MULTI-HEADED SELF-ATTENTION MECHANISM-BASED TRANSFORMER MODEL FOR PREDICTING BUS TRAVEL TIMES ACROSS MULTIPLE BUS ROUTES USING HETEROGENEOUS DATASETS

A Thesis Presented to the Faculty of the Graduate School

at the University of Missouri-Columbia

In Partial Fulfillment

of the Requirements for the Degree

Master of Science in Civil Engineering

By

MD AHNAF ZAHIN

Dr. Yaw Adu-Gyamfi, Thesis Supervisor

MAY 2023

The undersigned, appointed by the Dean of the Graduate School, have examined the thesis entitled:

MULTI-HEADED SELF-ATTENTION MECHANISM-BASED TRANSFORMER MODEL FOR PREDICTING BUS TRAVEL TIMES ACROSS MULTIPLE BUS ROUTES USING HETEROGENEOUS DATASETS

Presented by Md Ahnaf Zahin,

A candidate for the degree of Master of Science in Civil Engineering,

And hereby certify that, in their opinion, it is worthy of acceptance.

Dr. Yaw Adu-Gyamfi

Dr. Praveen Edara

Dr. Timothy Matisziw

DEDICATION

To my beloved parents Md Helal Uddin and Jasmin Akther

ACKNOWLEDGEMENTS

I am writing to express my sincere gratitude to all those who have contributed to the completion of my master’s thesis. First and foremost, I would like to thank my thesis supervisor Dr. Yaw Adu-Gyamfi, for his invaluable guidance and support throughout the research process. This project would not have been possible without his expertise and unwavering dedication.

I would also like to extend my gratitude to Dr. Praveen Edara and Dr. Timothy Matisziw for their valuable time and consideration as thesis committee members. Their insightful comments and suggestions have helped me to refine my ideas and improve the overall quality of this work.

Additionally, I want to thank my friends, Abdullah Al Maruf and Md Saber Hossain, for assisting with data processing and analysis when I occasionally ran into problems. My thesis was successfully completed with their guidance and motivation.

I want to thank my beloved parents Md Helal Uddin and Jasmin Akther, for their continuous support and encouragement, especially during those moments when I felt overwhelmed and unsure. Their love and encouragement have been a constant source of motivation and inspiration.

Finally, I would like to thank all my lab mates and other friends for helping, guiding, and supporting me throughout my master’s journey and making my life easier in Columbia.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS ii

TABLE OF CONTENTS iii

LIST OF FIGURES vi

LIST OF TABLES viii

ABSTRACT ix

CHAPTER 1: INTRODUCTION 1

1.1 BACKGROUND 1

1.2 PROBLEM STATEMENT 3

1.3 SCOPE OF THE STUDY 4

1.4 OBJECTIVES OF THE STUDY 5

1.5 CHAPTER REVIEW 6

CHAPTER 2: LITERATURE REVIEW 7

2.1 OVERVIEW 7

2.2 RELATED WORKS 8

2.2.1 Review of Different Methods for Collecting Transit Data 8

2.2.2 Traditional Approaches for Bus Travel Time Predictions 10

2.2.3 Development of Deep Learning Frameworks in Bus Travel Time Predictions 13

2.2.4 Application of Attention Mechanism in Bus Travel Time Predictions 15

2.3 RESEARCH GAP 15

2.4 CHAPTER REVIEW 17

CHAPTER 3: DATA DESCRIPTION, DATA ANALYSIS & PROPOSED METHODOLOGY 19

3.1 INTRODUCTION 19

3.2 DATA DESCRIPTION 19

3.2.1 Real-Time GTFS Data 20
3.2.2 Scheduled Data (Static Data) 21
3.2.3 Probe Data........ 23
3.3 PROPOSED METHODOLOGY 25
3.3.1 Mapping the Closest Stations........ 26
3.3.2 Assigning Stop IDs 26
3.3.3 Conflation 28
3.3.4 Route Construction and Travel Time Calculation 30
3.3.5 Mean Travel Time Computation........ 32
3.4 EXTRACTING DIFFERENT TRANSIT DATA ATTRIBUTES 33
3.4.1 Dwell Time 34
3.4.2 Headways........ 35
3.4.3 Bus Delays 35
3.4.4 Expected Travel Time........ 35
3.4.5 Volatility Measures........ 36
3.5 CHAPTER REVIEW........ 36
CHAPTER 4: DEEP LEARNING FRAMEWORKS FOR AVERAGE BUS TRAVEL TIME PREDICTION........ 38
4.1 INTRODUCTION 38
4.2 UNIVARIATE TRANSFORMER MODEL 39
4.2.1 Model Architecture 39
4.2.2 How the Transformer Model Works? 40
4.2.3 Input Data Normalization........ 41
4.2.4 Input Data Structuring........ 42
4.2.5 Model Training 43
4.2.6 Single-Step Forecasting 46
4.3 MULTIVARIATE GATED RECURRENT UNIT (GRU) MODEL 47
4.3.1 Model Training and Testing........ 48
4.4 MULTIVARIATE LONG SHORT-TERM MEMORY (LSTM) MODEL 51

4.5 HISTORICAL AVERAGE (HA) MODEL 53

4.6 XGBOOST MODEL 54

4.7 CHAPTER REVIEW 55

CHAPTER 5: COMPARATIVE ANALYSIS BETWEEN VARIOUS MODELS FOR AVERAGE BUS TRAVEL TIME PREDICTIONS: RESULTS AND INSIGHTS 56

5.1 INTRODUCTION 56

5.2 1-HOUR AHEAD TRAVEL TIME PREDICTION RESULTS 56

5.3 COMPARATIVE ANALYSIS BETWEEN DIFFERENT MODELS – ALL HOUR PREDICTIONS 59

5.3.1 One-Hour Ahead Predictions – All Hours 59

5.3.2 Transformer Model Outperforms Other Models: Discussions and Insights 68

5.3.3 XGBoost Model Outperformed GRU and LSTM: Discussions and Insights 69

5.4 MODEL PERFORMANCES FOR DIFFERENT STATIONS ALONG MULTIPLE ROUTES 70

5.5 MODEL PERFORMANCES FOR PEAK AND OFF-PEAK HOURS 72

5.6 COMPUTATION TIME OF DIFFERENT MODELS 73

5.7 IMPACT OF EXTERNAL FACTORS ON TRAVEL TIME PREDICTION 74

5.8 CHAPTER REVIEW 79

CHAPTER 6: CONCLUSIONS AND FUTURE RESEARCH 80

APPENDIX 83

REFERENCES 100

LIST OF FIGURES

Figure 3.1: Flowchart of GTFS Data Collection 20

Figure 3.2: Static Data Collection.. 22

Figure 3.3: Probe Data Collection 23

Figure 3.4: Probe Data Location around St. Louis 25

Figure 3.5: Flowchart of our Proposed Methodology.. 27

Figure 3.6: Mapping the Closest Stops – Schematic Diagram 28

Figure 3.7: Six Routes around St. Louis.. 30

Figure 3.8: Bus Travel Time Calculation 31

Figure 3.9: Mean Bus Travel Time (Seconds) Frequency for Six Routes........................ 33

Figure 3.10 : Dwell Time Calculation 34

Figure 4.1: Framework of the Proposed Modeling Approach 38

Figure 4.2: Transformer Model Architecture.. 39

Figure 4.3: Flow Diagram of Transformer Model 41

Figure 4.4: Input Data Structuring for Transformer Model 42

Figure 4.5: Model Training Process for Transformer Model 43

Figure 4.6: Train Loss and Valid Loss for Transformer Model 46

Figure 4.7: Training Loss and Testing Loss Plot for Transformer Model........................ 46

Figure 4.8: Testing Data Hourly Prediction Process 47

Figure 4.9: GRU Model Architecture 48

Figure 4.10: Model Training Process for GRU Model 49

Figure 4.11: Internal Structure of GRU Neural Network 50

Figure 4.12: GRU Model Convergence.. 51

Figure 4.13: LSTM Model Architecture 52

Figure 4.14: LSTM Model Convergence 53

Figure 4.15: XGBoost Model Algorithm 54

Figure 4.16: XGBoost Model Training 55

Figure 5.1: Sample 1-Hour Ahead Prediction Trends 57

Figure 5.2: One-hour prediction results for different models across multiple routes 62

Figure 5.3: 1-Hour Ahead Prediction MAPE Heatmaps for different Models across Multiple Routes 65

Figure 5.4: Error Bar Plots (Minimum, Maximum, and Median MAPEs) 66

Figure 5.5: Transformer and XGBoost Model Trend 66

Figure 5.6: Model Performances Along Different Routes – Heatmaps 71

Figure 5.7: Model Prediction Computation Times 74

Figure 5.8: Impact of external factors (GRU Model) 75

Figure 5.9: Impact of External Factors on Bus Travel Time Predictions 78

LIST OF TABLES

Table 3.1: Sample of Real-time GTFS Dataset..21

Table 3.2: Sample of Scheduled Data (Static Data) 22

Table 3.3: Sample of Collected Probe Data 24

Table 3.4: Sample of Integrated and Modified Dataset..29

Table 4.1: Model Hyperparameters for Transformer Model 44

Table 4.2: Baseline Configurations for Transformer Model 45

Table 4.3: Multivariate GRU Model Input Parameters...50

Table 4.4: GRU Model Hyperparameters 50

Table 4.5: LSTM Model Hyperparameters..53

Table 5.1: 9th Hour Predictions for the Six Routes 58

Table 5.2: Transformer Model Performance Comparison at the 9th hour 59

Table 5.3: Summarized Model Results 67

Table 5.4: Model Performances for Peak and Off-Peak Hours..72

Table 5.5: Computation Time for Different Models..73

ABSTRACT

Bus transit is a crucial component of transportation networks, especially in urban areas. Bus agencies must enhance the quality of their real-time bus travel information service to serve their passengers better and attract more travelers. Various models have recently been developed for estimating bus travel times to increase the quality of real-time information service. However, most are concentrated on smaller road networks due to their generally subpar performance in densely populated urban regions on a vast network and failure to produce good results with long-range dependencies. This paper develops a deep learning-based architecture using a single-step multi-station forecasting approach to predict average bus travel times for numerous routes, stops, and trips on a large-scale network using heterogeneous bus transit data collected from the GTFS database and the vehicle probe data. Over one week, data was gathered from multiple bus routes in Saint Louis, Missouri. This study developed a multi-headed self-attention mechanism-based Univariate Transformer neural network to predict the mean vehicle travel times for different hours of the day for multiple stations across multiple routes. In addition, we developed Multivariate GRU and LSTM neural network models for our research to compare the prediction accuracy and comprehend the robustness of the Transformer model. To validate the Transformer Model's performance more in comparison to the GRU and LSTM models, we employed the Historical Average Model and XGBoost model as benchmark models. Historical time steps and prediction horizon were set up to 5 and 1, respectively, which means that five hours of historical average travel time data were used to predict average travel time for the following hour. Only the historical average bus travel time was used as the input parameter for the Transformer model. Other features, including spatial and temporal information, volatility measures (e.g., the standard deviation and variance of travel time), dwell time, expected travel time, jam factors, hours of a day, etc., were captured from our dataset. These parameters were employed to develop the Multivariate GRU and LSTM models. The model's performance was evaluated based on a performance metric called Mean Absolute Percentage Error (MAPE). The results showed that the Transformer model outperformed other models for one-hour ahead prediction having minimum and mean MAPE values of 4.32% and 8.29%, respectively. We also investigated that the Transformer model performed the best during different traffic conditions (e.g., peak and off-peak hours). Furthermore, we also displayed the model computation time for the prediction; XGBoost was found to be the quickest, with a prediction time of 6.28 seconds, while the Transformer model had a prediction time of 7.42 seconds. The study's findings demonstrate that the Transformer model showed its applicability for real-time travel time prediction and guaranteed the high quality of the predictions produced by the model in the context of a complicated extensive transportation network in high-density urban areas and capturing long-range dependencies.

CHAPTER 1: INTRODUCTION

1.1 BACKGROUND

The efficient running of transportation systems is essential as daily transportation demand rises. Over the past ten years, the number of vehicle miles driven on US highways has increased by 10.1%, reaching 274.4 billion in January 2022 [1]. In the same period, Missouri has seen a more than two-fold increase in the number of miles driven. Missouri was the fourth-highest state for average annual mileage driven, with travelers covering 18,521 miles on average there [2]. In St. Louis, Missouri's second-largest city, there was a decrease in public transit use by 8%, with buses accounting up 64% of all trips taken on public transportation in 2018 [3]. Federal data gathered by the American Public Transportation Association for the United States as a whole reveal that there were 883 million fewer public-transit rides nationally in the third quarter of 2022 than there were in the same quarter in 2019. Given the continuously rising demand for transportation and the concurrent decline in the use of public transportation, it is crucial to make the service dependable and user-friendly. Apart from that, many problems have been brought on around the globe in recent years due to increased usage of private vehicles, including congested roads, increased greenhouse gas emissions, longer travel times, and a general degradation in the quality of life. Longer travel times have been a problem in many parts of the world because they make passengers wait longer at stops, which increases anxiety, fuel consumption, and pollution, stresses the transportation infrastructure, and decreases accessibility and mobility. To handle these concerns, including making the bus transit service user-friendly to people, accurate prediction of travel times is necessary.

Accurate prediction of bus travel times is essential for the efficient functioning of the Intelligent Public Transportation System (IPTS). IPTS often uses various sensors and technologies, including GPS, traffic cameras, and weather sensors, to collect real-time data from buses and the surrounding area. The IPTS software then processes and analyzes this data to precisely forecast the bus travel times. Therefore, it's important to precisely anticipate bus travel times, which also aids in the effective administration of the IPTS. However, predicting bus travel times accurately is challenging because of the complex interactions between various nonlinear factors such as traffic conditions, incidents, weather conditions, dwell times, passenger load, passenger boarding/alighting time, number of signalized intersections, etc. Conventional modeling techniques are unable to capture these interactions, and more sophisticated machine learning and deep learning algorithms that incorporate real-time data are needed to provide accurate predictions. Furthermore, finding the datasets required to create precise predictive models is typically impossible. Although using big data might produce useful results, their developments should be carefully considered before application [4]. Traditional methods have, therefore, only proven effective in estimating bus travel times for smaller road networks: for single routes, over a brief period, one or few stations at a time. Exploiting non-linear correlations is very rare in current studies [5] and for that reason the current study work looks at to characterize the link between the non-linear elements impacting bus travel times across a broad network having multiple routes. This framework makes use of recent developments in deep machine learning techniques and the current generation of graphical processing units (GPUs). A large, heterogeneous dataset that was obtained from regional transportation and traffic databases serves as the basis for the developed modeling approach. The resulting model forecasts average bus travel times on a network level for many routes, numerous buses, and multiple stop locations at once and at different hours of the day. The underlying model used a univariate multi-headed self-attention-based transformer neural network for single-step multi-station forecasting to estimate average bus travel times at various times of the day. The current study also developed deep learning frameworks using the multivariate GRU and LSTM algorithms. The suggested univariate Transformer model was empirically assessed and contrasted to these developed multivariate GRU and LSTM models along with other well-known models such as the XGBoost model and the Historical Average model. As far as the authors are aware, it was the first time an estimate of bus travel time in Saint Louis City had been made. Hence, an empirical investigation and comparison analysis were carried out to determine the applicability of a deep learning neural network model based on a multi-headed self-attention-based transformer.

1.2 PROBLEM STATEMENT

Bus transportation is an integral part of the public transit network in many countries. The comfort and contentment of passengers depend on buses arriving and departing on schedule. Yet, erratic bus travel times are a frequent problem that undermines the dependability of bus services, inconveniencing and upsetting passengers. To increase the dependability of bus transit, precise models must be created to forecast bus travel times.

Existing studies have attempted to solve this issue by using a variety of modeling techniques, such as various machine learning techniques, including regression models, artificial neural networks, the XGBoost method, and support vector machine techniques. Nowadays, the research includes different deep learning techniques like LSTM, GRU, Convolution Neural Networks, and different hybrid models. However, these models have some limitations in terms of accuracy and robustness due to the complexity of the bus travel time prediction. Generally, bus travel times are influenced by different external factors such as traffic conditions, weather conditions, passenger boarding or alighting, and route characteristics. Moreover, in terms of long-range dependencies, most models failed to develop robust models. That’s why maximum researchers tried to develop the model based on a smaller road network with fewer stations along a route. Furthermore, there is a lack of studies that focused on multiple routes; rather, they focused on single routes. As a result, this work aims to develop an extensive model using a multi-headed self-attention mechanism-based Transformer Neural Network Model for estimating mean bus travel times at different hours of the day by considering heterogeneous traffic conditions and multiple routes with multiple stations and long-range connections.

1.3 SCOPE OF THE STUDY

The scope of the study is discussed below:

- The scope of the study includes identifying and incorporating relevant data sources such as GTFS data, static data, and probe data. The GTFS data needs to be collected in a standardized way through cloud-based APIs by making it easier with other systems and services.
- The study can choose and develop the best deep learning architectures for predicting average bus travel times. It can also investigate how these model frameworks affect the precision of model predictions with long-range dependencies.

- Another scope of the study is to be able to extract different transit features (e.g., dwell time, headways, delays, etc.) that can be extracted from the datasets and how they may impact the prediction models.
- The scope of the study includes how different training techniques can improve the model's performances, such as normalization of data, shuffling of data, attention mechanism, and hyperparameter tuning.

1.4 OBJECTIVES OF THE STUDY

The objectives of the study are summarized below:

- **Prediction Accuracy:** The study's main objective is to develop multiple deep learning frameworks for predicting average bus travel times at different hours of the day that can improve the prediction accuracy compared to traditional machine learning and regression models.
- **Learning from the GTFS Data:** Another objective of our study is to be able to develop deep learning models that can learn from the General Transit Feed Specification (GTFS) data. GTFS data is rarely used for predicting bus travel times. Our study developed these predictive models using GTFS, scheduled transit, and probe data.
- **Model Performance on Different Traffic Conditions:** This study also investigates how the model performed during traffic conditions, such as Peak and Off-Peak hours.
- **Prediction Computation Time:** Reducing the computation time is another objective of the current study. We will develop different models and compare the computation time it takes to predict.

- **Impact of External Factors:** Bus travel time model predictions can be significantly influenced by spatio-temporal features (e.g., station distances, station sequences, hour of day, etc.) and external factors such as dwell time, jam factor, and expected travel time. The current study tries to examine how these features improve the predictability of outcomes.
- **Handling Complex Relationships:** Predicting bus travel times is a challenging issue that considers a variety of factors, such as traffic congestion, weather, passenger boarding/alighting, and route characteristics. Deep learning models are particularly suited for this purpose because they can capture intricate non-linear correlations between these factors. Our study aims to handle those complex relationships and develop different robust frameworks.

1.5 CHAPTER REVIEW

This chapter covers the background and necessity of predicting bus travel times, the problem statement of our research work, and the objectives and scopes of our research work. The remaining sections of the study are presented in the following order. Existing research works in bus travel time prediction of public transit buses are summarized in Chapter 2 in the “Literature Review” part. Chapter 3 describes data sources, data analysis, cleaning and preprocessing, the proposed methodology, and the various model features. The different model architectures, the input parameters we used to develop these models, and the model training and testing process will all be covered in the following chapter. Chapter 5 will discuss the findings of the different model performances, comparisons, and insights of the study. The future research directions and the study's conclusion will be presented in Chapter 6.

CHAPTER 2: LITERATURE REVIEW

2.1 OVERVIEW

Bus transit is an essential part of transportation networks, particularly in urban areas. Any intelligent transportation system must have precise real-time information on bus transit travel times. Bus travel time prediction is one practical method for enhancing service dependability, optimizing travel patterns, and reducing traffic issues. Bus travel time prediction is determining how long it will take a bus to travel between two points along a specific route, considering several variables that can affect travel time, including traffic congestion, weather, and the times when passengers board and alight the bus. The benefits of accurate real-time travel time information include decreased waiting times for passengers, reduced anxiety, improved ease of boarding or transferring buses, enriched types of public transit services, improved public transit's image and desirability, and a rational basis for scheduling. One can plan routes with fewer delays and lower stop waiting times by having accurate arrival and departure information. Accurate bus travel time forecasts can also assist transportation organizations or agencies in enhancing the general effectiveness and quality of their services by enabling them to more efficiently manage their fleets and allocate resources. However, it is very challenging to predict travel times precisely for a large-scale transportation network consisting of multiple routes, trips, and stops, due to the high degree of complexity.

2.2 RELATED WORKS

In recent years, numerous models have been developed for predicting bus arrival/ travel times. Various types of data, including Global Positioning System (GPS) data, Automatic Vehicle Location (AVL) data, real-time traffic data, etc., are used to develop these models. The predictive models are based on historical average models, linear regression models, and nonparametric regression models. Others include the filtering (Kalman) method, artificial neural networks, machine learning, and deep learning models. The following sections will give a brief idea about the works related to average bus travel time predictions.

2.2.1 Review of Different Methods for Collecting Transit Data

In the past, researchers have employed a range of input data from many sources including Global Positioning System (GPS) Data, Automatic Vehicle Location (AVL), manually gathered, surveys, mobile phone footprints, and social media data, etc. The emergence of big data technologies and its applications in traffic and public transportation have created a foundation for the delivery of data-driven solutions to connected issues. The Global Positioning System (GPS) data for buses is frequently used and a common data collection technique to build spatiotemporal models that can capture the intricate interactions between many factors and accurately predict bus travel times. Several studies used GPS based tracking system to collect the vehicle data for predicting bus travel times and arrival times [6]–[14]. Automatic Vehicle Location (AVL) Data is also widely used data for predicting bus travel times [15]. The location and movement of a bus are constantly tracked using AVL data, which is normally gathered using GPS technology. Using real-time AVL, Farooq et al. [16] presented a prediction system for public transport arrival time. GPS and AVL are examples of technology solutions, however, they are limited in their use of historical data and their disregard for space features. Smart card data and smartphone data were other sources of data that were used to develop frameworks for bus travel time predictions [17], [18]. Nowadays, Probe data and weather data are combined with different datasets to understand the weather effect and the effect of traffic incidents on bus travel time predictions. Data gathered from GPS sensors fitted in automobiles, lorries, and buses are referred to as probe data. Traffic patterns, such as speed, journey time, and congestion, can be studied using this data to track the flow of vehicles. The causes of traffic incidents, such as accidents and road closures, can also be determined and examined using probe data. This information can be employed to enhance traffic flow and optimize transportation operations. Weather data refers to data collected from weather monitoring stations, satellites, and other sources. This data includes information such as temperature, humidity, precipitation, wind speed, and visibility. Dhivyabharathi et al. [19] fitted probe vehicle data with the GPS data for predicting bus travel time which was also studied in India. Alam et al. [20] used GPS data and weather data for predicting arrival times of transit buses. In another study, Yu et al. [21] added weather data to the passenger board and alighting data collected from the APC system and AVL data to estimate bus travel times in Pennsylvania, USA.

Real time traffic data such as GTFS data is nowadays used for bus travel time predictions. GTFS real-time feed data are generated automatically through sensors at regular intervals. The scheduled data is typically published by transit agencies as a set of text files that conform to the GTFS standard. Shoman et al. [22] used vehicle probe data and massive heterogeneous bus transit data (GTFS) to construct a deep learning framework to predict bus delays on several routes. Barnes et al. [23] used GTFS real-time traffic data for predicting bus travel times and found significant results. Elliott and Lumley [24] also recently developed a model framework to predict transit vehicle travel times using GTFS-based road network data.

However, these large, heterogeneous datasets are rarely used to predict bus travel times. Most of the past researchers are focused on GPS and AVL datasets to develop bus travel time prediction models. In this study, we will use the real-time GTFS feed data, scheduled transit data (static data), and vehicle probe data to predict travel times across multiple routes.

2.2.2 Traditional Approaches for Bus Travel Time Predictions

Previously researchers predicted travel times using several traditional approaches such as the Linear Regression models, and Support Vector Machines (SVMs). For instance**,** Taparia and Brady [25] employed the Linear Regression model and compared the results to Gradient Boosting (XGBoost) approach to estimate the bus travel times between stops for a single route with numerous trips. He found that the Linear Regression Model showed approximately 1 min higher MaxAE (Maximum Absolute Error), 0.13 mins higher MAE, and 0.30 mins higher RMSE compared to Gradient Boosting method. In another study, Ashwini et al. [26] performed comparative analysis between different Linear Machine Learning models with Non-Linear Machine Learning models to predict the travel times, where he found that Non-Linear Machine Learning models totally outperformed the Linear Regression model. Yu et al. [27] investigated Random Forests Near Neighbors model achieving higher accuracy than Linear Regression model. Linear regression models generally use a mathematical function to predict travel times, which is formed by different independent variables. Explanatory variables must be statistically distinct from one another in linear regression models. Although many of the factors affecting transportation networks are highly connected [28]. For that reason, it is challenging to use regression models to handle complex non-linear interactions. Moreover, the primary limitation of these models is their delayed response to changes in traffic conditions, which makes them unreliable in the event of accidents or traffic jams. Support Vector Machines was another popular algorithm for predicting bus travel times [29]–[31] . Li et al. [30] developed a model using SVM combined with GPS to predict bus arrival times. Peng et al. [32] proposed a principal component analysis-genetic algorithm-support vector machine (PCA-GA-SVM) approach to precisely predict bus arrival time. However, these models are widely used as they are simple regarding the necessary data preparation and the required computational tools. Nowadays, these models are simply used as a baseline for comparison with different algorithms.

Many studies used nonparametric regression models for predicting bus arrival times/ travel times. Nonparametric models are easy to use due to the lesser number of estimated parameters. k-Nearest Neighbors is one of the most popular ones among them. Ashwini et al. [26] used the k-Nearest Neighbors method for predicting travel time based on time of day, day of week, and direction of travel as key inputs. Chang et al. [33] and Jairam et al. [34] used k-Nearest Neighbors method for their respective studies. Kumar et al. [35] employed k-NN classifier for real time bus travel time prediction. However, for larger datasets, k-NN doesn't perform well due to the limited accuracy. This model tends to provide good accuracy for smaller datasets.

Kalman Filtering (KFT) method and Gradient Boosting Decision Tree (GBDT) were followed for predicting bus travel times previously. KFT is the popular method among all of them, which uses a series of recursive estimate methods and the least mean square error as its best estimation criterion. Using the state-space model of signal and noise, it updates the estimation of state variables by using the estimated value of the past time and the observed value of the present time to obtain the estimated value of the present time. It can be used for computer operations and real-time processing. A multi-parameter, time-varying, complicated, large system with a high level of uncertainty, Kalman filters are excellent for conventional metropolitan transit systems**.** Zhang et al. [36] used KFT method for predicting travel times for a single bus based on a single line-detection and found to be performed well with higher accuracy in one-step prediction. Kumar et al. [37] also followed the same procedure to predict travel times under heterogeneous traffic conditions using day of the week as significant input and got better results. Achar et al. [38] was able to learn spatial and temporal correlations with bus arrival time prediction through the application of KFT. Schwinger explored that the combination of KFT with k-medoids can improve the quality of short-term bus travel time prediction. The linearity of Kalman filter models makes them computationally straightforward, but at the same time, it limits their ability to predict intricate nonlinear space-time algorithms. Additionally, the model doesn’t tend to perform well for sequential road segments or time steps. Recently, Cheng et al. [39] and Kawatani et al. [40] proposed a GBDT model for multi-step prediction for travel time forecasting.

Researchers also used statistical based probabilistic methods or clustering algorithms in transportation engineering to predict travel times [6], [41]–[44]. Ide and Kato [45] tested Gaussian Process Regression with realistic traffic data for probabilistic travel time prediction. Bayesian framework is another tool that has been used nowadays for forecasting. Isukapati et al. [46] developed a Bayesian Framework for bus dwell time prediction. In another study, Buchel and Corman [47] employed probabilistic bus delay prediction approach with Bayesian networks in Switzerland.

Artificial neural networks (ANN) have shown promising results in solving transportation challenges. The application of ANN in predicting bus arrival times reported better results compared to other methods [48], [49]. In a study, Kumar et al. [50] examined the effectiveness of the model-based data-driven artificial neural network (ANN) method and Kalman filter (KF) technique for predicting bus travel times. The experimental results demonstrated that the data-driven ANN can perform better than KF, but that the model requires a large amount of data to train its neural networks. ANNs can model complex nonlinear relationships between independent variables defining traffic flow and travel time along road segments. However, due to the computational difficulty, authors tried to use hybrid models to reduce the complexity. For example, Bai et al. [51] developed a combined model consisting of ANN and KFT for multiple bus routes to predict bus travel time and compare it with other models. The dynamic model outperformed all other models in terms of accuracy. Although ANN has shown some proven success in solving complex problems in recent times, it requires a larger dataset for training.

2.2.3 Development of Deep Learning Frameworks in Bus Travel Time Predictions

The transportation research community has recently become more interested in data-driven methodologies due to the growing amount of computer power presently available and the massive amount of data created by ITSs. By analyzing large amounts of urban traffic data, deep learning has an edge over traditional machine learning methods. For instance, Treethidtaphat et al. [52] found that the Deep Neural Network (DNN) model performed

55% better than the OLS regression model. Yuan et al. [53] implemented RNN and DNN for bus dynamic travel time prediction. However, LSTM neural network has become a popular deep learning tool for predicting bus travel times nowadays. Recently, numerous studies applied LSTM model to get better prediction results [54]–[59]. Agafonov and Yumaganov [55] tried to employ a recurrent LSTM neural network tool to predict bus arrival time for heterogeneous traffic conditions and compared it with a multilayer perceptron model. The LSTM model performed better than the multilayer perceptron model. Pang et al. [60] proposed to capture the long-range dependencies for predicting bus travel times with a combination of RNN and LSTM. Panyo et al. [56] compared the LSTM model with the SVM model in terms of accuracy for predicting the arrival times of 3 electric buses on 4 different routes. The results showed that LSTM had much higher accuracy than SVM. In another study, He et al. [61] used LSTM model to learn heterogeneous traffic patterns from the data while predicting bus travel times.

Hybrid models using Convolutional Neural Network are popular technique nowadays to predict travel times. Hou and Edara [62] presented long short-term memory (LSTM) and convolutional neural networks (CNN) to predict travel time in a road network. They found that the two models had approximately 3% higher accuracy than the baseline estimates and when compared to random forests (RFs) and gradient boosting machines (GBMs), the standard deviation of MAPE measures for both models were higher. Petersen et al. [63] proposed a multi-output, multi-time-step method for predicting bus trip times on a single bus line in Copenhagen using the convolutional LSTM. The findings suggested that the Convolutional LSTM model showed 2-7 % better accuracy than the Historical Average, LSTM, and Google Traffic predictions. Xin et al. [64] also produced a

Convolutional LSTM model for multi-step prediction of bus arrival time, and it was found to be best performer among other tested models. Khayyer et al. [65] constructed a hybrid deep neural network framework using LSTM, RNN and MLP for predicting public transit arrival times.

2.2.4 Application of Attention Mechanism in Bus Travel Time Predictions

Attention Mechanism is a powerful tool used in many forecasting models. Recently, it is applied in traditional deep learning models like LSTM and GRU to predict the bus travel times. One potential benefit of adding attention mechanism layer in bus travel time prediction models is to analyze the historical data of traffic conditions and other relevant factors, such as weather, time of day, and dwell time, that may affect the bus travel time. Wu et al. [66] explored a convolutional LSTM model with a self-attention mechanism that accurately predicted the travel time and waiting time at each station, ensuring the robustness of the model to capture long-range dependence in time series data. To capture temporal information and to satisfy the temporal dependence requirements for dynamic bus travel time predictions, Yuan et al. [67] developed an attention mechanism-based Recurrent Neural Network (RNN) model. According to the results, the approach performed better than conventional machine learning models and was 4.82% better than the Deep Neural Network used on the initial feature space.

2.3 RESEARCH GAP

According to the discussions in **Section 2.2**, most studies are concentrated on smaller road networks with single bus lines or routes and few stations for the development of predictive models. They were unable to predict accurately when a large-scale road network with multiple routes and stops was present. Very few studies have been able to solve these complex issues with long-range dependencies. As we have seen in **Section 2.2.3**, Wu et al. [66] tried to overcome this issue by applying an attention mechanism with convolutional LSTM. The model results got improved with this technique, however, there was not much significant improvement compared to LSTM and convolutional LSTM results. Due to this, in the current study, we proposed developing a model for predicting the average bus travel time using a Transformer Neural Network on a vast network in St. Louis, Missouri, with numerous routes and stations, which can demonstrate significantly improved performance compared to the GRU and LSTM models. The reason we chose the Transformer model is that it is unique as it relies solely on a self-attention mechanism, which allows it to capture the relationships between all elements of the input sequence in a single pass. This is different from other sequence-to-sequence models, such as recurrent neural networks (RNNs), which process the input sequence one element at a time. The self-attention mechanism in the Transformer model allows it to selectively attend to different parts of the input sequence, giving it the ability to model long-range dependencies and handle variable-length input sequences.

Moreover, from the above studies, we observed that most researchers used the data collected from the GPS-based tracking system (GPS data) and AVL data for developing models. Only few of them used the real-time GTFS data and so we tried to develop a framework that can learn from the real-time GTFS data incorporating with static transit data and vehicle probe data.

In our study, we also tried to examine the impact of spatio-temporal features and external factors on model performance. A very few studies discussed how spatiotemporal features and external factors affect prediction accuracy. For instance, Kumar et al. [68] and Shaji et al. [69] captured spatio-temporal correlations for dynamic bus travel time prediction. Lee et al. [70] also explored how the external factors like passenger-related variables, weather variables, and link-related variables played a massive role in improving the model's accuracy. Although we didn't include these parameters in the Transformer model, we tried to understand their impact through other developed models.

As a result, in our study, we'll propose a univariate Transformer Model and assess how it performs against other well-known deep learning models, such as multivariate GRU and LSTM based on the MAPE at various times of the day and under various traffic conditions. We will also observe how long it takes a model to compute predictions. The Transformer model performance will also be compared to some other traditional models such as XGBoost, and Historical Average Model. Additionally, we will illustrate the impact of adding the external features from the developed GRU model by showing the Mean Absolute Percentage Errors (MAPE) at different hours of the day. To the best of our knowledge, this is the first time a Transformer Model has been attempted to use to forecast bus travel times. We will ensure the high quality of the predictions made by our proposed Transformer Model and demonstrate its applicability for real-time travel time prediction of public transportation in the case of a complex transportation network.

2.4 CHAPTER REVIEW

Chapter 2 covers the existing works related to bus travel time predictions. We talked about the review of different data collection methods that are typically used for travel time prediction modeling, traditional approaches and algorithms, deep learning frameworks for modeling. We also highlighted the research gap of our current study and the application of attention-mechanism in travel time prediction modeling.

CHAPTER 3: DATA DESCRIPTION, DATA ANALYSIS & PROPOSED METHODOLOGY

3.1 INTRODUCTION

The purpose of this component of the study was to provide specifics regarding the data source, the methods of data collection, the various types of data, the preprocessing of datasets, and our suggested approach. By aggregating and geographically merging the various datasets, mapping the closest stations, and computing journey times at various stations along various routes on a vast network, the suggested methodology will demonstrate how to perform these things. It will also go through how to calculate the average travel time at various times of day and how to extract various features that can be extracted from the dataset.

3.2 DATA DESCRIPTION

There are a variety of data types that can be used in research focused on predicting transit network travel times. Bus GPS data is frequently used to develop spatiotemporal models that capture intricate interactions between many factors and accurately estimate bus journey times. GPS data is obtained from GPS devices fitted on buses. Predictive models are also trained using historical bus data, which allows researchers to spot patterns and trends that may be used to increase the model's forecasting accuracy. Real-time traffic data collected by sensors and cameras (e.g., GTFS Data, Probe Data) is another source of data that can reveal information about the trip and the vehicle, as well as arrival and departure times, levels of congestion, accidents, and other incidents that may have an impact on bus travel times. Weather information is sometimes used in studies to determine how weather variables like temperature, precipitation, wind speed, and snowfall affect bus travel times. Transit schedule data can also be used to develop models that consider how delays and adherence to the schedule affect bus travel times. The prediction architectures for our study was developed using data from three main sources: the real-time bus transit data from the GTFS, the scheduled bus transit data (static data), and probe data. The details of these three data sources are described in the following sections.

3.2.1 Real-Time GTFS Data

The real-time General Transit Feed Specification (GTFS) data was gathered through an API provided by St. Louis Metro requested for every 30 s from December 12, 2022, to December 19, 2022. We formatted the essential feed information after making a request to extract data from the St. Louis Metro API so that each request could deliver the data we needed for our research. Each request was inserted into a SQL Database and the request timer was set for 30 seconds. The complete Data Collection Python code was set up in this manner. The seven days' worth of data was then gathered in Cloud SQL after the code had been run on a Google Cloud Server via Compute Engine.

