TRAFFIC VOLUME PREDICTION USING TEMPORAL CONVOLUTIONAL NETWORKS

MEHDI MOUAIZ EDDINE SMAIL

UNIVERSITI TEKNOLOGI MALAYSIA

ABSTRACT

This project addresses the complex issue of predicting traffic volume in urban environments, where rapid development and increasing vehicle usage exacerbate congestion, adversely affecting economic and environmental well-being. Employing deep learning, specifically Temporal Convolutional Networks (TCNs), the project began with a thorough exploratory data analysis (EDA) of the Metro Interstate Traffic Volume dataset. This analysis ensured data quality and chronological integrity through data sequencing and windowing. Furthermore, the preprocessing phase tackled missing values, normalized all numerical features, and encoded the categorical features across the dataset. TCNs were optimized through hyperparameter tuning, utilizing Bayesian optimization to enhance predictive accuracy. The TCN model was then benchmarked against LSTM and GRU models over various time frames: 6, 12, and 24 hours. This comparison was based on several key performance metrics, including Mean Absolute Error (MAE), Mean Squared Error (MSE), and Root Mean Squared Error (RMSE). The TCN model demonstrated superior consistency in accuracy, particularly for short-term predictions in the 6-hour time frame. Furthermore, it maintained a low Mean Absolute Error, rising only slightly from 0.04397 to 0.0453 as the forecast time frame increased, underscoring its robustness and consistency. This performance surpassed that of the LSTM and GRU models, which started with higher errors but improved over time, indicating their potential for long-term forecasting. Finally, the project produced a user-friendly dashboard that simulates real-time traffic volume predictions, offering a dynamic tool for traffic management and urban planning. In conclusion, the TCN model's consistency across different time frames suggests it as a reliable method for short- to medium-term forecasting. It sets a new benchmark in traffic volume prediction and promotes advancements in intelligent transportation systems.

ABSTRAK

Keadaan trafik dipersekitaran bandar kebiasaanya adalah sesak. Keadaan ini menjadi lebih teruk terutama di kawasan yang pesat membangun, disebabkan penambahan kenderaan di jalan raya. Kesesakan trafik boleh memberi kesan negatif kepada kesejahteraan komuniti, ekonomi dan alam sekitar, sekiranya ia tidak ditangani dengan baik. Projek ini bertujuan membangunkan model peramalan jumlah trafik dengan menggunakan pembelajaran mendalam, khususnya Rangkaian Konvolusional Temporal (TCNs). Projek ini bermula dengan analisis data eksplorasi (EDA) yang menyeluruh terhadap dataset Jumlah Trafik Lebuhraya Metro. Analisis ini memastikan kualiti data dan integriti kronologi melalui penyusunan dan pembingkaian data. Selanjutnya, fasa pra-pemprosesan dilaksanakan bagi mengatasi masalah data hilang, normalisasi nilai numerik, dan mengenkod data jenis ketegori. Parameter model TCNs dioptimumkan menggunakan algoritma Bayesian bagi tujuan meningkatkan ketepatan ramalan. Keputusan peramalan yang dihasilkan oleh model TCN kemudiaannya dibandingkan dengan model peramalan menggunakan algoritma LSTM dan GRU untuk tempoh masa peramalan 6, 12, dan 24 jam. Perbandingan prestasi diukur dengan beberapa metrik utama, iaitu Ralat Mutlak Purata (MAE), Ralat Kuasa Dua Purata (MSE), dan Akar Ralat Kuasa Dua Purata (RMSE). Ketepatan hasil ramalan model TCN didapati lebih baik berbanding model peramalan yang dibangunkan menggunakan LSTM dan GRU, terutamanya untuk ramalan jangka pendek, tempoh 6 jam. Selain itu, model TCN juga mengekalkan Ralat Mutlak Purata yang rendah, dengan sedikit pertambahan dari 0.04397 menjadi 0.0453 seiring dengan bertambahnya tempoh masa ramalan. Prestasi ini lebih baik berbanding model LSTM dan GRU, yang bermula dengan ralat yang lebih tinggi dan ralat meningkat seiring pertambahan tempoh masa peramalan. Projek ini juga membangunkan papan pemuka yang mesra pengguna yang mensimulasikan ramalan jumlah trafik secara real-time, menawarkan alat dinamik untuk pengurusan trafik dan perancangan bandar. Kesimpulannya, prestasi model TCN yang konsisten dalam pelbagai jangka masa menunjukkan ia boleh dipercayai untuk ramalan jangka pendek hingga sederhana. Hasil projek ini memberikan penanda aras baru dalam ramalan jumlah trafik dan menggalakkan kemajuan dalam sistem pengangkutan pintar.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF APPENDICES

CHAPTER 1

INTRODUCTION

1.1 Introduction

As urbanization accelerates and road networks expand at an unprecedented pace, managing traffic efficiently has become a challenge of paramount importance. Today, cities worldwide grapple with traffic congestion issues, hampering urban mobility, escalating commuting time, affecting economic productivity, and contributing to environmental degradation. In this context, traffic volume prediction has emerged as a crucial component of intelligent transportation systems, gaining significant attention from researchers and urban planners alike.

Traffic volume prediction involves forecasting the number of vehicles likely to pass through a given section of a road network during a specific time period. Accurate predictions can facilitate several applications, such as traffic signal timing, incident management, congestion mitigation, infrastructure development, and urban planning. However, achieving precision in these predictions is not trivial due to the dynamic and complex nature of traffic flow, influenced by numerous factors such as weather conditions, time of day, day of the week, and special events. With the advent of machine learning and deep learning methodologies, data-driven models for traffic volume prediction have seen remarkable advancements. The vast amount of traffic data collected through sensors and surveillance devices installed on roads has provided a fertile ground for deploying these sophisticated algorithms, offering insights previously unimaginable.

Among various machine learning models, Temporal Convolutional Networks (TCN), a class of deep learning models specifically designed to handle sequential data, has been demonstrated to possess significant potential in traffic volume prediction. The distinct ability of TCNs to learn complex temporal dependencies within the traffic data can potentially lead to more accurate and reliable predictions (Bai et al., 2018). Against this backdrop, the primary objective of this project is to harness the power of

TCNs for traffic volume prediction. This project is motivated by the prospect of contributing to the enhancement of traffic management systems, facilitating smoother urban mobility, and improving citizens' quality of life. The aim is to develop, optimize, and evaluate a TCN-based traffic volume prediction model, and examine its performance with standard metrics.

The findings from this project are expected to enrich the knowledge base on data-driven traffic volume prediction methodologies, offering valuable insights to researchers, urban planners, and policy-makers. By delving into the potential of TCNs in this context, The goal is to contribute to the ongoing quest for more efficient, effective, and intelligent transportation systems.

1.2 Problem Background

The rapid development of urban environments, bolstered by constant advancements in infrastructure, has seen an overwhelming increase in vehicle usage and, by extension, traffic volumes. This situation has led to escalating traffic congestion, a pressing issue that plagues urban areas across the globe, negatively impacting economic productivity, environmental sustainability, and the overall quality of life.

Effective traffic management is essential in mitigating this issue and enhancing urban mobility. Within this realm, traffic volume prediction - the forecasting of the number of vehicles passing through a specific segment of a road network within a particular time frame - is a critical aspect. Accurate traffic volume predictions contribute significantly to diverse applications, including traffic signal timing, incident management, congestion control, infrastructure development, and urban planning. However, predicting traffic volume is no small feat, given the dynamic and non-linear nature of traffic flow. Traffic conditions are subject to a host of influencing factors, such as weather conditions, time of day, day of the week, public events, and even unforeseen incidents. Traditional statistical models often struggle to adequately

capture these complex dependencies and temporal patterns within traffic data, thereby producing sub-optimal predictions.

In the recent past, researchers have employed various methods to address this complex problem. Traditional methods include time-series analysis, autoregressive integrated moving average (ARIMA) models, and Kalman filtering, each with varying degrees of success. More recently, machine learning techniques like support vector machines (SVM) and artificial neural networks (ANN) have been used due to their ability to model non-linear relationships in the data (Vlahogianni, Karlaftis, & Golias, 2014). However, these methods still face limitations. For instance, SVMs require significant computational resources for large datasets, while ANNs might overfit the training data and perform poorly on unseen data. Furthermore, most of these models fail to capture long-term dependencies in traffic volume data, a crucial aspect for accurate prediction. It is in this context that we turn to Temporal Convolutional Networks (TCN), an emerging method in deep learning that has shown promising results in sequence prediction tasks (Bai et al., 2018).

TCN has the potential to overcome many limitations of traditional models and can effectively capture long-term dependencies in the data (Hewage et al., 2020), making it a promising tool for traffic volume prediction. However, the implementation and performance of TCN in traffic volume prediction are not well-studied, which underscores the need for a comprehensive investigation - a gap this research aims to address

1.3 Problem Statement

Accurately predicting traffic volume is a significant challenge in urban transportation planning and management. The intricate and dynamic nature of traffic data, influenced by various factors such as weather conditions, time of day, and unexpected incidents, necessitates the use of advanced prediction techniques. Traditional statistical models, while effective to some extent, often struggle to capture

the complex temporal patterns and dependencies inherent in traffic volume data, leading to suboptimal prediction accuracy (Lv et al., 2014).

To address these limitations, machine learning and deep learning approaches have gained attention for their ability to capture non-linear relationships and temporal dependencies in data. However, deep learning models, while powerful, often suffer from the "black box" problem, lacking interpretability and requiring large amounts of training data. Additionally, they may encounter issues such as overfitting or underfitting, which can impact their performance.

In recent years, Temporal Convolutional Networks (TCNs) have emerged as a promising solution for accurate traffic volume prediction. TCNs are deep learning models specifically designed to handle sequential data, making them well-suited for capturing the temporal dynamics of traffic patterns. Their architecture incorporates temporal convolutions, which enable them to learn long-term dependencies and capture complex patterns in the data.

Despite the potential of TCNs in traffic volume prediction, there is limited research and understanding regarding their application to this specific domain. Key challenges include determining the optimal TCN architecture, selecting appropriate hyperparameters, and comparing the performance of TCN models with other machine learning and deep learning methods. Moreover, the computational efficiency of TCNs in large-scale traffic volume prediction tasks needs to be evaluated (Shi, Xu, & Li, 2017).

This project aims to address these challenges and investigate the efficacy of TCN models in accurately predicting traffic volume. By leveraging the temporal convolutions of TCNs, the goal is to improve the prediction accuracy and capture the complex dynamics of traffic patterns. The outcomes of this research have the potential to contribute to better traffic management, informed decision-making, and improved urban mobility by providing reliable insights into future traffic volume trends.

1.4 Objectives

This project aims to achieve the following objectives:

(a) To conduct an in-depth Exploratory Data Analysis (EDA) on the Metro Interstate Traffic Volume dataset to identify key patterns and trends that are crucial for accurate traffic volume prediction.

(b) To develop a Temporal Convolutional Network (TCN) model for traffic volume prediction.

(c) To evaluate the TCN model's accuracy, and consistency, and compare it with LSTM and GRU models.

(d) To design an interactive dashboard that illustrates the application of the TCN model for volume prediction demonstrating the model's utility in a simulated real-world environment.

