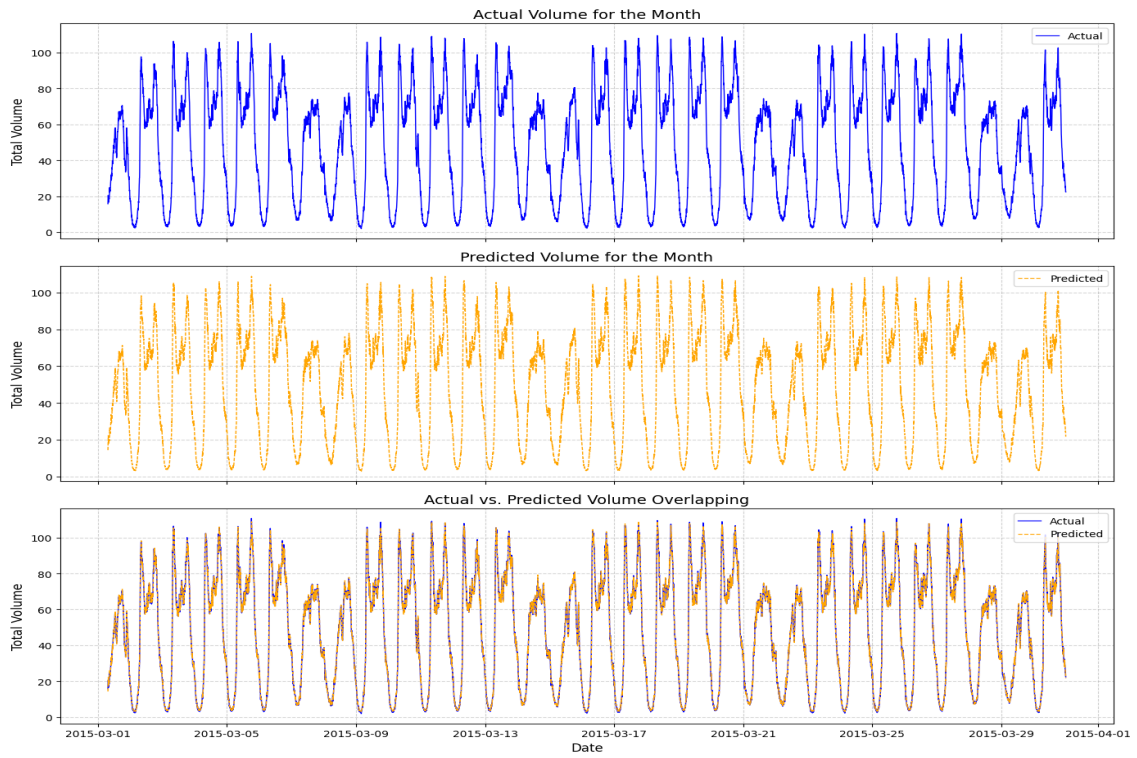


Figure 5.7: GRU 1 month Prediction (Actual vs Predicted)