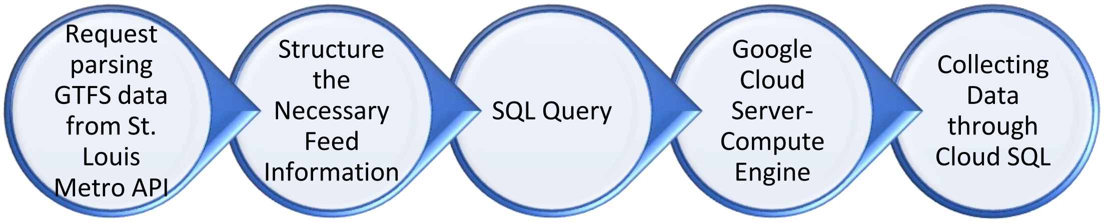

Request parsing GTFS data from St. Louis Metro API
Structure the Necessary Feed Information
SQL Query
Google Cloud Server-Compute Engine
Collecting Data through Cloud SQL

Figure 3.1: Flowchart of GTFS Data Collection

Figure 3.1 displays a general flowchart of the data collection process. Each request returns a range of trip-related details, including timestamps, request times, start dates and times, route ID, trip ID, vehicle ID, direction ID, shape ID, vehicle label, vehicle latitudes and longitudes, and some other information. A total of roughly 3 million records with a size of about 0.8 GB were stored, including data on 58 different routes and 260 distinct vehicles. ***Table 3.1*** shows the sample of our collected real-time GTFS data.

Table 3.1: Sample of Real-time GTFS Dataset

Feed Information	Values
trip_id	3031816
timestamp	2022-12-12 15:02:46
route_id	18125
latitude	38.6092300415091
longitude	-90.15199279785156
vehicle_id	7566
vehicle_label	41 Lee - SOUTH
request_time	2022-12-12 15:03:04
start_datetime	2022-12-12 14:55:00
block_id	69161
direction_id	1
shape_id	110609

3.2.2 Scheduled Data (Static Data)

The static data was collected from https://metrostlouis.org/Transit/google_transit.zip website, which contains schedules and arrival times for all vehicles operating the St. Louis metro region. This static data has very few updates per month and is smaller in size in comparison to the real-time GTFS data. For the same period that bus real-time GTFS data was being gathered, static data was also requested. Text files, including information about stops, stop times, routes, trips, schedules, shapes, calendars, and transit agencies, make up the static data. These text files are merged to get the combined information on different stops.

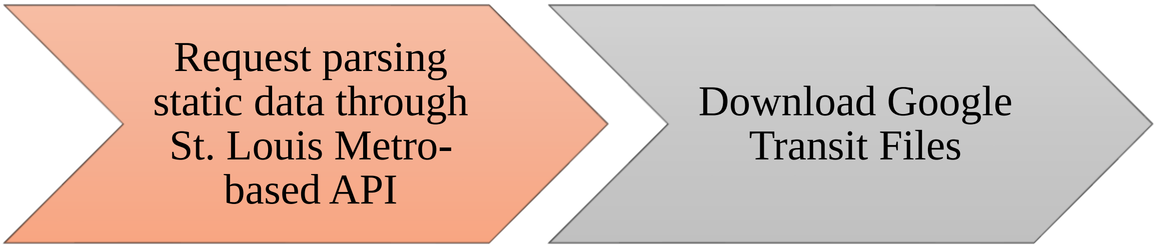

Request parsing static data through St. Louis Metro-based API
Download Google Transit Files

Figure 3.2: Static Data Collection

Figure 3.2 illustrates the static data collection method in our study. A sample of the static dataset is shown in ***Table 3.2***. With the use of this dataset, the closest stop locations for various vehicle time points in real-time data were obtained, and the travel time was then calculated.

Table 3.2: Sample of Scheduled Data (Static Data)

Items	Values
stop_id	127
stop_name	THEKLA @ EMERSON SB
stop_desc	NEAR SIDE THEKLA @ EMERSON SB
stop_lat	38.69486
stop_lon	-90.245546
trip_id	3027550
arrival_time	23:37:00

Items	Values
departure_time	23:37:00
stop_sequence	21
shape_dist_traveled	5295.833213
route_id	18125
service_id	2
direction_id	0
block_id	65345
shape_id	110478
trip_headsign	TO CIVIC CENTER TC

3.2.3 Probe Data

The Probe data was collected from Regional Integrated Transportation System (RITIS) website by queried through Google Cloud platform ***(Figure 3.3).*** The probe data provides information on traffic conditions for road segments such as capped speed, uncapped speed, free flow speed, congestion factor, segment length, start latitude and longitude, end latitude and longitude, etc.

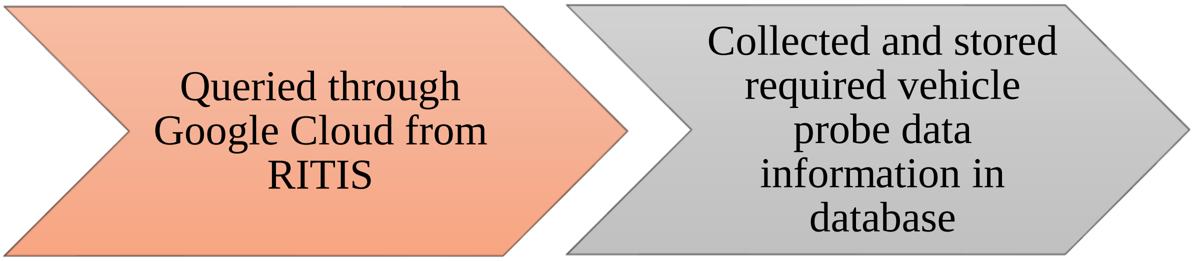

Queried through Google Cloud from RITIS
Collected and stored required vehicle probe data information in database

Figure 3.3: Probe Data Collection

Table 3.3 illustrates the sample of our collected probe data. The probe data was collected for the same time period what we did for the real-time GTFS data. While the probe data offers a lot of information about the road, we will use the capped and uncapped speed, segment length, and jam factor in our model and conflate "tmc" code when mapping bus stops (point) to relevant road segments (line). Our obtained probe data locations around St. Louis are shown in ***Figure 3.4***.

Table 3.3: Sample of Collected Probe Data

Items	Values
tmc	119+00842
link	12866
speed_capped	55.30
speed_uncapped	60.38
free_flow_speed	54.93
jam_factor	2.78940
confidence	0.73
main_road	I-55/I-64
cross_street	Clark Ave
direction	SOUTHBOUND
length	0.70725
start_lat	38.61884
start_long	-90.18619
end_lat	38.61681
end_long	-90.17825
county	ST LOUIS CITY
pub_millis	2022-12-12 17:02:50

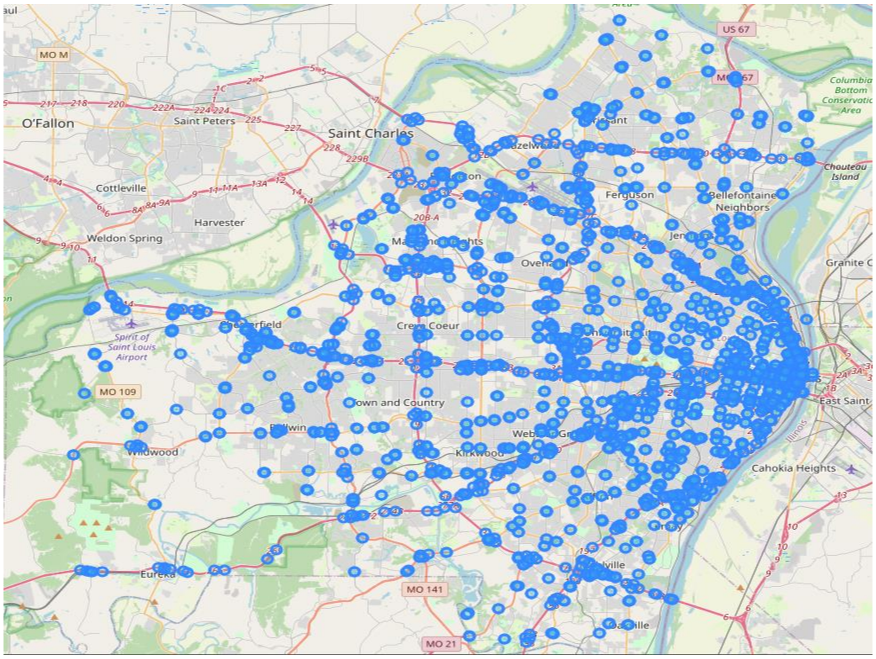


Figure 3.4: Probe Data Location around St. Louis

3.3 Proposed Methodology

A series of steps were undertaken to accomplish our research goals. ***Figure 3.5*** illustrates the critical elements of the research approach used to forecast mean bus travel times for various periods of the day. After collecting data, the first step was to preprocess all the data so that the data quality was improved and the chances for any missing values, dimensionality problems, and other errors were reduced. The next stage entails locating the closest stops or stations for each time point on all routes and constructing and choosing the top six routes with the most stations in St. Louis, Missouri. Next, we conflated the probe data by location into our dataset. In this way, we combined all the datasets, which mapped the datasets into a single data layer. In the third stage, we calculated variables such as bus travel times, station distances, etc. We next determined the mean travel time, our intended target variable, and set up the dataset for our models. From the dataset, we also extracted and computed some additional parameters, including volatility (standard deviation and variance of trip time), and dwell time. The deep learning models used these parameters as input parameters to make the model more robust.

3.3.1 Mapping the Closest Stations

In our research, our target variable is average bus travel times which we will predict using different input parameters. Then, we must map the closest stations to determine travel times. The nearest stops from the static data were mapped to each timepoint of the GTFS real-time feed using the haversine formula. The haversine theorem gives a great circle distance between two points on a sphere from their latitudes and longitudes. The haversine formula is given below:

$$d = 2r\ arcsin\sqrt{}(haversin(\varphi 2 - \varphi 1) + cos\varphi 1 cos\varphi 2 haversin(\lambda 2 - \lambda 1))$$

where d = distance; r = radius of the earth (6,378.1 km); $\varphi 1\ and\ \varphi 2$ indicate the latitudes of vehicles stops; $\lambda 1\ and\ \lambda 2$ indicate the longitudes of vehicles and stops. The haversine theorem maps the closest stations but the stop IDs were not assigned in a corrected way. So, we applied an algorithm that is described in the next section.

3.3.2 Assigning Stop IDs

We discovered that the vehicle points that passed a certain stop were awarded to that stop rather than the stop after it. As an illustration, stop 2 was assigned to the two circled vehicle spots in ***Figure 3.6***. To calculate the trip travel times for each station, we must designate the point that passes a particular stop as the next nearby stop. So, we followed an algorithm where we calculated the distances between two adjacent stops (*distance 1*), distances between a vehicle point and its previous stop (*distance 2*); and distances between a vehicle

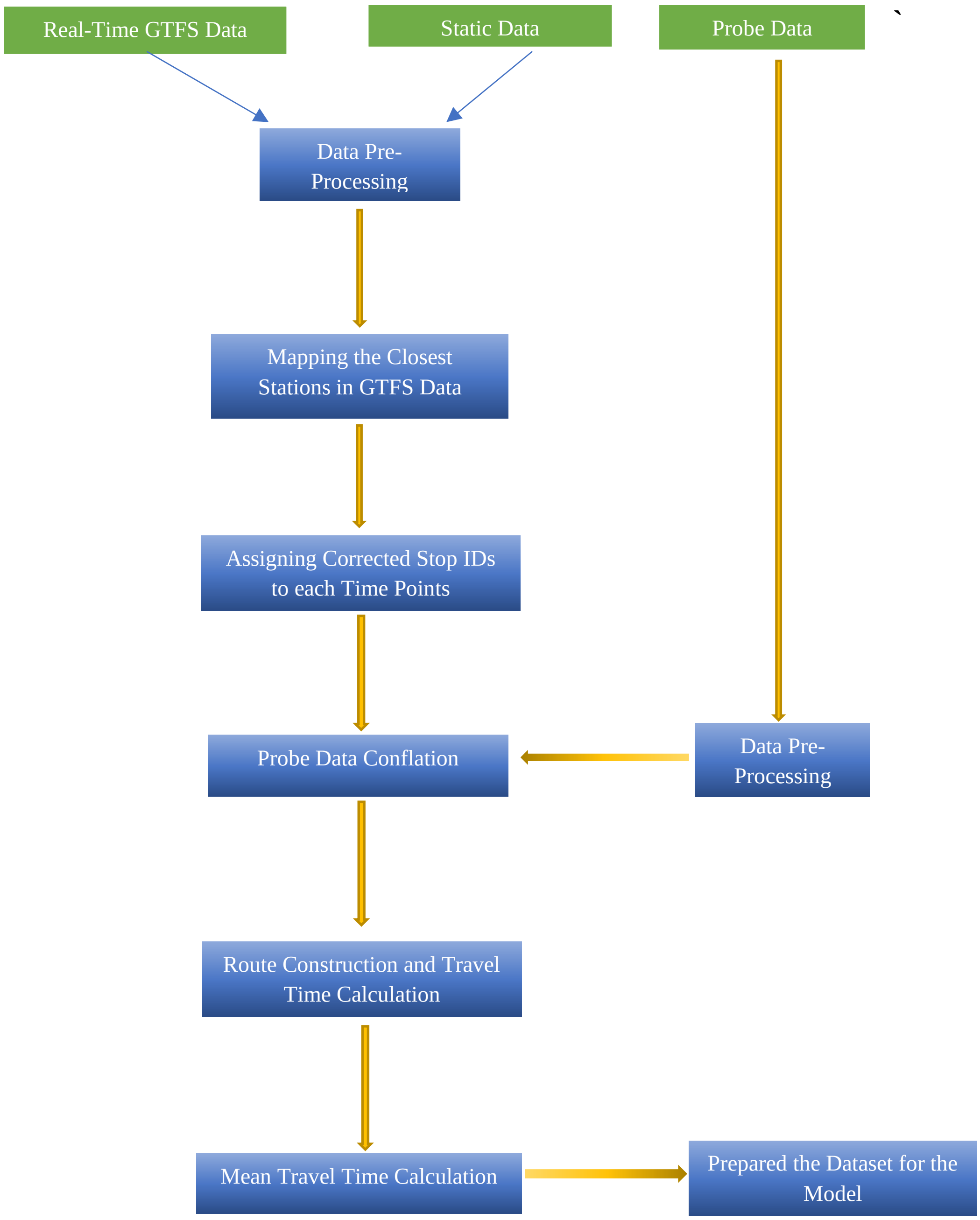

Real-Time GTFS Data
Static Data
Probe Data
`
Data Pre-Processing
Mapping the Closest Stations in GTFS Data
Assigning Corrected Stop IDs to each Time Points
Probe Data Conflation
Data Pre-Processing
Route Construction and Travel Time Calculation
Mean Travel Time Calculation
Prepared the Dataset for the Model

Figure 3.5: Flowchart of our Proposed Methodology

point and its next stop (*distance 3*). Now, we checked whether *distance 1* was greater than *distance 3* or not. If *distance 1* was greater than *distance 3*, we assigned the point to the next station. By following this algorithm, we were able to assign the circled points to the stop -3.

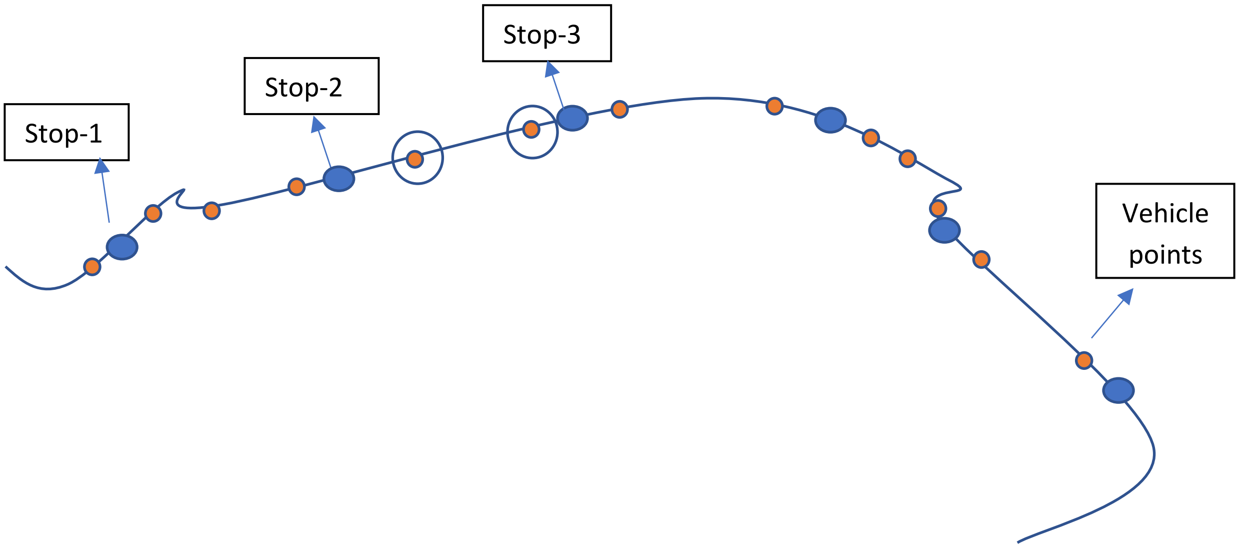


Figure 3.6: Mapping the Closest Stops – Schematic Diagram

Next, we identified the latitudes and longitudes of the beginning and ending points for each vehicle on each route. To identify the origin and destination of buses, we designated the name as "stop ID from -stop ID to" (***Table 3.4***). Then, using the Haversine formula, we determined the station distances for each station along each of the routes after giving each one a unique station sequence number. Station distance and the station sequence number will be used as model input parameters.

3.3.3 Conflation

Point-to-line conflation was used to combine the bus position (point) and probe (line segment) datasets after mapping the closest stops and obtaining the merged dataset between the real-time feed and static dataset. Using the Python "Geopandas" packages, we applied a spatial merging technique for point-to-line conflation. First, we created a "LineString" using the start latitude and longitude and end latitude and longitude information from the probe dataset. Finally, we attempted to match the stations' latitudes and longitudes within a 0.01 km radius of the newly formed line string. In this way, we combined all the datasets

Table 3.4: Sample of Integrated and Modified Dataset

Items	Values
route_id	18146
station_sequence	17
station_dist	0.453405509
latitude	38.6866493
longitude	-90.3587875
stop_id_from	2584
stop_id_to	2585
stop_lat_to	38.686609
stop_lon_to	38.686377
station	2584-2585
tmc	119+00068
link	4149
Speed_capped	35.71
Speed_uncapped	35.71
Free_flow_speed	39.64
Jam_factor	0.79461
main_road	I-170
cross_street	MO-340/Olive Blvd/Exit 3
direction	NORTHBOUND
length	0.70725

and managed to get the traffic speeds, segment length, and congestion information. ***Table 3.4*** displays an example of the final integrated and modified dataset.

3.3.4 Route Construction and Travel Time Calculation

The top 6 routes were selected based on the highest number of stations along a route. We wanted to explore our research around the St. Louis region, so we selected those six routes in the St. Louis area. The six selected route IDs are 18125, 18128, 1839, 18144, 18145, and 18146 (***Figure 3.7***). Then, to determine the bus travel times, we extracted the date from the timestamps and identified the unique trips for each route. By subtracting the previous station's closest timestamp from the next station's closest timestamp after performing "Unique Trip Identification," we can easily determine the travel times for each station. In our investigation, each trip ID and vehicle ID represent one unique trip each day. Each

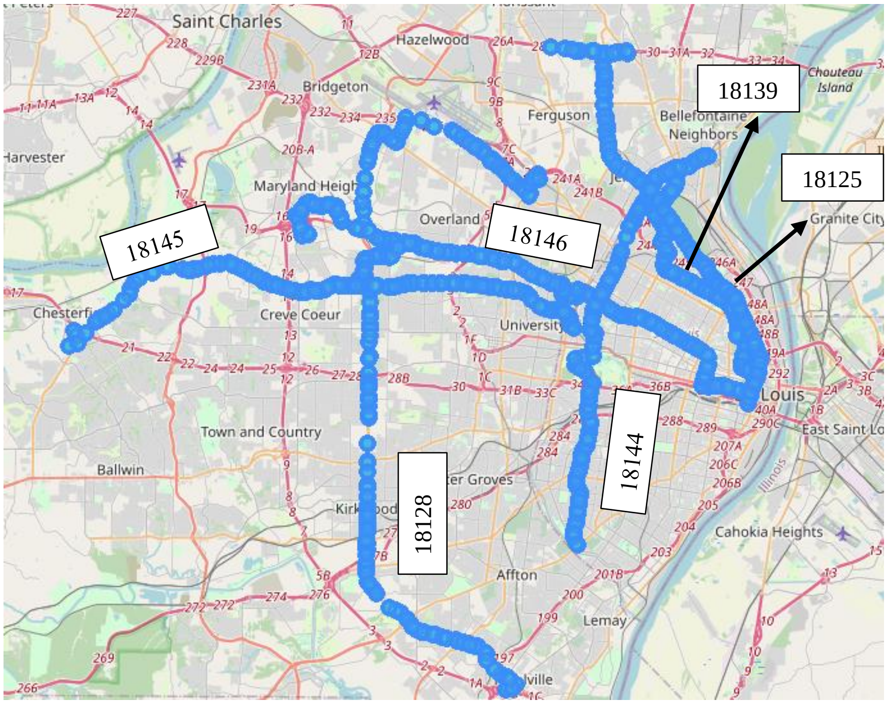


Figure 3.7: Six Routes around St. Louis

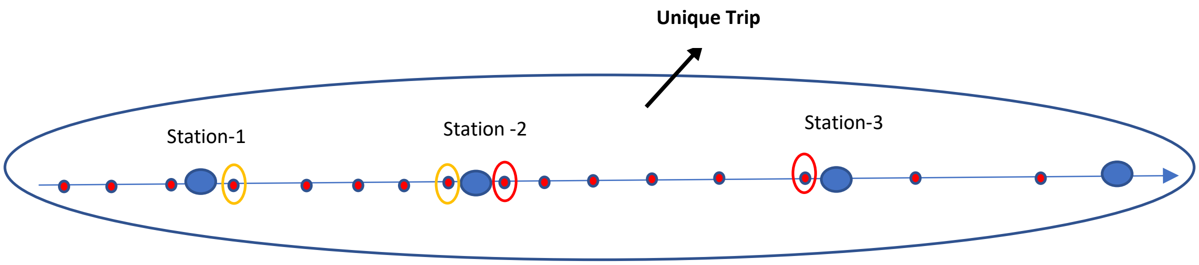


Figure 3.8: Bus Travel Time Calculation

individual trip may contain several stops, and there may be a number of time points between any two adjacent stations. The travel time for each station will then be calculated using the two points that are the furthest apart. The schematic diagram for calculating the travel time between each station is shown in ***Figure 3.8***. For example, we deducted the two yellow circled time points to calculate the travel time between station-1 and station -2, and for station - 2 to station -3, we used the red circled time points. Using this approach, we calculated the travel times for each station for each route.

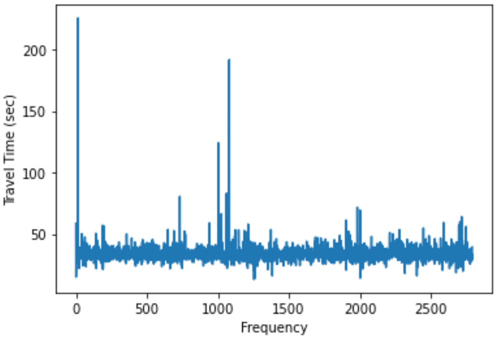
Upon closer inspection of the real-time feed being gathered, it became apparent that some trips had location problems, causing the record to include multiple locations far from the scheduled stops. These location errors can be caused due to one of the following reasons:

- The real-time feed didn’t return feed at that time.
- Bus rapidly departed from the stop.
- The driver disregarded the stop signal.

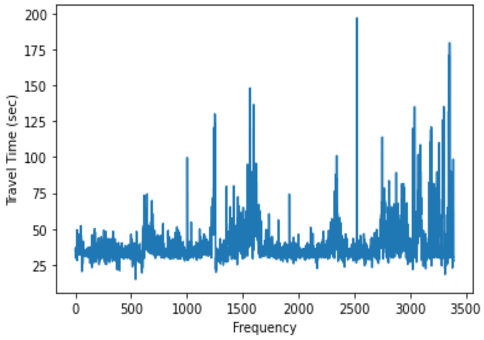
To resolve this problem, we interpolated the journey time based on the distance between the two time points and the station distance. For instance, in ***Figure 3.8***, the two time points that follow station 3 are far from the stations that are next to them. In this instance, we subtracted the time intervals and determined the distance between the two spots. We calculated the travel time between stations 3 and 4 using the distance and the estimated travel time.

3.3.5 Mean Travel Time Computation

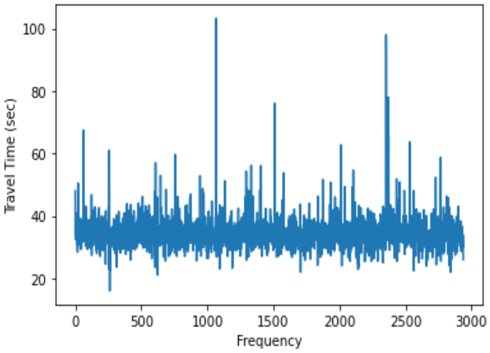
We extracted the hour and the days of the week from the timestamps. Next, we determined the mean bus travel time by aggregating the travel times for different hours of the day for each route and station and taking the mean of those. The mean travel time is our target variable, which we will be predicting using different input parameters.



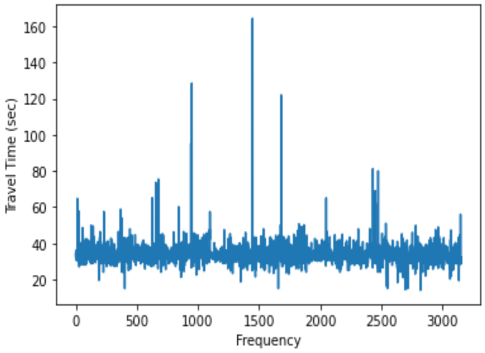
Route 18125



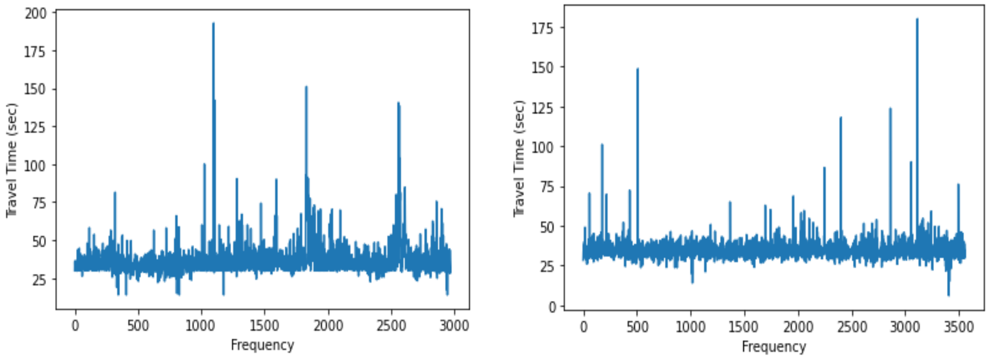
Route 18128



Route 18139



Route 18144



Route 18145

Route 18146

Figure 3.9: Mean Bus Travel Time (Seconds) Frequency for Six Routes

Figure 3.9 depicts the mean bus travel time-frequency plots in seconds for the six routes we selected. Mostly the mean travel time ranges between 20 to 75 seconds. There are few occasions where the travel time is much higher, which may be caused due to various factors such as congestion, dwell time or waiting time for passengers, weather conditions, and other traffic conditions.

3.4 EXTRACTING DIFFERENT TRANSIT DATA ATTRIBUTES

Several features from our dataset will be used as inputs to our model, including historical travel times, hours, station distances, traffic speed, jam factors, etc. In addition to these parameters, other transit attributes can be employed, such as dwell time, delays, headways, etc. to make the model learn quickly. The volatility measures such as standard deviation may also be helpful in predicting bus travel times. The details of how these features can be extracted from our dataset are presented in the following sections.

3.4.1 Dwell Time

Dwell time, sometimes known as terminal dwell time, is the amount of time a moving vehicle, such as a bus or train for public transportation, stays at a designated stop before going on to allow passengers to board or alight. Dwell time can have a significant impact on bus travel time prediction models because it affects the overall travel time of the bus. A bus's total travel time may be delayed in general if passengers waiting at bus stops for extended periods of time. Many factors, such as the number of people boarding or alighting, how they pay, and the bus stop's accessibility features, can contribute to this delay. Hence, it's critical to calculate dwell time precisely for estimating bus travel times. For computing dwell time, we first see if the latitude and longitudes of a vehicle in a station change or not. If the locations of the bus are not changing, but the timestamps are gradually changing, we can conclude it is waiting at that station. Then, we deduct the last timestamp from the first timestamp to calculate the dwell time **(*Figure 3.10*).** This approach is made for all the selected routes. For instance, a vehicle at station 1 has multiple time points, indicating that it is temporarily staying there. We can determine the dwell duration by deducting the final time point to the initial time point during its stay in that station.

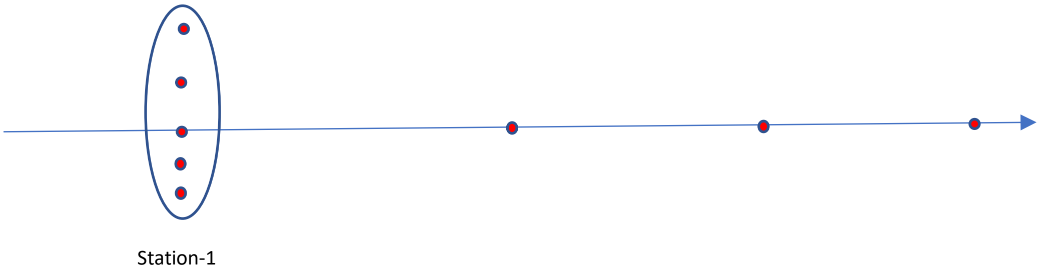


Figure 3.10 : Dwell Time Calculation

3.4.2 Headways

Bus headways are the intervals of time between the arrival of two buses in a row at a certain bus stop or station. Essentially, it refers to the interval between buses on a specific route. Headways are a crucial indicator of how frequently and reliably a bus service runs, and they can greatly impact passenger satisfaction and ridership. Buses arrive more frequently with a shorter headway, decreasing wait times and making transit more convenient and appealing to riders. Headways can be calculated by sorting the values based on timestamps and taking a difference between the arrival times of buses in each hour.

3.4.3 Bus Delays

The term "bus delay" describes the amount of time a bus takes to reach its destination after its scheduled or anticipated arrival time. Numerous things, including heavy traffic, blocked roads, collisions, inclement weather, bus technical problems, and other unforeseen occurrences, can cause delays. By sorting the timestamps and subtracting the scheduled departure time from the actual bus departure time, bus delays can be easily calculated from our dataset. The first timestamp after a bus leaves a station can be used to determine the actual bus departure time. The equation for the bus delays:

$$Bus\ Delays = Actual\ Bus\ Departure\ Time - Scheduled\ Bus\ Departure\ Time$$

3.4.4 Expected Travel Time

By the expected travel time, we mean the travel time we expect based on the probe data information. We were able to identify the tmc and link number for each station sequence (origin-destination) from our integrated dataset, which also provided the segment length and vehicle cap speed. We computed the segment travel time using the speed and the segment length. The expected travel time in seconds was then determined based on the station distance, segment length, and segment travel time.

$$segment\ travel\ time = \frac{segment\ length}{capped\ speed}$$

$$expected\ travel\ time = (\frac{segment\ travel\ time}{segment\ length}) \times (station\ distance)$$

3.4.5 Volatility Measures

The volatility measures – standard deviation and variance of bus travel time are also computed to add as our model inputs. Standard deviation provides an idea of how dispersed the value is concerning the mean travel time. A low standard deviation indicates values are clustered around the mean travel time, and a high standard deviation indicates the values are more spread out. Contrarily, variance is a measure of dispersion that considers the range of all the data set's values. It's basically the square of the standard deviation. These volatility measures may help the deep learning models quickly understand the relationship between inputs and target variables. The next chapters will discuss how the volatility measures and other model features will be used in our models and how they will improve the model results.

3.5 CHAPTER REVIEW

In this chapter, we talked about the different datasets such as the real-time GTFS dataset, scheduled (static) dataset, and probe dataset. We covered the data description and the data collection procedure of these datasets. The sample of the collected datasets and the final integrated datasets were illustrated in tables. In this chapter, we outlined the suggested methodology for our research. The procedure of mapping the closest stations in the dataset, the conflation and merging techniques of different datasets as well as the method of calculating the average bus travel times were also covered in this chapter. Finally, we talked about the many transit data attributes that may be retrieved from the dataset, including dwell time, bus headways, bus delays, expected travel times, and volatility measures.

CHAPTER 4: DEEP LEARNING FRAMEWORKS FOR AVERAGE BUS TRAVEL TIME PREDICTION

4.1 INTRODUCTION

This study's part aimed to discuss about numerous deep-learning neural network models using a single-step multi-station forecasting approach. For predicting bus travel times, a variety of models, including the univariate Transformer, multivariate LSTM and GRU neural network models, the XGBoost Model, and Historical Average Model have been developed. In the previous chapter, we discussed about our proposed methodology and how we aggregated the different datasets. In this chapter, we will discuss about the different deep learning model architectures, how we used our integrated dataset in the models, model training process, whereas the Chapter 5 will detail the Model Results and Analysis. The framework of our proposed approach has been shown in the ***Figure 4.1***.

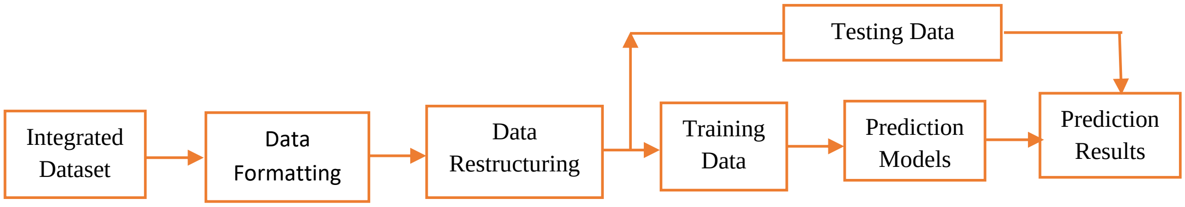


Figure 4.1: Framework of the Proposed Modeling Approach

4.2 UNIVARIATE TRANSFORMER MODEL

4.2.1 Model Architecture

The transformer model is a neural network model which was first introduced by Vaswani et al. [71]. The transformer model is highly capable of capturing long-range dependencies in a sequence, which is one of the main differences between the Transformer model and the conventional recurrent neural network (RNN) model. The Transformer's self-attention mechanism enables the model to concentrate on relevant segments of the input sequence and capture dependencies that are significantly longer than what is generally feasible with RNNs. In particular, the Transformer's multi-head attention mechanism allows the model to focus on several elements of the input sequence at once, which is crucial for accurately capturing long-distance connections. The model may determine associations between distant tokens and incorporate that knowledge into its predictions by paying attention to

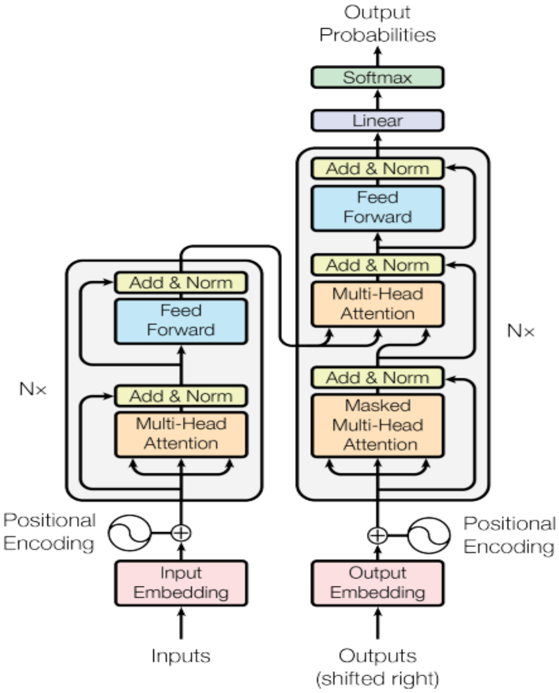


Figure 4.2: Transformer Model Architecture

various aspects of the input sequence. ***Figure 4.2*** shows the model architecture of a Transformer model. The encoder and the decoder are the two principal elements that make up the Transformer model architecture. Both elements are made up of multiple layers of feed-forward and self-attention neural networks.