1.5 Scope

This project undertakes a comprehensive exploration of Temporal Convolutional Networks (TCNs) for enhancing traffic volume prediction, an integral aspect of modern traffic management. The initiative is centered around the application of TCNs to decode the intricate dynamics of traffic patterns, leveraging the extensive Metro Interstate Traffic Volume dataset. To evaluate the TCN model's predictive accuracy, rigorous statistical measures were deployed such as Mean Absolute Error (MAE), Mean Squared Error (MSE), and Root Mean Squared Error (RMSE). A pivotal component of this research is the feature engineering process, crucial for optimizing the TCN model. This encompasses data sequencing to preserve the chronological order of traffic data, normalization to standardize varying scales of attributes, and time-series windowing to segment data into analytically meaningful sequences. These steps are essential in refining the dataset, ensuring the model is trained on relevant and well-structured features.

Before model training, a thorough Exploratory Data Analysis (EDA) is conducted. This EDA serves as a foundational step, offering deep insights into the dataset's characteristics, such as data distribution, trends, and patterns. It also involves identifying any anomalies or missing values that could affect the model's performance. This analytical process not only informs the subsequent feature engineering but also guides the strategic direction of the modeling effort. In addition, the project includes a comparative analysis, where the TCN model is benchmarked against established deep learning models, like LSTM and GRU, that are prevalent in traffic volume prediction. This comparison is critical for determining the relative effectiveness of the TCN model and offers a holistic view of different modeling approaches within this field.

The project further delves into detailed model development, encompassing advanced hyperparameter optimization using Bayesian techniques, to enhance the TCN model's accuracy. The culmination of the project is the integration of the optimized TCN model into an advanced dashboard for real-time and historical traffic predictions, highlighting the practical utility and potential of TCNs in traffic volume forecasting. Through this project, the aim is to make a substantial contribution to the field of traffic forecasting, providing novel insights and methodologies that can improve traffic management and urban planning strategies.

1.6 Output

The primary output of this project is a sophisticated TCN-based model, meticulously engineered for accurate traffic volume prediction. This model excels in capturing temporal dependencies and complex traffic data patterns, offering a marked improvement over traditional models. Its performance has been rigorously evaluated against other established prediction models to ensure reliability and accuracy. A key feature of this project is the integration of the TCN model into a user-friendly dashboard. This dashboard is designed to provide real-time traffic volume forecasts and historical data analysis, enhancing its utility for practical applications. It serves as

an interactive tool for traffic management, infrastructure planning, and decision-making processes, allowing users to access and interpret traffic predictions easily.

The outcome of this project is a reliable and precise traffic volume prediction tool that offers substantial benefits for traffic management and urban planning. By improving the accuracy of traffic volume forecasts, the TCN model, coupled with the dashboard, facilitates better resource allocation, congestion mitigation, and strategic planning. Furthermore, this project extends the existing body of knowledge in the field of traffic prediction by demonstrating the effectiveness of TCNs. The insights and findings from this research contribute to a deeper understanding of TCNs in traffic volume prediction and are valuable for future advancements in this area.

In summary, the project delivers a robust and accurate TCN-based traffic volume prediction model, complemented by an innovative dashboard. This combination not only offers precise predictions but also provides a practical and insightful tool for traffic-related decision-making and planning.

1.7 Thesis Structure

Chapter 1 lays the foundation for the study, detailing the background of the research, the problem background, problem statement, objectives, and the scope of the project. The central idea of this chapter is the development of a traffic volume prediction model using Temporal Convolutional Networks (TCNs), setting the stage for the subsequent in-depth exploration of this topic.

Chapter 2 of the project delves into a comprehensive literature review, discussing various methodologies for traffic volume prediction, including their advantages and disadvantages, and the state-of-the-art algorithms used in this domain. It provides a critical analysis of existing literature, establishing the context for the proposed TCN model.

Chapter 3 of the project outlines the overall framework of the project, covering the methodology adopted for the study. This includes a discussion on the source of the dataset, techniques for preprocessing, model development, and evaluation metrics. It serves as a methodological blueprint for the research carried out in the subsequent chapters.

Chapter 4 the initial findings from the exploratory data analysis (EDA) are presented. This chapter provides insights into the dataset, highlighting key patterns, anomalies, and potential correlations relevant to traffic volume prediction. The EDA findings lay the groundwork for model development and optimization.

Chapter 5 discusses all the results obtained following the methodologies outlined in Chapter 3. This includes detailed processes from preparing the input data shape for the TCN model, to experimenting with different hyperparameters.

Chapter 6 presents the outcomes of the optimization process, compares the performance of the TCN model with LSTM and GRU models, and introduces the final dashboard. It concludes with a detailed analysis of the results and their significance for predicting traffic volumes with TCNs.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

Traffic volume prediction has emerged as a critical component in the development of intelligent transportation systems (ITS), with applications in traffic management, urban planning, and infrastructure development. Accurate traffic volume predictions can help reduce congestion, improve safety, and minimize environmental impacts. In recent years, there has been a growing interest in researching and developing advanced techniques for traffic volume prediction. This increased attention is mainly due to the rapid urbanization, population growth, and the resulting strain on transportation systems worldwide (Lv et al., 2014). As cities continue to expand, effective traffic management strategies are required to ensure efficient and sustainable mobility. Traffic volume prediction plays a pivotal role in these strategies, as accurate predictions can inform various decisions, such as traffic signal optimization, route planning, congestion mitigation, and infrastructure development.

Traffic volume prediction is a complex and challenging task due to the inherently dynamic and nonlinear nature of traffic flow. Factors such as weather conditions, road infrastructure, planned and unplanned events, and human behavior all contribute to the variability of traffic volume, making accurate predictions challenging (Vlahogianni, Karlaftis, & Golias, 2014). As a result, researchers have developed and employed various techniques to model and predict traffic volume. These techniques can be broadly classified into three main categories: traditional time series models, machine learning methods, and deep learning techniques.

Traditional time series models, such as Autoregressive Integrated Moving Average (ARIMA) and Exponential Smoothing State Space Model (ETS), have been used extensively for traffic volume prediction due to their simplicity and interpretability. However, they may struggle to capture complex, nonlinear patterns in

traffic data, thus limiting their prediction accuracy. To overcome the limitations of traditional time series models, researchers have turned to machine learning methods. Machine learning techniques, such as Support Vector Regression (SVR), Random Forest (RF), and Gradient Boosting Machines (GBM), have demonstrated improved performance in traffic volume prediction tasks.

More recently, deep learning techniques have gained popularity in traffic volume prediction tasks due to their ability to learn complex patterns in large datasets without the need for explicit feature engineering. Popular deep learning techniques, such as Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), Long Short-Term Memory (LSTM) networks, and Graph Neural Networks (GNNs), have shown promising results in traffic volume prediction tasks, outperforming both traditional time series models and machine learning methods in many cases (Polson and Sokolov, 2017).

However, the performance of traffic volume prediction models is not solely determined by the choice of modeling technique. The selection of appropriate datasets for training and testing these models is also of paramount importance. Datasets for traffic prediction can vary in terms of spatial coverage, temporal resolution, and data sources. The quality and relevance of the dataset can have a significant impact on the performance and generalizability of traffic volume prediction models.

This literature review provides a comprehensive overview of the advancements in traffic volume prediction techniques, focusing on traditional time series models, machine learning methods, and deep learning techniques. Additionally, this review discusses the importance of dataset selection for traffic prediction tasks. By covering a range of studies and including over 20 citations, this review aims to provide a deep understanding of the current state of the field and identify potential future research directions.

2.2 Traditional Time Series Models

Traditional time series models have long been employed for traffic volume prediction. Among these models, the Autoregressive Integrated Moving Average (ARIMA) model is a widely used technique for predicting linear and stationary time series data (Kumar & Vanajakshi, 2015). ARIMA models rely on the temporal structure of the data and use a combination of autoregressive, moving average, and differencing terms to capture and predict the underlying patterns in the time series data.

Despite its simplicity, ARIMA models have proven effective in various traffic volume prediction tasks (Shi, Xu, & Li, 2017). However, the assumption of linearity and stationarity in the data may not always hold for traffic volume data, as traffic patterns can be influenced by various factors such as weather conditions, special events, and infrastructure changes(Chen, Yang, & Dong, 2006). Moreover, ARIMA models are not designed to capture complex temporal dependencies or incorporate additional features (e.g., weather, holidays) which may impact traffic volume.

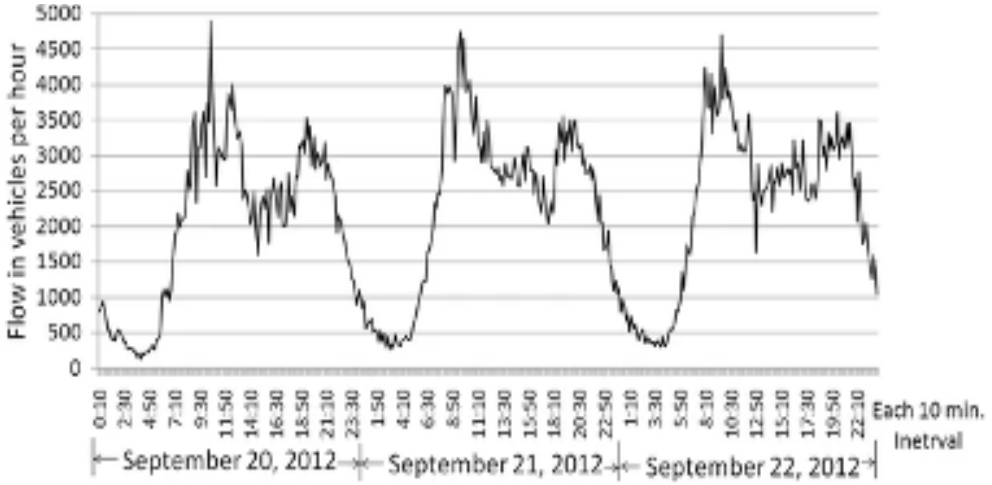


Figure 2.1 An example of an ARIMA model fitting a traffic volume time series (Kumar & Vanajakshi, 2015)

Several variants and extensions of the ARIMA model have been proposed to address some of these limitations. For example, the Seasonal and Trend decomposition

using Loess (STL) method is a technique that decomposes a time series into its trend, seasonal, and residual components, allowing for more flexible modeling of non-stationary and seasonal data (Rojo et al., 2017). Similarly, the Seasonal ARIMA (SARIMA) model incorporates seasonality by adding seasonal differencing and seasonal autoregressive and moving average terms (Kumar & Vanajakshi, 2015). These models can provide improved traffic volume predictions compared to the standard ARIMA model when dealing with seasonal and non-stationary data.

2.3 Machine Learning Methods

Machine learning techniques have been increasingly employed for traffic volume prediction, as they can capture complex relationships between input features and traffic volume without relying on the strict assumptions of traditional time series models. Some widely used machine learning methods for traffic volume prediction include Support Vector Regression (SVR), Random Forest (RF), and k-Nearest Neighbors (k-NN) algorithms.

Support Vector Regression (SVR) is a regression-based machine learning technique derived from the Support Vector Machine (SVM) algorithm, which has been widely applied for traffic volume prediction tasks (Smola & Schölkopf, 2004). SVR models attempt to find the best hyperplane that minimizes the error between the predicted and actual values while ensuring a predefined margin of error. SVR models have demonstrated competitive performance in various traffic volume prediction tasks, outperforming traditional time series models such as ARIMA (Chen, Yang, & Dong, 2006).