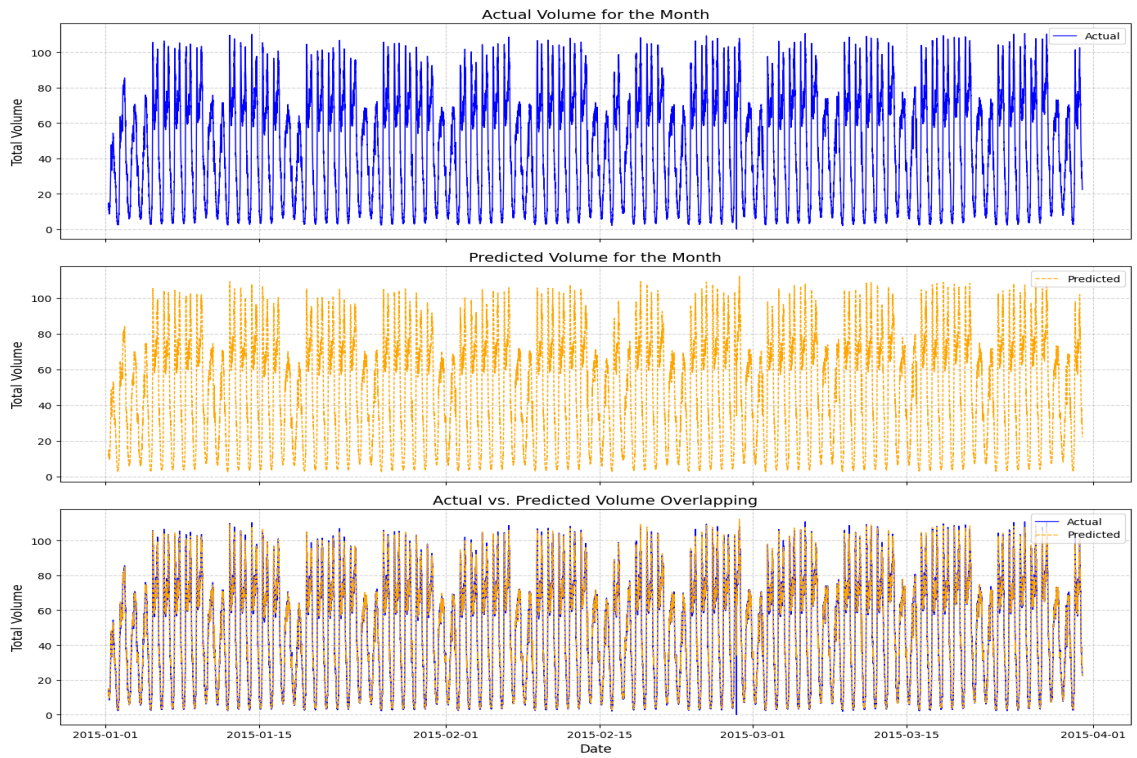


Figure 5.8: ANN 3 month Prediction (Actual vs Predicted)

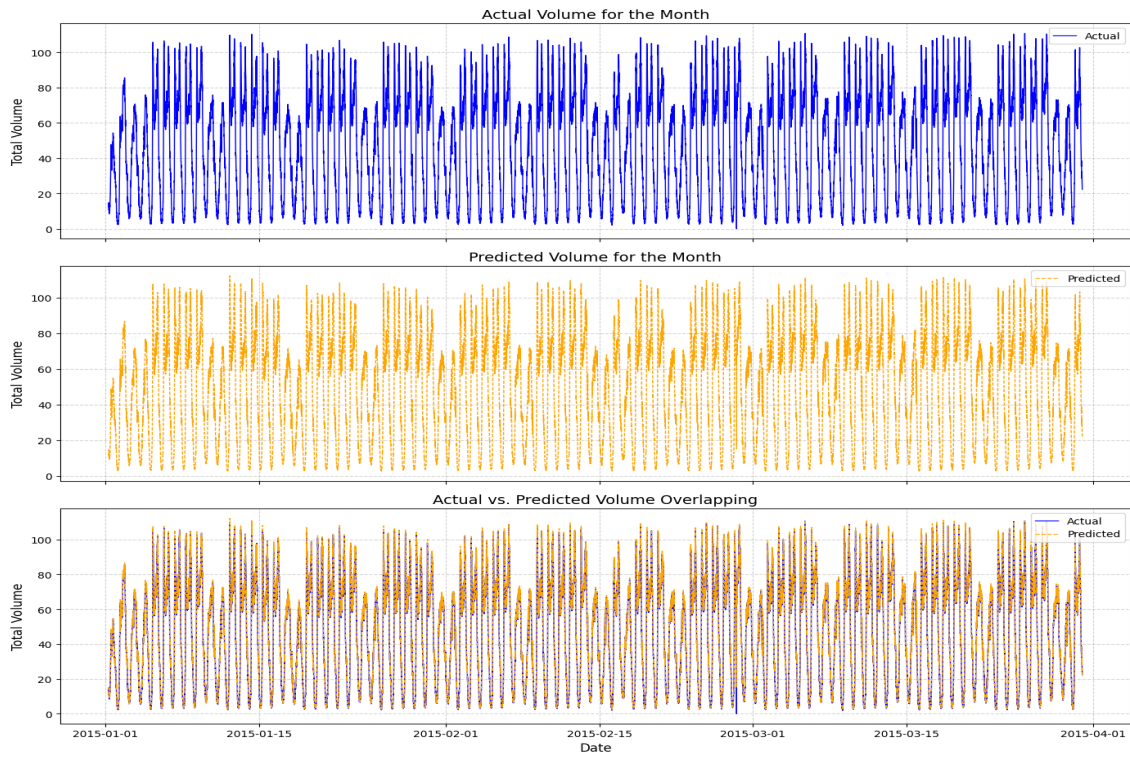


Figure 5.9: LSTM 3 month Prediction(Actual vs Predicted)