4.2.2 How the Transformer Model Works?

The brief points on how the transformer model works are given below:

- **Input Encoding:** The input is first tokenized into individual sub words or characters, and then these tokens are mapped to their corresponding embeddings. Positional encodings are added to the embeddings to convey information about the position of each token in the input sequence.
- **Multi-Head Self-Attention:** The encoded input sequence is fed into the multi-head self-attention layer, where each token attends to every other token in the input sequence. This is done by computing a weighted sum of all the other tokens' embeddings, where the weights are determined by the similarity between the current token and the other tokens.
- **Layer Normalization:** After computing the self-attention weights, the output is passed through a layer normalization step to normalize the activations.
- **Feedforward Neural Network:** The normalized activations are then passed through a feedforward neural network with two linear layers and a non-linear activation function in between them.
- **Residual Connections and Layer Normalization:** The output of the feedforward network is added to the input embeddings to create a residual connection. This is followed by another layer normalization step to normalize the activations.

- **Encoder Stacking:** The above steps are repeated for a fixed number of times, typically referred to as the number of encoder layers.
- **Encoder-Decoder Attention (for Decoder Only):** In the decoder, another multi-head attention layer is introduced, which takes as input the encoder outputs and attends to them in addition to the self-attention layer.
- **Output Layer:** Finally, the output of the last layer of the decoder is passed through a linear layer and a SoftMax activation function to produce the final probability distribution over the output vocabulary.

Figure 4.3 illustrates the flow diagram of how the Transformer model works.

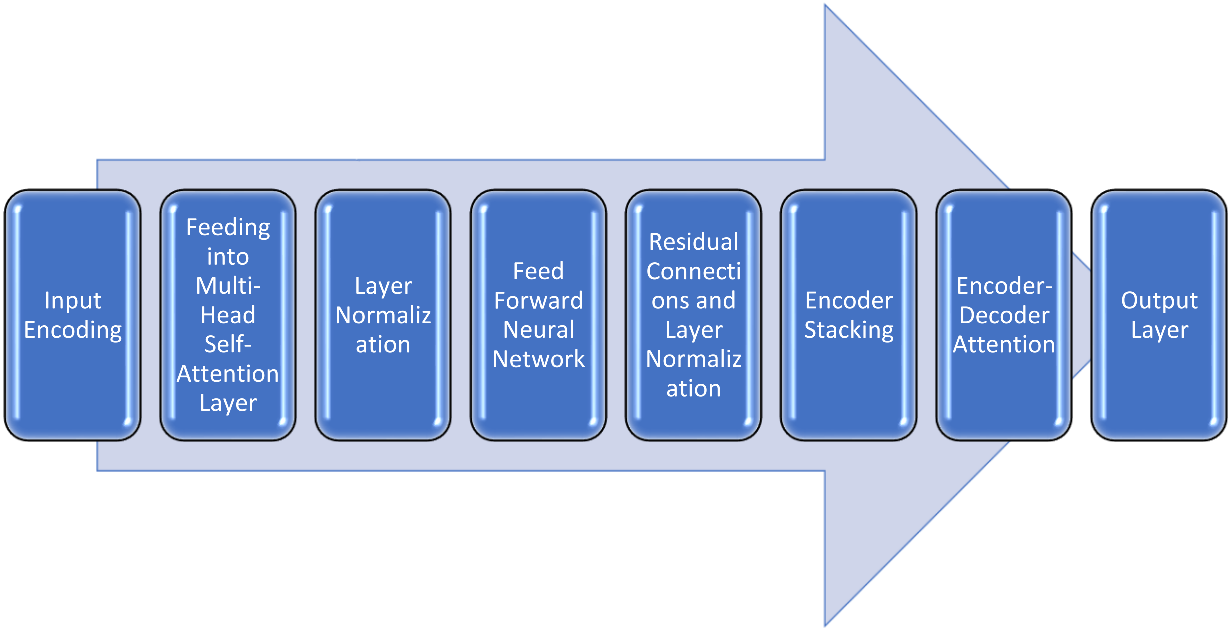

Input Encoding
Feeding into Multi-Head Self-Attention Layer
Layer Normalization
Feed Forward Neural Network
Residual Connections and Layer Normalization
Encoder Stacking
Encoder-Decoder Attention
Output Layer

Figure 4.3: Flow Diagram of Transformer Model

4.2.3 Input Data Normalization

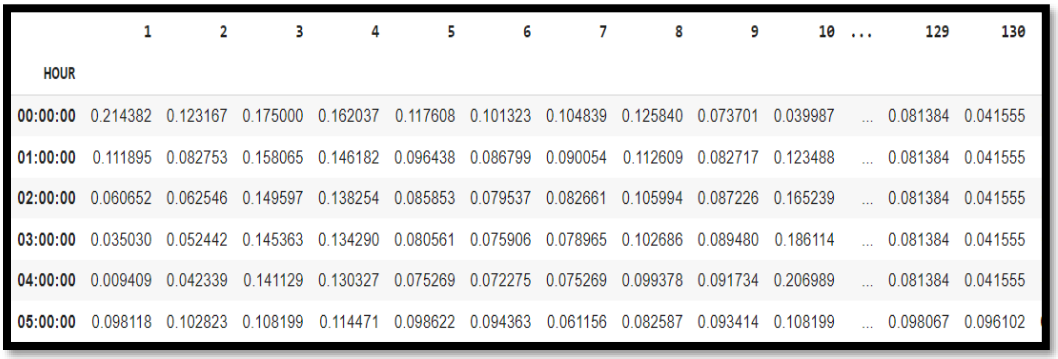
Before reshaping, we normalized all the variables of our dataset. Normalization is an important data processing technique that helps to reduce the scale and variability of input features, making it simpler for the model to learn the underlying patterns in the data. Normalization can also improve the deep learning model performance by reducing the likelihood of vanishing or exploding gradients, which can cause the model to converge too slowly or not at all. It can help prevent overfitting by decreasing the impact of outliers or extreme values in the input data. Our calculated historical average travel time data was used as the input of our Transformer model. To normalize our dataset variables, the following formula is used:

$$x_{norm} = (\, x_i - x_{min})/(\, x_{max} - x_{min})$$

where x_{norm} = The ith normalized value in the dataset, x_i = The ith value in the dataset, x_{max} = The maximum value in the dataset, x_{min} = The minimum value in the dataset

4.2.4 *Input Data Structuring*

We reshaped our dataset to fit it inside the Transformer model. The dataset was reshaped and structured as in ***Figure 4.4***. The data was compiled using a one-hour interval, which is represented by the reshaped data frame's index. The column numbers represent the order of stations along a route and the inside values are the normalized travel time values.



	1	2	3	4	5	6	7	8	9	10	...	129	130
HOUR													
00:00:00	0.214382	0.123167	0.175000	0.162037	0.117608	0.101323	0.104839	0.125840	0.073701	0.039987	...	0.081384	0.041555
01:00:00	0.111895	0.082753	0.158065	0.146182	0.096438	0.086799	0.090054	0.112609	0.082717	0.123488	...	0.081384	0.041555
02:00:00	0.060652	0.062546	0.149597	0.138254	0.085853	0.079537	0.082661	0.105994	0.087226	0.165239	...	0.081384	0.041555
03:00:00	0.035030	0.052442	0.145363	0.134290	0.080561	0.075906	0.078965	0.102686	0.089480	0.186114	...	0.081384	0.041555
04:00:00	0.009409	0.042339	0.141129	0.130327	0.075269	0.072275	0.075269	0.099378	0.091734	0.206989	...	0.081384	0.041555
05:00:00	0.098118	0.102823	0.108199	0.114471	0.098622	0.094363	0.061156	0.082587	0.093414	0.108199	...	0.098067	0.096102

Figure 4.4: Input Data Structuring for Transformer Model

4.2.5 Model Training

After reshaping and restructuring dataset, the model was trained using 80 % of the dataset. The ratio of data used for training, validation, and testing is 0.8:0.9:1.0. Historical timesteps and prediction horizon were set up to 5 and 1, respectively, which means that previous 5-time steps of our data were used to predict the mean bus travel times for the following hour. We used data loader for training, which provides an efficient way to load large datasets into memory in batches, reducing memory usage and improving training speed. We used a batch size of 2, which makes the batch input size of (2,5,1) and batch output size of (2,1). Then the datasets were pushed for model training (see ***Figure 4.5***).

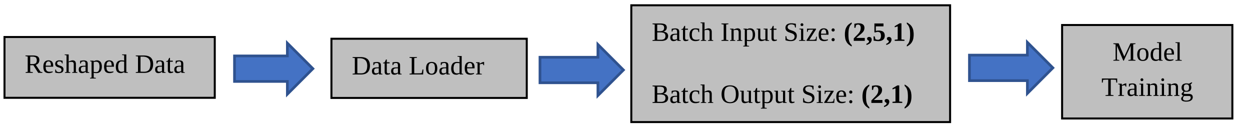


Figure 4.5: Model Training Process for Transformer Model

Table 4.1 summarizes the hyperparameters we used for training the model and ***Table 4.2*** shows the baseline model configuration.

- **NUM_COL** - The number of features or columns in the input data. This will be the number of station sequences along a route.
- **INPUT_LEN** - The number of timesteps in the input sequence. It will be 5 in our case, as we used the previous five timesteps.
- **PRED_LEN** - The number of timesteps in the output or prediction sequence. It will be 1 as our prediction horizon is 1.
- **LEARNING_RATE** - The learning rate used in the optimizer during training. We used 0.0001 as learning rate.

- **NUM_EPOCHS** - The number of epochs or passes through the entire training dataset. We ran 100 epochs for training our dataset.
- **MIN_DELTA** - The minimum change in the monitored metric to qualify as an improvement, used in early stopping.
- **PATIENCE** - The number of epochs with no improvement after which training will be stopped, used in early stopping.

There are some other baseline configurations, which are listed below:

- **BASELINE_MODEL_DIMENSION:** The number of dimensions used for the input and output of each Transformer block.
- **BASELINE_MODEL_NUMBER_OF_HEADS:** The number of attention heads used in each multi-head attention layer in the Transformer blocks.
- **BASELINE_MODEL_NUMBER_OF_LAYERS:** The number of Transformer blocks used in the model.
- **BASELINE_MODEL_DROPOUT_PROB:** The probability of dropping out each input element during training, which is used to prevent overfitting.

Table 4.1: Model Hyperparameters for Transformer Model

Hyperparameters	**Values**
NUM_COL	Varies with different routes
INPUT_LEN	5
PRED_LEN	1
LEARNING_RATE	0.0001

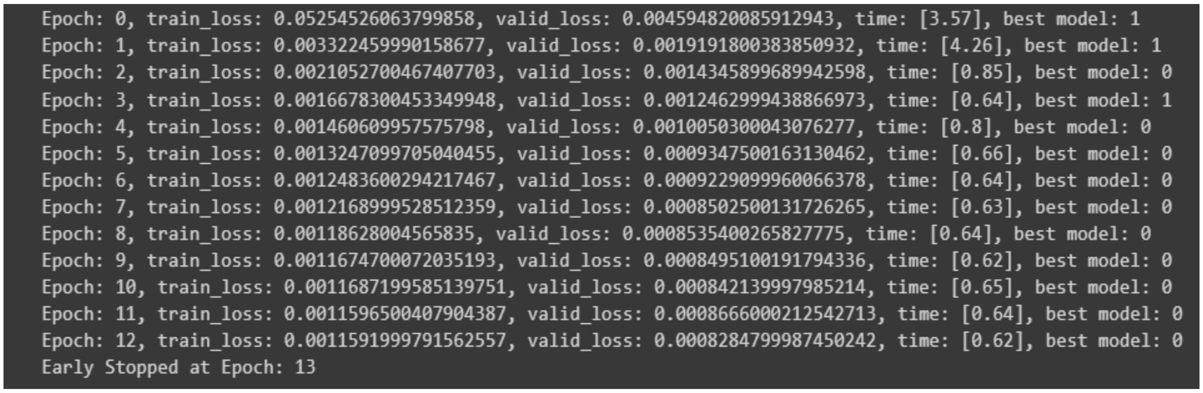
Hyperparameters	Values
NUM_EPOCHS	100
MIN_DELTA	0.0005
PATIENCE	10

Table 4.2: Baseline Configurations for Transformer Model

Baseline Configurations	Values
BASELINE_MODEL_DIMENSION	256
BASELINE_MODEL_NUMBER_OF_HEADS	8
BASELINE_MODEL_NUMBER_OF_LAYERS	6
BASELINE_MODEL_DROPOUT_PROB	0.3

4.2.5.1 Training and Testing Losses

We used an early stopping approach to run the model for 100 iterations. After running for 14 epochs, the model was stopped early, and the best model was preserved based on train loss and valid test loss. The ***Figure 4.6*** shows the train and valid test loss and how it was converged and the process of early stopped. ***Figure 4.7*** plots the training and testing losses, and the plot demonstrates how the training and testing losses converged.



```
Epoch: 0, train_loss: 0.05254526063799858, valid_loss: 0.004594820085912943, time: [3.57], best model: 1
Epoch: 1, train_loss: 0.003322459990158677, valid_loss: 0.0019191800383850932, time: [4.26], best model: 1
Epoch: 2, train_loss: 0.0021052700467407703, valid_loss: 0.0014345899689942598, time: [0.85], best model: 0
Epoch: 3, train_loss: 0.0016678300453349948, valid_loss: 0.0012462999438866973, time: [0.64], best model: 1
Epoch: 4, train_loss: 0.001460609957575798, valid_loss: 0.0010050300043076277, time: [0.8], best model: 0
Epoch: 5, train_loss: 0.0013247099705040455, valid_loss: 0.0009347500163130462, time: [0.66], best model: 0
Epoch: 6, train_loss: 0.0012483600294217467, valid_loss: 0.0009229099960066378, time: [0.64], best model: 0
Epoch: 7, train_loss: 0.0012168999528512359, valid_loss: 0.0008502500131726265, time: [0.63], best model: 0
Epoch: 8, train_loss: 0.00118628004565835, valid_loss: 0.0008535400265827775, time: [0.64], best model: 0
Epoch: 9, train_loss: 0.0011674700072035193, valid_loss: 0.0008495100191794336, time: [0.62], best model: 0
Epoch: 10, train_loss: 0.0011687199585139751, valid_loss: 0.000842139997985214, time: [0.65], best model: 0
Epoch: 11, train_loss: 0.0011596500407904387, valid_loss: 0.0008666000212542713, time: [0.64], best model: 0
Epoch: 12, train_loss: 0.0011591999791562557, valid_loss: 0.0008284799987450242, time: [0.62], best model: 0
Early Stopped at Epoch: 13
```

Figure 4.6: Train Loss and Valid Loss for Transformer Model

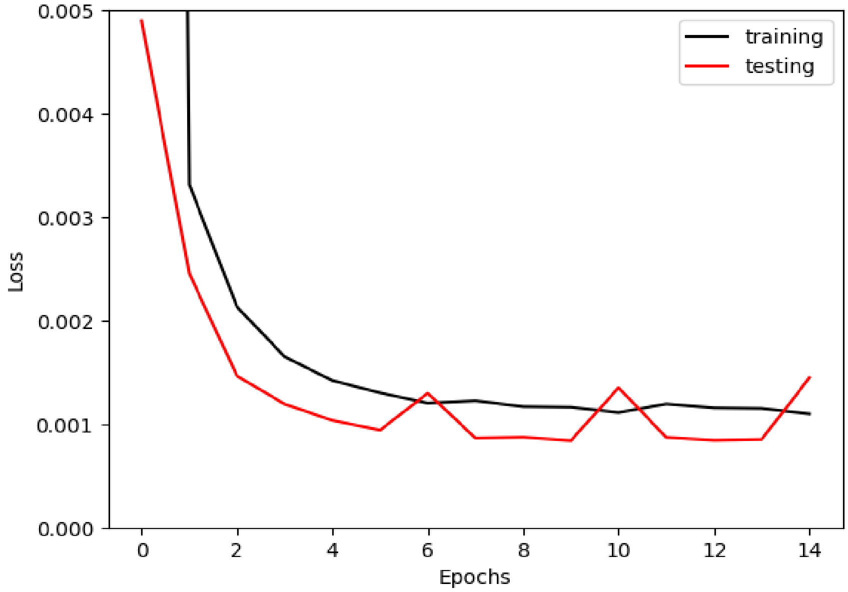


Figure 4.7: Training Loss and Testing Loss Plot for Transformer Model

4.2.6 Single-Step Forecasting

This section evaluates the performance of the trained Transformer model against a test dataset which consisted of the full dataset for six routes. We perform single step forecasting throughout all hours of the day, using the previous 5 hours to predict the future one hour. For example, we predicted the 5th hour predictions using 0-4th hour travel time values and predicted the 6th hour predictions using 1st – 5th hour values and so on **(*Figure 4.8*)**. The main task is to use the Transformer model algorithm in predicting the average bus travel times on a vast transportation network having multiple stations. This procedure of ***Figure 4.8*** is followed for each station along a route. Eventually, we forecast the average bus travel times on every hour of the whole day for each station along a route by performing a single-step multi-station forecasting approach (***Figure 4.8***).

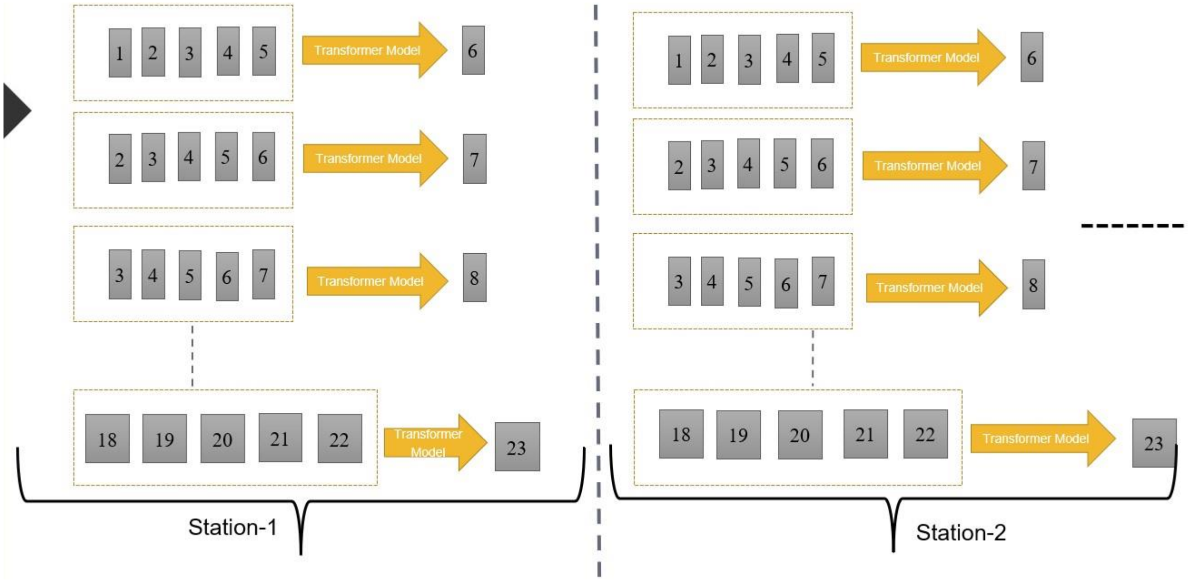


Figure 4.8: Testing Data Hourly Prediction Process

4.3 MULTIVARIATE GATED RECURRENT UNIT (GRU) MODEL

The second model we developed to validate the accuracy and robustness of our proposed Transformer Model is the multivariate GRU model. A Gated Recurrent Unit (GRU) is a type of recurrent neural network (RNN) architecture that is developed to improve the limitations of standard RNNs such as vanishing gradients, and long-term dependencies. A typical RNN is appropriate for sequence-to-sequence problems because the output of each hidden state is fed back into the subsequent iteration. Standard RNNs, however, find it challenging to maintain long-term dependencies due to the vanishing gradient problem, which makes modeling long-term sequences challenging. ***Figure 4.9*** shows the GRU model architecture.

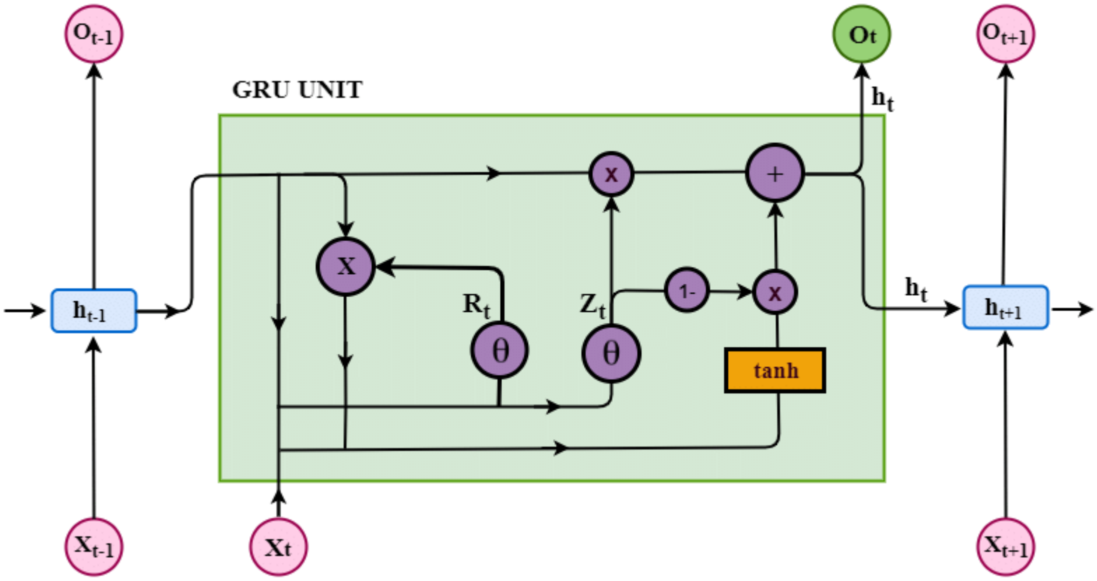


Figure 4.9: GRU Model Architecture

By including gating mechanisms that control the flow of information into and out of the hidden state, GRU addresses this limitation. The reset gate and the update gate are two of the gates found in the GRU cell. The reset gate determines how much of the previous hidden state to forget, while the update gate determines how much of the current input should be included in the new hidden state. This process continues like a relay system, producing the desired output.

4.3.1 Model Training and Testing

As before, we used the previous five hours of historical data inputs to predict the travel time for the next hour, where our horizon is one (H =1). The ratio of the training and testing dataset was 0.8: 0.2. To train our model, we first reshaped our inputs into an N × T × D format, where N denotes the number of rows, T is the number of historical data points (5 in our study) we utilized for predicting, in this case, and D denotes the number of input channels which is 8 and they are historical travel times, hour, station sequences, station distances, standard deviation of travel time, dwell time, expected travel time and jam factor. We experimented with developing the GRU model by adding input features in various combinations. As an illustration, we initially created the model with just the Historical Travel Times, and then we included the Hour and other variables. When all 8 features were used as inputs, the GRU Model output had the lowest Mean Absolute Percentage Error (MAPE). The input details are listed in ***Table 4.3***. The data loader technique was used with a batch size of 512. In a GRU model, the input data is typically represented as a sequence of feature vectors. The final shape size of our input data is (512, 5, 8), and the output data is (512, 1). The flowchart for model training process is shown in the graph (***Figure 4.10***).

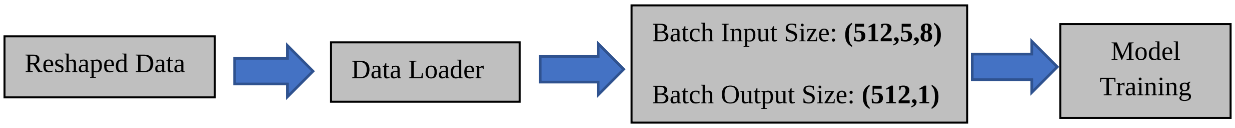


Figure 4.10: Model Training Process for GRU Model

Mean Squared Error (MSE) loss criterion and the torch Adam optimizer optimization method were used for training. We ran 1000 epochs for training the model. ***Table 4.4*** contains a summary of the model hyperparameters that we used to create this GRU model. 20 Hidden Layers and 2 Recurrent Neural Network Layers were used to develop the model. ***Figure 4.11*** shows the internal structure of how the hidden layers and recurrent layers of a GRU model looks like. ***Figure 4.12*** illustrates how the model was converged and the loss plot with epochs.

Table 4.3: Multivariate GRU Model Input Parameters

Input Parameters	Normalization
Historical Travel Times	Normalized
Hour	Normalized
Station Sequences	Not Normalized
Station Distances	Not Normalized
Standard Deviation	Not Normalized
Dwell Time	Normalized
Expected Travel Time	Normalized
Jam Factor	Normalized

Table 4.4: GRU Model Hyperparameters

Hyperparameters	Values
Hidden Layers	20
RNN Layers	2
Learning Rate	0.0001

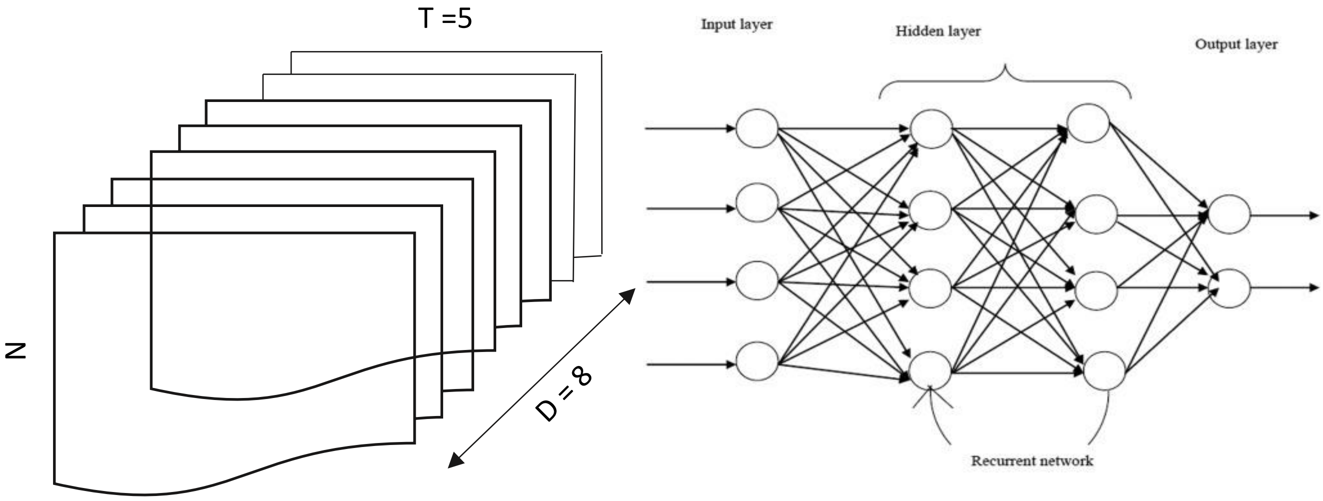
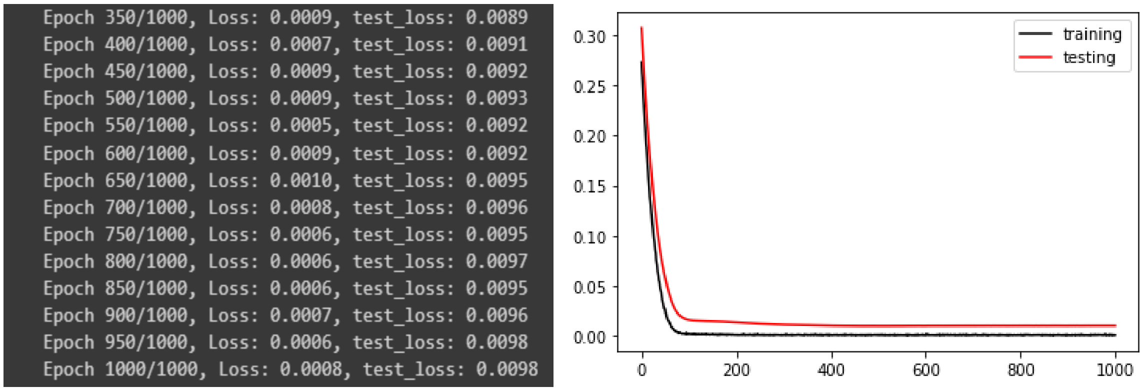


Figure 4.11: Internal Structure of GRU Neural Network

The single step forecasting approach was used for model testing as we did before in the Transformer model. We used the full dataset for testing and tried to find out the mean absolute percentage error for average bus travel times in each hour for each station along every route. We used this multivariate GRU model to predict our average bus travel time values.



```
Epoch 350/1000, Loss: 0.0009, test_loss: 0.0089
Epoch 400/1000, Loss: 0.0007, test_loss: 0.0091
Epoch 450/1000, Loss: 0.0009, test_loss: 0.0092
Epoch 500/1000, Loss: 0.0009, test_loss: 0.0093
Epoch 550/1000, Loss: 0.0005, test_loss: 0.0092
Epoch 600/1000, Loss: 0.0009, test_loss: 0.0092
Epoch 650/1000, Loss: 0.0010, test_loss: 0.0095
Epoch 700/1000, Loss: 0.0008, test_loss: 0.0096
Epoch 750/1000, Loss: 0.0006, test_loss: 0.0095
Epoch 800/1000, Loss: 0.0006, test_loss: 0.0097
Epoch 850/1000, Loss: 0.0006, test_loss: 0.0095
Epoch 900/1000, Loss: 0.0007, test_loss: 0.0096
Epoch 950/1000, Loss: 0.0006, test_loss: 0.0098
Epoch 1000/1000, Loss: 0.0008, test_loss: 0.0098
```

Figure 4.12: GRU Model Convergence

4.4 MULTIVARIATE LONG SHORT-TERM MEMORY (LSTM) MODEL

The third model we developed to validate the accuracy and robustness of our proposed Transformer Model is multivariate LSTM model. Long Short-Term Memory (LSTM) is another type of neural network that has three types of gates that control the flow of information into and out of the cell. They are forget gate, input gate, and output gate. ***Figure 4.13*** shows the LSTM model architecture. The forget gate determines which information to discard from the cell's memory. Input gate decided which information

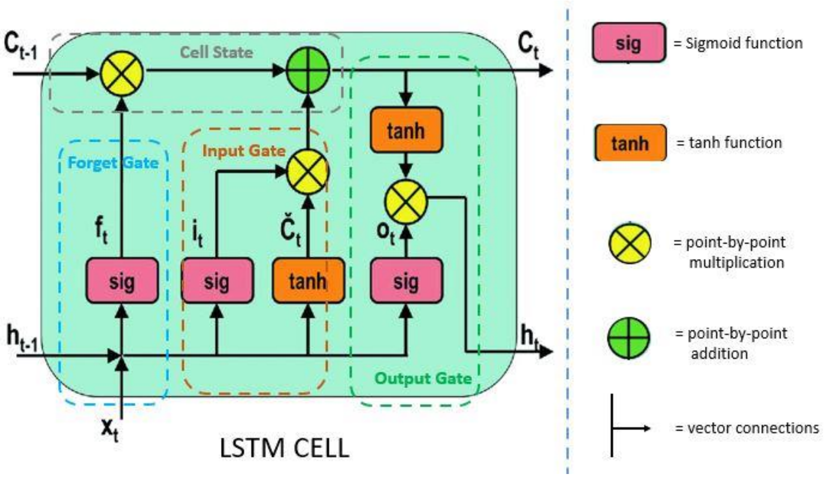
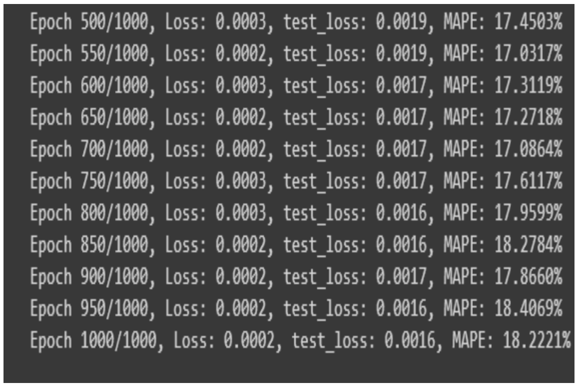


Figure 4.13: LSTM Model Architecture

should be added to the memory cell. It outputs a value between 0 and 1 for each component of the memory cell after receiving as inputs the previous output and the current input. A value of 0 indicates that the information shouldn't be added, whereas a value of 1 indicates that it should. Output gate determines which information to output from the memory cell. It outputs a value between 0 and 1 for each memory cell element based on the inputs of the previous output and the current input. The current output is multiplied by this value before being forwarded to the following time step.

The multivariate LSTM model was trained in the same way how we trained the multivariate GRU model in **section 4.3.1**. The same number of features were used to train the LSTM model. Like GRU, the LSTM model also showed the best output when we used the eight features that were used in the GRU model. ***Figure 4.14*** shows how the model and test loss was converged, and the graph shows the plot of loss vs epochs. The model hyperparameters we used for LSTM Model Architecture was 50 hidden layers, 3 RNN layers and 0.0001 learning. ***Table 4.5*** summarized the model hypermeters.



```
Epoch 500/1000, Loss: 0.0003, test_loss: 0.0019, MAPE: 17.4503%
Epoch 550/1000, Loss: 0.0002, test_loss: 0.0019, MAPE: 17.0317%
Epoch 600/1000, Loss: 0.0003, test_loss: 0.0017, MAPE: 17.3119%
Epoch 650/1000, Loss: 0.0002, test_loss: 0.0017, MAPE: 17.2718%
Epoch 700/1000, Loss: 0.0002, test_loss: 0.0017, MAPE: 17.0864%
Epoch 750/1000, Loss: 0.0003, test_loss: 0.0017, MAPE: 17.6117%
Epoch 800/1000, Loss: 0.0003, test_loss: 0.0016, MAPE: 17.9599%
Epoch 850/1000, Loss: 0.0002, test_loss: 0.0016, MAPE: 18.2784%
Epoch 900/1000, Loss: 0.0002, test_loss: 0.0017, MAPE: 17.8660%
Epoch 950/1000, Loss: 0.0002, test_loss: 0.0016, MAPE: 18.4069%
Epoch 1000/1000, Loss: 0.0002, test_loss: 0.0016, MAPE: 18.2221%
```

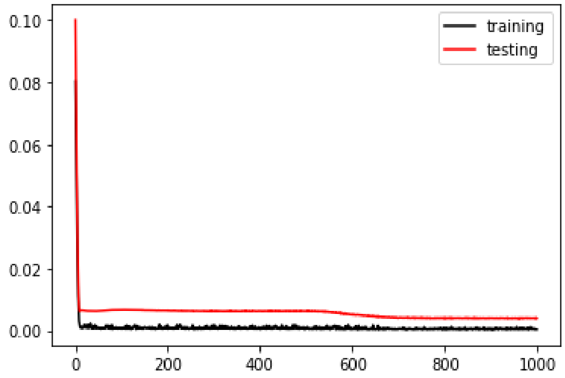


Figure 4.14: LSTM Model Convergence

Table 4.5: LSTM Model Hyperparameters

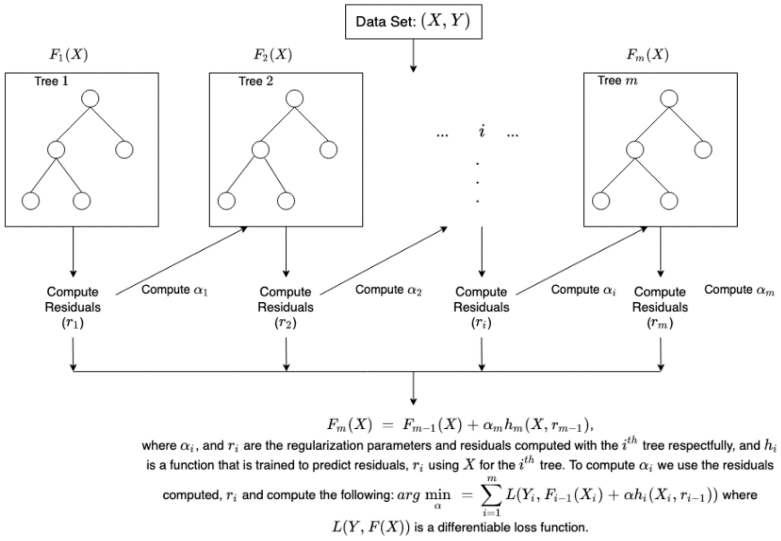
Hyperparameters	Values
Hidden Layers	50
RNN Layers	3
Learning Rate	0.0001

4.5 HISTORICAL AVERAGE (HA) MODEL

The first benchmark model used to validate the accuracy and robustness of our proposed Transformer model is a simple historical average model. Historical Average model is a simple forecasting technique that uses the average of past observations to predict the future values. In our study, we considered the previous 5 hours of mean travel times for each station to predict the mean bus travel time for the next hour of that station. For each station, the historical mean travel time data was filtered based on the selected route id. After filtering the data, the model calculates the mean of previous five travel time values for each hour and used it as the forecast value. Then the predicted values and ground truth values for each hour for each station along each route were evaluated based on MAPE.