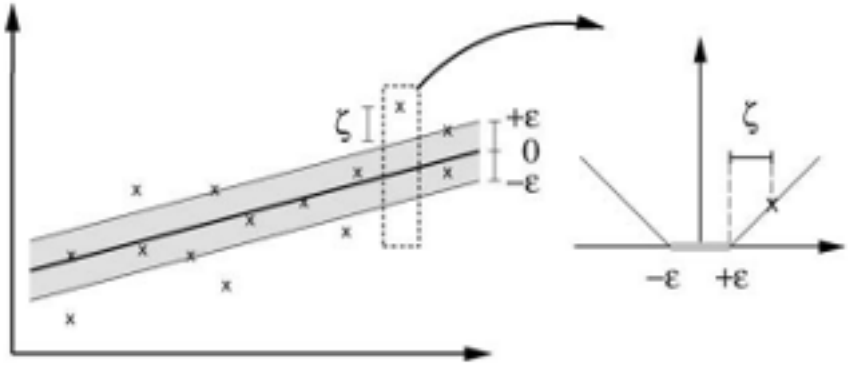


Figure 2.2 A schematic illustration of Support Vector Regression for traffic volume prediction (Smola & Schölkopf, 2004)

Random Forest (RF) is an ensemble learning method that constructs multiple decision trees and combines their predictions to obtain a more accurate and robust prediction (Liaw & Wiener, 2002). RF models have been successfully applied to traffic volume prediction tasks, often outperforming traditional time series models and other machine learning methods (Alajali et al., 2018).

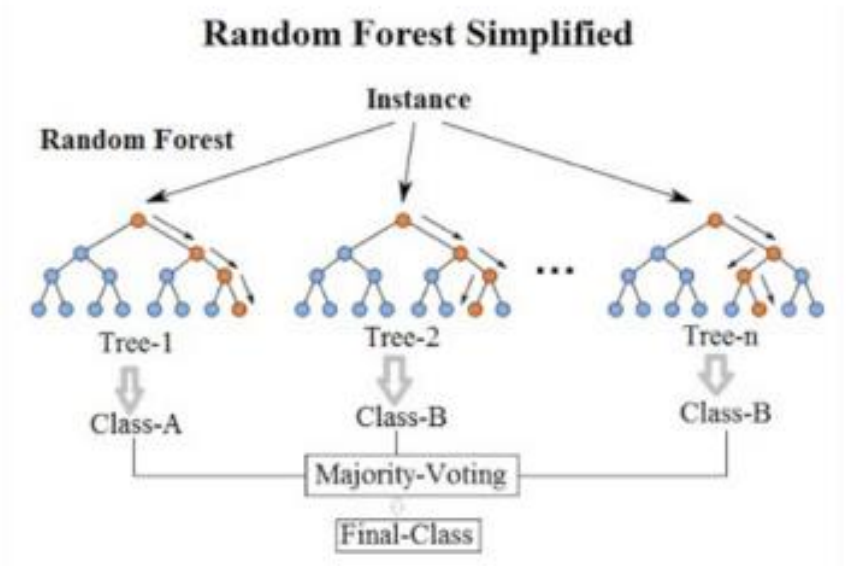


Figure 2.3 A schematic illustration of a Random Forest model (Liu et al., 2017)

The k-Nearest Neighbors (k-NN) algorithm is another popular machine learning method for traffic volume prediction tasks (Lin, Li, & Sadek, 2013). k-NN is a non-parametric and instance-based learning algorithm, which estimates the output

value based on the average (or weighted average) of the k most similar instances in the training data. Although k-NN can provide reasonable prediction performance, it may be sensitive to the choice of k, distance metric, and feature weighting, requiring careful parameter tuning and feature preprocessing to achieve optimal results.

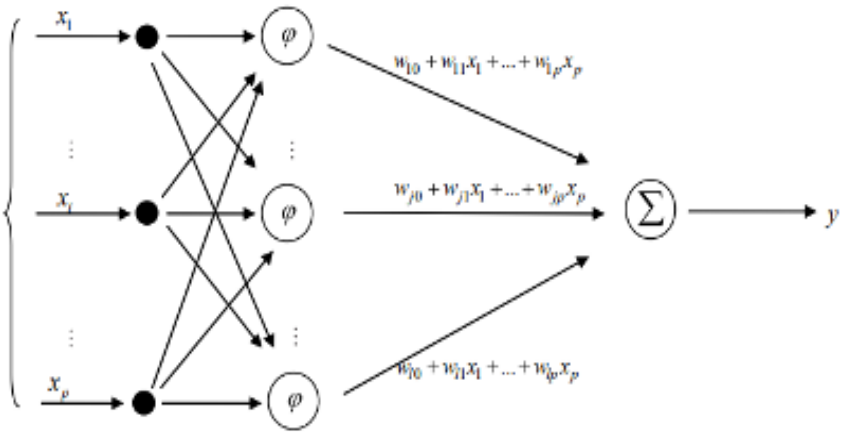


Figure 2.4 A schematic illustration of the k-Nearest Neighbors algorithm (Lin, Li, & Sadek, 2013)

2.4 Deep Learning Techniques

In recent years, deep learning techniques have gained significant attention in the field of traffic volume prediction, owing to their ability to automatically learn hierarchical feature representations from raw data. Convolutional Neural Networks (CNNs), Long Short-Term Memory (LSTM) networks, and Graph Convolutional Networks (GCNs) are some of the commonly used deep learning techniques for traffic volume prediction. Convolutional Neural Networks (CNNs) have been successfully applied to traffic volume prediction tasks due to their ability to capture local spatial and temporal dependencies in the data (Fei, Hu, Wei, & Chen, 2022). CNNs consist of multiple layers of convolution and pooling operations, which can effectively learn hierarchical feature representations for traffic volume prediction.

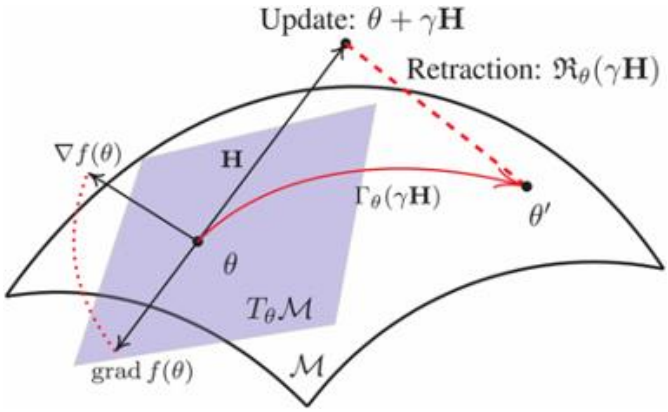


Figure 2.5 A schematic illustration of a Convolutional Neural Network for traffic volume prediction (Fei, Hu, Wei, & Chen, 2022)

Long Short-Term Memory (LSTM) networks are a type of recurrent neural network (RNN) that have been widely used for sequence-to-sequence prediction tasks, including traffic volume prediction (Park et al., 2018). LSTM networks are designed to address the vanishing and exploding gradient problems in standard RNNs, enabling the efficient learning of long-range temporal dependencies in the data.

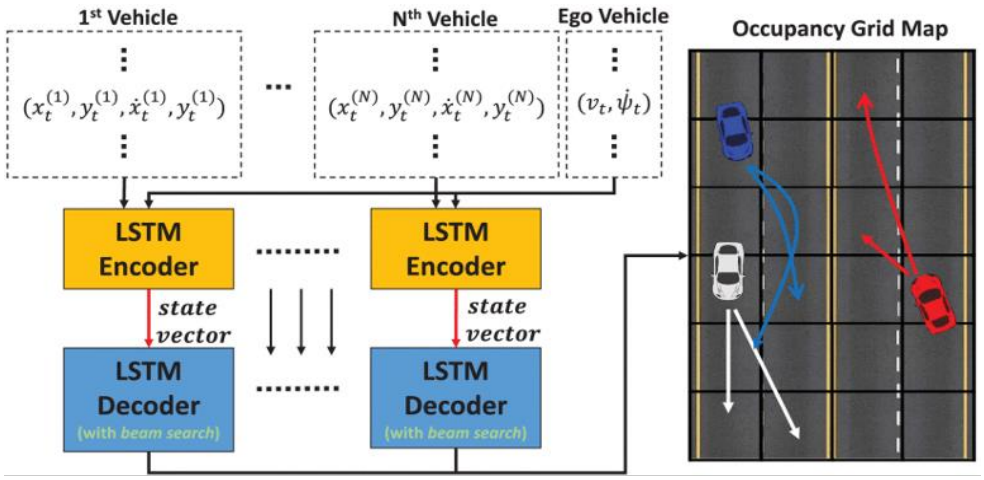


Figure 2.6 A schematic illustration of a Long Short Term Memory network for traffic volume prediction (Park et al., 2018)

Graph Convolutional Networks (GCNs) have been employed for traffic volume prediction tasks to capture spatial dependencies between different locations in a traffic network (Zhang, Zheng, & Qi, 2017). GCNs leverage graph-based

representations of the traffic network and incorporate neighborhood information in the learning process, providing more accurate and robust predictions compared to methods that consider only temporal dependencies.

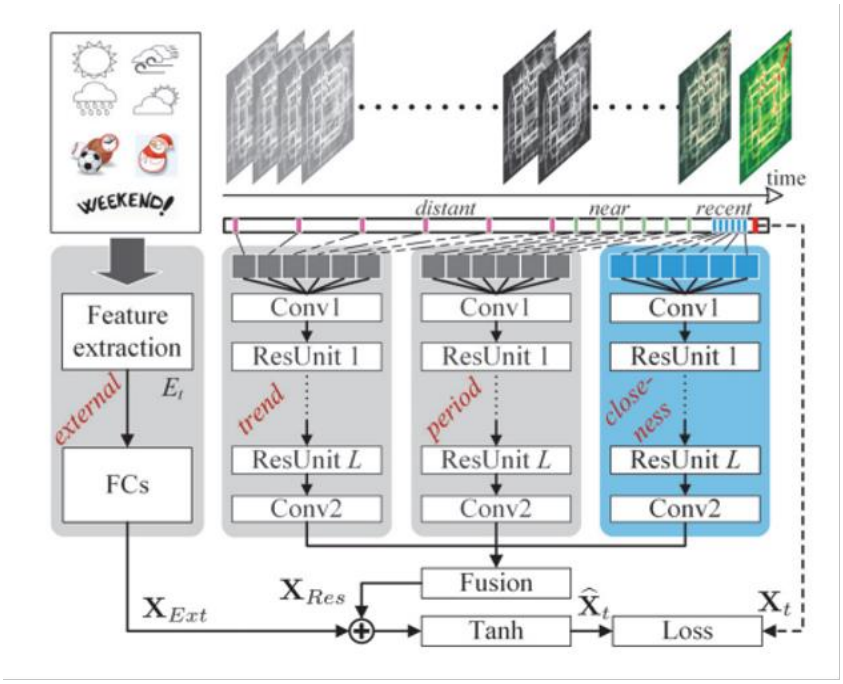


Figure 2.7 A schematic illustration of Graph Convolutional Network for traffic volume prediction (Zhang, Zheng, & Qi, 2017)

2.5 LSTM And GRU For Traffic Volume Prediction

The utilization of deep learning methodologies, particularly Long Short-Term Memory (LSTM) and Gated Recurrent Units (GRU), represents a paradigm shift in the domain of traffic volume prediction. These advanced neural network models excel in capturing the temporal dependencies and dynamic fluctuations inherent in traffic flows, offering a substantial improvement over conventional machine learning and time-series forecasting techniques. LSTM, a specialized form of recurrent neural network (RNN), is ingeniously designed to overcome the vanishing gradient dilemma that plagues standard RNNs, thereby enabling it to learn from long-term dependencies across time. This capability is attributed to its sophisticated architecture, which incorporates forget, input, and output gates, allowing the network to selectively retain or discard information through its memory cells. Such a design proves to be

exceptionally beneficial for modelling the sequential and time-sensitive nature of traffic volume data, enabling accurate predictions over extended periods.