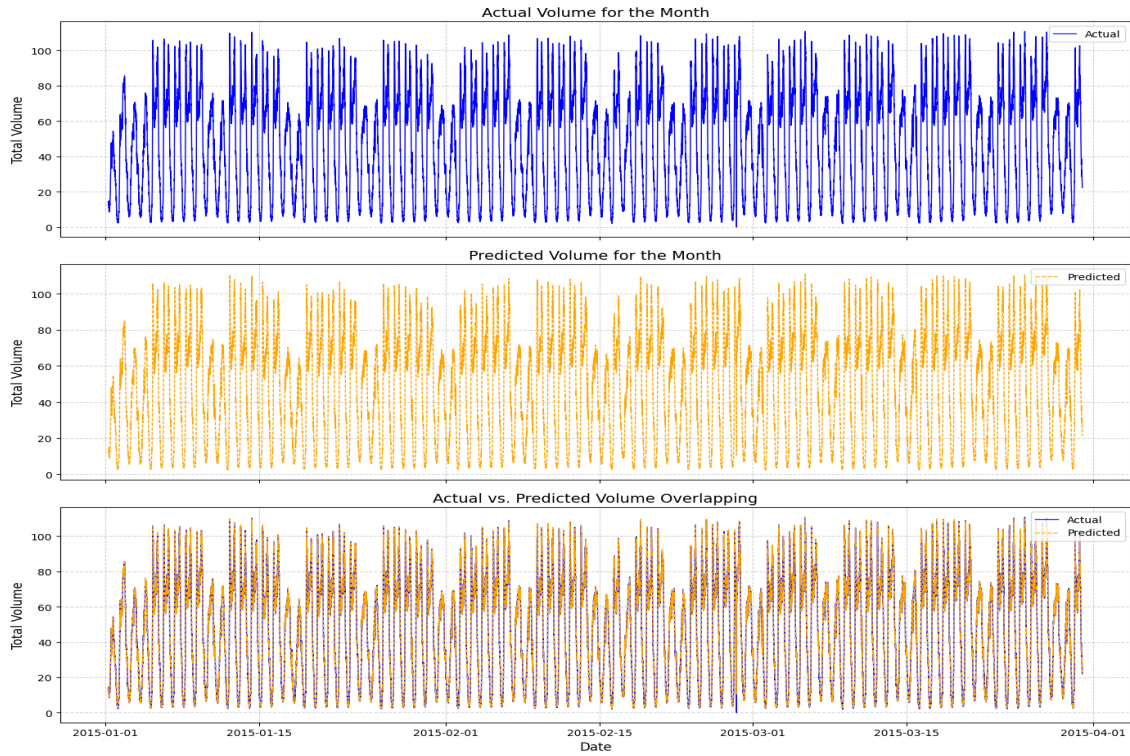


Figure 5.10: Bi-LSTM 3 month Prediction(Actual vs Predicted)

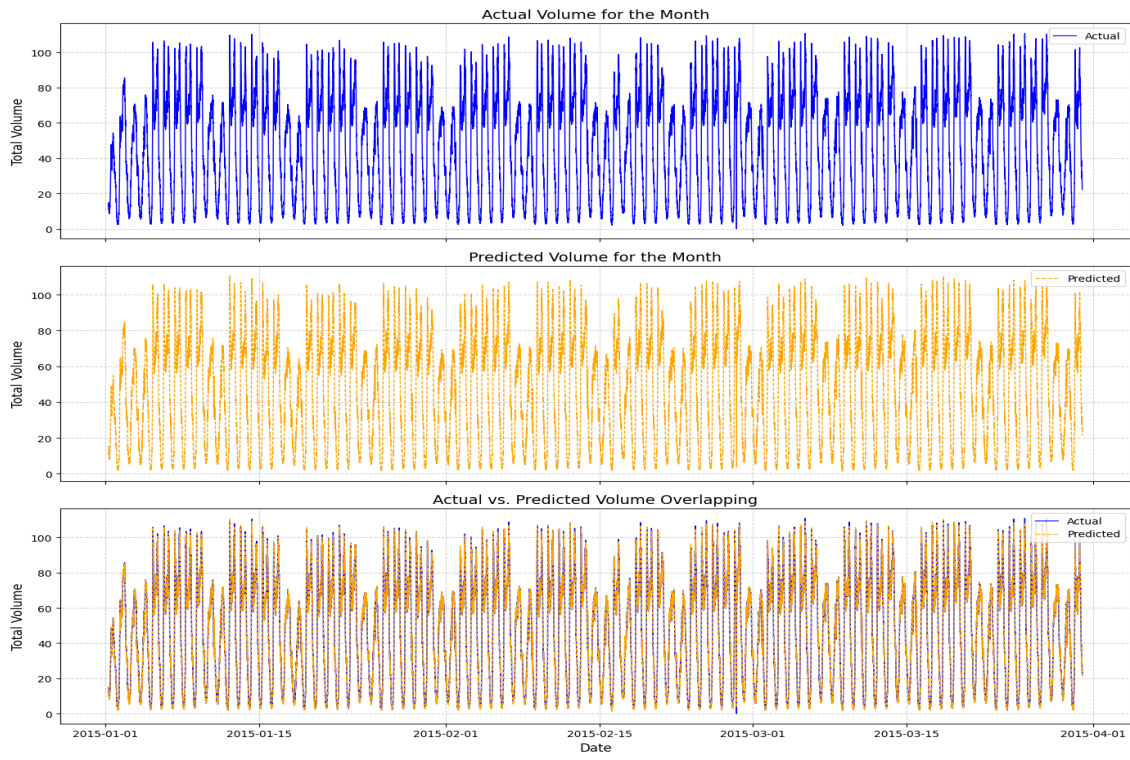


Figure 5.11: RNN 3 month Prediction(Actual vs Predicted)