4.6 XGBOOST MODEL

The second benchmark model used to validate the accuracy and robustness of our proposed Transformer model is an XGBoost model. XGBoost (Extreme Gradient Boosting) is a scalable, distributed gradient-boosted decision tree (GBDT) machine learning algorithm that is widely used for forecasting modeling. A series of decision trees are trained repeatedly by XGBoost, with each new tree being trained to fix the flaws of the prior one. This process, called boosting, aids in enhancing the model's accuracy over time. Other methods including regularization, learning rate shrinkage, and feature subsampling are also incorporated by XGBoost to increase the model's accuracy. These methods enhance the model's ability for generalization and prevent overfitting. ***Figure 4.15*** gives a brief illustration how the gradient boosting tree works.



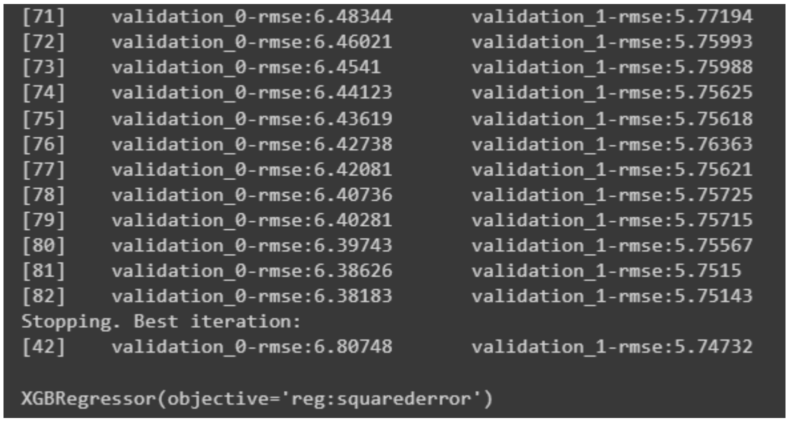
$$F_m(X) \; = \; F_{m-1}(X) + \alpha_m h_m(X, r_{m-1}),$$

where α_i, and r_i are the regularization parameters and residuals computed with the i^{th} tree respectfully, and h_i is a function that is trained to predict residuals, r_i using X for the i^{th} tree. To compute α_i we use the residuals computed, r_i and compute the following: $arg \min_{\alpha} \; = \sum_{i=1}^{m} L(Y_i, F_{i-1}(X_i) + \alpha h_i(X_i, r_{i-1}))$ where $L(Y, F(X))$ is a differentiable loss function.

Figure 4.15: XGBoost Model Algorithm

(**Source**: https://docs.aws.amazon.com/sagemaker/latest/dg/xgboost-HowItWorks.html)

We used “XGBRegressor” class from the XGBoost library in python. Root Mean Squared Error (RMSE) was used as the evaluation metrics for training the model. The early stopping round was set to 40, which means that training will stop if the RMSE on the testing dataset doesn’t improve for 40 consecutive rounds. The model training was converged and stopped when the best iteration on validation RMSE was found. ***Figure 4.16*** shows the snippet of how the model was trained. The same input parameters were used to train the model as we did for the GRU and LSTM model for predicting average bus travel times on different hours.



```
[71]    validation_0-rmse:6.48344       validation_1-rmse:5.77194
[72]    validation_0-rmse:6.46021       validation_1-rmse:5.75993
[73]    validation_0-rmse:6.4541        validation_1-rmse:5.75988
[74]    validation_0-rmse:6.44123       validation_1-rmse:5.75625
[75]    validation_0-rmse:6.43619       validation_1-rmse:5.75618
[76]    validation_0-rmse:6.42738       validation_1-rmse:5.76363
[77]    validation_0-rmse:6.42081       validation_1-rmse:5.75621
[78]    validation_0-rmse:6.40736       validation_1-rmse:5.75725
[79]    validation_0-rmse:6.40281       validation_1-rmse:5.75715
[80]    validation_0-rmse:6.39743       validation_1-rmse:5.75567
[81]    validation_0-rmse:6.38626       validation_1-rmse:5.7515
[82]    validation_0-rmse:6.38183       validation_1-rmse:5.75143
Stopping. Best iteration:
[42]    validation_0-rmse:6.80748       validation_1-rmse:5.74732

XGBRegressor(objective='reg:squarederror')
```

Figure 4.16: XGBoost Model Training

4.7 CHAPTER REVIEW

We talked about many model frameworks in this chapter, including the Transformer Model, GRU, LSTM, Historical Average Model, and XGBoost Model. We discussed the many parameters and hyperparameters we utilized to develop these models as well as how the models operate. We also discussed how we trained these models to forecast the average bus travel times. To give readers a quick overview of how the models operated, the process diagrams, tables, figures, and algorithms were added.

CHAPTER 5: COMPARATIVE ANALYSIS BETWEEN VARIOUS MODELS FOR AVERAGE BUS TRAVEL TIME PREDICTIONS: RESULTS AND INSIGHTS

5.1 INTRODUCTION

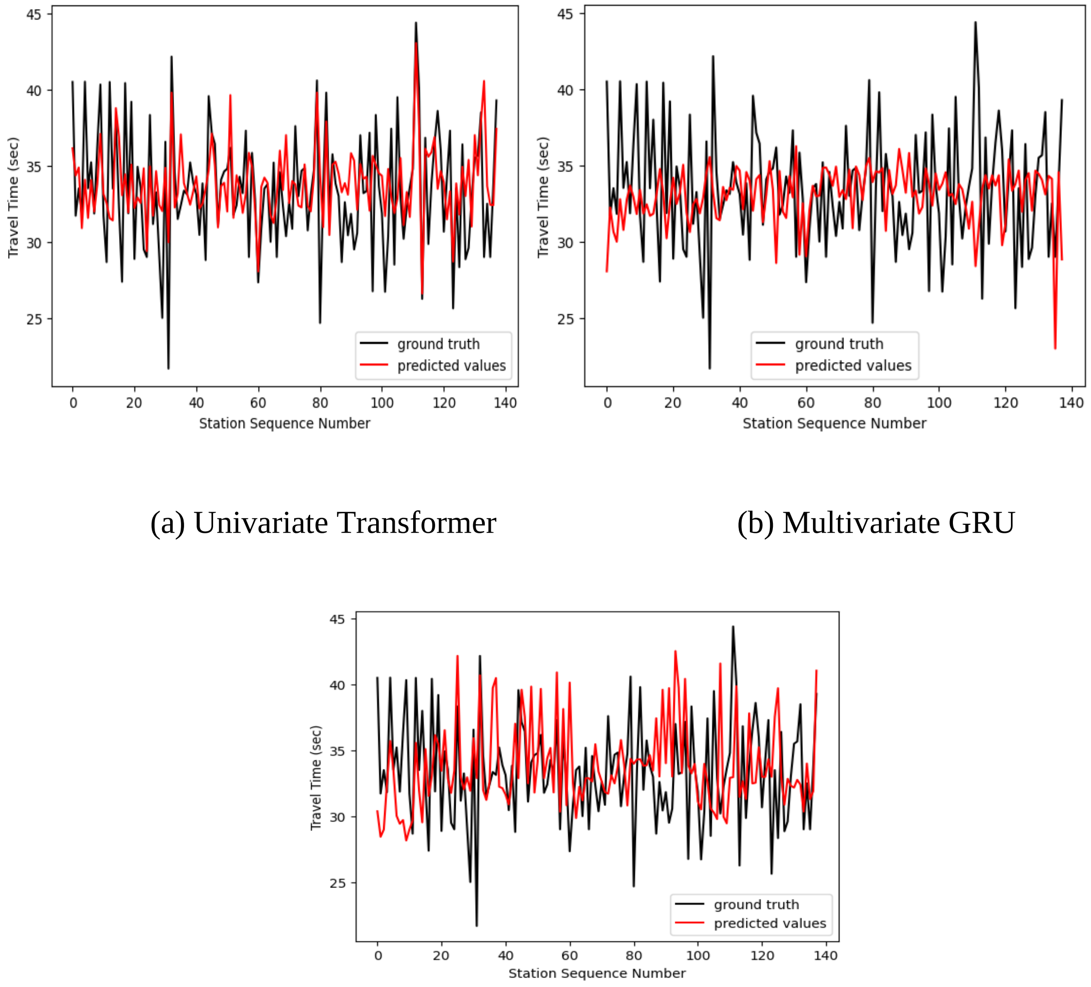
The results, analyses, and findings from the various models described in Chapter 4 will be covered in this chapter. In Chapter 4, we talked about the different model architect features we used to develop the Univariate Transformer Model, Multivariate GRU Model, and Multivariate LSTM Model. We also discussed our benchmark models – The historical Average Model and the XGBoost Model, which will be used to compare the performances of our developed models. The models' performances will be evaluated based on a metric called Mean Absolute Percentage Error (MAPE), a measure of the prediction accuracy of a forecasting model. We will demonstrate how the results of the different models vary across multiple routes. The performances of our model in different traffic conditions, such as peak hours and off-peak hours, and the comparison of computation times for prediction will also be illustrated in this chapter. Additionally, we will demonstrate the impact of adding external variables such as the expected travel time, dwell time and jam factors to our models and discuss how it improves the model results.

5.2 1-HOUR AHEAD TRAVEL TIME PREDICTION RESULTS

We will illustrate the one-hour ahead travel time predictions on the six routes we selected. The predictions results will be evaluated based on a performance metric called Mean Absolute Percentage Error (MAPE). MAPE is a commonly used measure of accuracy for forecasted data. It is a percentage-based error metric that measures the average absolute percentage difference between the actual and predicted values of a time series. MAPE is computed as the following equation:

$$MAPE = \frac{1}{N}\sum_{i=1}^{N}\left|\frac{t_i - \widehat{t_i}}{t_i}\right|$$

Where t_i is the actual travel time, $\widehat{t_i}$ is the predicted or forecasted travel time, and N is the total number of observations.



(a) Univariate Transformer

(b) Multivariate GRU

(c) Multivariate LSTM

Figure 5.1: Sample 1-Hour Ahead Prediction Trends

Figures of 5.1 illustrates an example of how the model trends and patterns look for our developed Transformer, GRU, and LSTM models. The univariate Transformer model was able to pick up the sharp peaks much better than the other two models, which indicates that even with using only the travel times as input, the Transformer model was able to handle complex patterns more. Again, the Mean Absolute Percentage errors for the three models are respectively 7.15%, 10.61%, and 10.99%, meaning that the Transformer model had the lowest MAPE. ***Table 5.1*** provides a summary of the Mean Absolute Percentage Errors (MAPE) for all the models for our forecasts at the ninth hour along each route. The outcomes show that the Univariate Transformer Model had the highest prediction accuracy across all six of the routes we considered for our study.

Table 5.1: 9th Hour Predictions for the Six Routes

Model Names	**MAPE (%)**					
	Route 18125	**Route 18128**	**Route 18139**	**Route 18144**	**Route 18145**	**Route 18146**
Univariate Transformer Model	8.1931 %	8.1885 %	7.2594 %	5.9545 %	6.7964 %	5.5367 %
Multivariate GRU Model	11.6446 %	13.5734 %	7.8969 %	7.3952 %	13.1124%	8.5709 %
Multivariate LSTM Model	11.3841 %	12.0381 %	8.3290 %	8.3385 %	13.542 %	9.5329 %
Historical Average Model	10.8579 %	12.3125 %	9.2378 %	7.6359 %	7.6359 %	7.6729 %
XGBoost Model	9.7582 %	8.4419 %	7.6414 %	6.9381 %	6.9381 %	7.2245 %

Table 5.2 summarizes the percentages of better prediction accuracy Transformer Model gives than all other models for predicting the average bus travel times at the 9th Hour which can be considered as busy traffic hour. As can be shown, for all six routes, the univariate Transformer Model outperformed the multivariate GRU and LSTM models.

Also, compared to the Historical Average Model and XGBoost Model, it offered greater accuracy. The ground truth and predicted value plots from ***Figure 5.1*** also show that the Transformer Model can pick the shapes much better compared to the GRU and LSTM models. The second-best model for all the six routes at the 9th hour according to the ***Table 5.1 and 5.2*** was the XGBoost model, which performed better than the Multivariate GRU and LSTM models in most cases. This result is shown for a single hour, and the results for the entire day will be demonstrated in the following section.

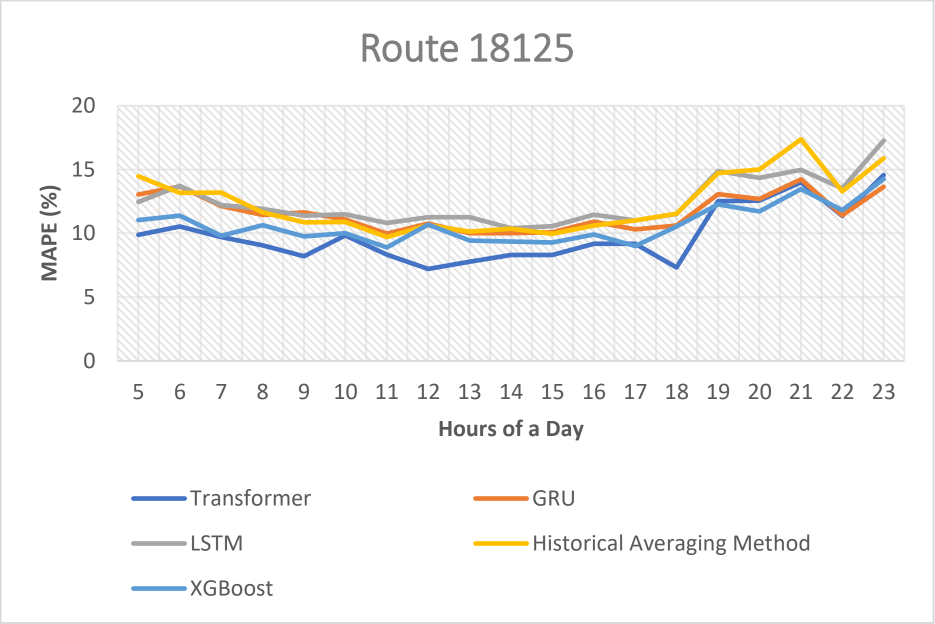
Table 5.2: Transformer Model Performance Comparison at the 9th hour

Models	**Route 18125**	**Route 18128**	**Route 18139**	**Route 18144**	**Route 18145**	**Route 18146**
Multivariate GRU	29.64 %	39.67 %	8.07 %	19.48 %	48.17 %	35.40 %
Multivariate LSTM	28.03 %	31.97 %	12.84%	28.59 %	49.81 %	41.92 %
Historical Average	24.54 %	33.49 %	21.42 %	22.02 %	10.99 %	27.84 %
XGBoost Model	16.04 %	3 %	5 %	14.18 %	2.04 %	23.36 %

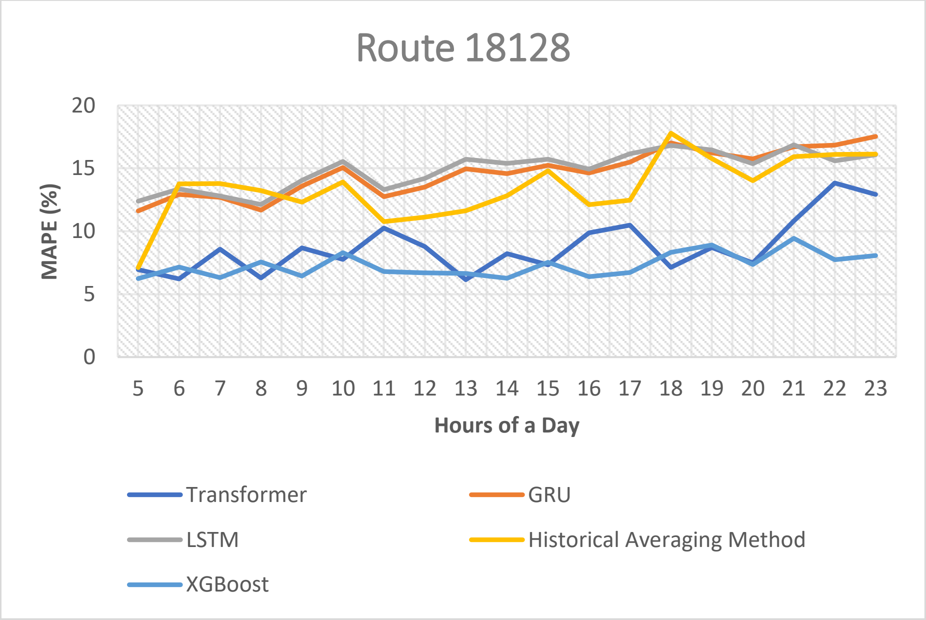
5.3 COMPARATIVE ANALYSIS BETWEEN DIFFERENT MODELS – ALL HOUR PREDICTIONS

5.3.1 One-Hour Ahead Predictions – All Hours

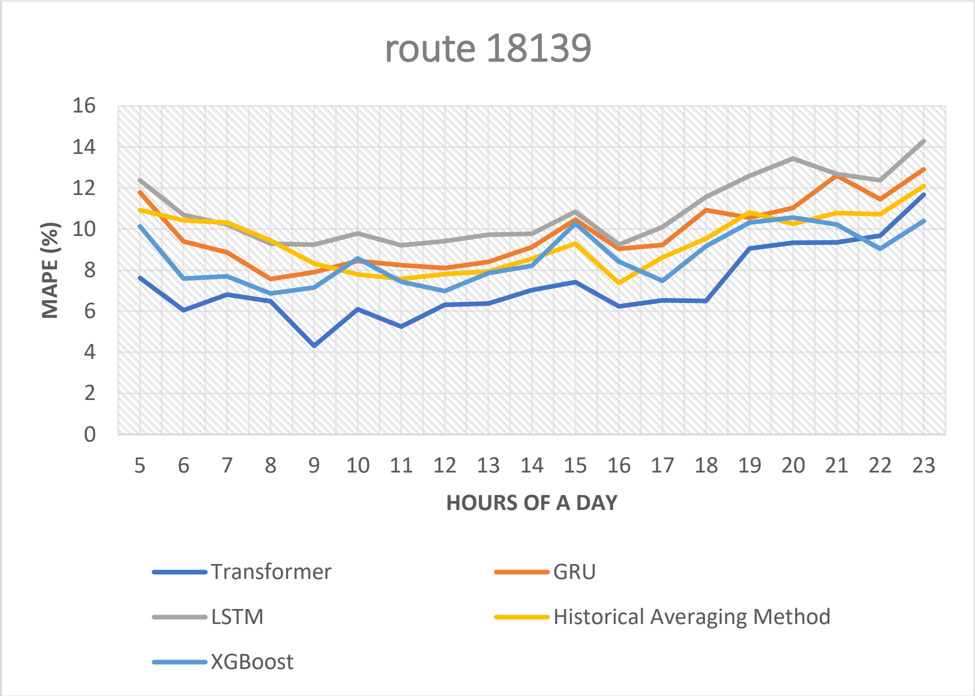
By demonstrating various comparative analyses based on the model performances, this part analyzes the robustness of our suggested univariate Transformer model. We displayed the outcomes for a specific hour (9th Hour) in the previous section. To see how the Transformer model outperforms than other models, we plotted the average MAPEs for all the routes from 5 am – 12 pm at various times of the day. The graph plots indicate how the average travel time looks at different hours of a day.



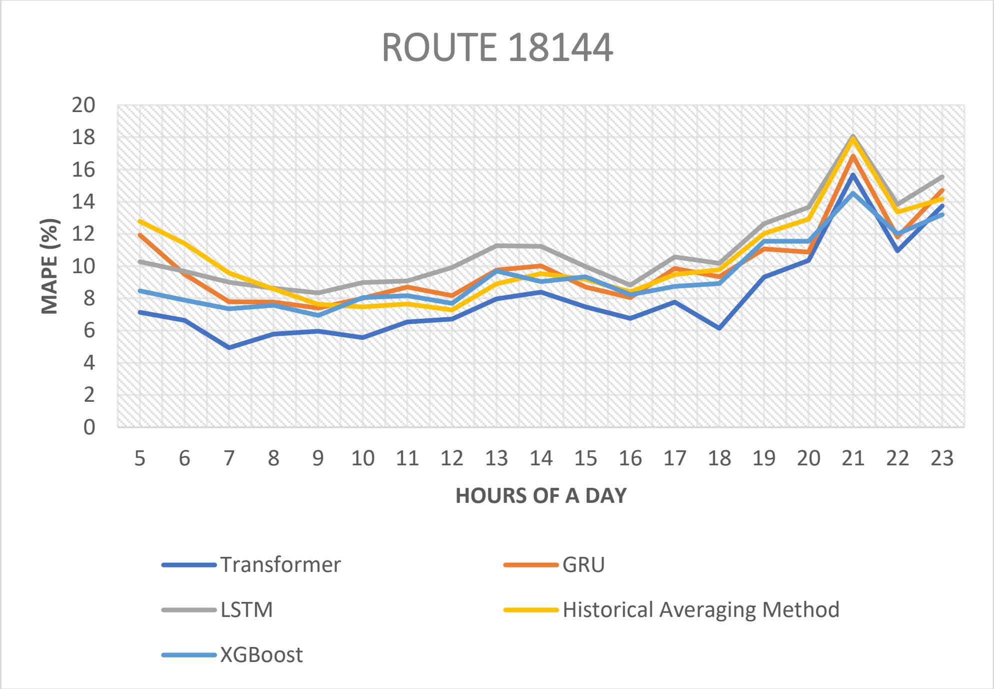
(a) Comparative Analysis between different models on Route 18125



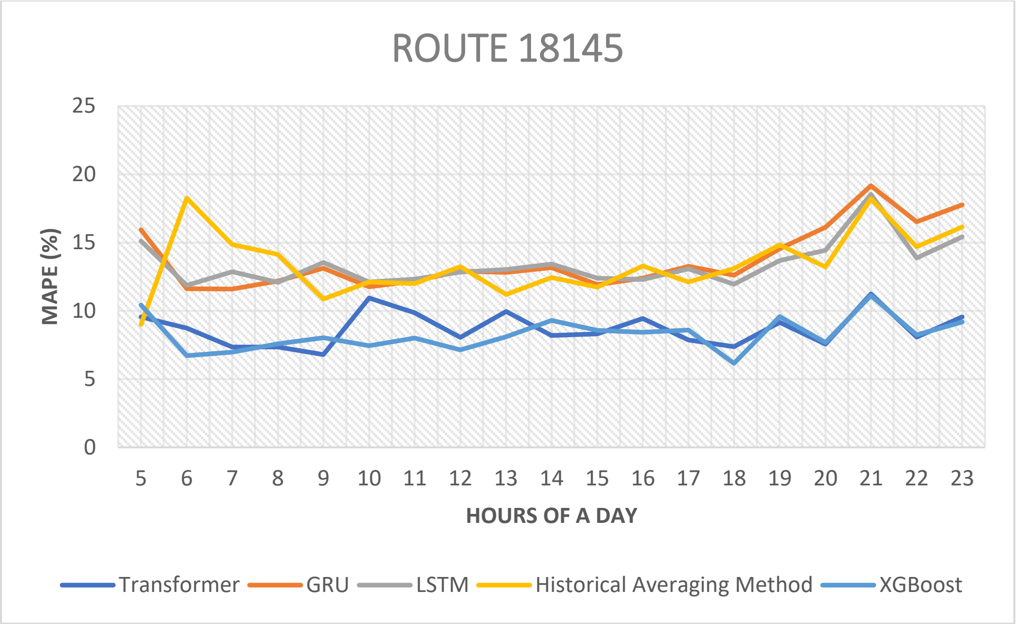
(b) Comparative Analysis between different models on Route 18128



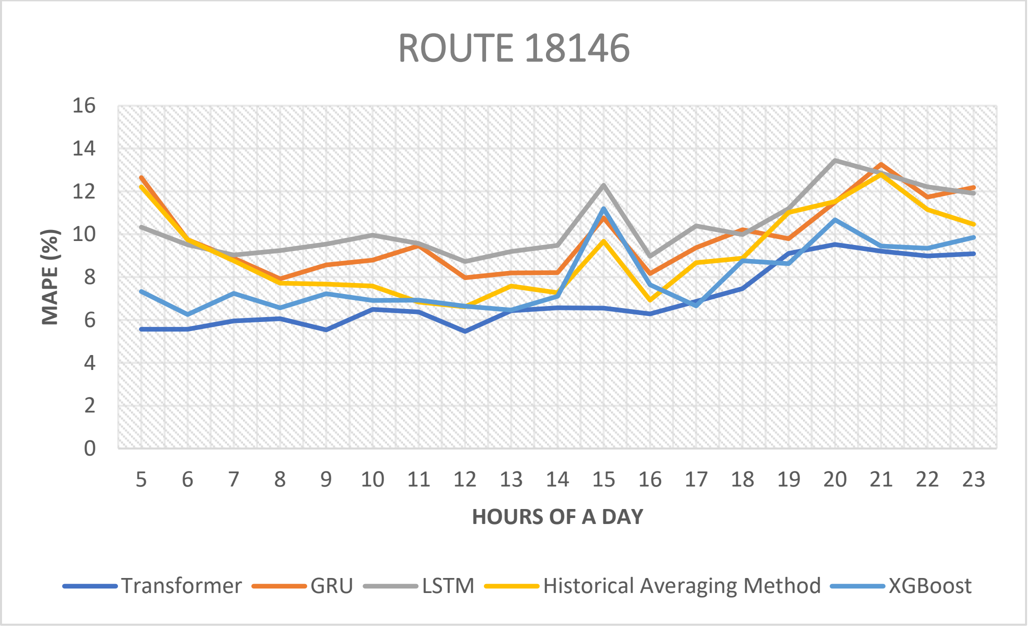
(c) Comparative Analysis between different models on Route 18139



(d) Comparative Analysis between different models on Route 18144



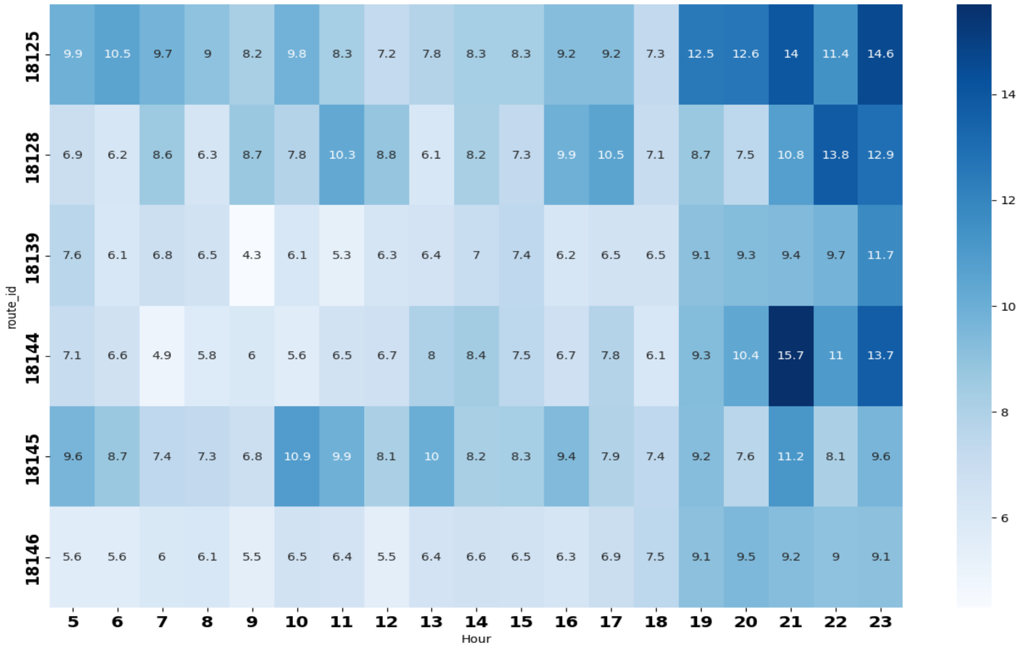
(e) Comparative Analysis between different models on Route 18145



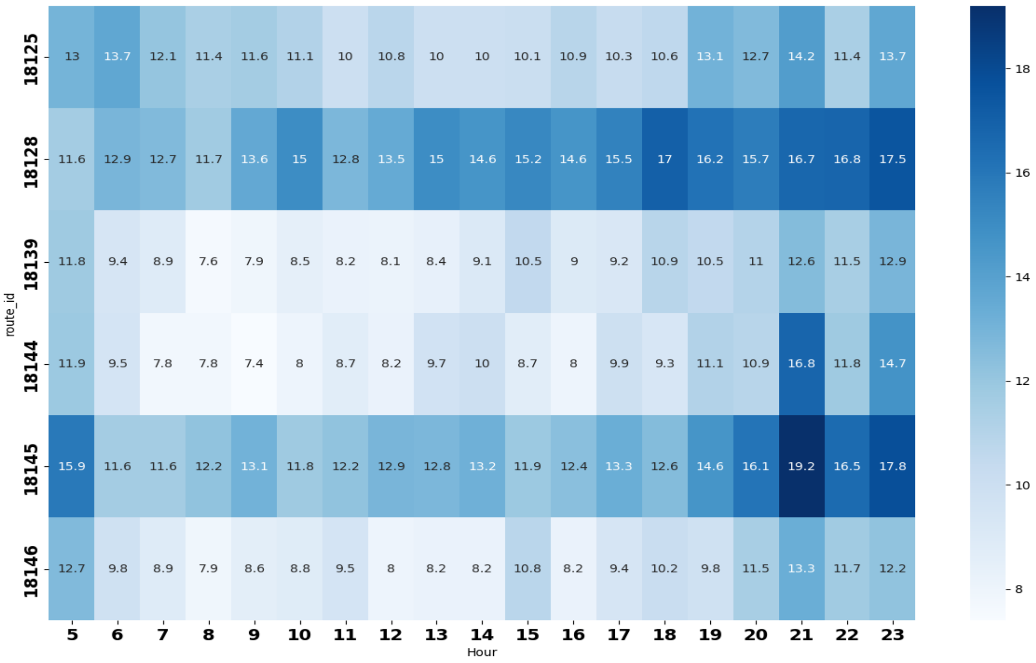
(f) Comparative Analysis between different models on Route 18146

Figure 5.2: One-hour prediction results for different models across multiple routes

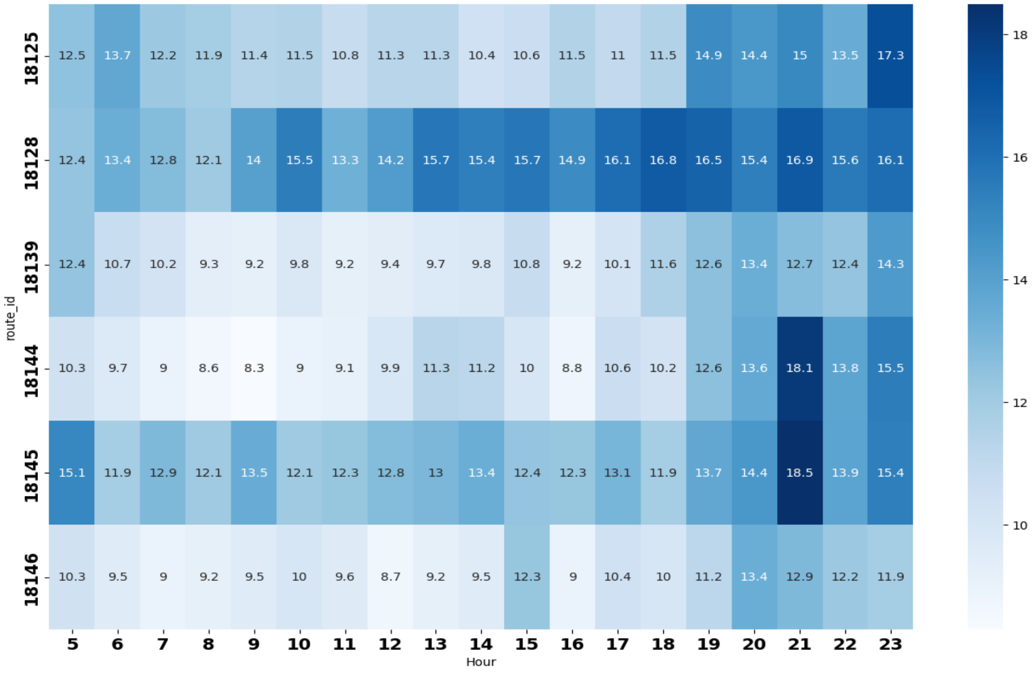
Figures 5.2 (a) to 5.2 (f) show that the Transformer model demonstrated greater accuracy than other models in most of the scenarios. There are only a few instances for Routes 18128 and 18145 where the XGBoost displayed improved results for a few hours. Transformer model might have overfitted on some of the data, which would explain why it performed worse than XGBoost in those cases. The model hypermeters might be another factor. It's likely that the hyperparameters selected for the XGBoost model were better suited in some of the situations where XGBoost shown greater performance. ***Figures 5.3 (a) – 5.3 (e)*** show the heatmaps for Mean Absolute Percentage Errors for different models we developed for predicting average bus travel times at different hours of a day.