Similarly, GRU simplifies the LSTM model by merging the forget and input gates into two new gates: the update and reset gates. This modification reduces the model's complexity without compromising its ability to process temporal sequences, making GRU an efficient alternative to LSTM for traffic prediction tasks. Both LSTM and GRU models have demonstrated remarkable proficiency in traffic volume forecasting, as they adeptly handle the sequential dynamics of traffic data, accommodating for sudden changes in traffic patterns due to various factors such as weather conditions, holidays, and peak hours. The comparative analysis conducted by Das (2023) sheds light on the efficacy of these models in the realm of traffic management. Through extensive experimentation on the Metro Interstate Traffic Volume dataset, it illustrates how LSTM and GRU models outshine traditional forecasting models by providing more accurate and reliable traffic volume predictions. This research underscores the potential of LSTM and GRU to revolutionize traffic management systems by enhancing the precision of traffic forecasts, thus facilitating more effective planning and optimization of traffic flows.

The significance of adopting LSTM and GRU models in traffic volume prediction cannot be overstated. By leveraging the historical traffic data and learning the intricate patterns, these models offer predictive insights that are crucial for the proactive management of traffic congestion, road safety improvements, and the efficient allocation of road resources. Moreover, the flexibility and adaptability of LSTM and GRU models allow for their application across various traffic scenarios and conditions, further evidencing their utility in creating intelligent transportation systems that can dynamically respond to changing traffic environments.

In conclusion, the ground-breaking work in applying LSTM and GRU models for traffic volume prediction paves the way for innovative traffic management solutions. By accurately forecasting traffic volumes, these models contribute to the development of smarter, more resilient transportation infrastructures capable of addressing the challenges posed by urbanization and increasing vehicle populations.

Furthermore, the results from the two models are perfect to use for benchmarking against the TCN model that will be developed in this project.

2.6 Temporal Convolutional Networks

In recent years, deep learning techniques have demonstrated their potential in handling time series data, such as traffic volume prediction. In particular, Temporal Convolutional Networks (TCNs) have emerged as a promising deep-learning approach for time series prediction tasks (Hewage et al., 2020). TCNs are a type of convolutional neural network (CNN) designed specifically for time series data by incorporating causal convolutions and dilated convolutions.

Causal convolutions ensure that the output at a specific time step only depends on the inputs from the past and the present, preventing information leakage from the future. This is crucial for accurate predictions in time series applications. Dilated convolutions allow the network to increase its receptive field exponentially with depth, allowing it to capture long-range dependencies in the input data (Oord et al., 2016).

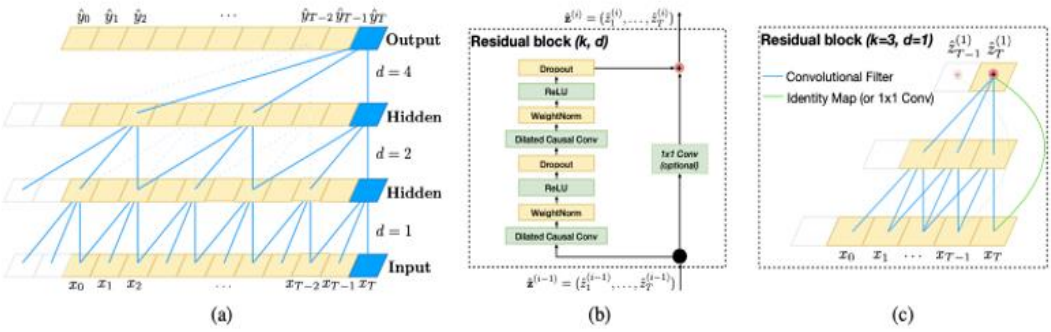


Figure 2.8 A graphical representation of a TCN model (Bai et al., 2018)

TCNs have several advantages over other deep learning techniques for time series prediction, such as RNNs and LSTMs. Firstly, TCNs exhibit more stable training behavior and are less susceptible to vanishing or exploding gradient problems (Bai et al., 2018). Secondly, TCNs can be easily parallelized, making them more computationally efficient compared to RNNs and LSTMs, which require sequential computation (Hewage et al., 2020). Finally, TCNs have been shown to outperform

other deep learning techniques in various time series prediction tasks, including traffic volume prediction (Zhao et al., 2019).

Despite these advantages, there has been limited research on applying TCNs to traffic volume prediction tasks. However, the few studies that have explored TCNs in this context have reported promising results. For example, Zhao et al. (2019) demonstrated that TCNs outperformed traditional time series models, machine learning methods, and other deep learning techniques in predicting traffic volume on a highway. They applied a T-GCN-based model to predict traffic volume on urban roads in China, achieving improved performance compared to several baseline models, including ARIMA, SVR, and LSTM.

Given the potential benefits of TCNs in traffic volume prediction tasks, further research is warranted to explore the suitability of TCNs in various traffic scenarios, as well as to investigate potential improvements and modifications to the architecture.

2.7 Dataset

The selection of appropriate datasets for training and testing traffic volume prediction models is of paramount importance. Datasets for traffic prediction can vary in terms of spatial coverage, temporal resolution, and data sources. The quality and relevance of the dataset can have a significant impact on the performance and generalizability of traffic volume prediction models. Furthermore, the choice of the dataset can influence the selection of suitable prediction techniques, as different techniques may have varying strengths and weaknesses depending on the data characteristics.

A widely-used dataset for traffic volume prediction research is the Metro Interstate Traffic Volume dataset, available on the UCI Machine Learning Repository (Das, 2023). The dataset includes hourly traffic volume data collected from a stationary sensor located on the Interstate 94 highway in Minneapolis-St. Paul,

Minnesota, USA, between 2012 and 2018. The dataset also includes several additional features, such as weather conditions, holidays, and time-related variables, which can be used as input features for traffic volume prediction models.

Researchers have employed various prediction techniques, such as ARIMA, SVR, RF, LSTM, and CNN, to model and predict traffic volume using this dataset (Cao, Li, & Chan, 2020). These studies have demonstrated the importance of selecting appropriate techniques, feature engineering, and model architectures to achieve accurate traffic volume predictions. However, the application of TCNs on this dataset for traffic volume prediction remains under-explored.

In addition to the Metro Interstate Traffic Volume dataset, several other datasets have been used in traffic volume prediction research, such as the Caltrans Performance Measurement System (PeMS) dataset (Chen, 2002) and the Beijing Transportation Dataset (Zhao et al., 2020). These datasets provide traffic volume data from different geographic locations and traffic scenarios, offering opportunities for evaluating and comparing the performance of different prediction techniques in various contexts.

Selecting an appropriate dataset for traffic volume prediction research is critical, as the choice of the dataset can influence the model's performance and generalizability. Future research in traffic volume prediction should not only focus on the development of new techniques but also consider the evaluation of existing techniques on diverse datasets to gain a deeper understanding of their strengths and limitations.

2.8 Model Optimization

In the realm of hyperparameter tuning, Bayesian optimization has emerged as a superior method, renowned for its strategic and resource-efficient approach. This technique, as opposed to the exhaustive nature of grid search, is rooted in a probabilistic model that adeptly navigates the hyperparameter space to swiftly identify the minimum of an objective function. It's particularly effective in scenarios where the

evaluation of the function is computationally intensive (Shahriari et al., 2016). Bayesian optimization differs from grid search by using a probabilistic model to predict the performance of hyperparameters, thereby guiding the search to more promising areas of the hyperparameter space.

The power of Bayesian optimization lies in its iterative process, which incorporates prior knowledge and sequentially refines this understanding with actual observations to better estimate the objective function. It strikes a judicious balance between exploration of uncertain regions and exploitation of areas predicted to yield strong results. This methodology significantly curtails the number of evaluations to pinpoint an optimal solution, thus conserving computational resources and expediting the optimization process.

On the other hand, grid search suffers from scalability issues, especially as the hyperparameter dimensions escalate, which exponentially increases the volume of evaluations—a phenomenon often cited as the "curse of dimensionality" (Snoek, Larochelle, & Adams, 2012), Grid search also overlooks the dependencies between hyperparameters, which can be pivotal since the optimal value for one parameter might be contingent on the values of others.

Given these considerations, Bayesian optimization is typically the preferred choice for hyperparameter tuning within machine learning, due to its efficiency in iteration and its capacity to uncover superior hyperparameter values in fewer trials. Hence, why it would be a great fit for this project's purposes.

2.9 Benchmarks

In this section, a performance comparison of various machine learning and deep learning models for different traffic analysis tasks is outlined, including sequence modeling, traffic volume prediction, and traffic affecting environments classification. The evaluation metrics used in these papers include accuracy, mean squared error (MSE), mean absolute error (MAE), negative log-likelihood (NLL), and root mean

squared error (RMSE). The models evaluated in these papers include Temporal Convolutional Networks (TCN), Long Short-Term Memory (LSTM), Gated Recurrent Unit (GRU), Decision Tree (DT), Support Vector Machine (SVM), k-Nearest Neighbors (KNN).

Table 2.1 provides an overview of the benchmark results, including the source of the paper, the dataset used, the evaluation metrics employed, and the specific models utilized.

Table 2.1 Different Methods of Traffic Prediction

Source	Dataset	Evaluation Metrics	Model
An Empirical Evaluation of Genetic Convolutional and Recurrent Networks for Sequence Modelling (Bai, 2018)	Adding Problem, Sequential MNIST, Copy Memory, JSB Chorales, Nottingham, PennTreebank, Wikitext-103, LAMBADA, text8	Accuracy (100%), MSE, NILL	TCN, LSTM, GRU, CNN
Traffic Volume Prediction using Memory-Based Recurrent Neural Networks: A comparative analysis of LSTM and GRU (Das, 2023)	Metro Interstate Traffic Volume	MSE, MAE, MAPE	LSTM, GRU
Application of Data Mining Techniques for Classification of Traffic Affecting Environments (Khiewwan et al.,2020)	Metro Interstate Traffic Volume	Accuracy (75%)	Decision Tree (DT), Support Vector Machine (SVM), k-Nearest Neighbors (KNN)

Genetic Convolutional and Recurrent Networks exhibited other architectures, but it is still infantile in certain aspects, which might be why it isn't advanced in research as other architectures. The lack of optimization schemes means that researchers head towards other methods, although this can also mean that more work into this would greatly improve the performance. The use of memory-based and recurrent neural networks seems to be promising, with very high accuracy in prediction. Although this is the case, the pre-processing or preparing of the data sets for the algorithm was reported to be a challenging task since it needs another dimension. This is also accompanied by a relatively high computational power, which might not be an issue with more advanced processors but may trail behind other simpler methods. In application data mining, using a decision tree also used time for prediction, thus can predict traffic jams early, allowing for more time in accounting for them. With an accuracy of about 75% in traffic jam prediction at certain points, this method can be used beneficially in preparing for congestion at bottleneck points.

2.10 Conclusion

In conclusion, traffic volume prediction is a crucial task in the domain of intelligent transportation systems, with significant implications for traffic management and planning. Accurate traffic volume prediction models can facilitate congestion mitigation, route planning, and infrastructure investment decisions. Over the past few years, a wide range of techniques, including traditional time series models, machine learning methods, and deep learning techniques, have been employed to predict traffic volume with varying degrees of success.