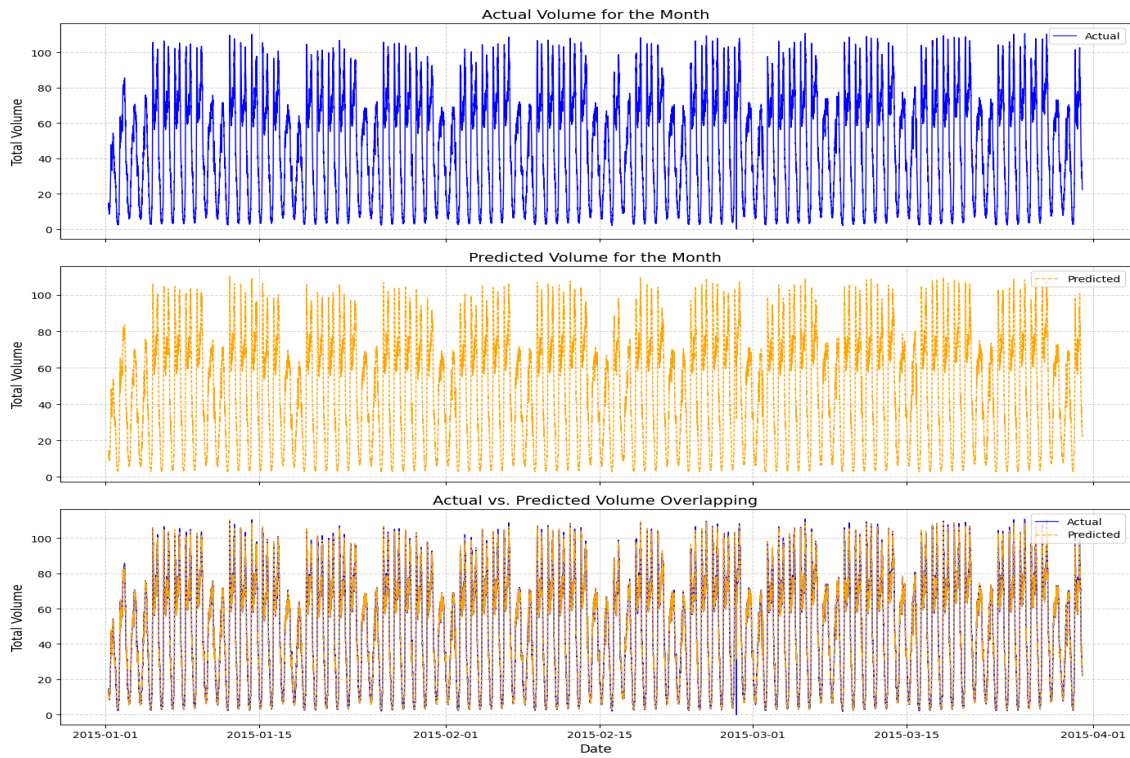
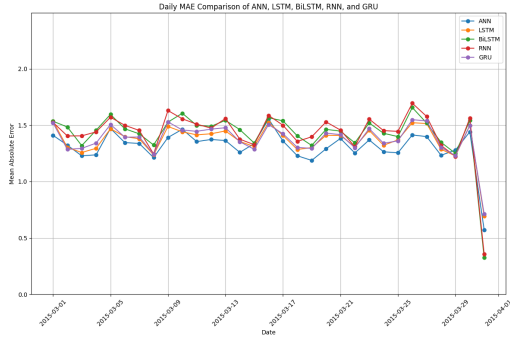


Figure 5.12: GRU 3 month Prediction(Actual vs Predicted)

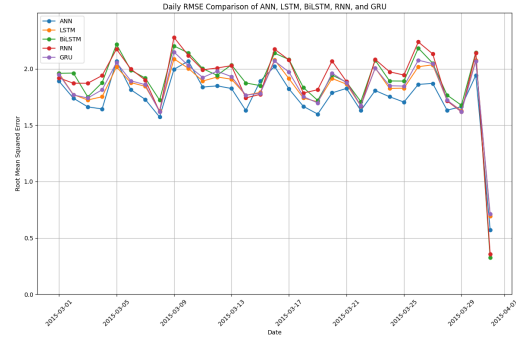
5.3.2 Daily Evaluation Metrics Comparison

Phase 1(4 month data)

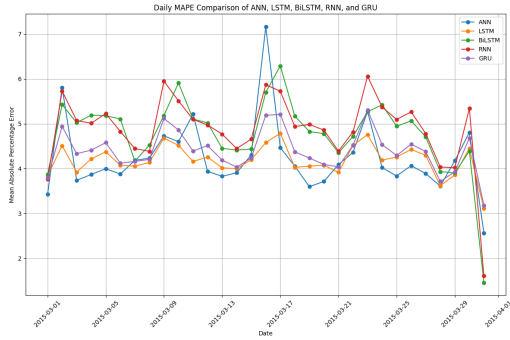
Figure 5.13 illustrates the daily performance metrics for all models during Phase 1 (4-month data). The graphs provide a comprehensive view of the day-to-day prediction accuracy and variability for each model, allowing for a detailed comparison of their performance across different days. By analyzing these metrics, including MAE, RMSE, MAPE, and R-squared values, we can assess the consistency and reliability of each model in capturing the underlying patterns



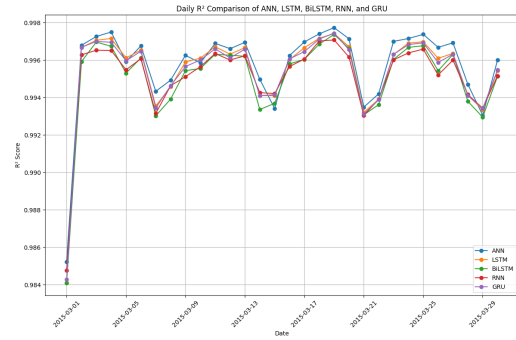
(a) Comparison of daily MAE



(b) Comparison of daily RMSE



(c) Comparison of daily MAPE



(d) Comparison of daily R-squared

Figure 5.13: Daily Evaluation Matrices Value Comparison for all models(Phase 1)

Phase 2(15 month data)

Figure 5.14 showcases the daily performance metrics for all models during Phase 2 (15-month data). This extended dataset allows us to evaluate the models' prediction performance and stability over a longer period. These visualizations help in identifying trends, periodicities, and potential anomalies in the models' predictions, providing insights into their robustness and adaptability to the extended dataset.