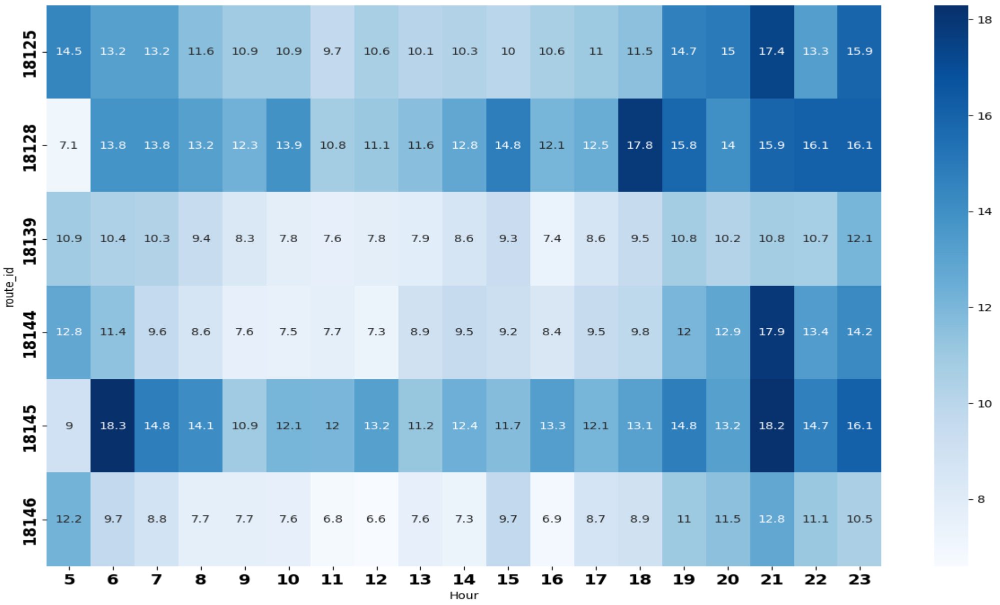
(a) Heatmap of 1-Hour Ahead Prediction Results – Transformer



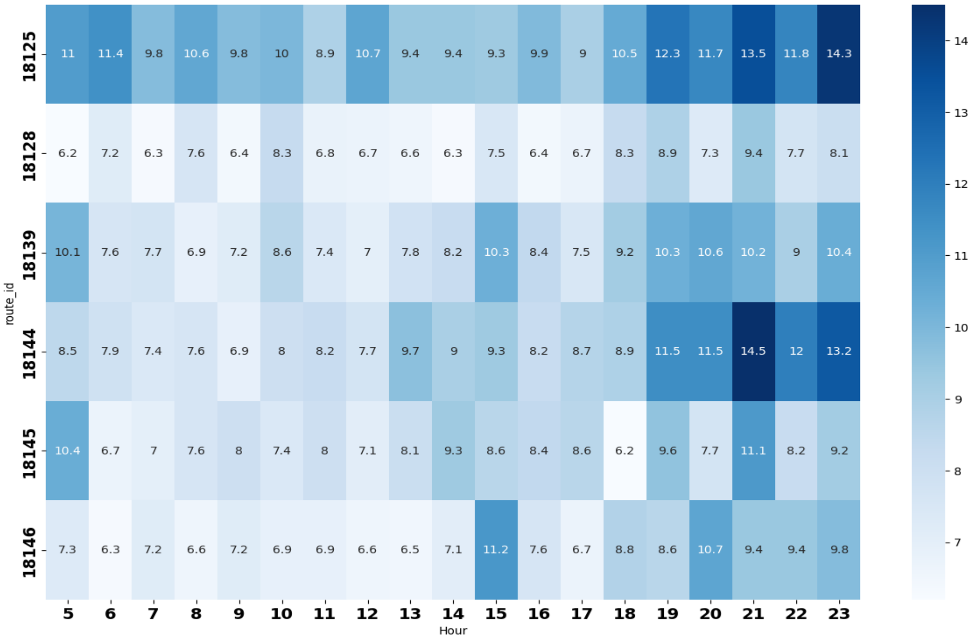
(b) Heatmap of 1-Hour Ahead Prediction Results – GRU



(c) Heatmap of 1-Hour Ahead Prediction Results – LSTM



(d) Heatmap of 1-Hour Ahead Prediction Results – HA Model



(e) Heatmap of 1-Hour Ahead Prediction Results – XGBoost Model

Figure 5.3: 1-Hour Ahead Prediction MAPE Heatmaps for different Models across Multiple Routes

In contrast to other models, the Transformer model has the lowest minimum, maximum, and median MAPE values, according to the error bar plots for the minimum, maximum, and median MAPEs in ***Figure 5.4***. The transformer model was able to capture

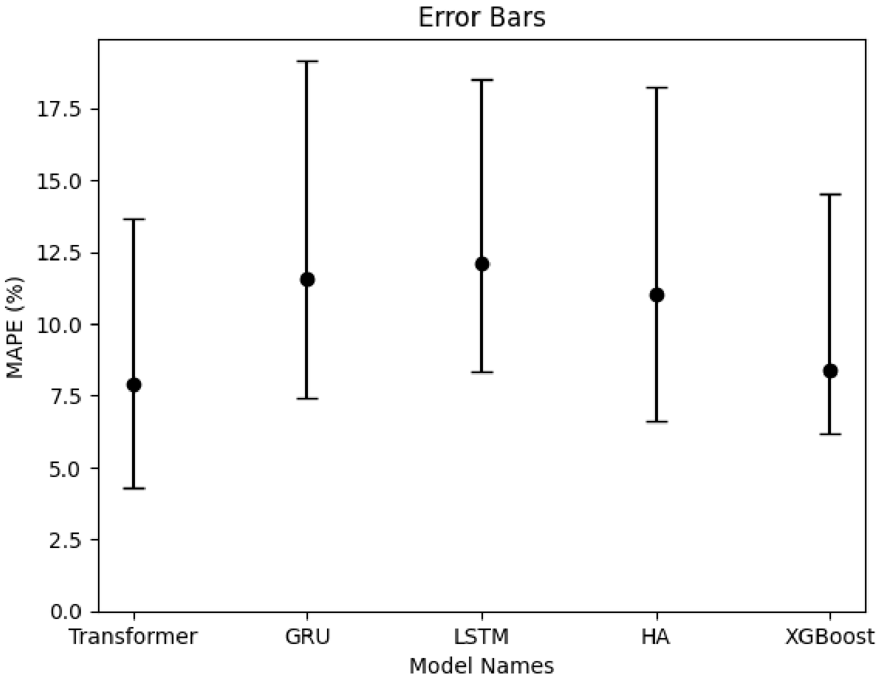
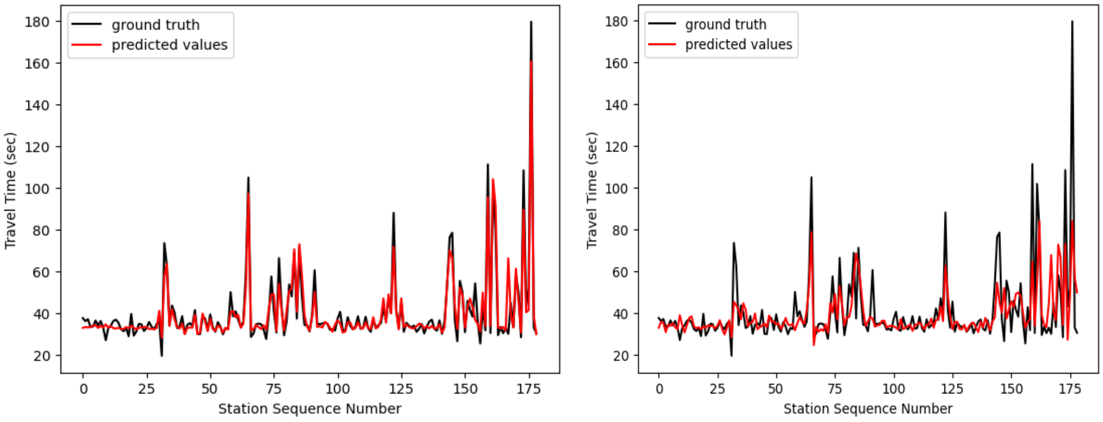


Figure 5.4: Error Bar Plots (Minimum, Maximum, and Median MAPEs)



(a) Univariate Transformer Model

(b) XGBoost Model

Figure 5.5: Transformer and XGBoost Model Trend

the sharp trends better than the XGBoost model, even though their medians are extremely close to each other (see ***Figure 5.5***). The Transformer model can handle complex patterns well even when only using the mean travel times as input.

Table 5.3: Summarized Model Results

Models	**MAPE**		
	Minimum	**Mean**	**SD**
Univariate Transformer	4.32 %	8.29 %	2.19 %
Multivariate GRU	7.40 %	11.62 %	2.71 %
Multivariate LSTM	8.34 %	12.18 %	2.38 %
Historical Average	8.61 %	12.36 %	2.84 %
XGBoost	6.15 %	8.71 %	1.82 %

Note: SD = Standard Deviation

Section 5.2 and 5.3 demonstrates that the Univariate Transformer model showed the best results among all other models. We also figured out the Minimum, Mean, and Standard Deviation of MAPEs for all the models, which is summarized in ***Table 5.3***. The minimum MAPE measures of 1-Hour ahead average travel time prediction for the Transformer Model is 4.32%, which is the lowest compared to other models. The mean MAPE measure of the Transformer model is 8.29 %, which is close to XGBoost Model, but performs much better than the GRU, LSTM, and HA models. However, the Transformer model had a higher standard deviation of MAPE than the XGBoost model but lower than other deep learning models. Given that there are more parameters to estimate in the deep learning training process than in the XGBoost, the results of deep learning prediction should have a higher level of randomness. The deep learning models also have many hyperparameters that need to be tuned. The optimal value of these hyperparameters can vary depending on the dataset, which may cause higher standard deviation.

5.3.2 Transformer Model Outperforms Other Models: Discussions and Insights

From the discussions in the **sections 5.2 and 5.3**, it became clear that the Transformer delivered the best results. Below are some hypotheses as to why Transformer Model may be working so well:

- The Transformer Model relies solely on a multi-headed self-attention mechanism, which allows it to capture the relationships between all input sequence elements in a single pass. This differs from other sequence-to-sequence models, such as GRU and LSTM, which process the input sequence one element at a time. The self-attention mechanism in Transformer model allows it to selectively attend to different parts of the input sequence, giving the ability to model long-range dependencies and handle variable-length input sequences.
- The multiheaded self-attention mechanism enables the Transformer to learn various representations of the input sequence simultaneously. The ability of each attention head to focus on a distinct feature of the input enables the model to capture various dependencies and interactions between the sequence's components.
- Transformers also take advantage of residual connections, letting the model remember data from the previous layers and simplifying learning long-term dependencies. The residual connections also help to mitigate the vanishing

gradient problem that can occur in deep neural networks. The residual connections can be represented mathematically as:

$$Output = LayerNorm(Sublayer(Input) + Input)$$

The multi-headed self-attention sub-layer or the feedforward sub-layer are the two sub-layer kinds employed in the Transformer. The original input is added to the sub-layer's output, and the sum is then put through a layer normalization step. The network's next sub-layer receives this output as input.

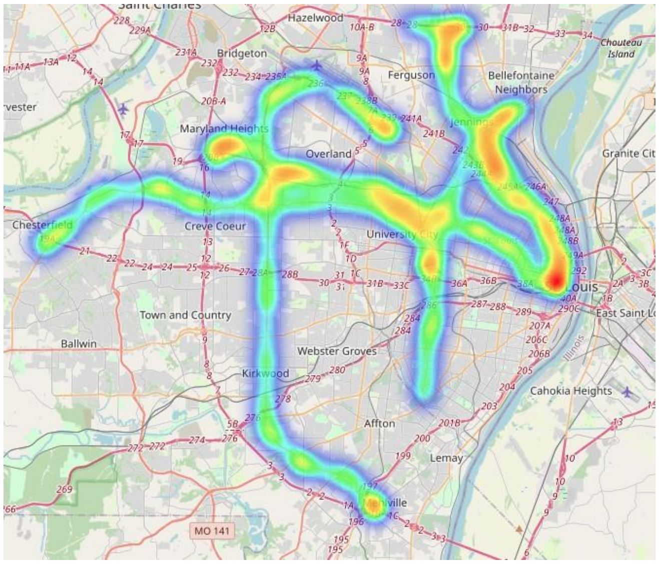
5.3.3 XGBoost Model Outperformed GRU and LSTM: Discussions and Insights

The second-best model from the above discussions, we got was the XGBoost model, which outperformed the GRU and LSTM models. These are the possible reasons behind:

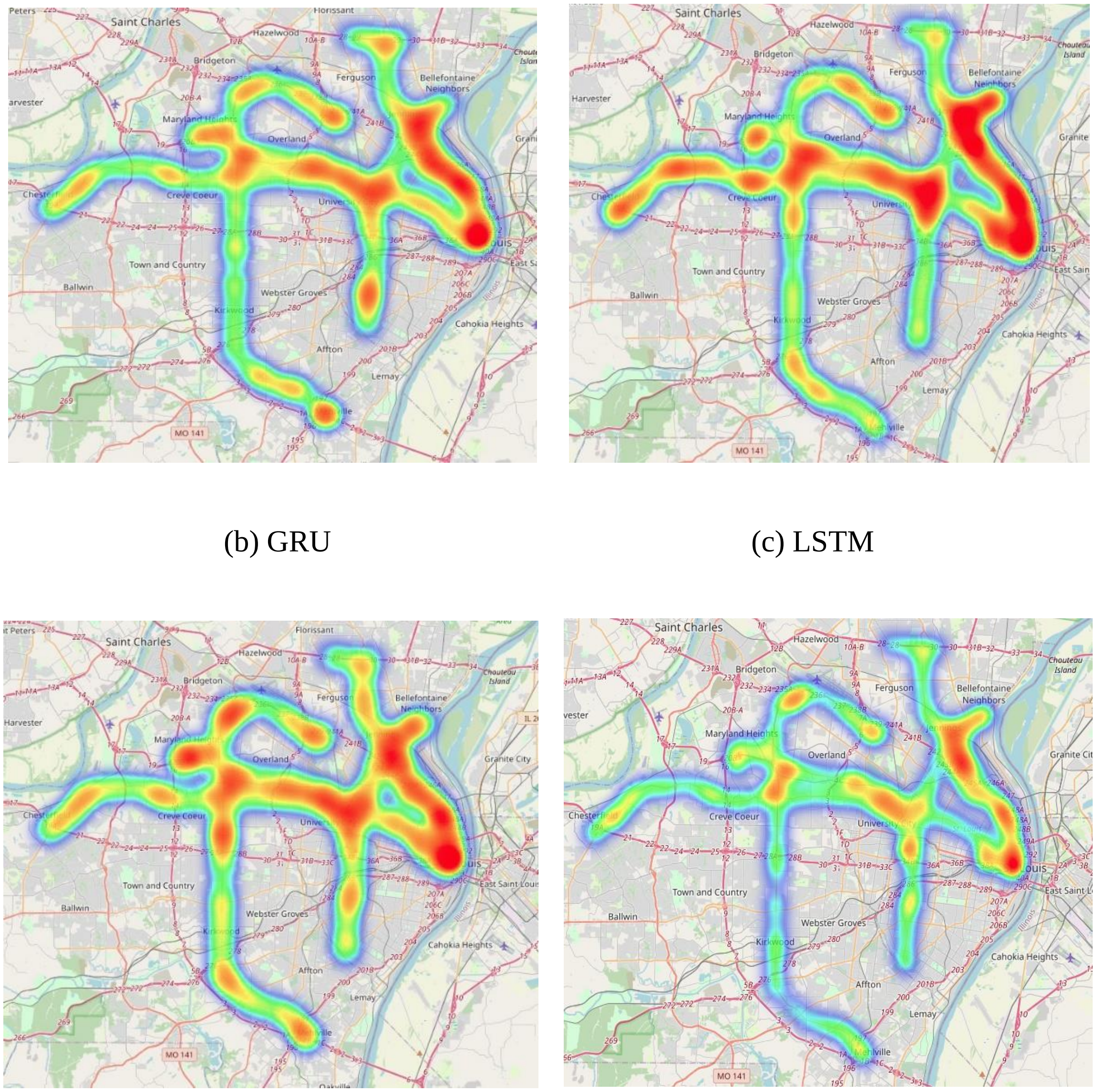
- XGBoost is a gradient boosting algorithm that can handle non-linear patterns in the data, which is crucial in capturing complex long-range dependencies. In contrast, GRU and LSTM models can also handle non-linear patterns well. Still, they can suffer from vanishing or exploding gradients when trying to propagate information over long sequences.
- XGBoost is faster and computationally efficient than GRU and LSTM, especially when dealing with larger datasets. This is because XGBoost parallelizes the training process, while GRU and LSTM are inherently sequential.

5.4 MODEL PERFORMANCES FOR DIFFERENT STATIONS ALONG MULTIPLE ROUTES

The previous section showed the MAPE along different routes in particular hours of a day. This section will illustrate how the MAPE varies for different models in different stations along the six different routes we selected. We produced some heatmaps based on model performances for multiple stations along different routes. We calculated the MAPEs for each station along each route and using the latitude and longitude, we showed the values in the map. The red marks in the heatmap indicate areas of high intensity or higher values of MAPE. The heatmap color scale typically goes from blue (low intensity) to yellow (medium intensity) to red intensity (high intensity).



(a) Transformer



(b) GRU

(c) LSTM

(d) HA

(e) XGBoost

Figure 5.6: Model Performances Along Different Routes – Heatmaps

Figure 5.6 (a) shows that Historical Average Model, LSTM, and GRU all have significantly more red marks than the Transformer model. Notably, the XGBoost outperformed GRU, LSTM, and the Historical Average Model rather favorably. The XGBoost model, however, performed worse than the Transformer model. The results are like those of the MAPE comparison of several models for hourly forecasts in earlier parts. The Transformer Model fared best, with the XGBoost Model coming in second.

5.5 MODEL PERFORMANCES FOR PEAK AND OFF-PEAK HOURS

The model's performances under different traffic conditions, such as peak and off-peak hours, are shown in ***Table 5.4***. Peak hour is regarded as the hours of 7:00 to 10:00 am in the morning and 4:00 to 7:00 pm in the evening. It is observed that the Transformer model performed significantly better than other models in both traffic conditions for all the routes.

Table 5.4: Model Performances for Peak and Off-Peak Hours

	Route 18125		**Route 18128**		**Route 18139**		**Route 18144**		**Route 18145**		**Route 18146**	
Model	**Peak**	**Off-Peak**	**Peak**	**Off-Peak**	**Peak**	**Off-Peak**	**Peak**	**Off-Peak**	**Peak**	**Off-Peak**	**Peak**	**Off-Peak**
Transformer	8.8%	10.4%	6.51%	8.87%	6.14%	7.78%	6.22%	8.96%	7.69%	9.17%	6.36%	7.30%
GRU	11.2%	11.82%	14.17%	14.89%	8.92%	10.19%	8.36%	10.77%	12.52%	14.33%	8.85%	10.3%
LSTM	11.58%	12.84%	14.47%	15.07%	9.94%	11.32%	9.25%	11.85%	12.63%	13.77%	9.53%	10.8%
HA	11.47%	12.73%	13.61%	13.37%	8.94%	9.61%	8.91%	11.12%	13.05%	13.62%	8.11%	9.57%
XGBoost	8.94%	11.04%	6.96%	9.46%	7.8%	9.04%	7.96%	10.08%	7.63%	8.58%	7.35%	8.21%

Table 5.4 further shows that for all models, peak hour forecast accuracy is consistently higher than off-peak hour prediction accuracy**.** This is expected because during the peak hours the availability of data was much more compared to the off-peak hours. As a result, the models were able to train with more data during the peak hours resulted in better accuracy.

5.6 COMPUTATION TIME OF DIFFERENT MODELS

Computation time is important in bus travel time prediction models. To provide accurate predictions the prediction model must be able to process large amounts of data quickly. Moreover, it is important to optimize resource utilization to run the bus travel time prediction models. This includes using the available computing resources effectively, such as minimizing memory usage and GPU utilization. The models are also required to be updated regularly to ensure that they continue to provide accurate predictions. If the computation time is too slow, it may take a long time to update the model, which can result in outdated predictions. ***Table 5.5*** summarizes the average computation time to predict the travel time for the five models on an 11th Gen Intel(R) Core (TM) i9-11900 @ 2.50GHz processor with 32-GB random access memory. We ran the models on Google Colab using Graphics Processing Unit (GPU) with standard GPU class and runtime shape.

Table 5.5: Computation Time for Different Models

Model	Prediction Time
Univariate Transformer	7.42 s
Multivariate GRU	6.99 s
Multivariate LSTM	8.67 s
Historical Average	6.34 s
XGBoost	6.28 s

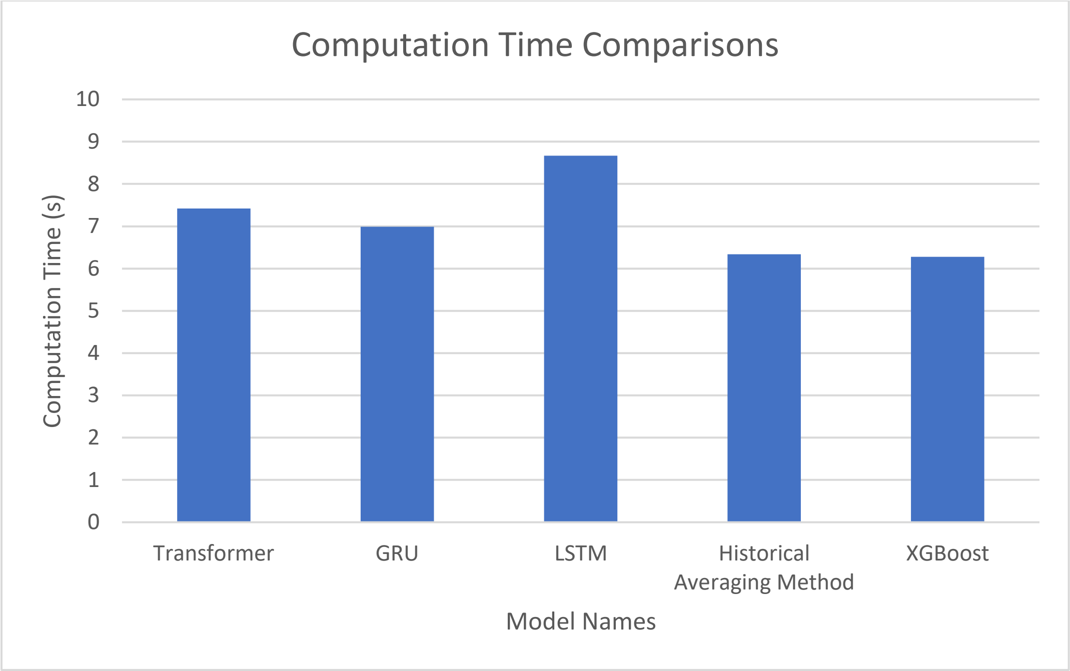
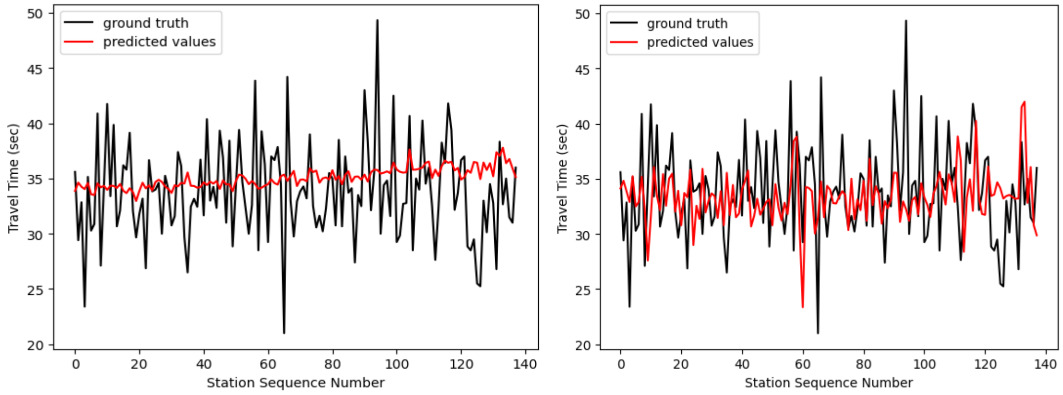


Figure 5.7: Model Prediction Computation Times

Table 5.5 and ***Figure 5.7*** show that LSTM model has the highest prediction computation time while XGBoost has the lowest. That is expected as XGBoost is faster and trained quickly due to lesser complexities than deep learning models. The average computation time for the Transformer model is about 7.42 s, which isn't much that much longer than the XGBoost, GRU and Historical Average Models.

5.7 IMPACT OF EXTERNAL FEATURES ON TRAVEL TIME PREDICTION

One of the additional insights we found from our study is the impact of different external transit features on bus travel time prediction. External factors such as the expected travel time from one station to another, dwell time, and jam factor were all input features that we used while developing the GRU and LSTM models. Nonetheless, as we suggested the univariate Transformer model in this study, we didn't take them into account when building the Transformer model. However, by using the GRU and LSTM models that we developed,

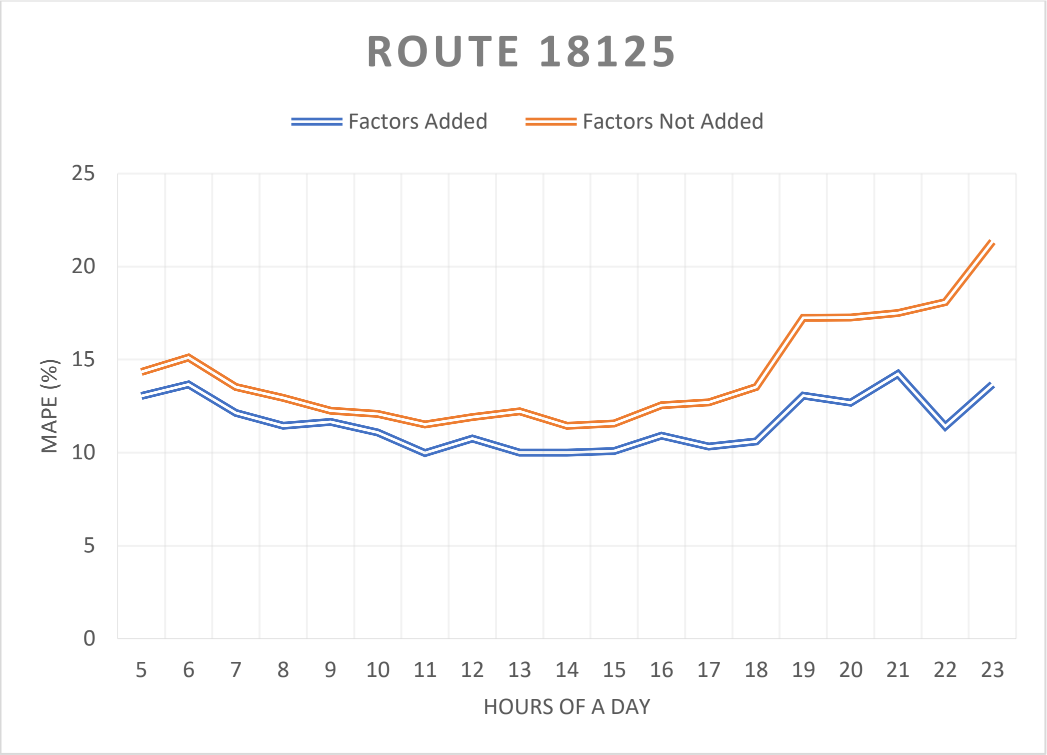


(a) Without adding external factors

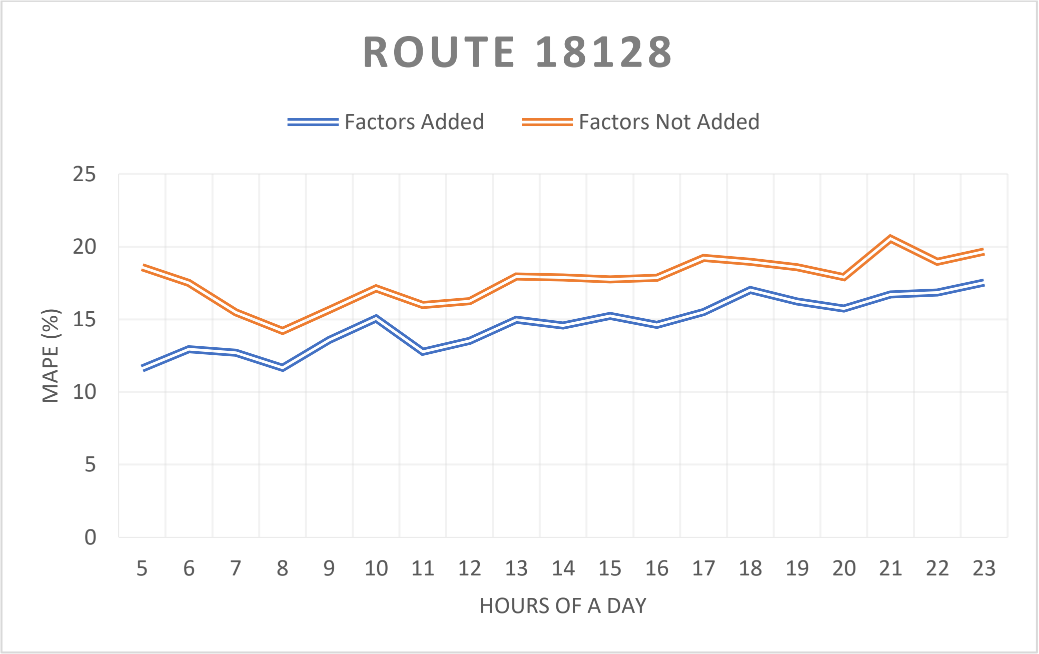
(b) After adding the external factors

Figure 5.8: Impact of external factors (GRU Model)

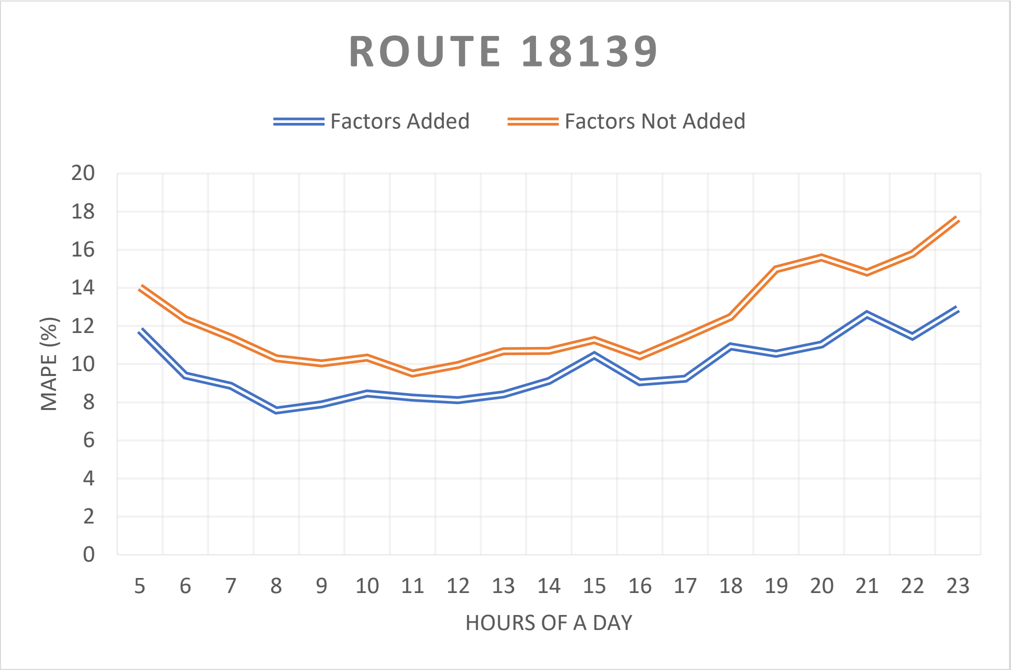
we can see how well these features work to improve the accuracy of these models. The addition of those elements improved the performance of both. In this section, we are showing the performances of GRU model to understand the impact of these factors. The instances in ***Figure 5.8*** illustrates how the model was able to learn the complex trends and patterns better after adding these external factors. In ***Figure 5.8 (a)***, we didn't consider the factors like expected travel time, dwell time, and jam factor. On the other hand, we included those factors in ***Figure 5.8 (b)***, which also boosted the model's performance by a good margin. The Mean Absolute Percentage Error in ***Figure 5.8 (a)*** is around 14%, while it is approximately 8.5% in ***Figure 5.8 (b)***. Compared to the first instance, where the external parameters weren't considered, it shows a 40% improvement in accuracy.



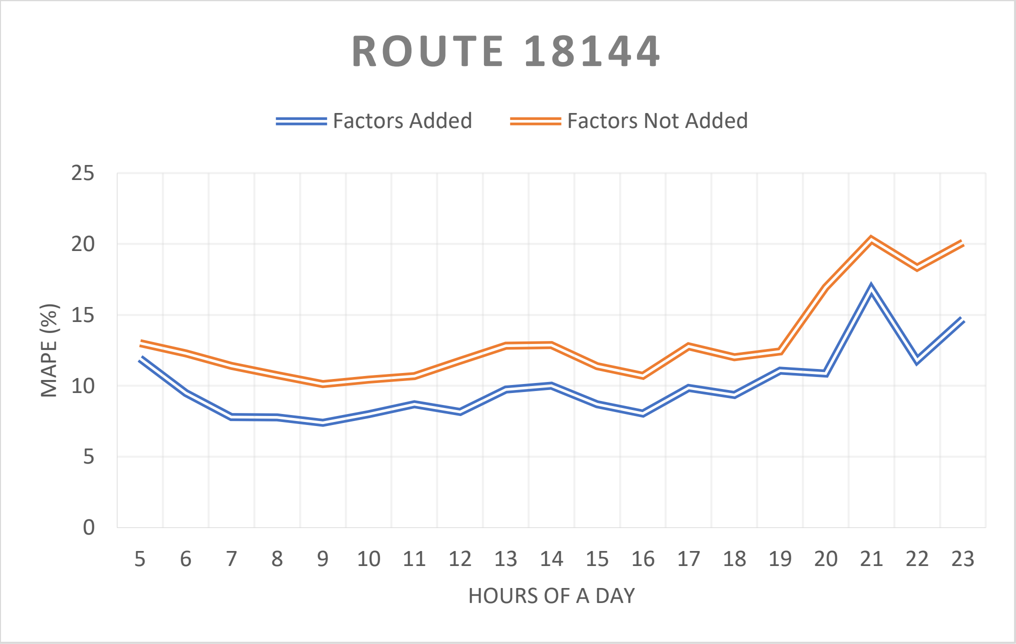
(a) Route 18125



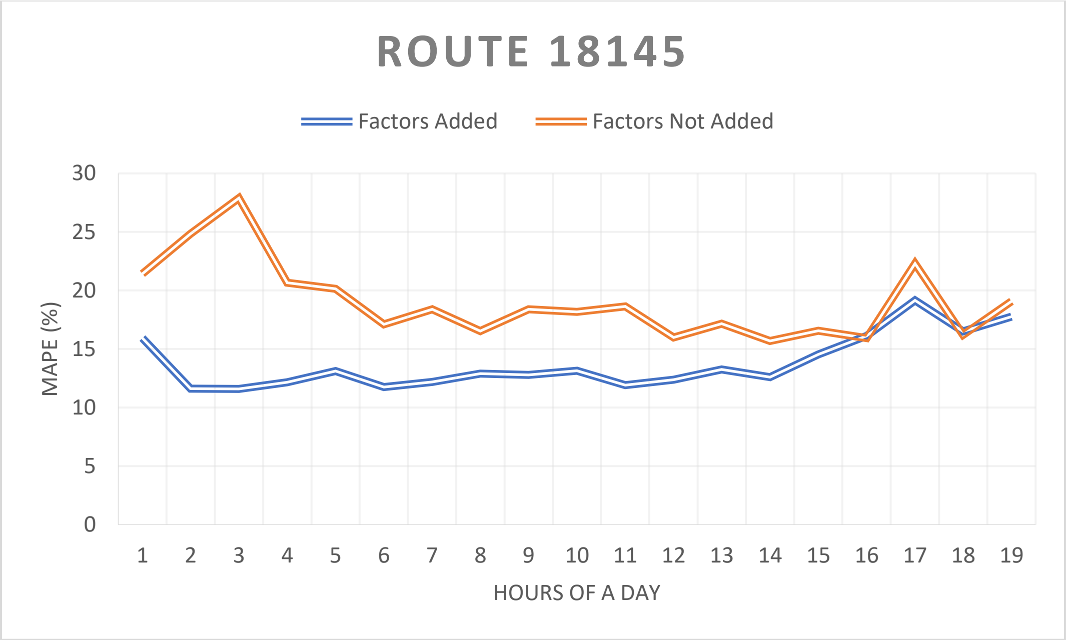
(b) Route 18128



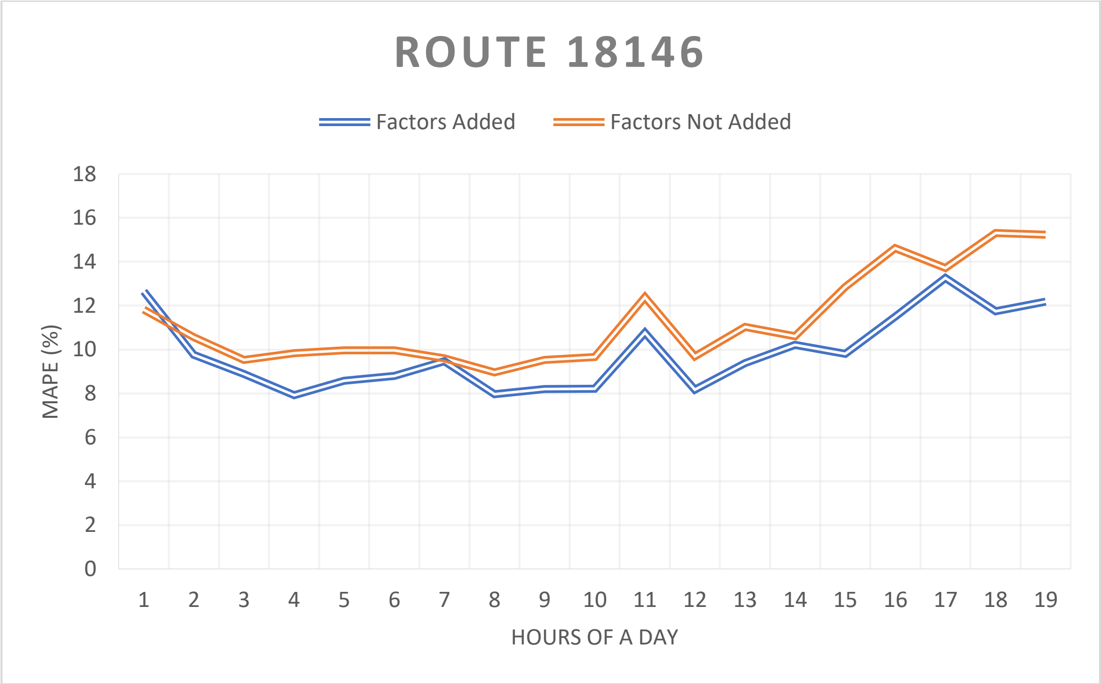
(c) Route 18139



(d) Route 18144



(e) Route 18145



(f) Route 18146

Figure 5.9: Impact of External Factors on Bus Travel Time Predictions

From the ***Figures 5.7*** above, it is notable that for all the routes, the bus travel time predictions are much accurate when we added the external factors such as dwell time, the expected time and jam factor. Jam factor refers to the congestion level on the roadways. High levels of traffic congestion can slow down buses and increase travel times. Dwell time refers to the amount of time a bus spends at a station to pick up or drop off passengers. Longer dwell times can add to the overall time of the bus. Expected travel time is the time we expect to travel from one station to another based on the probe data information. The improved outcomes with the addition of these features suggest that the created GRU model was able to learn the trends and patterns of bus trip times more effectively, indicating the stronger effectiveness of these external features.