Among the deep learning techniques, Temporal Convolutional Networks (TCNs) have emerged as a promising approach for time series prediction tasks, such as traffic volume prediction. TCNs offer several advantages over other deep learning techniques, such as more stable training behavior, computational efficiency, and the ability to capture long-range dependencies in the data. Despite these benefits, research on applying TCNs to traffic volume prediction tasks has been limited, and further

investigation is warranted to evaluate the suitability of TCNs for various traffic scenarios and datasets.

The choice of dataset is a critical factor in traffic volume prediction research, as it influences the performance and generalizability of prediction models. The Metro Interstate Traffic Volume dataset is a widely-used dataset for traffic volume prediction studies. However, other datasets, such as the Caltrans Performance Measurement System (PeMS) dataset and the Beijing Transportation Dataset, provide opportunities to evaluate and compare different prediction techniques in diverse contexts.

Future research in traffic volume prediction should focus on exploring the potential of TCNs and other deep learning techniques for diverse datasets and traffic scenarios, as well as investigating potential improvements and modifications to model architectures. Additionally, researchers should continue to evaluate the performance of existing techniques on a variety of datasets to gain a deeper understanding of their strengths and limitations, ultimately leading to the development of more accurate and robust traffic volume prediction models.

By investigating the potential of TCNs and other deep learning techniques for traffic volume prediction, researchers can contribute to the ongoing development and refinement of accurate, efficient, and robust traffic prediction models, ultimately improving traffic management and planning efforts worldwide.

CHAPTER 3

RESEARCH METHODOLOGY

3.1 Introduction

Understanding traffic volume patterns is critical for efficient traffic management and planning. Achieving accurate traffic volume prediction can lead to improved road safety, reduced congestion, and enhanced driver convenience. The proposed traffic volume prediction project taps into the prowess of Temporal Convolutional Networks (TCNs), a deep learning model recognized for its expertise in handling sequential data. The detailed methodology for this project, stretching from data acquisition to model evaluation, aims to foster the development of an accurate and robust model, capable of trustworthy predictions.

Driven by the aspiration to craft a comprehensive, data-driven solution for traffic volume prediction, this project seeks to combine the power of deep learning with the richness of available traffic data. The aim is to devise a model that not only excels in academic metrics but also proves practically applicable for traffic management authorities. The aim of detailing the methodology is to offer transparency into the working process, promoting a thorough understanding of how the TCN model for traffic volume prediction will be developed and evaluated.

The details of the research framework for this study are shown in Figure 3.1.

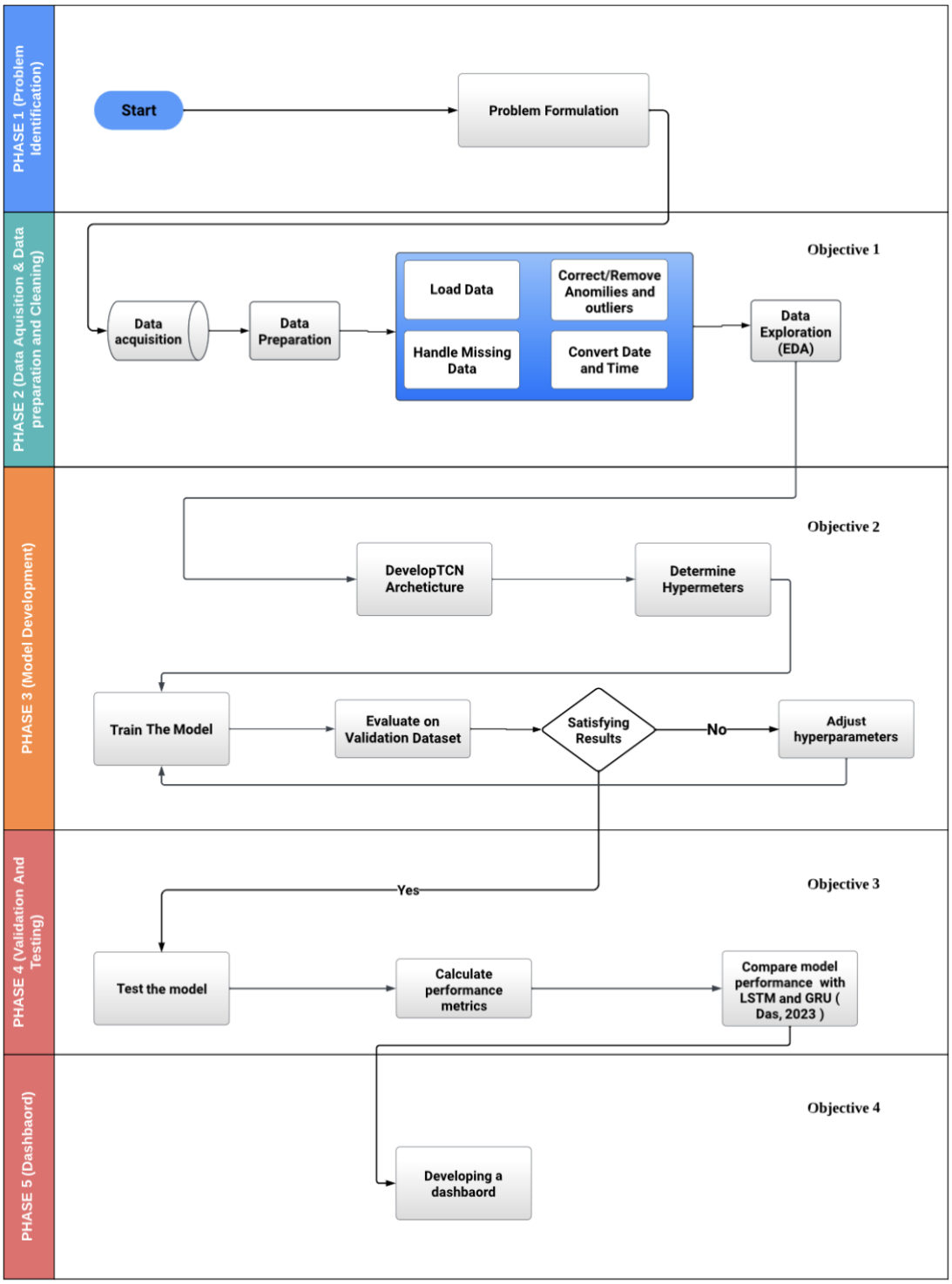


Figure 3.1 Proposed Framework

3.2 Problem Identification

The primary objective of this study is to investigate the implementation of Temporal Convolutional Networks (TCNs) in the domain of traffic volume prediction. Despite the considerable potential of TCNs, there are certain challenges that need to be addressed to produce high-quality, accurate traffic predictions.

1. Prediction of atypical traffic patterns: One of the most significant hurdles in traffic volume prediction lies in accurately predicting irregular or atypical traffic patterns. These could be due to exceptional events such as public holidays, major sports events, or unexpected road closures. In these instances, the traffic volume deviates significantly from the regular patterns, often falling into the 'long tail' area. The TCN model should be adept at capturing these unusual fluctuations in traffic volume and make accurate predictions, even in the face of such uncommon occurrences.

2. Adjustment to rapid changes in traffic volume: Traffic volume is not constant; it is influenced by numerous factors, including weather conditions, road works, accidents, or changes in local events. These factors can lead to rapid changes in traffic volume, presenting a significant challenge for any prediction model. A successful traffic volume prediction model should have the capacity to adapt quickly and accurately reflect these sudden changes in its predictions.

Addressing these challenges effectively is a critical part of this study, with the aim of developing a robust TCN model capable of making accurate traffic volume predictions under varying conditions. The subsequent sections will discuss in detail how each stage of the methodology contributes to tackling these problems and enhancing the model's performance.

3.3 Data Acquisition And Preparation

Data acquisition is the first and foremost step in any data-driven project. In the context of traffic volume prediction, acquiring reliable and accurate traffic data is crucial to the success of the project. The dataset chosen for this project is the Metro

Interstate Traffic Volume dataset from the UCI Machine Learning Repository. It contains traffic volume data collected from a particular location on a major interstate highway, spanning several years. The dataset comprises of various attributes, such as date, time, traffic volume, weather conditions, and holidays. Furthermore, Table 3.1 provides a summary of these attributes. The use of a comprehensive dataset with various influencing factors helps in building a more robust and accurate prediction model.

Table 3.1 Dataset Attributes

Attribute	Description
holiday	US National holidays plus regional holiday
temp	Average temperature in Kalvin
rain_1h	Amount in mm of rain that occurred in the hour
Snow_1h	Amount in mm of snow that occurred in the hour
Clouds_all	Percentage of cloud cover
weather_main	Short textual description of the current weather
Weather_description	Longer textual description of the current weather
Date_time	Hour and date of the data collected in local CST time
Traffic_volume	Hourly count of the number of vehicles on the westbound segment

By using a publicly available dataset, researchers can ensure the reproducibility of their work and facilitate comparison with other existing models in the literature. The dataset can be downloaded from the UCI Machine Learning Repository's website and stored in a suitable format for further processing, such as a CSV file or a DataFrame in Python. Data acquisition should be performed in a manner that maintains the integrity of the original dataset while complying with any usage restrictions or licensing agreements. Moreover, data privacy and security concerns should be considered, especially if working with sensitive or personally identifiable information.

The data used in this project is obtained from an open-source traffic volume prediction dataset available for download at (https://archive.ics.uci.edu/dataset/492/metro+interstate+traffic+volume). The dataset provides detailed hourly information about various factors affecting traffic volume, including weather conditions, time, and holiday periods. It is a valuable asset for building a robust and accurate traffic volume prediction model using Temporal Convolutional Networks (TCN).

Data preprocessing is a vital step in building this prediction model and involves several processes:

1. Preliminary Analysis: An initial examination of the dataset helps understand its structure, the variables it contains, and any evident patterns or anomalies. This step is crucial for informed decision-making during the modeling phase.

2. Data Cleaning & Validation: Ensuring the dataset is complete and relevant is critical. All missing or irrelevant data should be addressed appropriately. For instance, it might be necessary to fill in 'None' values for categorical features or median values for numerical ones. Similarly, any outliers within the data, such as abnormal traffic volumes or weather conditions, should be identified and managed to prevent skewing the model's predictions.

3. Handling Categorical Data: The dataset includes categorical data such as 'weather_main' and 'holiday'. These variables may need to be encoded as numerical values to be effectively used in the TCN model.

4. Normalizing Numerical Data: Numerical data, such as temperature or clouds_all, may need to be normalized to ensure all variables are on a similar scale. This process prevents certain variables from disproportionately influencing the model.

5. Data Windowing: This method is particularly useful when working with sequential data for tasks such as forecasting or recognizing patterns over time. The idea is to create a series of overlapping windows that can be fed into a

model for training, allowing the model to learn from a sequence of data points as one single input.

6. Splitting the Data: The dataset should be divided into a training set, used to build the model, and a testing set, used to evaluate the model's performance.

The above steps are crucial in building a traffic volume prediction model, as they ensure the data is appropriately prepared for use in training the TCN model. This preparation is essential for maximizing the accuracy and effectiveness of the prediction model. The specific data preprocessing steps applied are presented in Table 3.2.

Table 3.2 Purpose of data pre-processing method

Data Pre-processing methods	Purpose
Preliminary analysis	To examine the provided data set and gain knowledge that can be applied to the modeling phase.
Data Cleaning & validation	To ensure NA values are filled in a way that simplifies future modeling, such as inputting 'None' for categorical features or the median value for numerical features.
Handling categorical data	To convert categorical variables like 'weather_main' and 'holiday' into numerical values.
Normalizing numerical data	To ensure all variables are on the same scale, preventing certain variables from dominating the model.
Data windowing	To allow the model to learn from a sequence of data points as one single input.
Splitting the data	To divide the dataset into a training set for building the model and a testing set for evaluating its performance.