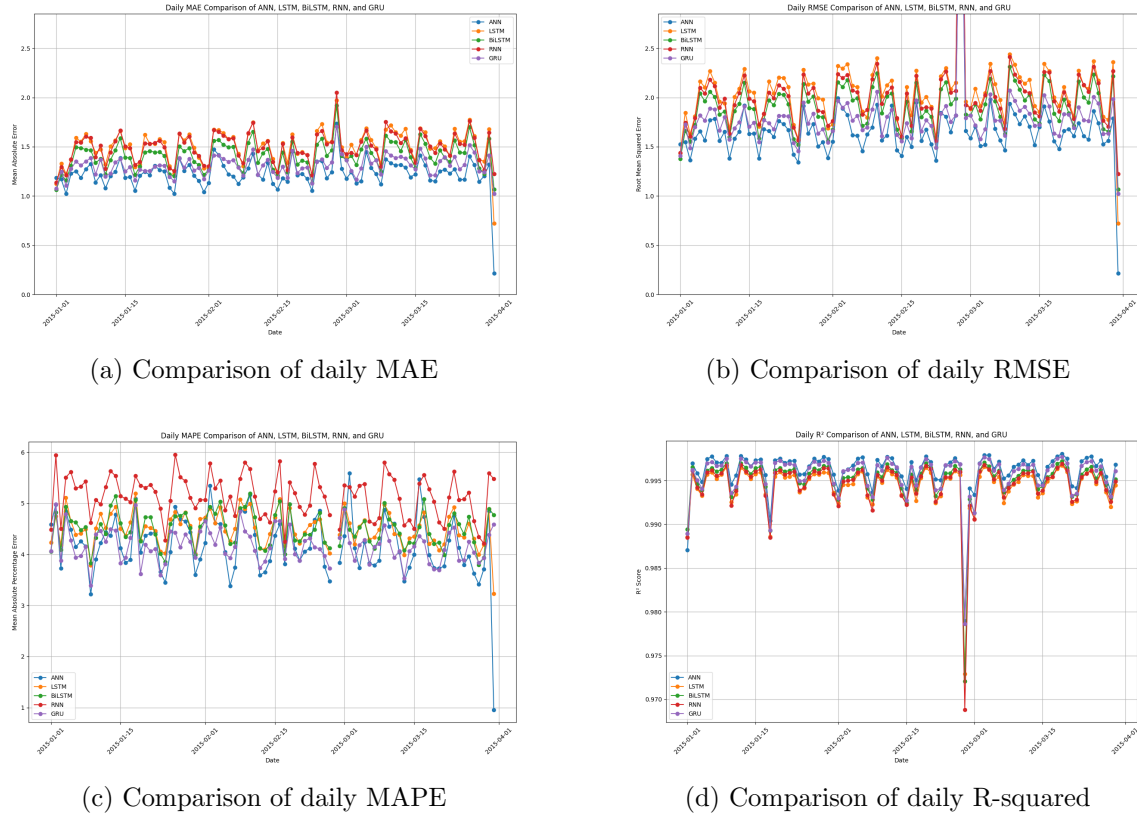


Figure 5.14: Daily Evaluation Matrices Value Comparison for all models(Phase 2)

5.3.3 First Day, Best Day & Worst Day Prediction

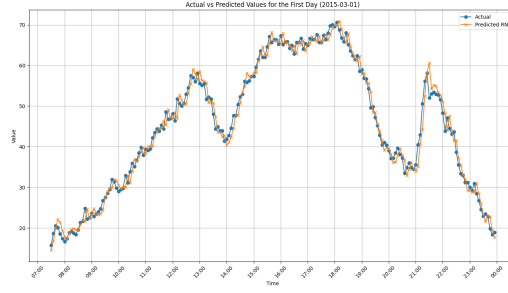
In this analysis, the terms **First Day**, **Best Day**, and **Worst Day** are used to categorize specific days based on the model's prediction performance. The **First Day** refers to the initial day of the testing period, serving as the starting point for evaluating the model's predictive capability. The **Best Day** is defined as the day where the model achieved the highest accuracy, evidenced by the lowest Mean Absolute Error (MAE) among all days in the dataset. Conversely, the **Worst Day** represents the day with the least accurate predictions, indicated by the highest MAE value, highlighting the model's greatest deviation from the actual traffic volume. These categorizations help to understand the variability in the model's performance across different days.