5.8 CHAPTER REVIEW

Chapter 5 covers the results and insights of the different models and comparative analysis between different models. In this chapter, we demonstrated the one-hour ahead prediction results from 5th hour to 23rd hour for all the routes. We showed the graphical comparisons and heatmaps to compare the model results in this chapter. Apart from that graphical heatmaps were also shown for all models to identify the intensity of MAPE values in different stations across multiple routes. Chapter 5 included illustrations of how the models performed during Peak Hour and Off-Peak Hour traffic. This chapter also discussed the computation time for each prediction model.

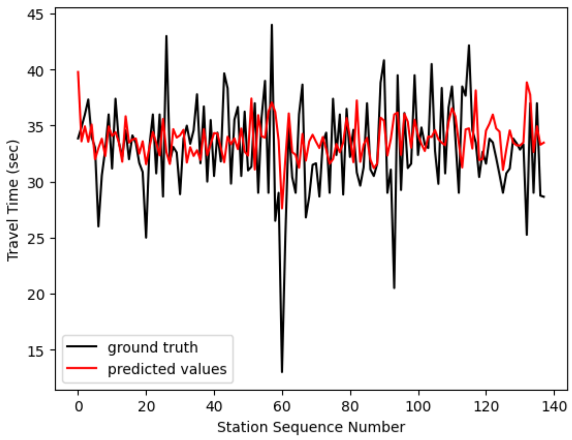
CHAPTER 6: CONCLUSIONS AND FUTURE RESEARCH

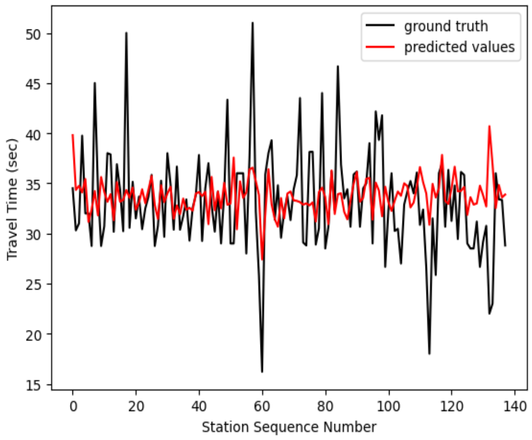
The scope of forecasting bus travel times more accurately has greatly increased due to the quick development of machine learning and high-performance computing. In the current study, we implemented three deep learning-based algorithms - Univariate Transformer, Multivariate GRU, and Multivariate LSTM for predicting average bus travel times incorporating heterogeneous datasets across multiple bus routes on an extensive network in the St. Louis region. Using the information of the previous 5-time steps (5 hours) of one station-to-station segment along a route, we predicted the average bus travel time for the following hour for the same segment. We followed this approach for all six routes, which were selected based on the greatest number of stops. We attempted to determine how well these models work when it comes to predicting travel times for long-range dependencies and contrasted our findings with those of other conventional machine learning models, such as the Historical Average Model and XGBoost model. The results showed that the Transformer model outperformed other models in prediction accuracy when we compared it with the one-hour ahead prediction results for each hour. We computed the minimum, mean, and standard deviation of MAPE for each model to summarize the results. The results show that the minimum and mean MAPE values of the Univariate Transformer Model were 4.32% and 8.29%, respectively. The second-best performer was XGBoost, which had values of 6.15% and 8.71%, respectively. We also plotted the heatmaps for all the routes to understand the intensity of MAPE values for each station along each route. The heatmap results also indicated that the Transformer model had the lowest intensity of MAPE values compared to other models. Additionally, we illustrated how these models perform during different traffic conditions, such as peak and off-peak Hours. The outputs demonstrated that the prediction accuracy during the peak hours was always higher than the off-peak hours. We also identified that for both conditions Transformer model performed significantly better than other models. Finally, we showed the model computation time for the prediction, where we found that XGBoost Model had the quickest computing time having a 6.28-second prediction time. The univariate Transformer model had a 7.42-second prediction time which is not far from the XGBoost model. The Univariate Transformer model was by far the best performer among all the frameworks we developed. In conclusion, it can be said that even with only historical travel times as input, a Transformer model can still capture some degree of long-range dependency through its self-attention mechanism. The model can use self-attention to weigh the importance of different historical travel times in the context of the entire bus route, allowing it to understand better the complex relationships between different parts of the route and their impact on travel times. Other factors that may affect travel times, such as traffic congestion, dwell times, and unexpected delays, could significantly impact the travel times of buses. By including this additional information in the model's input, the model may be better able to capture long-range dependencies and make more accurate predictions.

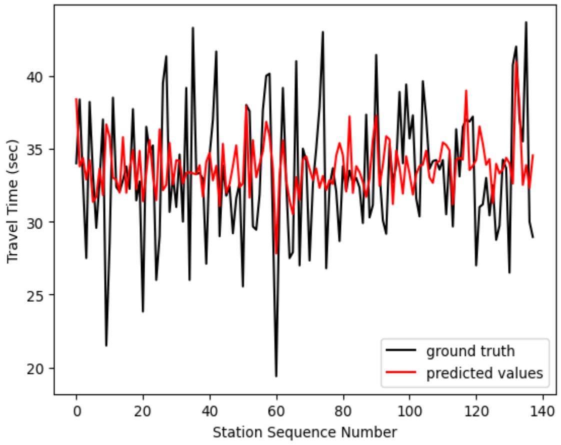
The future of bus travel time prediction modeling using transformer models looks promising. With the advancement in technology and the availability of large-scale datasets, these models have the potential to improve the accuracy of bus travel time predictions significantly. In the current study, we used the univariate Transformer model to predict the bus travel times for long-range scale road networks. However, in the future, the multivariate transformer model can be developed to predict the bus travel times using all the factors we used for the GRU and LSTM deep learning neural network models, such as jam factors, traffic speeds, expected travel time, dwell time, etc. to better capture long-range dependencies and make more accurate predictions. Researchers could also look at the use of cutting-edge data sources to enhance the conventional sources of data utilized in the Transformer Model, such as real-time GPS data or social media data. Furthermore, researchers could explore the use of ensemble techniques to combine multiple transformer models and multi-step forecasting approach to improve the accuracy of bus travel time prediction modeling. Ensemble methods have been shown to be effective in improving prediction accuracy in other domains, and their application in bus travel time prediction modeling could yield similar benefits.

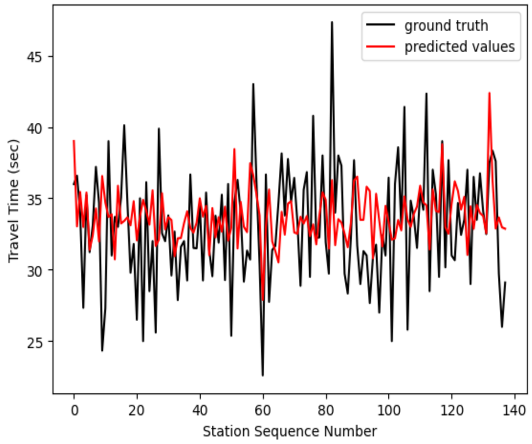
APPENDIX

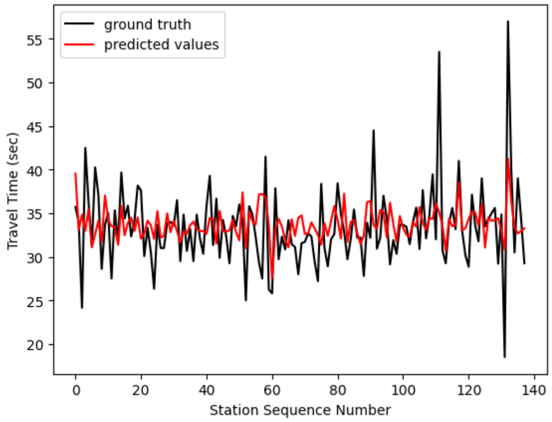
Transformer Model Predictions for 5th Hour to 23rd Hour – Route 18125

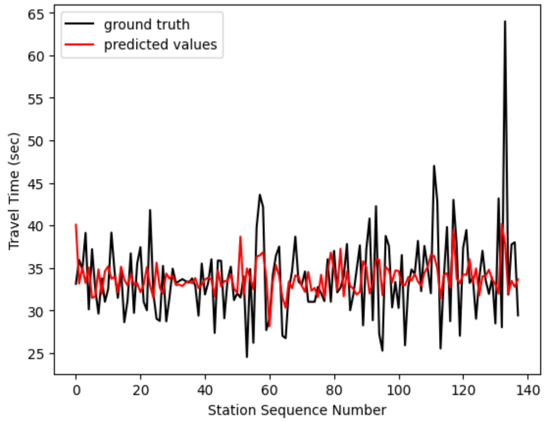


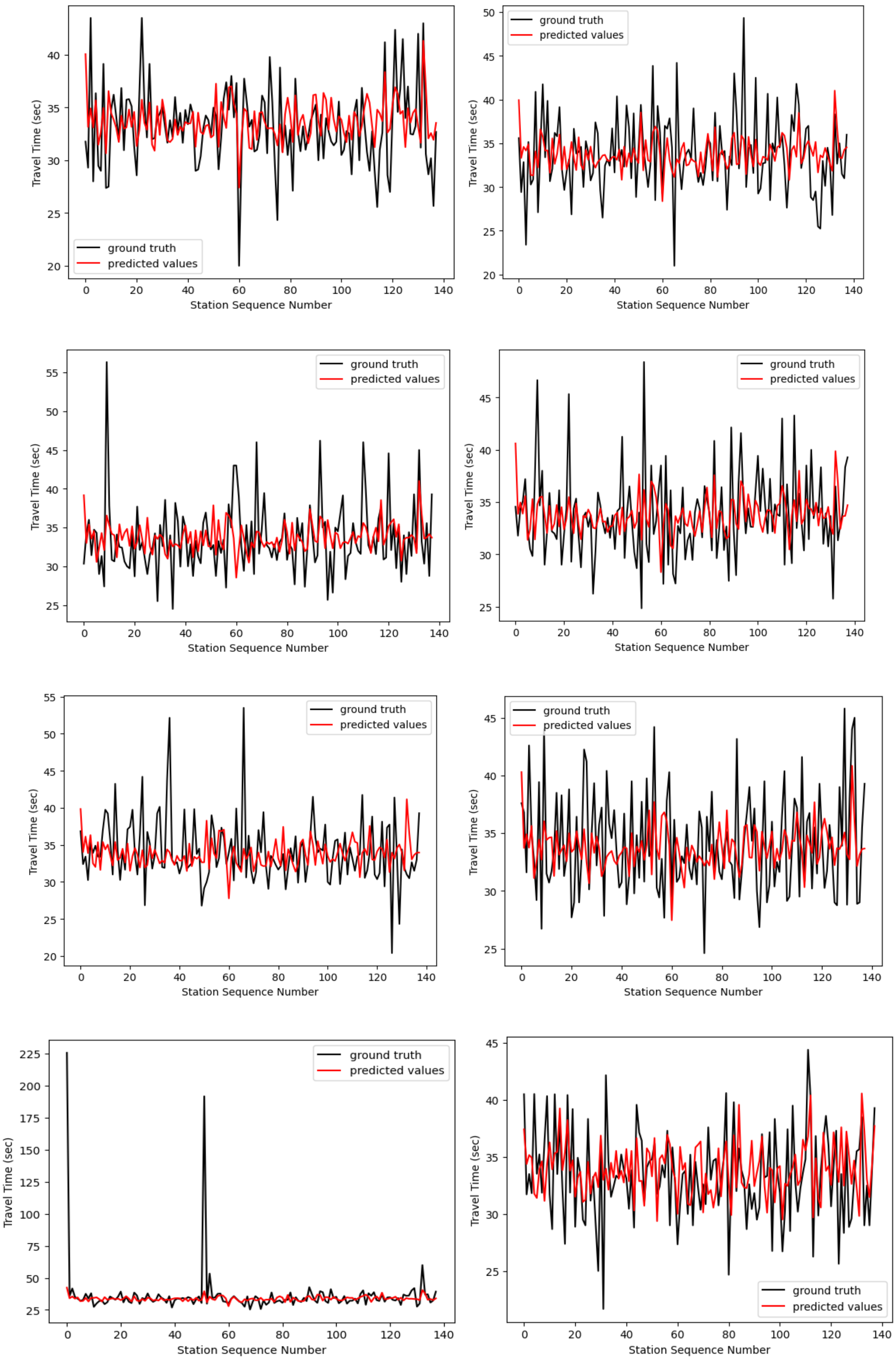


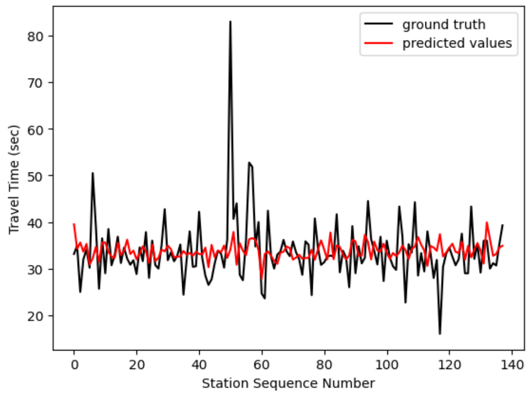


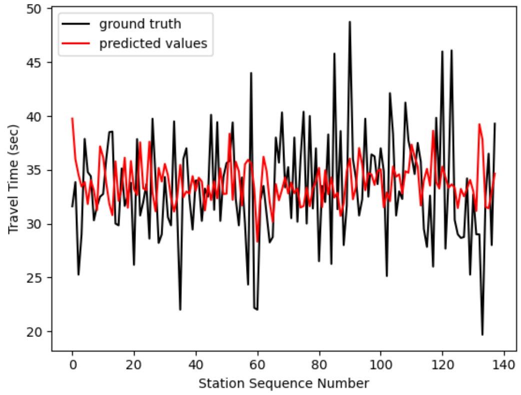


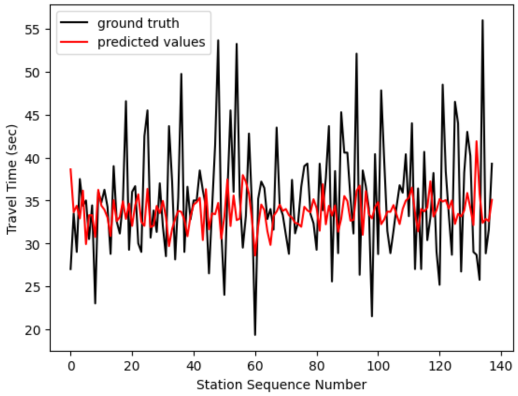


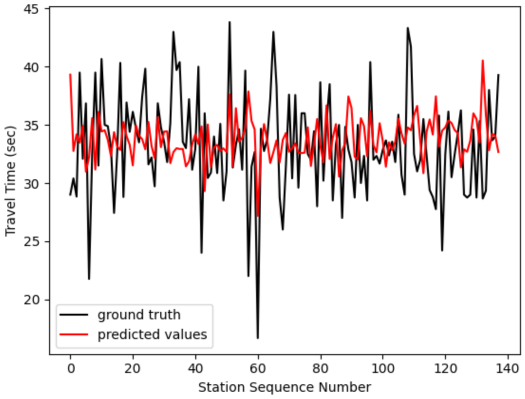


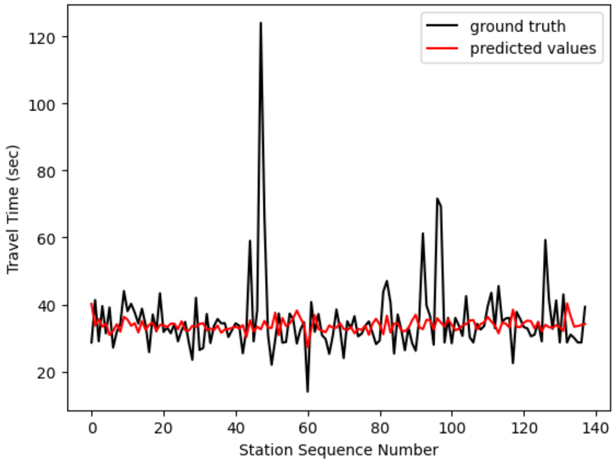

Travel Time (sec)
40
35
30
25
20
ground truth
predicted values
0
20
40
60
80
100
120
140
Station Sequence Number
Travel Time (sec)
50
45
40
35
30
25
20
ground truth
predicted values
0
20
40
60
80
100
120
140
Station Sequence Number
Travel Time (sec)
55
50
45
40
35
30
25
ground truth
predicted values
0
20
40
60
80
100
120
140
Station Sequence Number
Travel Time (sec)
45
40
35
30
25
ground truth
predicted values
0
20
40
60
80
100
120
140
Station Sequence Number
Travel Time (sec)
55
50
45
40
35
30
25
20
ground truth
predicted values
0
20
40
60
80
100
120
140
Station Sequence Number
Travel Time (sec)
45
40
35
30
25
ground truth
predicted values
0
20
40
60
80
100
120
140
Station Sequence Number
Travel Time (sec)
225
200
175
150
125
100
75
50
25
ground truth
predicted values
0
20
40
60
80
100
120
140
Station Sequence Number
Travel Time (sec)
45
40
35
30
25
ground truth
predicted values
0
20
40
60
80
100
120
140
Station Sequence Number


ground truth
predicted values
Travel Time (sec)
80
70
60
50
40
30
20
0
20
40
60
80
100
120
140
Station Sequence Number

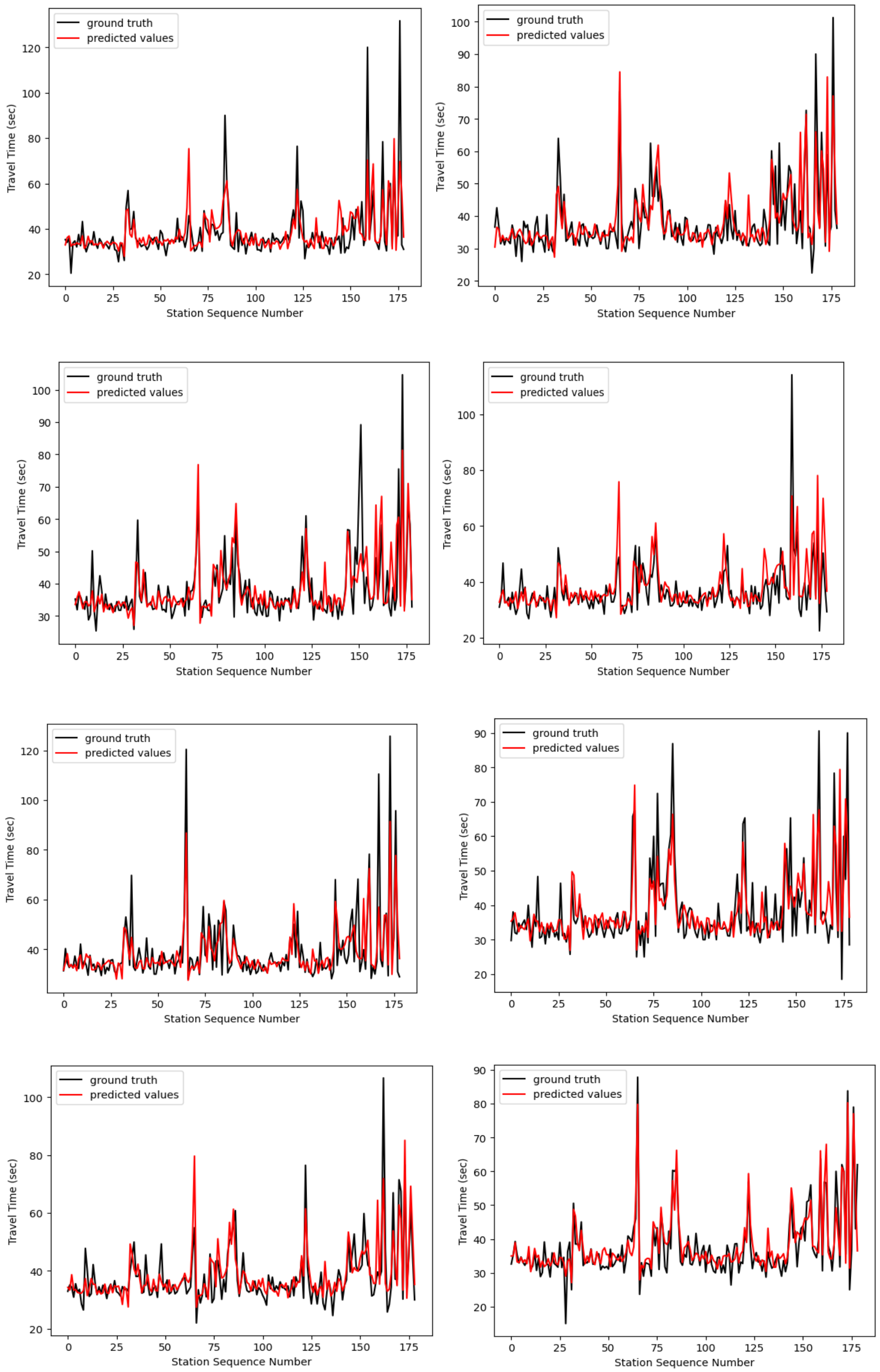

ground truth
predicted values
Travel Time (sec)
50
45
40
35
30
25
20
0
20
40
60
80
100
120
140
Station Sequence Number

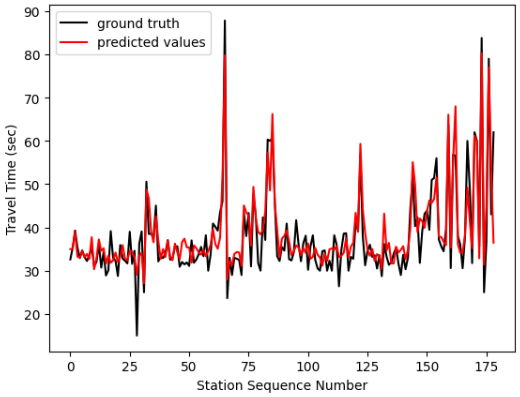

ground truth
predicted values
Travel Time (sec)
55
50
45
40
35
30
25
20
0
20
40
60
80
100
120
140
Station Sequence Number

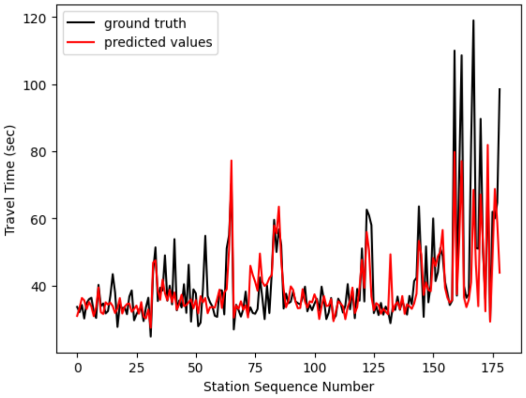

ground truth
predicted values
Travel Time (sec)
45
40
35
30
25
20
0
20
40
60
80
100
120
140
Station Sequence Number

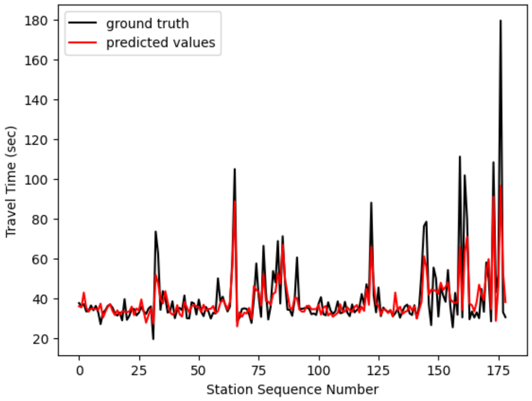

ground truth
predicted values
Travel Time (sec)
120
100
80
60
40
20
0
20
40
60
80
100
120
140
Station Sequence Number

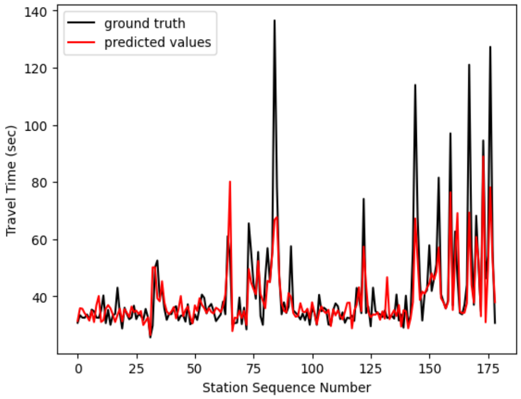
Transformer Model Predictions for 5th Hour to 23rd Hour – Route 18128

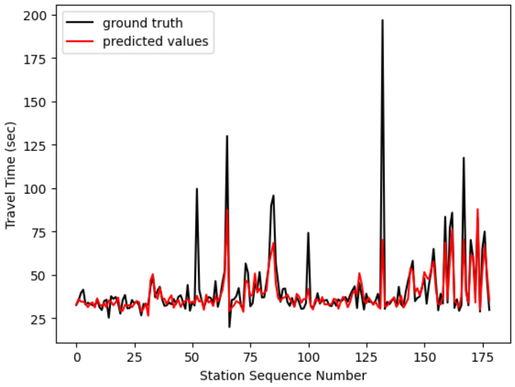


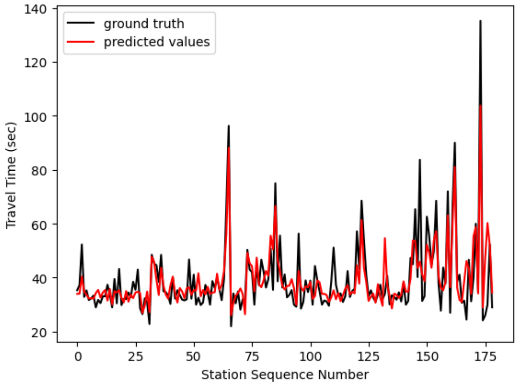

ground truth
predicted values
Travel Time (sec)
90
80
70
60
50
40
30
20
0
25
50
75
100
125
150
175
Station Sequence Number

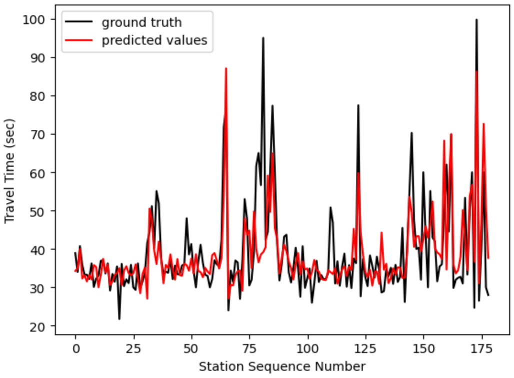

ground truth
predicted values
Travel Time (sec)
120
100
80
60
40
0
25
50
75
100
125
150
175
Station Sequence Number

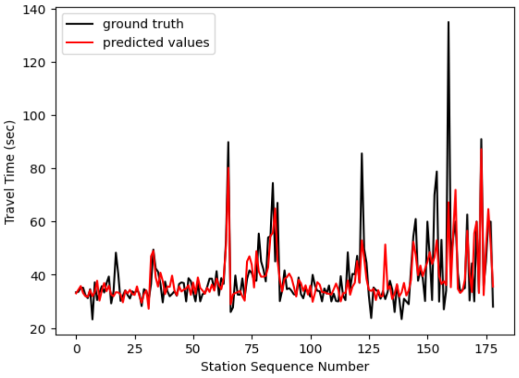

ground truth
predicted values
Travel Time (sec)
180
160
140
120
100
80
60
40
20
0
25
50
75
100
125
150
175
Station Sequence Number

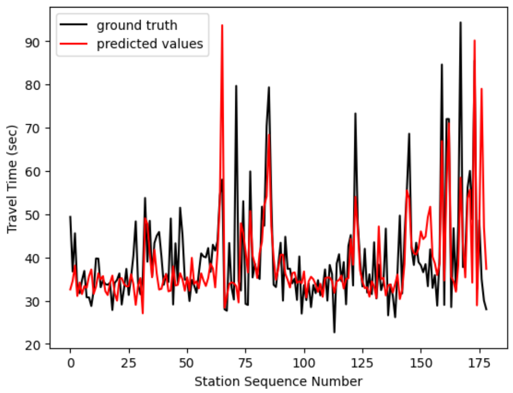

ground truth
predicted values
Travel Time (sec)
140
120
100
80
60
40
0
25
50
75
100
125
150
175
Station Sequence Number

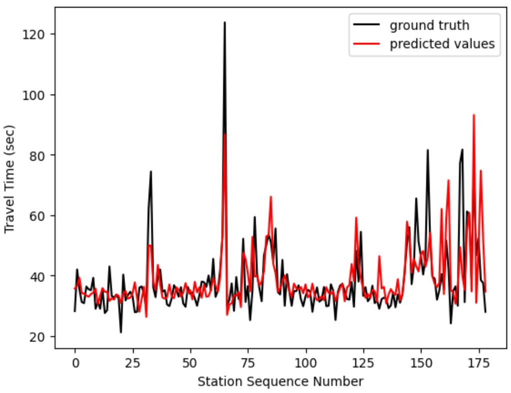

ground truth
predicted values
Travel Time (sec)
200
175
150
125
100
75
50
25
0
25
50
75
100
125
150
175
Station Sequence Number

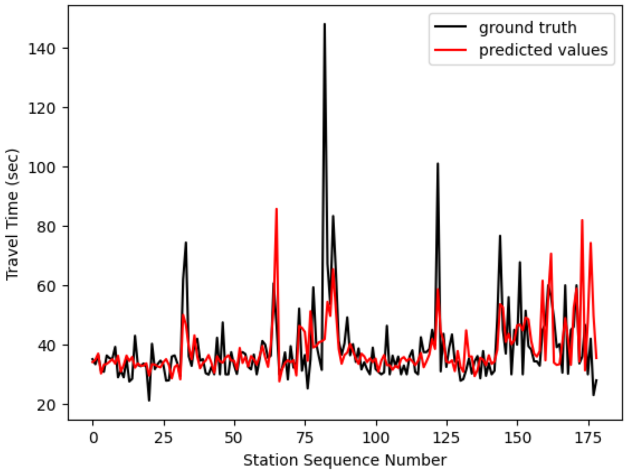

ground truth
predicted values
Travel Time (sec)
140
120
100
80
60
40
20
0
25
50
75
100
125
150
175
Station Sequence Number


ground truth
predicted values
Travel Time (sec)
100
90
80
70
60
50
40
30
20
0
25
50
75
100
125
150
175
Station Sequence Number

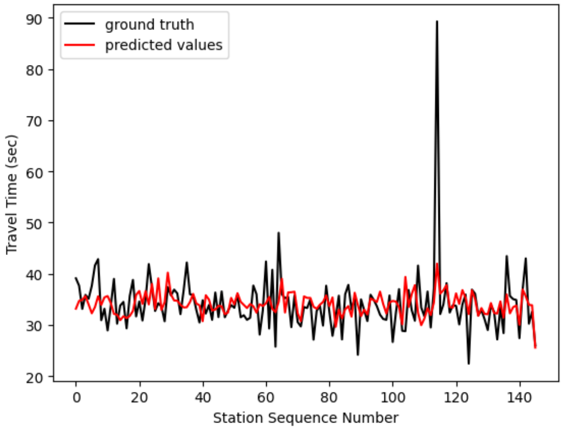

ground truth
predicted values
Travel Time (sec)
140
120
100
80
60
40
20
0
25
50
75
100
125
150
175
Station Sequence Number

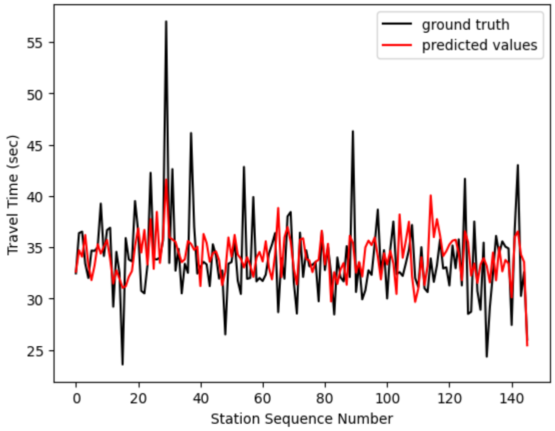


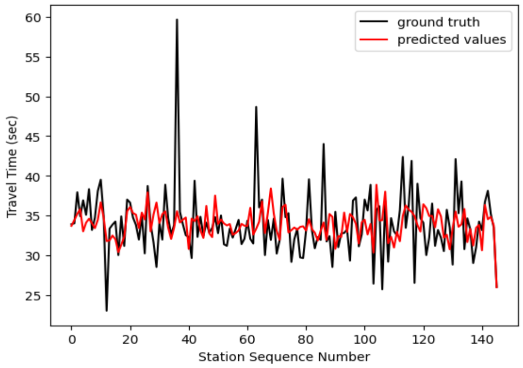


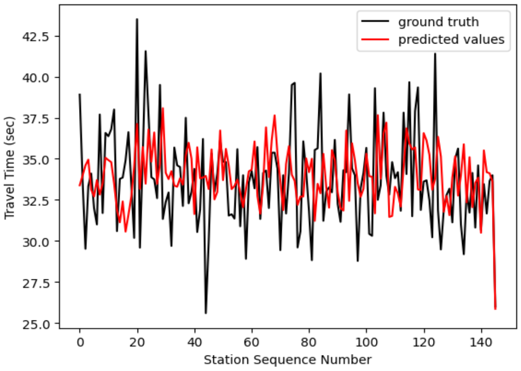


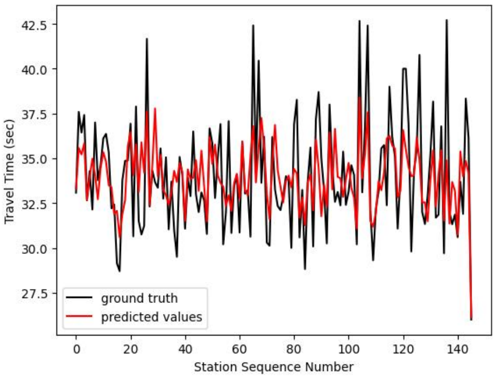
Transformer Model Predictions for 5th Hour to 23rd Hour – Route 18139

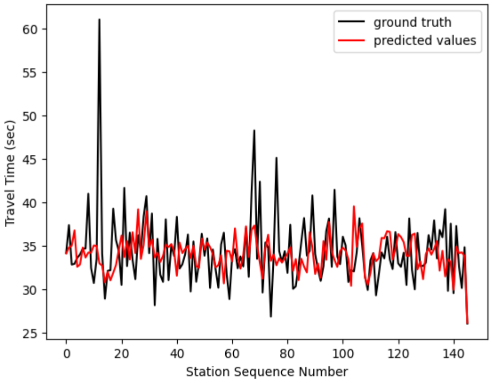


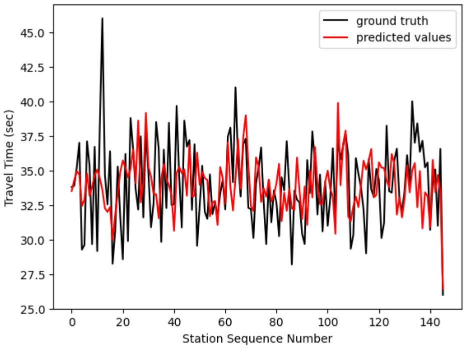


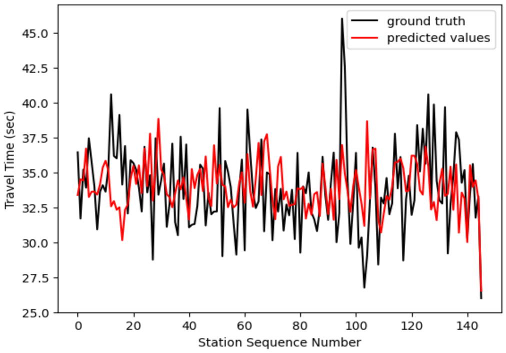

ground truth
predicted values
Travel Time (sec)
60
55
50
45
40
35
30
25
0
20
40
60
80
100
120
140
Station Sequence Number