3.4 Model Development

In the context of this project, the model development revolves around the implementation of a Temporal Convolutional Network (TCN) specifically for

predicting traffic volume. TCNs have emerged as an effective deep learning model tailored to handle sequential data, making them highly suitable for time series prediction tasks such as ours.

The process of TCN model development involves several integral steps:

1. Designing the Model Architecture: The model architecture is constituted by three primary layers - the input layer, the convolutional layers, and the output layer. The input layer is designed to accommodate the preprocessed features, while the convolutional layers carry the responsibility of discerning complex temporal patterns within the data. The final predictions, be it a single value or a sequence of values contingent on the problem at hand, are produced by the output layer.

2. Feature Engineering: Feature engineering, a crucial aspect of model development, entails choosing and modifying variables in the dataset to enhance the performance of the model. In this scenario, feature engineering will comprise the sequencing of data, along with scaling and normalization, followed by time series windowing.

3. Hyperparameter Determination: Choosing appropriate hyperparameters, such as the kernel size, the number of filters, and the dilation rate, is a crucial element in achieving optimal performance from the model. These hyperparameters can be fine-tuned manually or via automated techniques like Bayesian optimization. It's important to create a balance between model complexity and computational efficiency, ensuring that the model can effectively learn from the available data.

4. Model Compilation: Upon determining the TCN architecture and its hyperparameters, the model needs to be compiled using a fitting loss function and optimization algorithm. The choice of loss function, such as Mean Squared Error (MSE), and an optimizer like Adam hinges upon the specific problem and dataset characteristics. Thus, in this case it is suitable.

The following Figure 3.2 presents the base architecture for the TCN model to be used in this project. Note that these parameters may require modifications or adjustments based on performance results during the model development process.

Figure 3.2 Base TCN Architecture

This proposed architecture encapsulates key components from (Bai, 2018). The model is equipped with dilated and causal convolutions, which respectively help capture long-term dependencies and preserve the temporal order of events. Residual connections aid in avoiding vanishing or exploding gradient issues, enhancing the model's training efficacy. As with the previous model, the output layer consists of one node, providing the predicted traffic volume.

The methodical and accurate development of the TCN model, focusing on effective feature engineering, optimal hyperparameters, and appropriate model compilation, forms the backbone of this project's pursuit for precise traffic volume prediction.

3.5 Model Evaluation

The post-training evaluation of the Temporal Convolutional Network (TCN) model's performance is an essential step in determining its ability to generalize to unseen data. This offers insights into its potential real-world applicability. In this project, the performance metrics employed for assessing the model's predictive accuracy in the context of traffic volume prediction include mean absolute error (MAE), mean squared error (MSE), and R-squared.

1. Mean Absolute Error (MAE): The MAE measures the average absolute difference between the predicted and the actual values. It is a linear score, which means that all the individual differences are weighted equally in the average. The equation for calculating MAE is:

$$MAE = (1/N) * \Sigma|y\_i - \hat{y}\_i| \quad (3.1)$$

2. Mean Squared Error (MSE): The MSE assesses the average squared difference between the predicted and actual values. Unlike MAE, MSE is a quadratic scoring rule that squares the average distance to the target. The equation for calculating MSE is:

$$MSE = (1/N) * \Sigma(y\_i - \hat{y}\_i)\text{^}2 \quad (3.2)$$

3. Root Mean Squared Error (RMSE): is a standard way to measure the error of a model in predicting quantitative data. It is commonly used in regression analysis to verify experimental results. RMSE serves to aggregate the magnitudes of the errors in predictions into a single measure of predictive power. The equation for calculating RMSE is:

$$RMSE = n1\sum i = 1n(yi - y\text{^}i)2 \quad (3.3)$$

A lower RMSE value is indicative of a more accurate model. RMSE is particularly useful when large errors are particularly undesirable.

Visual representations of the model's predictions compared with the actual values from the test dataset provide significant insights into the model's performance and help to identify potential areas for improvement. Various techniques such as line plots, scatter plots, or heatmaps can be utilized for this purpose. In addition, comparing the performance of the TCN model with other machine learning and deep learning models explored in contemporary research is integral to the model evaluation process. This comparison allows for a better understanding of the strengths and limitations of different techniques.

CHAPTER 4

EXPLORATORY DATA ANALYSIS

4.1 Introduction

This chapter delves into the initial findings and observations obtained from the exploratory data analysis (EDA) process. It serves as a crucial bridge between the rigorous data preparation and model development stages, providing insightful preliminary results. The discussion covers the inherent patterns, anomalies, and potential correlations in the data that could significantly influence traffic volume predictions. The observations gathered during the EDA process will help shape the subsequent model development and training phases, aiming for the most accurate and robust traffic volume predictions possible.

The chapter concludes with an assessment of the results, identifying the insights gained and the way forward in light of these findings. This concluding section marks the transition from data exploration to the application of these findings in the context of Temporal Convolutional Networks (TCNs), setting the stage for the subsequent chapter on model development and evaluation.

4.2 Data Understanding

Embarking on the exploratory data analysis journey, the primary aim is to understand the intricacies of the traffic volume data and unearth hidden patterns that might aid in the prediction tasks ahead. EDA is an essential precursor to any modeling effort. It gives a sense of what the data is communicating and guides the subsequent steps of the process.

Firstly, it is extremely important to know the dataset well and statistics in this process is very useful, as it provides descriptive measures that demonstrate the main characteristics of the data the project is dealing with.