Phase 1(4 month data)

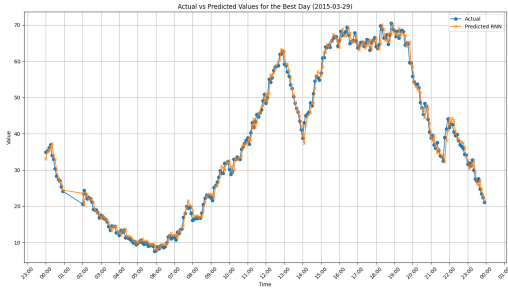
In Phase 1, we focused on evaluating the performance of the Recurrent Neural Network (RNN) model, which demonstrated the best results among the tested models. The visualization includes three specific days: the first day, the best day, and the worst day of prediction, based on the daily Mean Absolute Error (MAE) values.

- **First Day Prediction (Figure 5.15a):** The graph illustrates the comparison between the actual and predicted traffic volumes for the first day of the dataset. The model shows a strong correlation between actual and predicted values, indicating its ability to capture the initial patterns in the data.
- **Best Day Prediction (Figure 5.15b):** On March 29, 2015, the RNN model achieved its best performance with the lowest daily MAE of 1.2231. The graph highlights the close alignment between the actual and predicted values, showcasing the model's high accuracy on this particular day.
- **Worst Day Prediction (Figure 5.15c):** The worst performance occurred on March 26, 2015, with a daily MAE of 1.6946. Despite the higher error, the graph still demonstrates a reasonable level of accuracy, although the deviations between actual and predicted values are more pronounced.

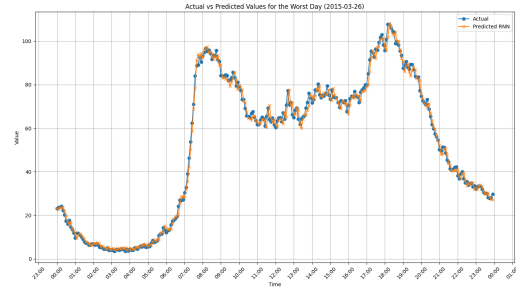
These visualizations emphasize the RNN model's robustness and consistency in predicting traffic volume, with notable variations in accuracy on different days.



(a) First day Prediction



(b) Best Day Prediction



(c) Worst Day Prediction

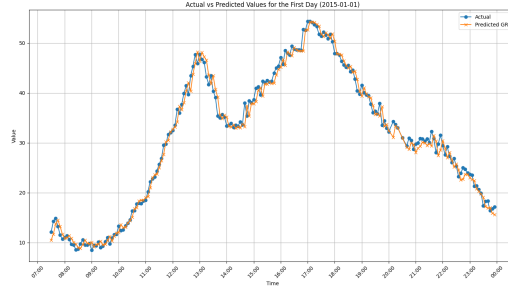
Figure 5.15: First day, Best day & Worst Day Prediction by RNN(Phase 1)

Phase 2(15 month data)

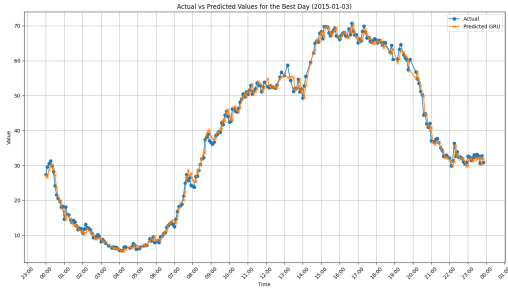
In Phase 2, we extended the analysis period to 15 months and evaluated the performance of the Gated Recurrent Unit (GRU) model, which showed the best performance. Similar to Phase 1, we visualized the first day, the best day, and the worst day of prediction based on the daily MAE values.

- **First Day Prediction (Figure 5.16a):** The graph for the first day of the dataset shows the GRU model's predictions closely matching the actual traffic volumes, reflecting the model's effectiveness from the start.
- **Best Day Prediction (Figure 5.16b):** The GRU model's best performance was recorded on January 3, 2015, with the lowest daily MAE of 1.1046. The graph demonstrates an almost perfect alignment between actual and predicted values, highlighting the model's excellent predictive capability on this day.
- **Best Day Prediction (Figure 5.16c):** The worst day for the GRU model was February 27, 2015, with a daily MAE of 1.7064. Although the error is higher, the model still captures the general trend of traffic volume, with some deviations observed in the predictions.

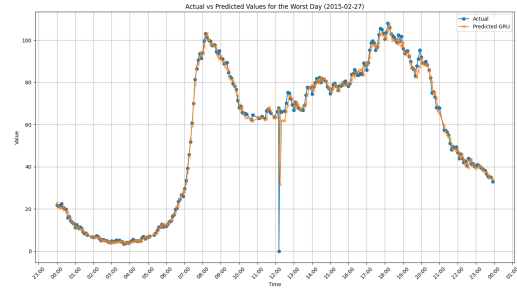
These visualizations emphasize the RNN model's robustness and consistency in predicting traffic volume, with notable variations in accuracy on different days.