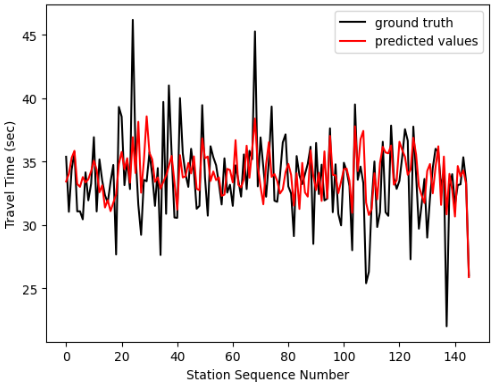

ground truth
predicted values
Travel Time (sec)
42.5
40.0
37.5
35.0
32.5
30.0
27.5
25.0
0
20
40
60
80
100
120
140
Station Sequence Number

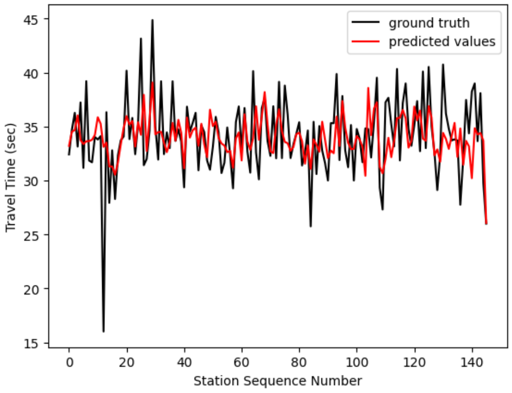

ground truth
predicted values
Travel Time (sec)
42.5
40.0
37.5
35.0
32.5
30.0
27.5
0
20
40
60
80
100
120
140
Station Sequence Number

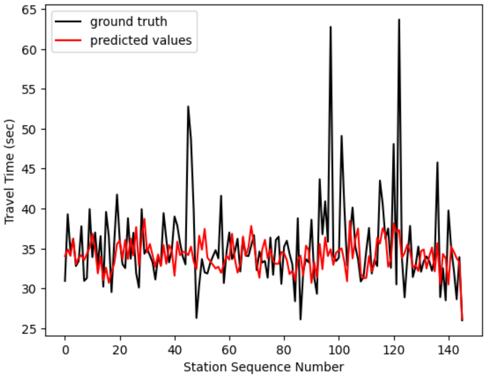

ground truth
predicted values
Travel Time (sec)
60
55
50
45
40
35
30
25
0
20
40
60
80
100
120
140
Station Sequence Number

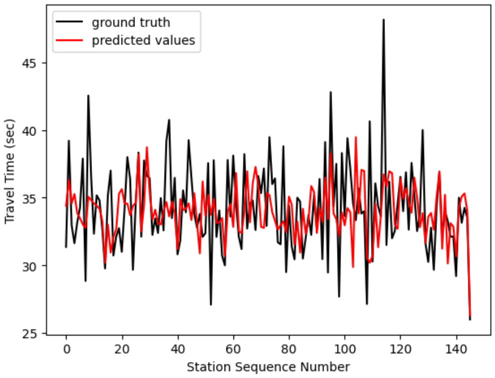

ground truth
predicted values
Travel Time (sec)
45.0
42.5
40.0
37.5
35.0
32.5
30.0
27.5
25.0
0
20
40
60
80
100
120
140
Station Sequence Number

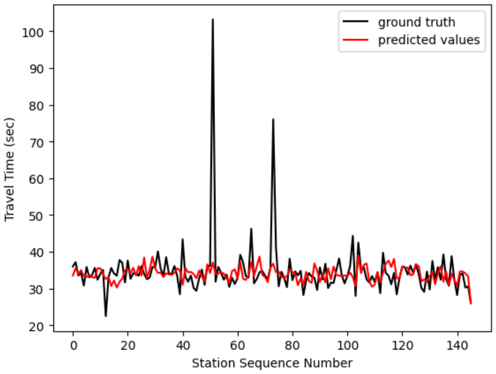

ground truth
predicted values
Travel Time (sec)
45.0
42.5
40.0
37.5
35.0
32.5
30.0
27.5
25.0
0
20
40
60
80
100
120
140
Station Sequence Number

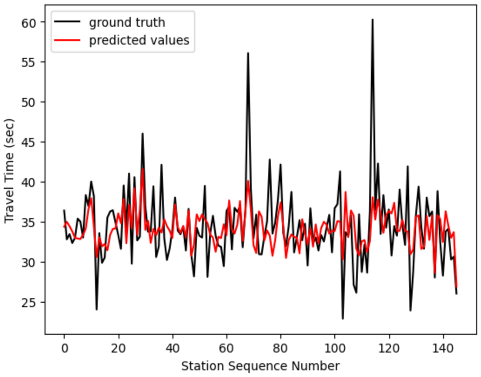

ground truth
predicted values
Travel Time (sec)
45
40
35
30
25
0
20
40
60
80
100
120
140
Station Sequence Number

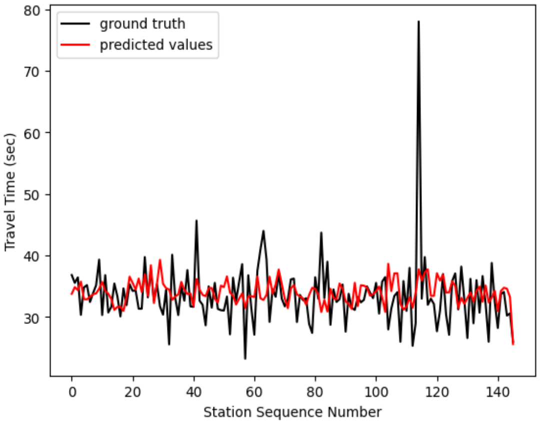

ground truth
predicted values
Travel Time (sec)
45
40
35
30
25
20
15
0
20
40
60
80
100
120
140
Station Sequence Number

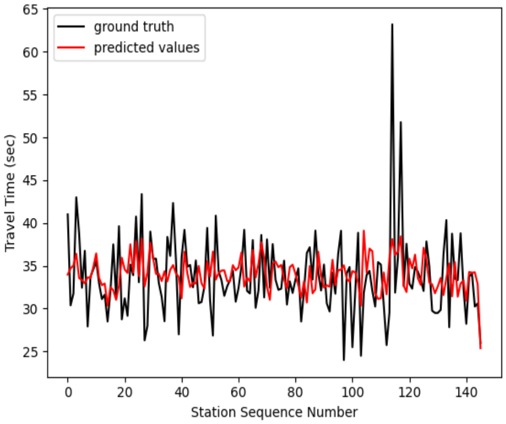

ground truth
predicted values
Travel Time (sec)
65
60
55
50
45
40
35
30
25
0
20
40
60
80
100
120
140
Station Sequence Number

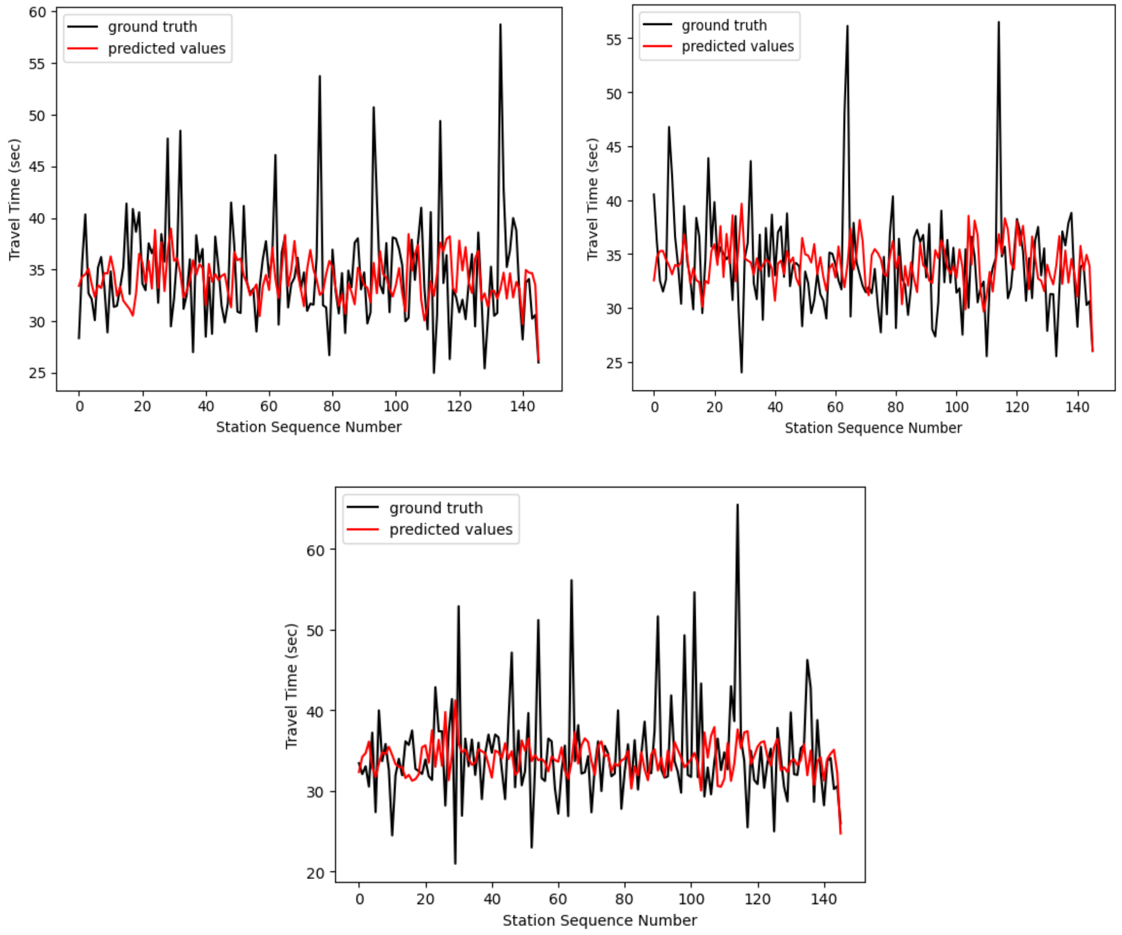

ground truth
predicted values
Travel Time (sec)
45
40
35
30
25
0
20
40
60
80
100
120
140
Station Sequence Number


ground truth
predicted values
Travel Time (sec)
100
90
80
70
60
50
40
30
20
0
20
40
60
80
100
120
140
Station Sequence Number

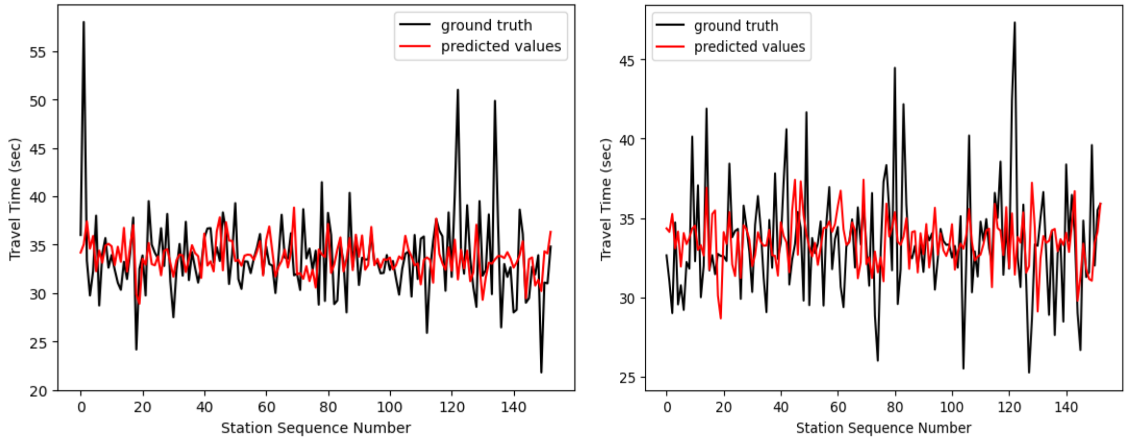

ground truth
predicted values
Travel Time (sec)
60
55
50
45
40
35
30
25
0
20
40
60
80
100
120
140
Station Sequence Number

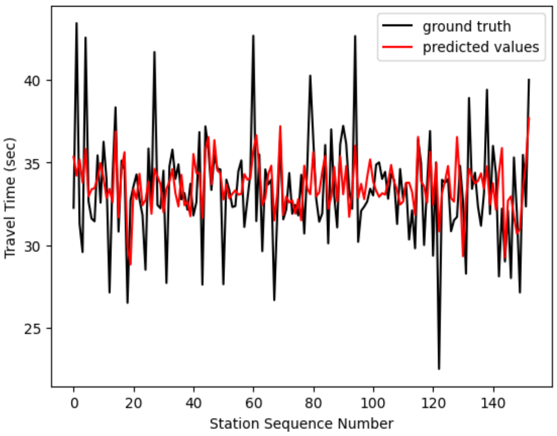

ground truth
predicted values
Travel Time (sec)
80
70
60
50
40
30
0
20
40
60
80
100
120
140
Station Sequence Number

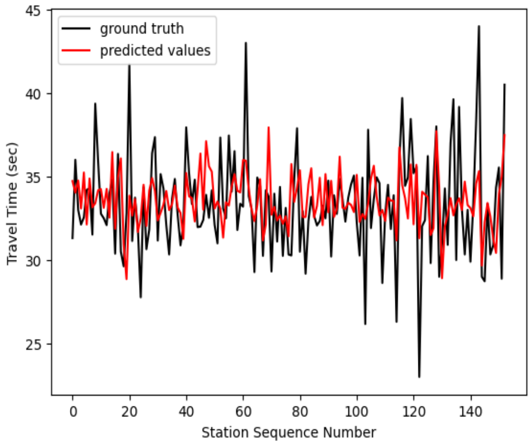

ground truth
predicted values
Travel Time (sec)
65
60
55
50
45
40
35
30
25
0
20
40
60
80
100
120
140
Station Sequence Number

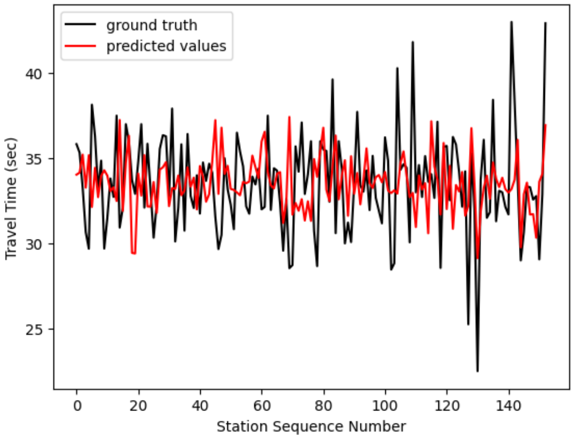


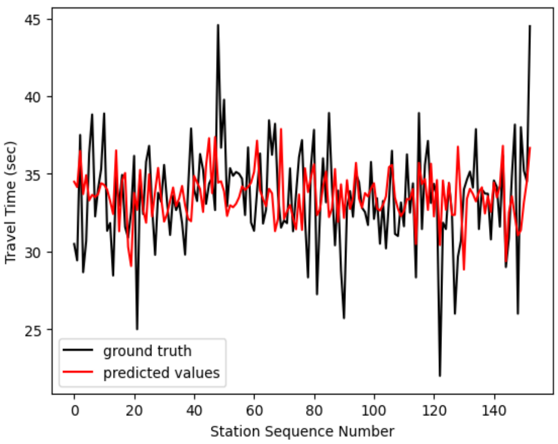
Transformer Model Predictions for 5th Hour to 23rd Hour – Route 18144

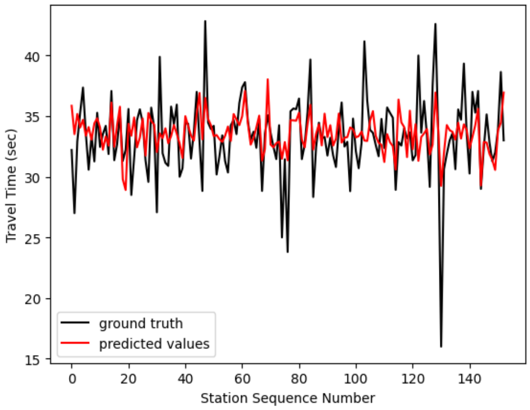


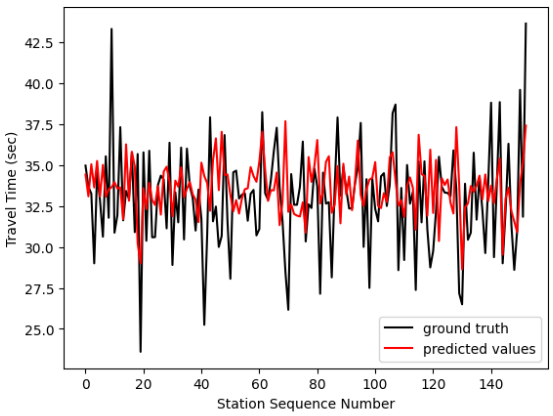

ground truth
predicted values
Travel Time (sec)
25
30
35
40
0
20
40
60
80
100
120
140
Station Sequence Number

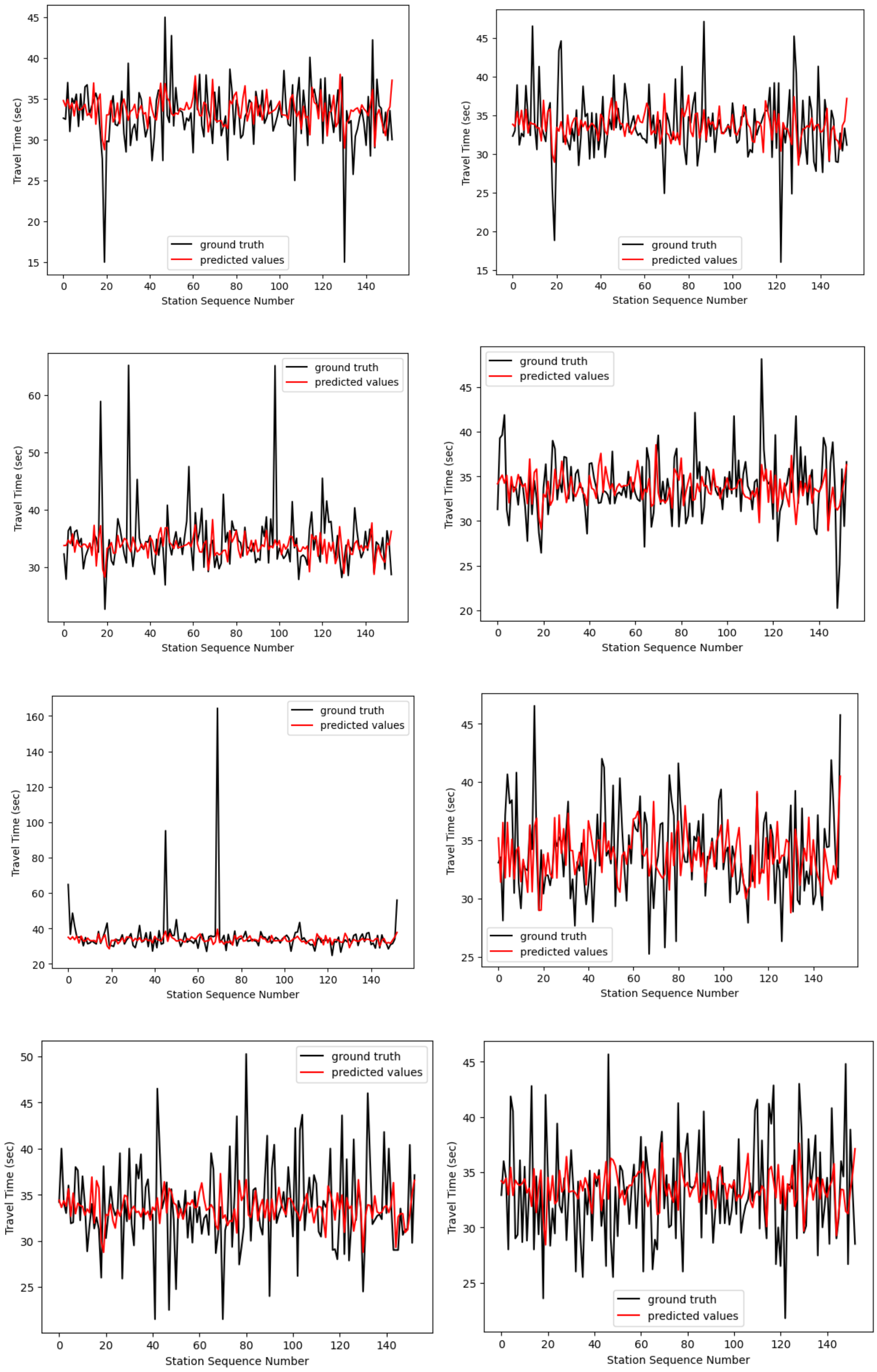

ground truth
predicted values
Travel Time (sec)
25
30
35
40
45
0
20
40
60
80
100
120
140
Station Sequence Number

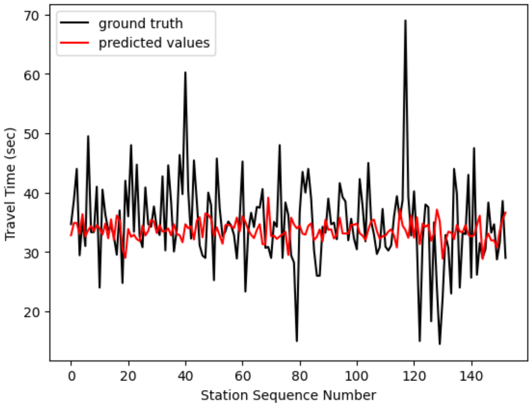

ground truth
predicted values
Travel Time (sec)
25
30
35
40
0
20
40
60
80
100
120
140
Station Sequence Number

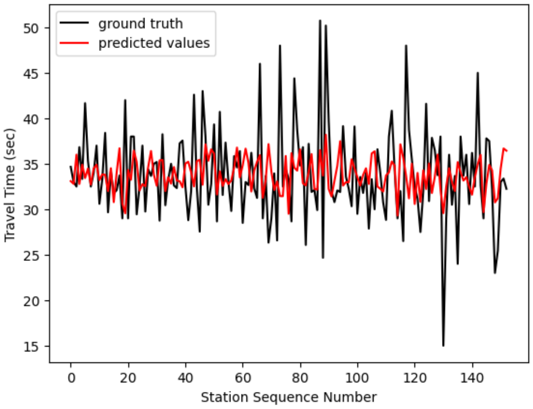

ground truth
predicted values
Travel Time (sec)
25
30
35
40
45
0
20
40
60
80
100
120
140
Station Sequence Number

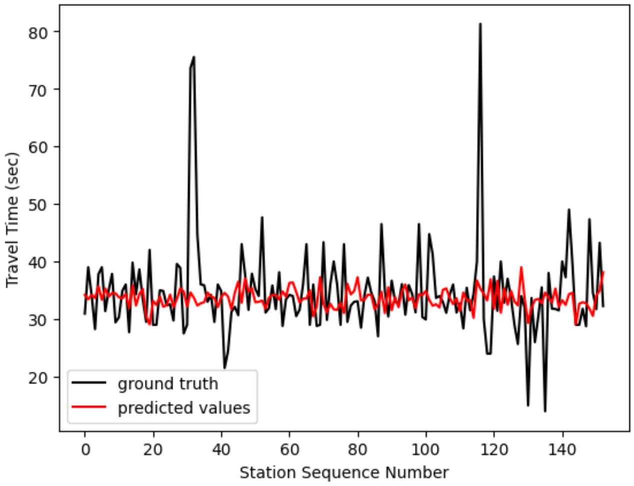

ground truth
predicted values
Travel Time (sec)
15
20
25
30
35
40
0
20
40
60
80
100
120
140
Station Sequence Number


ground truth
predicted values
Travel Time (sec)
25.0
27.5
30.0
32.5
35.0
37.5
40.0
42.5
0
20
40
60
80
100
120
140
Station Sequence Number

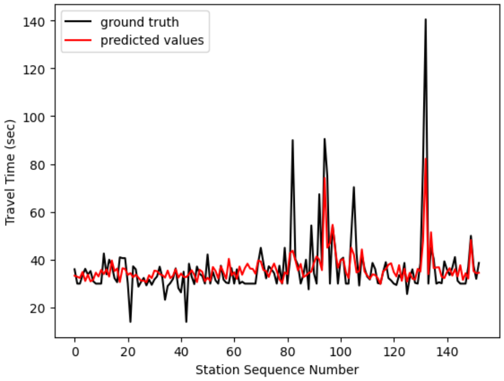

Travel Time (sec)
Station Sequence Number
ground truth
predicted values
15
20
25
30
35
40
45
0
20
40
60
80
100
120
140
30
40
50
60
160
140
120
100
80
60
40
20
50

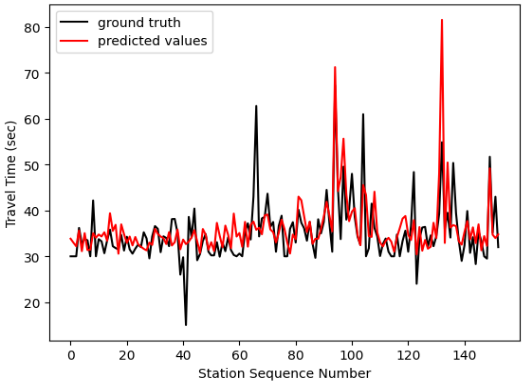


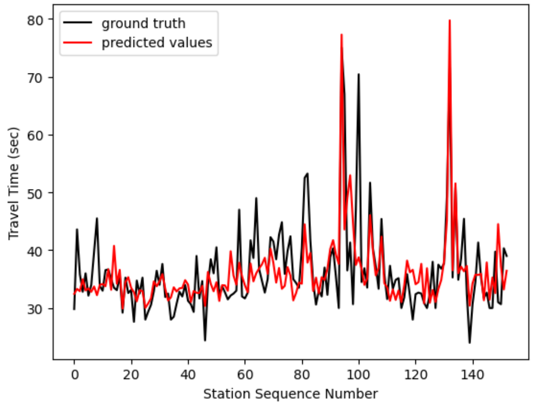


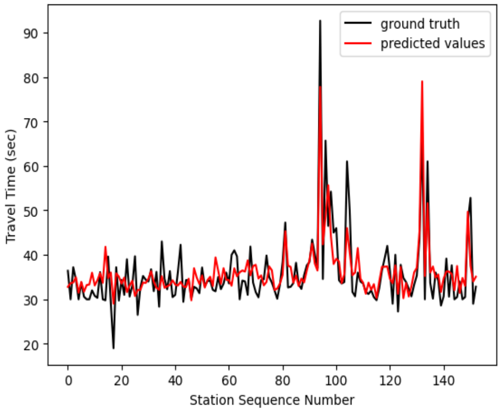


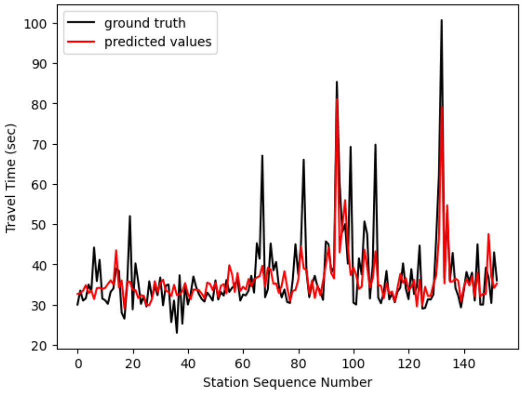
Transformer Model Predictions for 5th Hour to 23rd Hour – Route 18145

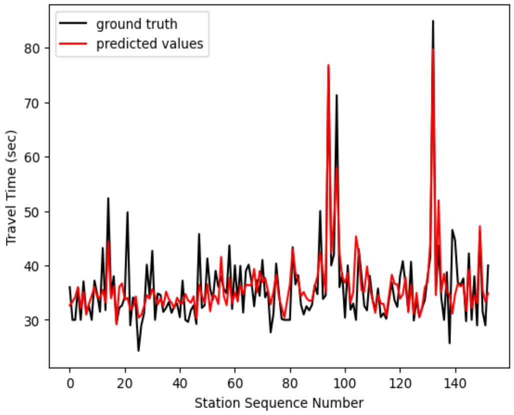


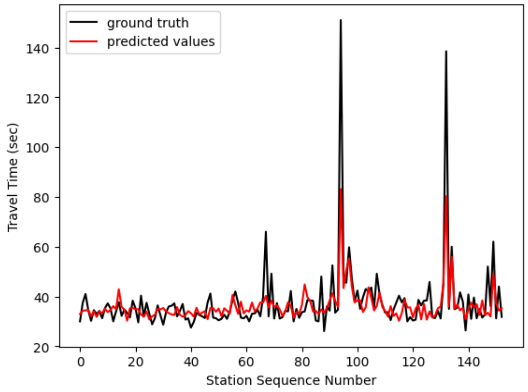


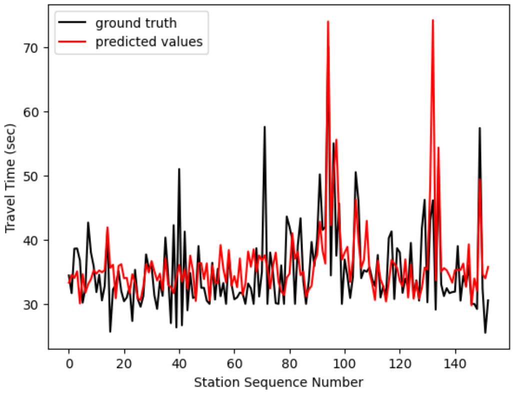

ground truth
predicted values
Travel Time (sec)
80
70
60
50
40
30
0
20
40
60
80
100
120
140
Station Sequence Number

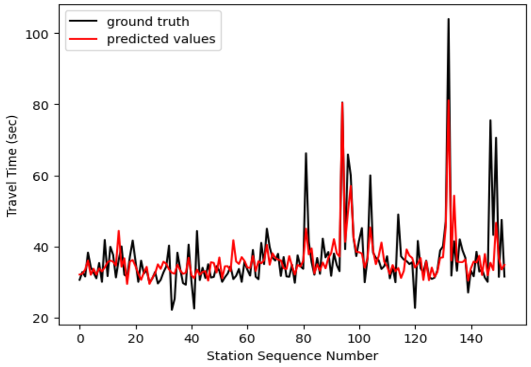

ground truth
predicted values
Travel Time (sec)
90
80
70
60
50
40
30
20
0
20
40
60
80
100
120
140
Station Sequence Number

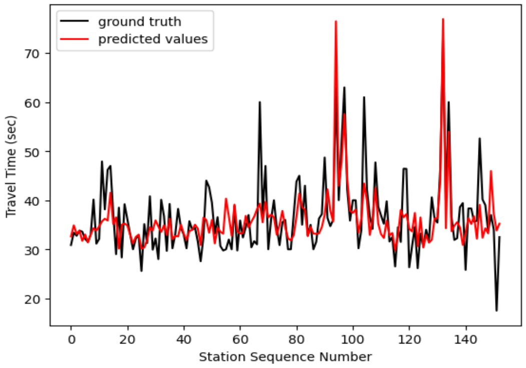

ground truth
predicted values
Travel Time (sec)
100
90
80
70
60
50
40
30
20
0
20
40
60
80
100
120
140
Station Sequence Number

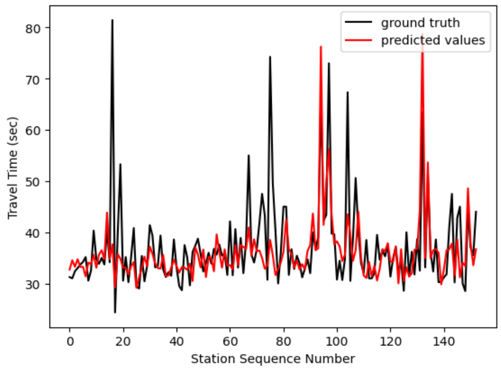

ground truth
predicted values
Travel Time (sec)
80
70
60
50
40
30
0
20
40
60
80
100
120
140
Station Sequence Number

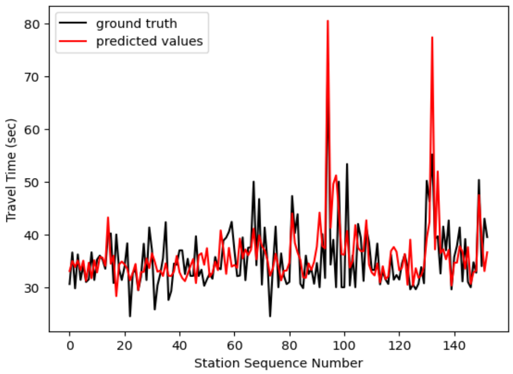

ground truth
predicted values
Travel Time (sec)
140
120
100
80
60
40
20
0
20
40
60
80
100
120
140
Station Sequence Number

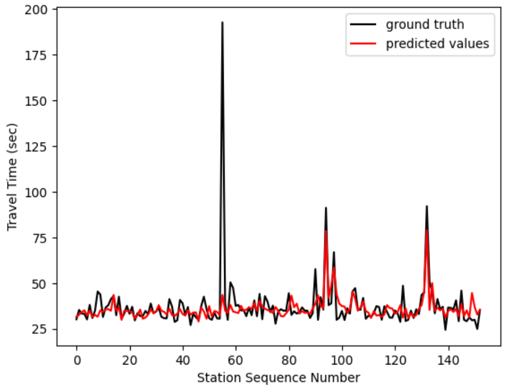

ground truth
predicted values
Travel Time (sec)
70
60
50
40
30
0
20
40
60
80
100
120
140
Station Sequence Number

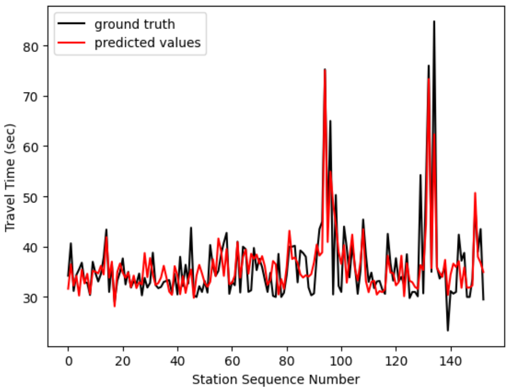

ground truth
predicted values
Travel Time (sec)
100
80
60
40
20
0
20
40
60
80
100
120
140
Station Sequence Number

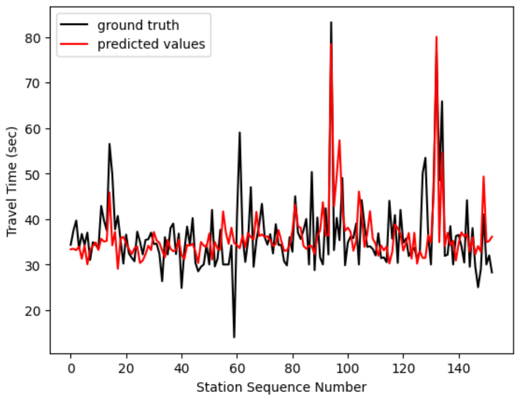

ground truth
predicted values
Travel Time (sec)
70
60
50
40
30
20
0
20
40
60
80
100
120
140
Station Sequence Number

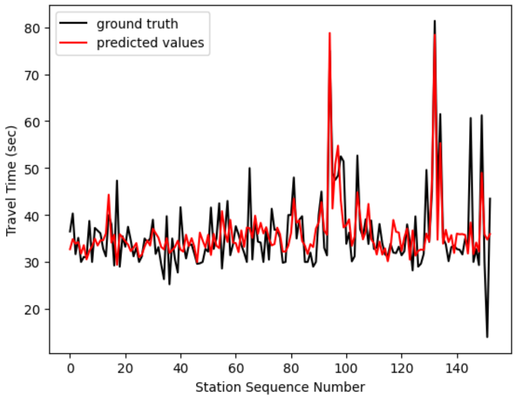

ground truth
predicted values
Travel Time (sec)
80
70
60
50
40
30
0
20
40
60
80
100
120
140
Station Sequence Number