After loading the dataset into and displaying the basic information about each column. Figure 4.1 provides an overview of the dataset to be used for traffic volume prediction using a Temporal Convolutional Network (TCN). This dataset comprises 48,204 entries, each with nine features. It's essential to note that the dataset is complete, with no missing values across any of the categories.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 48204 entries, 0 to 48203
Data columns (total 9 columns):
 #   Column               Non-Null Count  Dtype
---  ------               --------------  -----
 0   holiday              48204 non-null  object
 1   temp                 48204 non-null  float64
 2   rain_1h              48204 non-null  float64
 3   snow_1h              48204 non-null  float64
 4   clouds_all           48204 non-null  int64
 5   weather_main         48204 non-null  object
 6   weather_description  48204 non-null  object
 7   date_time            48204 non-null  object
 8   traffic_volume       48204 non-null  int64
dtypes: float64(3), int64(2), object(4)
memory usage: 3.3+ MB
```

Figure 4.1 Overview on the dataset attributes

The features comprise both numerical and categorical data types. Numerical features include temp, rain_1h, snow_1h, clouds_all, and the target variable traffic_volume. Categorical features are holiday, weather_main, and weather_description. date_time is currently listed as a categorical variable but will need conversion into a timestamp for appropriate use in the time-series-based TCN model.

Figure 4.2 presents a statistical summary of the numerical attributes: 'temp', 'rain_1h', 'snow_1h', 'clouds_all', and 'traffic_volume'. The 'temp' attribute, denoting temperature in Kelvin, ranges from 0K to 310.07K with an average value of 281.21K, reflecting a wide array of weather conditions. The 'rain_1h' column, indicating hourly rainfall levels, has a notably large maximum value, potentially suggesting outliers. On the other hand, the 'snow_1h' attribute, which represents hourly snowfall, is nearly always zero, possibly indicating that snowfall might have a negligible effect on traffic

volume. The 'clouds_all' column, reflecting the percentage of cloud cover, varies from 0 to 100%, implying a diversity of weather conditions. Finally, the target variable 'traffic_volume' spans from 0 to 7280 vehicles with an average of roughly 3259.8. The substantial standard deviation signals considerable variance in traffic volume, a key aspect to account for in the subsequent Temporal Convolutional Network (TCN) model construction.

	temp	rain_1h	snow_1h	clouds_all	traffic_volume
count	48204.000000	48204.000000	48204.000000	48204.000000	48204.000000
mean	281.205870	0.334264	0.000222	49.362231	3259.818355
std	13.338232	44.789133	0.008168	39.015750	1986.860670
min	0.000000	0.000000	0.000000	0.000000	0.000000
25%	272.160000	0.000000	0.000000	1.000000	1193.000000
50%	282.450000	0.000000	0.000000	64.000000	3380.000000
75%	291.806000	0.000000	0.000000	90.000000	4933.000000
max	310.070000	9831.300000	0.510000	100.000000	7280.000000

Figure 4.2 Statistical summary of the numerical values

Figure 4.3 provides information on the categorical features of the dataset: 'holiday', 'weather_main', 'weather_description', and 'date_time'. The 'holiday' feature encapsulates 12 unique values, the most frequent of which is 'None'. This is a clear indication that most of the data points fall on non-holiday days. The 'weather_main' feature includes 11 unique conditions, with 'Clouds' appearing most frequently. This might suggest that cloud cover is a prevalent condition in the dataset, an aspect that could potentially impact traffic volume. However, since a deep-learning model will be used that won't be an issue.

The 'weather_description' feature brings additional granularity, encompassing 38 unique conditions. Here, the 'sky is clear' condition appears most frequently, suggesting that this particular weather condition may have an influence on the traffic volume. The 'date_time' feature, encapsulating the timestamp of each data point, holds 40,575 unique values. Its most frequent entry is '2013-04-18 22:00:00', occurring six times. Furthermore, factors like time of day and date can significantly influence traffic

volume, highlighting peak and off-peak periods—a consideration that will be important when training the TCN model. Thus, it needs to be converted to timestamp format.

	holiday	weather_main	weather_description	date_time
count	48204	48204	48204	48204
unique	12	11	38	40575
top	None	Clouds	sky is clear	2013-04-18 22:00:00
freq	48143	15164	11665	6

Figure 4.3 Distribution and frequency of the categorical features

4.2.1 Univariate Analysis

In this section, a comprehensive analysis of each variable is conducted. Beginning with the 'holiday' column, Figure 4.4 illustrates a substantial predominance of normal days throughout the years covered in the dataset. Consequently, holidays aren't visually discernible in the plot. To effectively analyze them, it becomes necessary to filter out the normal days. Consequently, the 'holiday' column should be re-categorized into two classes: 'holiday' and 'non-holiday'. This binary classification would simplify further analysis. However, the binary classification is only done to analyze the column and won’t be a part of the final data. Proceeding to Figure 4.5, it depicts the count distribution of each holiday present within the dataset, granting valuable insights into their frequency.

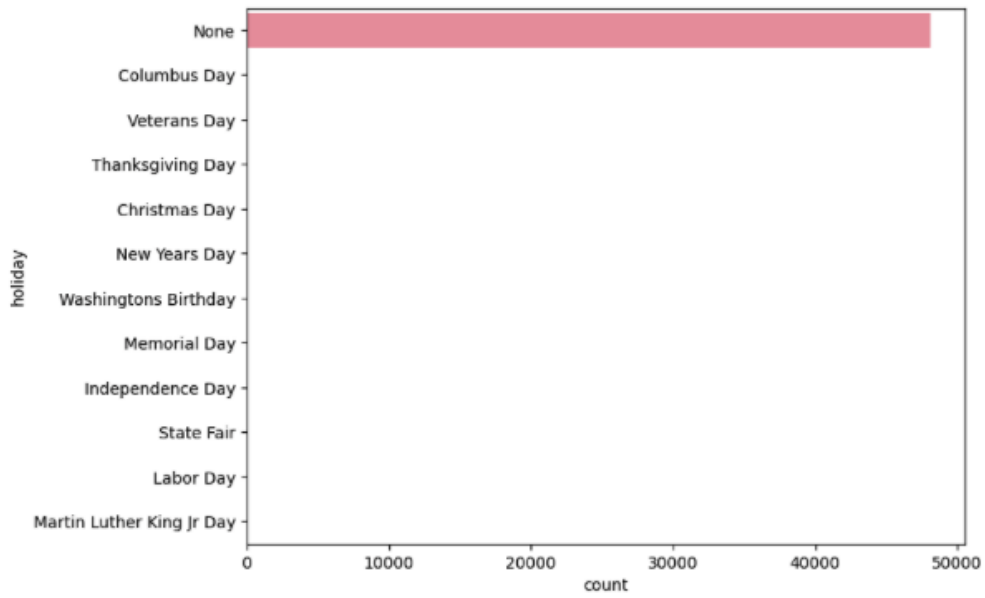


Figure 4.4 Count of the holiday column

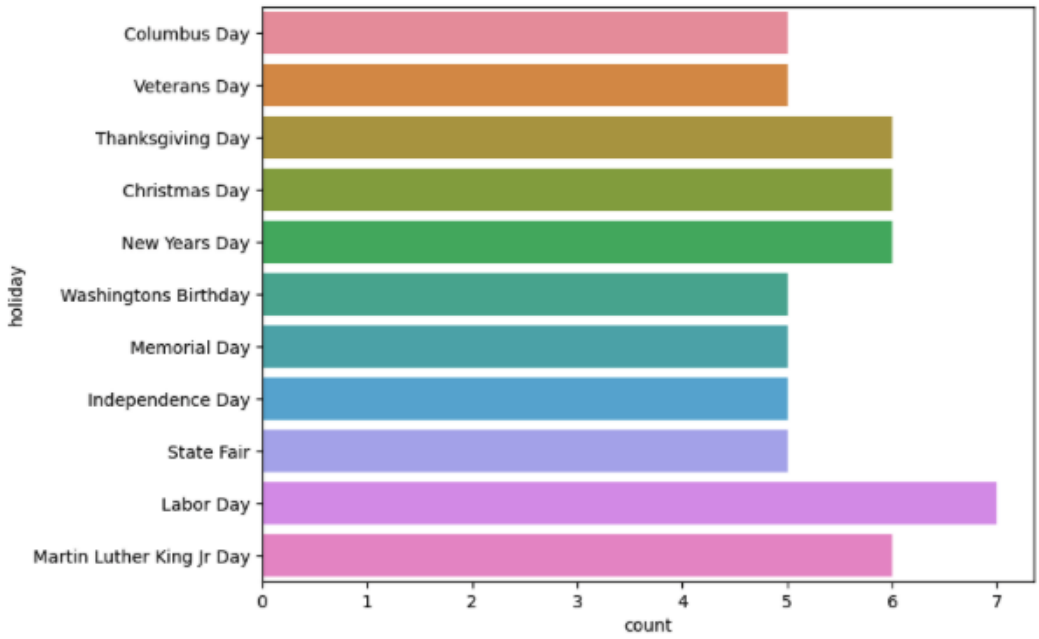


Figure 4.5 The count of each holiday

For the 'temp' column, which records temperature values, an initial observation reveals that these temperatures were originally recorded in Kelvin units. To make the data more intuitively understandable, these values were converted to Celsius. It's important to note that, as mentioned earlier, some outliers were observed at absolute zero. These likely resulted from errors during the data capture process. It is essential

to address these outliers by removing them to ensure they do not adversely affect the overall analysis and subsequent prediction model's performance.

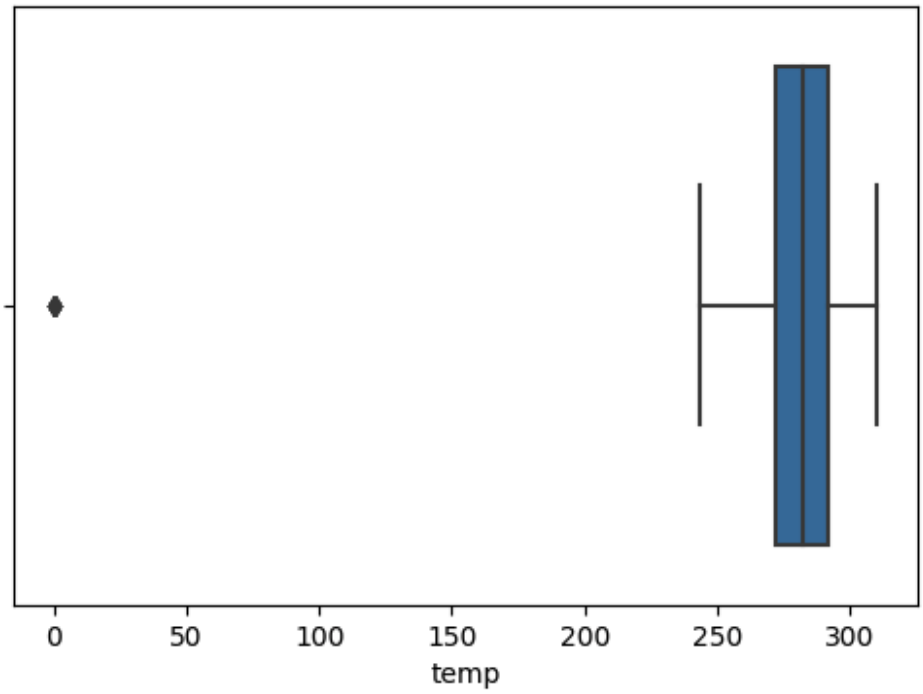


Figure 4.6 The distribution of the temperature column