(a) First Day Prediction



(b) Best Day Prediction



(c) Worst Day Prediction

Figure 5.16: First day, Best day & Worst Day Prediction by GRU(Phase 2)

Chapter 6

Conclusion

6.1 Summary

In this study, we applied five deep learning models—Artificial Neural Network (ANN), Long Short-Term Memory (LSTM), Bidirectional Long Short-Term Memory (BiLSTM), Recurrent Neural Network (RNN), and Gated Recurrent Unit (GRU)—to forecast traffic data. Our goal was to utilize the deep learning models’ capability to capture complex non-linear and temporal dependencies. We gathered extensive historical traffic volume data from various sensors and monitoring systems in Porto City, Portugal. This comprehensive dataset allowed us to train and evaluate our models effectively.

During Phase 1, which involved four months of data, the RNN model demonstrated superior performance compared to the other models, showcasing its ability to effectively capture short-term temporal patterns. The RNN outperformed other models in terms of Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and R-squared (R^2) metrics, indicating its robustness in forecasting traffic volume for shorter durations.

In Phase 2, which extended the dataset to 15 months, the GRU model excelled in handling the extended data sequence, providing the most accurate predictions among the five models. The GRU’s gating mechanism enabled it to maintain performance over longer sequences, making it well-suited for the more extended period considered in Phase 2.

Overall, our deep learning models consistently outperformed traditional forecasting methods, highlighting the potential of these advanced techniques in enhancing traffic management by providing more accurate and reliable traffic volume predictions.

6.2 Limitation

Despite the encouraging outcomes, our study is not without limitations that warrant further exploration. One significant limitation is the model’s ability to handle rare and unpredictable events, such as the COVID-19 pandemic, public holidays, or accidents, which can drastically alter traffic patterns. Our models, trained on historical data spanning several years, may not fully capture the dynamics introduced by these atypical events. This

gap in our research suggests a need for developing models that are robust to such anomalies and capable of adapting to sudden changes in traffic behaviour. Additionally, while our initial model configurations provided a good fit for the training data, the impact of hyperparameter tuning remains underexplored. The lack of a comprehensive hyperparameter optimization strategy means that the models' full potential in terms of performance and generalization has yet to be realized. Furthermore, our approach was limited by the static nature of the training dataset, which may not reflect recent traffic conditions. As such, there is a critical need to evaluate our models on more current data to validate their robustness and accuracy in real-world scenarios.

6.3 Future Work

Future research will address these limitations through several strategic initiatives. First, we plan to enhance the model's robustness by incorporating strategies to handle rare and unforeseen events. This could involve integrating anomaly detection algorithms or developing specialized training protocols that account for unusual traffic conditions. By doing so, we aim to improve the model's adaptability and resilience in the face of unexpected disruptions.

Moreover, we will systematically explore the impact of hyperparameter tuning on model performance. This will involve conducting extensive experiments using techniques such as grid search, random search, and Bayesian optimization to identify the optimal hyperparameters that maximize the models' predictive accuracy. Such fine-tuning is expected to enhance the models' ability to generalize to new, unseen data, thereby improving their reliability and performance in diverse traffic scenarios. We also intend to update our dataset with more recent traffic data to ensure that the models remain relevant and effective. This will involve collecting and integrating new data streams to reflect current traffic patterns, enabling us to evaluate the models' performance in real-time conditions. Additionally, we will investigate the application of our models for filling missing data points collected from sensors, enhancing the completeness and reliability of traffic datasets.

Furthermore, our research aims to expand the applicability of our predictive models globally. We will explore the feasibility of replicating our approach in different cities worldwide, provided that appropriate sensors are used to gather traffic data. This expansion will be crucial for demonstrating the scalability and versatility of our models in various urban environments. Ultimately, our data-driven approach to traffic volume forecasting holds the promise of transforming traffic management practices worldwide. By offering scalable, accurate, and efficient solutions, our models have the potential to significantly enhance traffic flow, reduce congestion, and improve the overall quality of urban life. We are confident that our continued research and development will lead to innovative solutions that address the evolving challenges of modern traffic management.