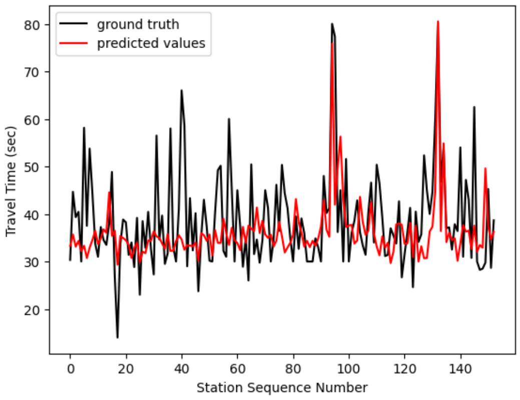

ground truth
predicted values
Travel Time (sec)
80
70
60
50
40
30
0
20
40
60
80
100
120
140
Station Sequence Number

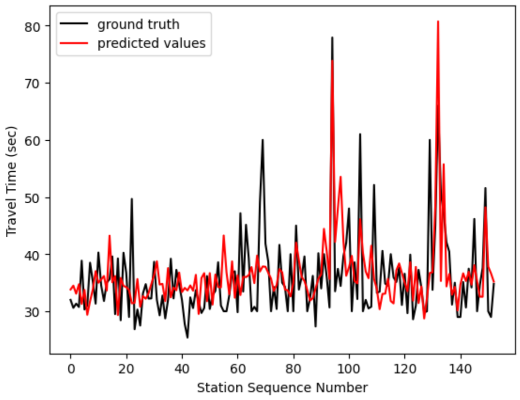

ground truth
predicted values
Travel Time (sec)
200
175
150
125
100
75
50
25
0
20
40
60
80
100
120
140
Station Sequence Number

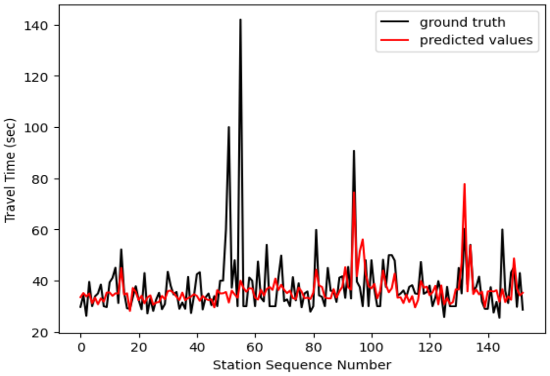

ground truth
predicted values
Travel Time (sec)
80
70
60
50
40
30
0
20
40
60
80
100
120
140
Station Sequence Number


ground truth
predicted values
Travel Time (sec)
80
70
60
50
40
30
20
0
20
40
60
80
100
120
140
Station Sequence Number

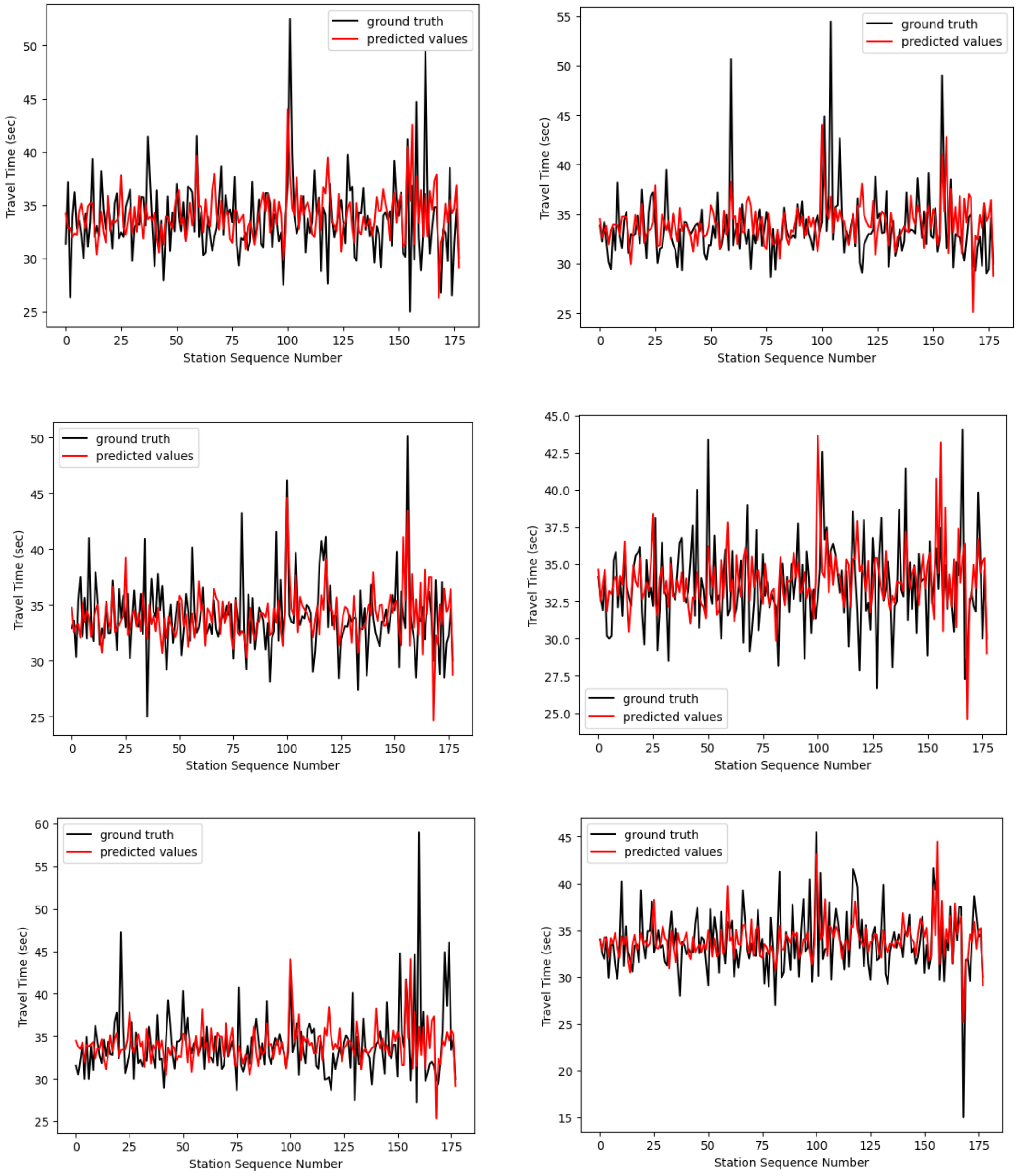

ground truth
predicted values
Travel Time (sec)
80
70
60
50
40
30
20
0
20
40
60
80
100
120
140
Station Sequence Number

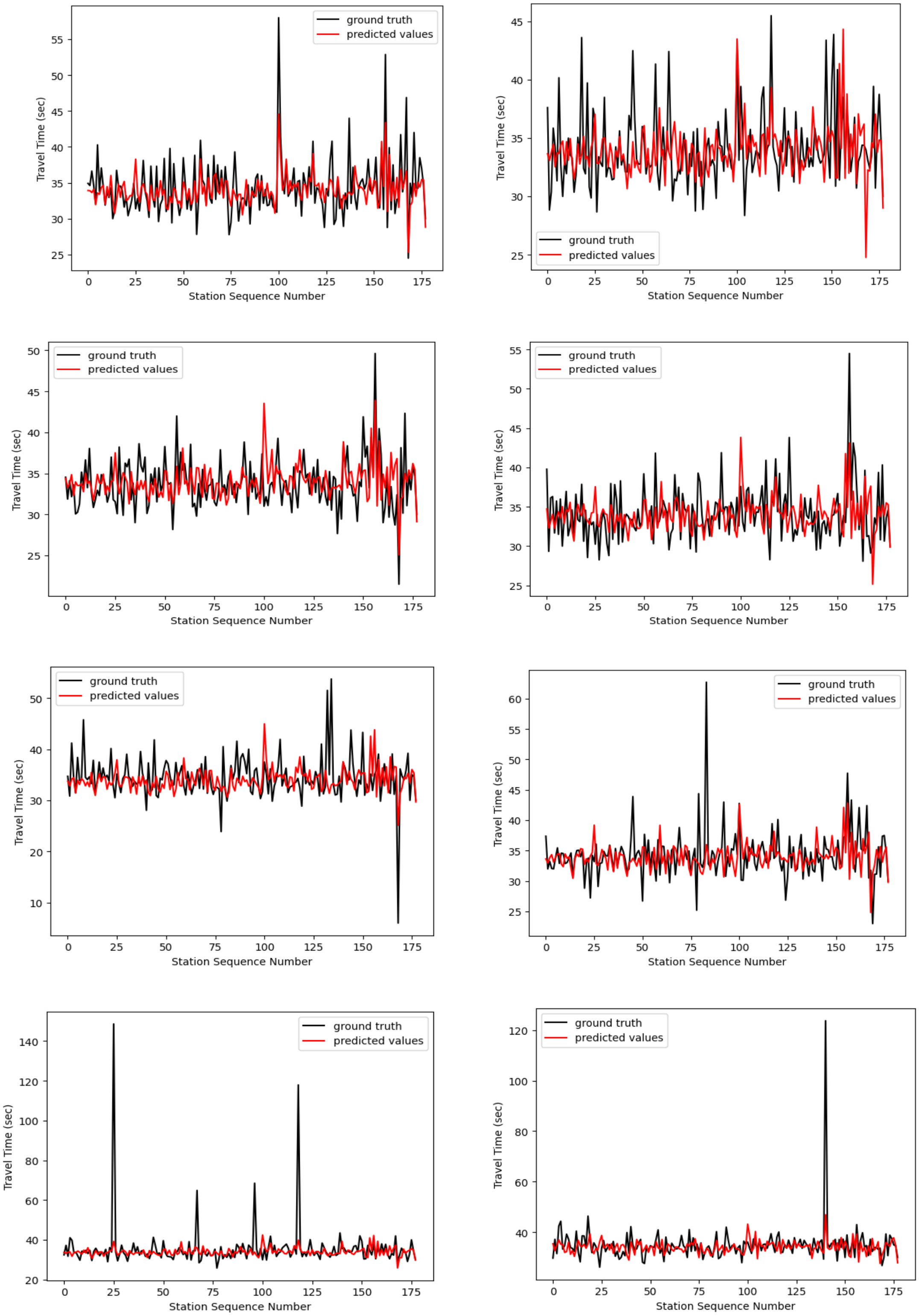

ground truth
predicted values
Travel Time (sec)
80
70
60
50
40
30
20
0
20
40
60
80
100
120
140
Station Sequence Number

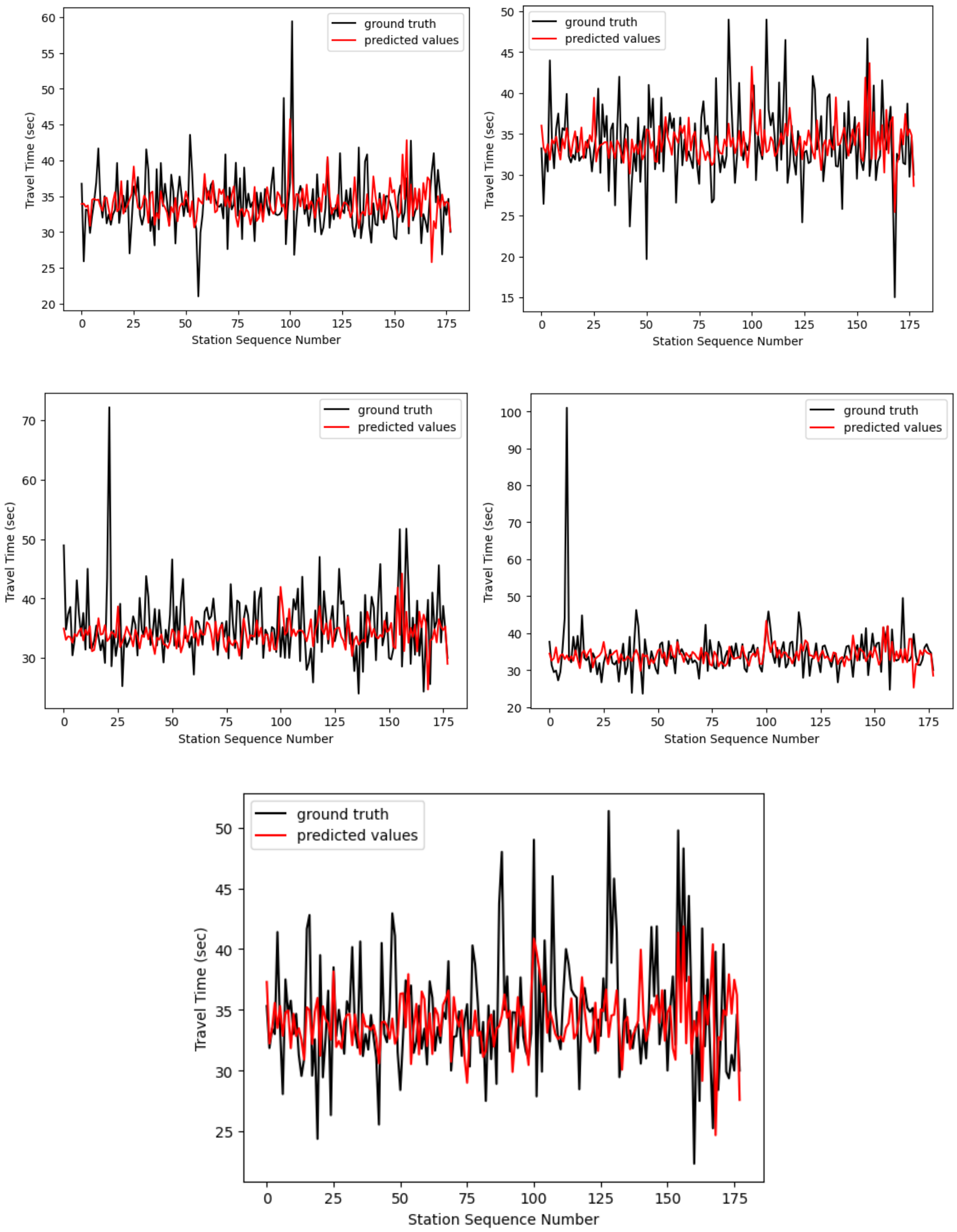

ground truth
predicted values
Travel Time (sec)
80
70
60
50
40
30
0
20
40
60
80
100
120
140
Station Sequence Number



Transformer Model Predictions for 5th Hour to 23rd Hour – Route 18146




ground truth
predicted values
Travel Time (sec)
Station Sequence Number
25
30
35
40
45
50
55
0
25
50
75
100
125
150
175
ground truth
predicted values
Travel Time (sec)
Station Sequence Number
25
30
35
40
45
0
25
50
75
100
125
150
175
ground truth
predicted values
Travel Time (sec)
Station Sequence Number
25
30
35
40
45
50
0
25
50
75
100
125
150
175
ground truth
predicted values
Travel Time (sec)
Station Sequence Number
25
30
35
40
45
50
55
0
25
50
75
100
125
150
175
ground truth
predicted values
Travel Time (sec)
Station Sequence Number
10
20
30
40
50
0
25
50
75
100
125
150
175
ground truth
predicted values
Travel Time (sec)
Station Sequence Number
25
30
35
40
45
50
55
60
0
25
50
75
100
125
150
175
ground truth
predicted values
Travel Time (sec)
Station Sequence Number
20
40
60
80
100
120
140
0
25
50
75
100
125
150
175
ground truth
predicted values
Travel Time (sec)
Station Sequence Number
40
60
80
100
120
0
25
50
75
100
125
150
175


ground truth
predicted values
Travel Time (sec)
Station Sequence Number
60
55
50
45
40
35
30
25
20
0
25
50
75
100
125
150
175
ground truth
predicted values
Travel Time (sec)
Station Sequence Number
50
45
40
35
30
25
20
15
0
25
50
75
100
125
150
175
ground truth
predicted values
Travel Time (sec)
Station Sequence Number
70
60
50
40
30
0
25
50
75
100
125
150
175
ground truth
predicted values
Travel Time (sec)
Station Sequence Number
100
90
80
70
60
50
40
30
20
0
25
50
75
100
125
150
175
ground truth
predicted values
Travel Time (sec)
Station Sequence Number
50
45
40
35
30
25
0
25
50
75
100
125
150
175

REFERENCES

[1] Fhwa, “Federal Highway Administration (FHWA) Research and Technology Update Newsletter, July 2022, Spring Issue.” doi: https://doi.org/10.21949/1521915.

[2] FHWA, “BUDGET ESTIMATES FISCAL YEAR 2023 FEDERAL HIGHWAY ADMINISTRATION SUBMITTED FOR THE USE OF THE COMMITTEES ON APPROPRIATIONS,” 2023.

[3] One STL, “OneSTL - Indicator_ Transit Ridership,” 2020, Accessed: Apr. 18, 2023. [Online]. Available: https://www.onestl.org/indicators/connected/metric/transit-ridership

[4] G. P. Griffin, M. Mulhall, C. Simek, and W. W. Riggs, “Mitigating Bias in Big Data for Transportation,” *Journal of Big Data Analytics in Transportation*, vol. 2, no. 1, pp. 49–59, Apr. 2020, doi: 10.1007/s42421-020-00013-0.

[5] A. Achar, R. Regikumar, and B. A. Kumar, “Dynamic Bus Arrival Time Prediction exploiting Non-linear Correlations,” in *2019 International Joint Conference on Neural Networks (IJCNN)*, IEEE, Jul. 2019, pp. 1–8. doi: 10.1109/IJCNN.2019.8852358.

[6] G. Huang, J. Xing, L. Meng, F. Li, and L. Ma, “Travel time prediction for Bus Rapid Transmit using a statistics-based probabilistic method,” in *2010 2nd International Conference on Signal Processing Systems*, IEEE, Jul. 2010, pp. V1-457-V1-460. doi: 10.1109/ICSPS.2010.5555535.

[7] Zychowski Adam, Junosza-Szaniawski Konstanty, and Kosicki Aleksander, *Proceedings of the 10th International Conference on Computer Recognition Systems CORES 2017*, vol. 578. Cham: Springer International Publishing, 2018. doi: 10.1007/978-3-319-59162-9.

[8] H. Zhang and B. Shi, “Travel Time of Buses Based on GPS Trajectory Data: Analysis and Prediction,” in *CICTP 2019*, Reston, VA: American Society of Civil Engineers, Jul. 2019, pp. 1172–1183. doi: 10.1061/9780784482292.104.

[9] A. Khamparia and R. Choudhary, “Prediction of Bus Arrival Time Using Intelligent Computing Methods,” in *Pervasive Computing: A Networking*

Perspective and Future Directions, Singapore: Springer Singapore, 2019, pp. 127–143. doi: 10.1007/978-981-13-3462-7_12.

[10] X. Zhang and Z. Liu, “Prediction of Bus Arrival Time based on GPS Data: Taking No. 6 Bus in Huangdao District of Qingdao City as an Example,” in *2019 Chinese Control Conference (CCC)*, IEEE, Jul. 2019, pp. 8789–8794. doi: 10.23919/ChiCC.2019.8866558.

[11] J. Liu and G. Xiao, “Efficient Bus Arrival Time Prediction Based on Spark Streaming Platform,” in *2019 IEEE 23rd International Conference on Computer Supported Cooperative Work in Design (CSCWD)*, IEEE, May 2019, pp. 416–421. doi: 10.1109/CSCWD.2019.8791859.

[12] D. Liu, J. Sun, and S. Wang, “BusTime: Which is the Right Prediction Model for My Bus Arrival Time?,” in *2020 5th IEEE International Conference on Big Data Analytics (ICBDA)*, IEEE, May 2020, pp. 180–185. doi: 10.1109/ICBDA49040.2020.9101265.

[13] A. Kakarla, V. S. K. R. Munagala, T. Ishizaka, A. Fukuda, and S. Jana, “Travel Time Prediction and Route Performance Analysis in BRTS based on Sparse GPS Data,” in *2021 IEEE 93rd Vehicular Technology Conference (VTC2021-Spring)*, IEEE, Apr. 2021, pp. 1–5. doi: 10.1109/VTC2021-Spring51267.2021.9448832.

[14] A. Khadhir, B. Anil Kumar, and L. D. Vanajakshi, “Analysis of global positioning system based bus travel time data and its use for advanced public transportation system applications,” *J Intell Transp Syst*, vol. 25, no. 1, pp. 58–76, Jan. 2021, doi: 10.1080/15472450.2020.1754818.

[15] C. Coghlan *et al.*, “Assigning Bus Delay and Predicting Travel Times using Automated Vehicle Location Data,” *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2673, no. 3, pp. 624–636, Mar. 2019, doi: 10.1177/0361198119832866.

[16] M. U. Farooq, A. Shakoor, and A. B. Siddique, “GPS based Public Transport Arrival Time Prediction,” in *2017 International Conference on Frontiers of Information Technology (FIT)*, IEEE, Dec. 2017, pp. 76–81. doi: 10.1109/FIT.2017.00021.

[17] Y. Zhou, L. Yao, Y. Chen, Y. Gong, and J. Lai, “Bus arrival time calculation model based on smart card data,” *Transp Res Part C Emerg Technol*, vol. 74, pp. 81–96, Jan. 2017, doi: 10.1016/j.trc.2016.11.014.

[18] P. Wepulanon, A. Sumalee, and W. H. K. Lam, “A real-time bus arrival time information system using crowdsourced smartphone data: a novel framework and simulation experiments,” *Transportmetrica B: Transport Dynamics*, vol. 6, no. 1, pp. 34–53, Jan. 2018, doi: 10.1080/21680566.2017.1353449.

[19] B. Dhivyabharathi, B. Anil Kumar, L. Vanajakshi, and M. Panda, “Particle Filter for Reliable Bus Travel Time Prediction Under Indian Traffic Conditions,” *Transportation in Developing Economies*, vol. 3, no. 2, p. 13, Oct. 2017, doi: 10.1007/s40890-017-0043-z.

[20] O. Alam, A. Kush, A. Emami, and P. Pouladzadeh, “Predicting irregularities in arrival times for transit buses with recurrent neural networks using GPS coordinates and weather data,” *J Ambient Intell Humaniz Comput*, vol. 12, no. 7, pp. 7813–7826, Jul. 2021, doi: 10.1007/s12652-020-02507-9.

[21] Z. Yu, J. S. Wood, and V. V. Gayah, “Using survival models to estimate bus travel times and associated uncertainties,” *Transp Res Part C Emerg Technol*, vol. 74, pp. 366–382, Jan. 2017, doi: 10.1016/j.trc.2016.11.013.

[22] M. Shoman, A. Aboah, and Y. Adu-Gyamfi, “Deep Learning Framework for Predicting Bus Delays on Multiple Routes Using Heterogenous Datasets,” *Journal of Big Data Analytics in Transportation*, vol. 2, no. 3, pp. 275–290, Dec. 2020, doi: 10.1007/s42421-020-00031-y.

[23] R. Barnes, S. Buthpitiya, J. Cook, A. Fabrikant, A. Tomkins, and F. Xu, “BusTr,” in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, New York, NY, USA: ACM, Aug. 2020, pp. 3243–3251. doi: 10.1145/3394486.3403376.

[24] T. Elliott and T. Lumley, “Modelling the travel time of transit vehicles in real-time through a GTFS-based road network using GPS vehicle locations,” *Aust N Z J Stat*, vol. 62, no. 2, pp. 153–167, Jun. 2020, doi: 10.1111/anzs.12294.

[25] A. Taparia and M. Brady, “Bus Journey and Arrival Time Prediction based on Archived AVL/GPS data using Machine Learning,” in *2021 7th International Conference on Models and Technologies for Intelligent Transportation Systems (MT-ITS)*, IEEE, Jun. 2021, pp. 1–6. doi: 10.1109/MT-ITS49943.2021.9529328.

[26] B. P. Ashwini, R. Sumathi, and H. S. Sudhira, “Bus Travel Time Prediction: A Comparative Study of Linear and Non-Linear Machine Learning Models,” in *Journal of Physics: Conference Series*, IOP Publishing Ltd, Jan. 2022. doi: 10.1088/1742-6596/2161/1/012053.

[27] B. Yu, H. Wang, W. Shan, and B. Yao, “Prediction of Bus Travel Time Using Random Forests Based on Near Neighbors,” *Computer-Aided Civil and Infrastructure Engineering*, vol. 33, no. 4, pp. 333–350, Apr. 2018, doi: 10.1111/mice.12315.

[28] S. I.-J. Chien, M. Asce, Y. Ding, and C. Wei, “Dynamic Bus Arrival Time Prediction with Artificial Neural Networks”, doi: 10.1061/ASCE0733-947X2002128:5429.

[29] N. Servos, X. Liu, M. Teucke, and M. Freitag, “Travel Time Prediction in a Multimodal Freight Transport Relation Using Machine Learning Algorithms,” *Logistics*, vol. 4, no. 1, p. 1, Dec. 2019, doi: 10.3390/logistics4010001.

[30] Y. Li, C. Huang, and J. Jiang, “Research of bus arrival prediction model based on GPS and SVM,” in *2018 Chinese Control And Decision Conference (CCDC)*, IEEE, Jun. 2018, pp. 575–579. doi: 10.1109/CCDC.2018.8407197.

[31] A. K. Bachu, K. K. Reddy, and L. Vanajakshi, “BUS TRAVEL TIME PREDICTION USING SUPPORT VECTOR MACHINES FOR HIGH VARIANCE CONDITIONS,” *Transport*, vol. 36, no. 3, pp. 221–234, Aug. 2021, doi: 10.3846/transport.2021.15220.

[32] Z. Peng, Y. Jiang, X. Yang, Z. Zhao, L. Zhang, and Y. Wang, “BUS ARRIVAL TIME PREDICTION BASED ON PCA-GA-SVM,” *Neural Network World*, vol. 28, no. 1, pp. 87–104, 2018, doi: 10.14311/NNW.2018.28.005.

[33] H. Chang, D. Park, S. Lee, H. Lee, and S. Baek, “Dynamic multi-interval bus travel time prediction using bus transit data,” *Transportmetrica*, vol. 6, no. 1, pp. 19–38, Jan. 2010, doi: 10.1080/18128600902929591.

[34] J. R, B. A. Kumar, S. S. Arkatkar, and L. Vanajakshi, “Performance Comparison of Bus Travel Time Prediction Models across Indian Cities,” *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2672, no. 31, pp. 87–98, Dec. 2018, doi: 10.1177/0361198118770175.

[35] B. A. Kumar, R. Jairam, S. S. Arkatkar, and L. Vanajakshi, “Real time bus travel time prediction using *k* -NN classifier,” *Transportation Letters*, vol. 11, no. 7, pp. 362–372, Jul. 2019, doi: 10.1080/19427867.2017.1366120.

[36] X. Zhang, L. Lauber, H. Liu, J. Shi, M. Xie, and Y. Pan, “Travel time prediction of urban public transportation based on detection of single routes,”

PLoS One, vol. 17, no. 1 January 2022, Jan. 2022, doi: 10.1371/journal.pone.0262535.

[37] B. A. Kumar, L. Vanajakshi, and S. C. Subramanian, "Pattern-based time-discretized method for bus travel time prediction," *J Transp Eng*, vol. 143, no. 6, Jun. 2017, doi: 10.1061/JTEPBS.0000029.

[38] A. Achar, D. Bharathi, B. A. Kumar, and L. Vanajakshi, "Bus Arrival Time Prediction: A Spatial Kalman Filter Approach," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 3, pp. 1298–1307, Mar. 2020, doi: 10.1109/TITS.2019.2909314.

[39] J. Cheng, G. Li, and X. Chen, "Research on Travel Time Prediction Model of Freeway Based on Gradient Boosting Decision Tree," *IEEE Access*, vol. 7, pp. 7466–7480, 2019, doi: 10.1109/ACCESS.2018.2886549.

[40] T. Kawatani, T. Yamaguchi, Y. Sato, R. Maita, and T. Mine, "Prediction of Bus Travel Time over Intervals between Pairs of Adjacent Bus Stops Using City Bus Probe Data," *International Journal of Intelligent Transportation Systems Research*, vol. 19, no. 2, pp. 456–467, Jun. 2021, doi: 10.1007/s13177-021-00251-8.

[41] A. S. M. M. Islam, M. Shirazi, and D. Lord, "Finite mixture Negative Binomial-Lindley for modeling heterogeneous crash data with many zero observations," *Accid Anal Prev*, vol. 175, p. 106765, Sep. 2022, doi: 10.1016/j.aap.2022.106765.

[42] T. Cristóbal, G. Padrón, A. Quesada-Arencibia, F. Alayón, G. de Blasio, and C. R. García, "Bus Travel Time Prediction Model Based on Profile Similarity," *Sensors*, vol. 19, no. 13, p. 2869, Jun. 2019, doi: 10.3390/s19132869.

[43] L. Ricard, G. Desaulniers, A. Lodi, and L.-M. Rousseau, "Predicting the probability distribution of bus travel time to measure the reliability of public transport services," *Transp Res Part C Emerg Technol*, vol. 138, p. 103619, May 2022, doi: 10.1016/j.trc.2022.103619.

[44] A. S. M. M. Islam, M. Shirazi, and D. Lord, "Grouped Random Parameters Negative Binomial-Lindley for accounting unobserved heterogeneity in crash data with preponderant zero observations," *Anal Methods Accid Res*, vol. 37, p. 100255, Mar. 2023, doi: 10.1016/j.amar.2022.100255.

[45] T. Idé and S. Kato, "Travel-Time Prediction using Gaussian Process Regression: A Trajectory-Based Approach," in *Proceedings of the 2009*

SIAM International Conference on Data Mining, Philadelphia, PA: Society for Industrial and Applied Mathematics, Apr. 2009, pp. 1185–1196. doi: 10.1137/1.9781611972795.101.

[46] I. K. Isukapati, C. Igoe, E. Bronstein, V. Parimi, and S. F. Smith, "Hierarchical Bayesian Framework for Bus Dwell Time Prediction," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 5, pp. 3068–3077, May 2021, doi: 10.1109/TITS.2020.2979390.

[47] B. Buchel and F. Corman, "Probabilistic Bus Delay Predictions with Bayesian Networks," in *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*, IEEE, Sep. 2021, pp. 3752–3758. doi: 10.1109/ITSC48978.2021.9564537.

[48] H. Xu and J. Ying, "Bus arrival time prediction with real-time and historic data," *Cluster Comput*, vol. 20, no. 4, pp. 3099–3106, Dec. 2017, doi: 10.1007/s10586-017-1006-1.

[49] P. Ranjitkar, L.-S. Tey, E. Chakravorty, and K. L. Hurley, "Bus Arrival Time Modeling Based on Auckland Data," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2673, no. 6, pp. 1–9, Jun. 2019, doi: 10.1177/0361198119840620.

[50] B. A. Kumar, L. Vanajakshi, and S. C. Subramanian, "Pattern-Based Time-Discretized Method for Bus Travel Time Prediction," *J Transp Eng A Syst*, vol. 143, no. 6, Jun. 2017, doi: 10.1061/JTEPBS.0000029.

[51] C. Bai, Z.-R. Peng, Q.-C. Lu, and J. Sun, "Dynamic Bus Travel Time Prediction Models on Road with Multiple Bus Routes," *Comput Intell Neurosci*, vol. 2015, pp. 1–9, 2015, doi: 10.1155/2015/432389.

[52] W. Treethidtaphat, W. Pattara-Atikom, and S. Khaimook, "Bus arrival time prediction at any distance of bus route using deep neural network model," in *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, IEEE, Oct. 2017, pp. 988–992. doi: 10.1109/ITSC.2017.8317891.

[53] Y. Yuan *et al.*, "Bus Dynamic Travel Time Prediction: Using a Deep Feature Extraction Framework Based on RNN and DNN," *Electronics (Basel)*, vol. 9, no. 11, p. 1876, Nov. 2020, doi: 10.3390/electronics9111876.

[54] P. He, G. Jiang, S.-K. Lam, and D. Tang, "Travel-Time Prediction of Bus Journey With Multiple Bus Trips," *IEEE Transactions on Intelligent*

Transportation Systems, vol. 20, no. 11, pp. 4192–4205, Nov. 2019, doi: 10.1109/TITS.2018.2883342.

[55] A. A. Agafonov and A. S. Yumaganov, “Bus Arrival Time Prediction Using Recurrent Neural Network with LSTM Architecture,” *Optical Memory and Neural Networks*, vol. 28, no. 3, pp. 222–230, Jul. 2019, doi: 10.3103/S1060992X19030081.

[56] K. Panyo, J. Bootkrajang, P. Inkeaw, and J. Chaijaruwanich, “Bus Arrival Time Estimation for Public Transportation System Using LSTM,” in *2020 - 5th International Conference on Information Technology (InCIT)*, IEEE, Oct. 2020, pp. 134–138. doi: 10.1109/InCIT50588.2020.9310940.

[57] H. Liu, H. Xu, Y. Yan, Z. Cai, T. Sun, and W. Li, “Bus Arrival Time Prediction Based on LSTM and Spatial-Temporal Feature Vector,” *IEEE Access*, vol. 8, pp. 11917–11929, 2020, doi: 10.1109/ACCESS.2020.2965094.

[58] S. Nadeeshan and A. S. Perera, “Multi-Step Bidirectional LSTM for Low Frequent Bus Travel Time Prediction,” in *2021 Moratuwa Engineering Research Conference (MERCon)*, IEEE, Jul. 2021, pp. 462–467. doi: 10.1109/MERCon52712.2021.9525709.

[59] P. K. Biswas, S. M. Khan, K. Piratla, and M. Chowdhury, “Development and Evaluation of Statistical and Machine-Learning Models for Queue-Length Estimation for Lane Closures in Freeway Work Zones,” *J Constr Eng Manag*, vol. 149, no. 5, May 2023, doi: 10.1061/JCEMD4.COENG-12648.

[60] J. Pang, J. Huang, Y. Du, H. Yu, Q. Huang, and B. Yin, “Learning to Predict Bus Arrival Time From Heterogeneous Measurements via Recurrent Neural Network,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 9, pp. 3283–3293, Sep. 2019, doi: 10.1109/TITS.2018.2873747.

[61] P. He, G. Jiang, S.-K. Lam, and Y. Sun, “Learning heterogeneous traffic patterns for travel time prediction of bus journeys,” *Inf Sci (N Y)*, vol. 512, pp. 1394–1406, Feb. 2020, doi: 10.1016/j.ins.2019.10.073.

[62] Y. Hou and P. Edara, “Network Scale Travel Time Prediction using Deep Learning,” *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2672, no. 45, pp. 115–123, Dec. 2018, doi: 10.1177/0361198118776139.

[63] N. C. Petersen, F. Rodrigues, and F. C. Pereira, “Multi-output bus travel time prediction with convolutional LSTM neural network,” *Expert Syst Appl*, vol. 120, pp. 426–435, Apr. 2019, doi: 10.1016/j.eswa.2018.11.028.

[64] Z.-Y. Xie, Y.-R. He, C.-C. Chen, Q.-Q. Li, and C.-C. Wu, “Multistep Prediction of Bus Arrival Time with the Recurrent Neural Network,” *Math Probl Eng*, vol. 2021, pp. 1–14, Mar. 2021, doi: 10.1155/2021/6636367.

[65] A. Khayyer, D. F. Silva, and A. Vinel, “Predicting Public Transit Arrival Times: A Hybrid Deep Neural Network Approach,” *Journal of Big Data Analytics in Transportation*, vol. 2, no. 3, pp. 291–305, Dec. 2020, doi: 10.1007/s42421-020-00032-x.

[66] J. Wu, Q. Wu, J. Shen, and C. Cai, “Towards Attention-Based Convolutional Long Short-Term Memory for Travel Time Prediction of Bus Journeys,” *Sensors*, vol. 20, no. 12, p. 3354, Jun. 2020, doi: 10.3390/s20123354.

[67] Y. Yuan *et al.*, “Bus Dynamic Travel Time Prediction: Using a Deep Feature Extraction Framework Based on RNN and DNN,” *Electronics (Basel)*, vol. 9, no. 11, p. 1876, Nov. 2020, doi: 10.3390/electronics9111876.

[68] B. A. Kumar, A. Achar, D. Bharathi, and L. Vanjakshi, “A Seasonal Modelling Approach Capturing Spatio-Temporal Correlations for Dynamic Bus Travel Time Prediction,” in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, IEEE, Oct. 2019, pp. 503–508. doi: 10.1109/ITSC.2019.8917055.

[69] H. E. Shaji, A. K. Tangirala, and L. Vanajakshi, “Prediction Of Trends In Bus Travel Time Using Spatial Patterns,” *Transportation Research Procedia*, vol. 48, pp. 998–1007, 2020, doi: 10.1016/j.trpro.2020.08.128.

[70] G. Lee, S. Choo, S. Choi, and H. Lee, “Does the Inclusion of Spatio-Temporal Features Improve Bus Travel Time Predictions? A Deep Learning-Based Modelling Approach,” *Sustainability*, vol. 14, no. 12, p. 7431, Jun. 2022, doi: 10.3390/su14127431.

[71] A. Vaswani *et al.*, “Attention Is All You Need,” Jun. 2017.