References

- [1] Linchao Li, Lingqiao Qin, Xu Qu, Jian Zhang, Yonggang Wang, and Bin Ran. Day-ahead traffic flow forecasting based on a deep belief network optimized by the multi-objective particle swarm algorithm. *Knowledge-Based Systems*, 172:1–14, 2019.
- [2] Lei Lin, John C Handley, Yiming Gu, Lei Zhu, Xuejin Wen, and Adel W Sadek. Quantifying uncertainty in short-term traffic prediction and its application to optimal staffing plan development. *Transportation Research Part C: Emerging Technologies*, 92:323–348, 2018.
- [3] Yisheng Lv, Yanjie Duan, Wenwen Kang, Zhengxi Li, and Fei-Yue Wang. Traffic flow prediction with big data: A deep learning approach. *IEEE Transactions on Intelligent Transportation Systems*, 16(2):865–873, 2014.
- [4] Nicholas G Polson and Vadim O Sokolov. Deep learning for short-term traffic flow prediction. *Transportation Research Part C: Emerging Technologies*, 79:1–17, 2017.
- [5] Xiaolei Ma, Zhuang Dai, Zhengbing He, Jihui Ma, Yong Wang, and Yunpeng Wang. Learning traffic as images: A deep convolutional neural network for large-scale transportation network speed prediction. *Sensors*, 17(4):818, 2017.
- [6] Xiaolei Ma, Zhimin Tao, Yinhai Wang, Haiyang Yu, and Yunpeng Wang. Long short-term memory neural network for traffic speed prediction using remote microwave sensor data. *Transportation Research Part C: Emerging Technologies*, 54:187–197, 2015.
- [7] Ishteaque Alam, Dewan Md Farid, and Rosaldo JF Rossetti. The prediction of traffic flow with regression analysis. In *Emerging Technologies in Data Mining and Information Security: Proceedings of IEMIS 2018, Volume 2*, pages 661–671. Springer, 2019.
- [8] Azzedine Boukerche and Jiahao Wang. Machine learning-based traffic prediction models for intelligent transportation systems. *Computer Networks*, 181:107530, 2020.
- [9] Yuanli Gu, Wenqi Lu, Xinyue Xu, Lingqiao Qin, Zhuangzhuang Shao, and Hanyu Zhang. An improved bayesian combination model for short-term traffic prediction with deep learning. *IEEE Transactions on Intelligent Transportation Systems*, 21(3):1332–1342, 2019.

- [10] Anirudh Ameya Kashyap, Shravan Raviraj, Ananya Devarakonda, Shamanth R Nayak K, Santhosh KV, and Soumya J Bhat. Traffic flow prediction models—a review of deep learning techniques. *Cogent Engineering*, 9(1):2010510, 2022.
- [11] Pedro Lara-Benítez, Manuel Carranza-García, and José C Riquelme. An experimental review on deep learning architectures for time series forecasting. *International journal of neural systems*, 31(03):2130001, 2021.
- [12] Haifeng Zheng, Feng Lin, Xinxin Feng, and Youjia Chen. A hybrid deep learning model with attention-based conv-lstm networks for short-term traffic flow prediction. *IEEE Transactions on Intelligent Transportation Systems*, 22(11):6910–6920, 2020.
- [13] Yuankai Wu, Huachun Tan, Lingqiao Qin, Bin Ran, and Zhuxi Jiang. A hybrid deep learning based traffic flow prediction method and its understanding. *Transportation Research Part C: Emerging Technologies*, 90:166–180, 2018.
- [14] Yang Zhang, Tao Cheng, and Yibin Ren. A graph deep learning method for short-term traffic forecasting on large road networks. *Computer-Aided Civil and Infrastructure Engineering*, 34(10):877–896, 2019.
- [15] Bing Yu, Haoteng Yin, and Zhanxing Zhu. Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. *arXiv preprint arXiv:1709.04875*, 2017.
- [16] AA Agafonov. Short-term traffic data forecasting: a deep learning approach. *Optical Memory and Neural Networks*, 30(1):1–10, 2021.
- [17] Bryan Lim and Stefan Zohren. Time-series forecasting with deep learning: a survey. *Philosophical Transactions of the Royal Society A*, 379(2194):20200209, 2021.
- [18] Charalampos Bratsas, Kleanthis Koupidis, Josep-Maria Salanova, Konstantinos Giannakopoulos, Aristeidis Kaloudis, and Georgia Aifadopoulou. A comparison of machine learning methods for the prediction of traffic speed in urban places. *Sustainability*, 12(1):142, 2019.
- [19] S Elsayed, D Thyssens, A Rashed, HS Jomaa, and L Schmidt-Thieme. Do we really need deep learning models for time series forecasting? arxiv 2021. *arXiv preprint arXiv:2101.02118*.
- [20] Eric L Manibardo, Ibai Laña, and Javier Del Ser. Deep learning for road traffic forecasting: Does it make a difference? *IEEE Transactions on Intelligent Transportation Systems*, 23(7):6164–6188, 2021.
- [21] Zhuoning Yuan, Xun Zhou, and Tianbao Yang. Hetero-convlstm: A deep learning approach to traffic accident prediction on heterogeneous spatio-temporal data. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 984–992, 2018.

-
- [22] Jianhu Zheng and Mingfang Huang. Traffic flow forecast through time series analysis based on deep learning. *IEEE Access*, 8:82562–82570, 2020.
 - [23] Jürgen Grieser, Silke Trömel, and C-D Schönwiese. Statistical time series decomposition into significant components and application to european temperature. *Theoretical and applied climatology*, 71:171–183, 2002.
 - [24] Arzoo Miglani and Neeraj Kumar. Deep learning models for traffic flow prediction in autonomous vehicles: A review, solutions, and challenges. *Vehicular Communications*, 20:100184, 2019.
 - [25] Yan Zheng, Chunjiao Dong, Daiyue Dong, and Shengyou Wang. Traffic volume prediction: A fusion deep learning model considering spatial-temporal correlation. *Sustainability*, 13(19):10595, 2021.
 - [26] Xueyan Yin, Genze Wu, Jinze Wei, Yanming Shen, Heng Qi, and Baocai Yin. Deep learning on traffic prediction: Methods, analysis, and future directions. *IEEE Transactions on Intelligent Transportation Systems*, 23(6):4927–4943, 2021.