Upon inspecting the 'rain_1h' and 'snow_1h' columns of the dataset, which quantify the amount of rain and snow per hour, we observe a significant skew in their distribution, prominently featuring a preponderance of zeroes as illustrated in Figure 4.7. This skewness suggests that these variables have a high concentration of values at the lower end of the range, which could potentially skew the model's performance if not addressed.

To ensure the robustness of the predictive model and mitigate the undue influence of these skewed distributions, an interquartile range (IQR) method will be employed to identify and exclude outliers. The IQR is a measure of statistical dispersion and is calculated as the difference between the 75th and 25th percentiles (Q3 and Q1, respectively) of the data.

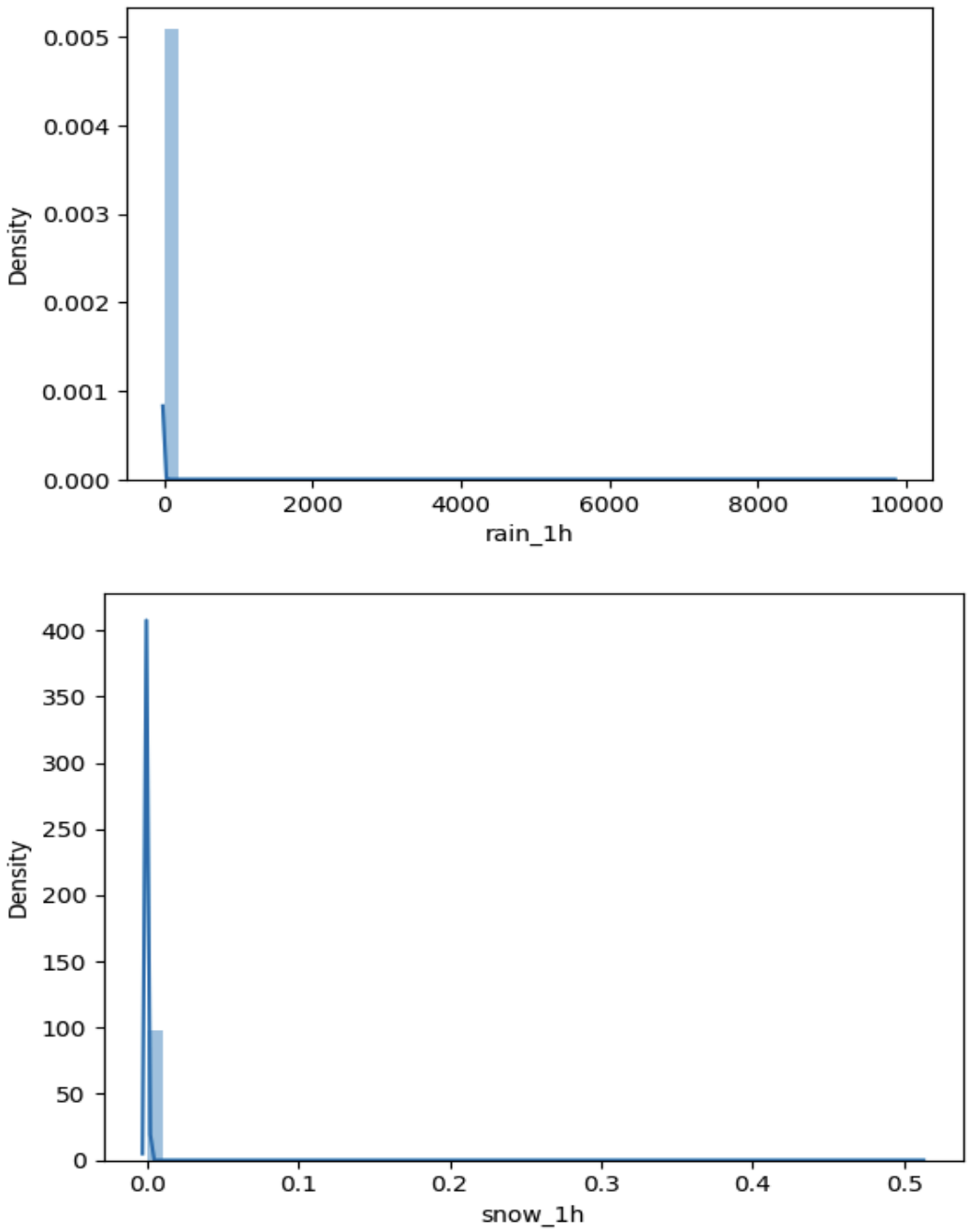


Figure 4.7 The distribution the rain_1h and snow_1h columns

In relation to the 'clouds_all' column depicted in Figure 4.8, the data shows a noticeable number of zero values. These values represent periods with no cloud cover. However, unlike the rain and snow variables, the distribution of this data does not raise concerns about inconsistency. Rather, it represents a wide spectrum of possible cloud

cover percentages, from clear skies (0%) to overcast conditions (100%). This range of values suggests that this variable may provide useful information for the traffic volume prediction task, as weather conditions could potentially influence traffic patterns.

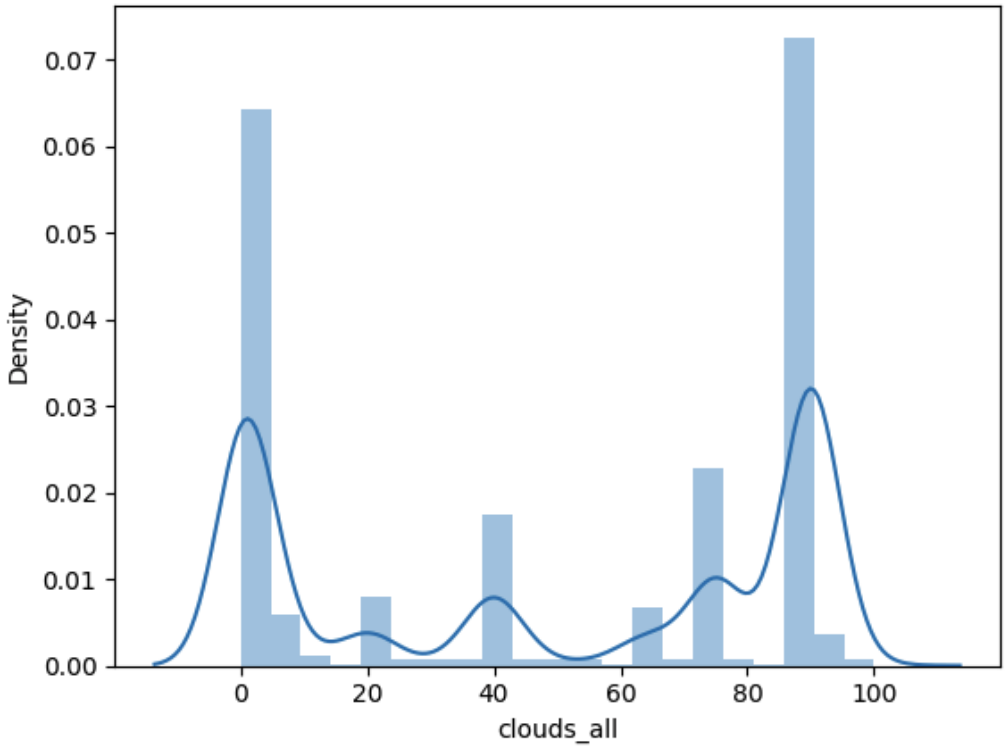


Figure 4.8 The distribution the ‘clouds_all’ column

Upon closer inspection of the 'weather_main' and 'weather_description' columns within the dataset, it is noted that while 'weather_description' offers a more detailed elaboration of 'weather_main', both features present nuanced aspects of the weather conditions. Given the depth and complexity that deep learning models, such as Temporal Convolutional Networks (TCN), can accommodate, retaining both columns could enrich the model's input and potentially enhance its predictive capabilities. This is especially pertinent considering the high-dimensional feature space that deep learning models can effectively handle.

Figures 4.9 and 4.10 visually depict the variation and the additional granularity provided by the 'weather_description' column as compared to 'weather_main'. In the context of traffic volume prediction, the detailed weather descriptions may capture

specific atmospheric conditions that have a distinct impact on traffic patterns, which the more general 'weather_main' categories could overlook.

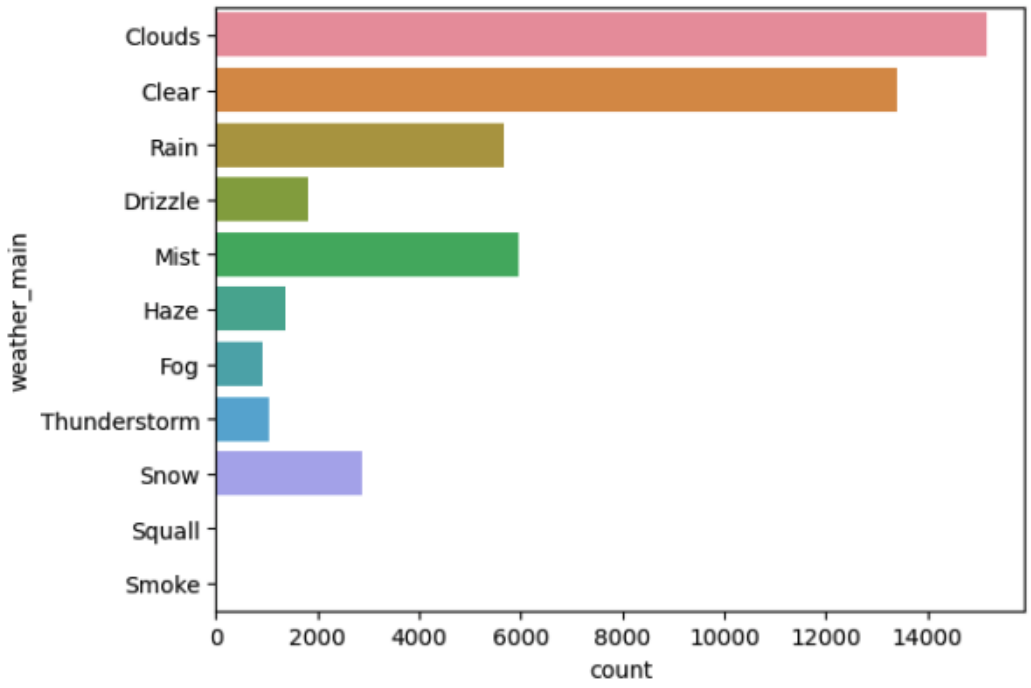


Figure 4.9 The count of each value in the ‘weather_main’ column

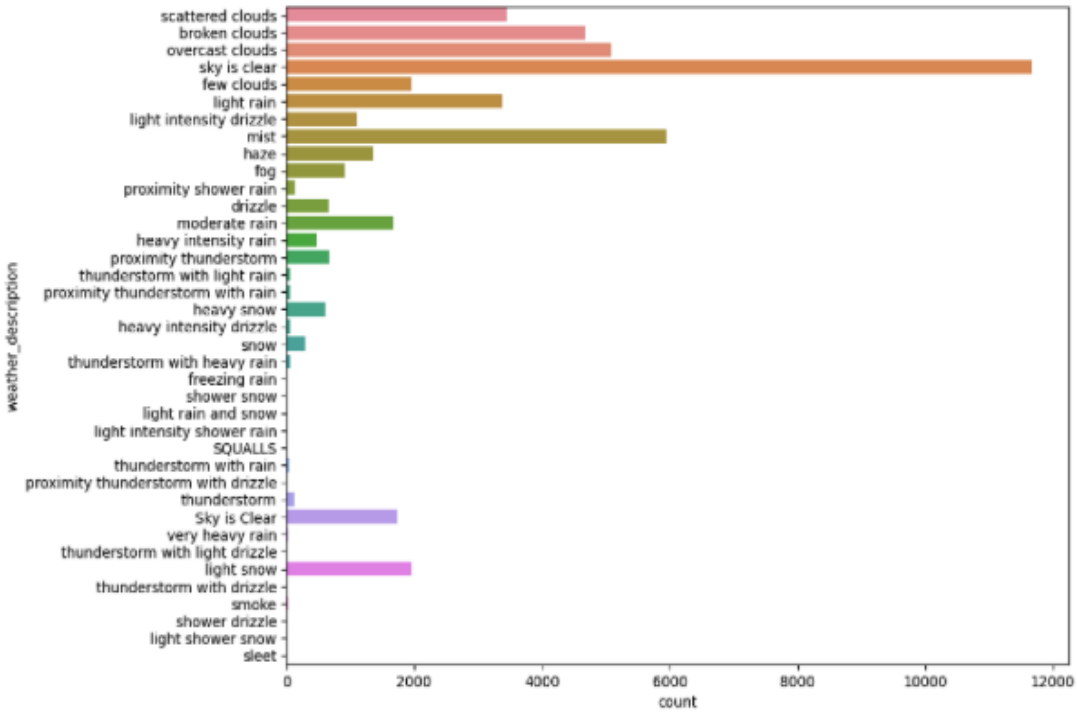


Figure 4.10 The count of each value in the 'weather_description' column

4.2.2 Bivariate Analysis

In order to gain deeper insights into the traffic volume data, a bivariate analysis] was conducted, focusing on the relationship between the traffic volume and various factors such as the hour of the day, the month, and year, and the distinction between weekdays and weekends. This analysis aimed to examine how the traffic volume varies in relation to these factors and uncover any patterns or trends that may exist. By exploring the associations between the traffic volume and these variables, a better understanding of the temporal dynamics and seasonal variations of traffic volume can be obtained, facilitating more accurate prediction models and informed decision-making in traffic management.

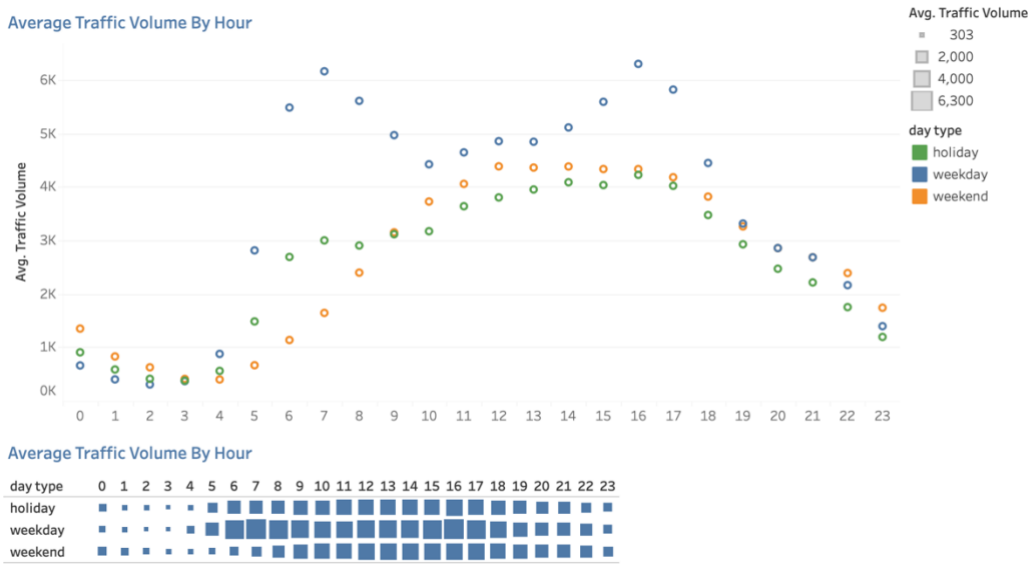


Figure 4.11 Average traffic volume per hour

The analysis of the average traffic volume data for different hours and day types in Figure 4.11 reveals several key insights. Firstly, weekdays exhibit distinct rush hour patterns with higher volumes during morning and evening peak hours. This information can aid traffic management systems in efficiently allocating resources and implementing congestion alleviation strategies during these periods. Secondly, weekends demonstrate different traffic patterns, with relatively high volumes throughout the day, indicating a more leisure-oriented and less time-constrained travel behavior. Adjusting traffic signal timings and managing traffic flow during weekends can accommodate these distinct demand patterns. Thirdly, nighttime traffic shows higher volumes during late-night and early morning hours on holidays and weekends, suggesting late-night activities or travel during these periods. Understanding nighttime traffic patterns enables the optimization of road infrastructure and the implementation of appropriate safety measures. Additionally, fluctuations in traffic volume throughout the day reflect variations in travel demand and user preferences, highlighting the need for informed decisions regarding infrastructure development, public transportation schedules, and traffic management strategies. By leveraging these insights, transportation authorities can allocate resources effectively, optimize traffic signal timings, adjust public transportation frequencies, and enhance overall urban mobility.

The analysis of average traffic volume per hour by month and day type in Figure 4.12 provides valuable insights for traffic volume prediction. The variations observed across different months indicate the presence of seasonal patterns in traffic volume, with higher volumes during certain months like August and lower volumes in months like January. This information is crucial for developing predictive models that can capture and account for these seasonal fluctuations. Furthermore, the analysis reveals variations in traffic volume based on the time of day, with peak volumes occurring during specific hours. For instance, there is a noticeable increase in traffic volume during morning and evening rush hours, suggesting the influence of commuting patterns. This temporal dependency underscores the importance of incorporating time-dependent features in prediction models to accurately capture the fluctuations in traffic volume throughout the day. Additionally, the correlation between traffic volume and temperature highlights the impact of weather conditions on traffic patterns. Higher traffic volumes tend to coincide with favorable weather conditions, such as moderate temperatures. This finding emphasizes the need to consider weather data as a significant factor in traffic volume prediction models.

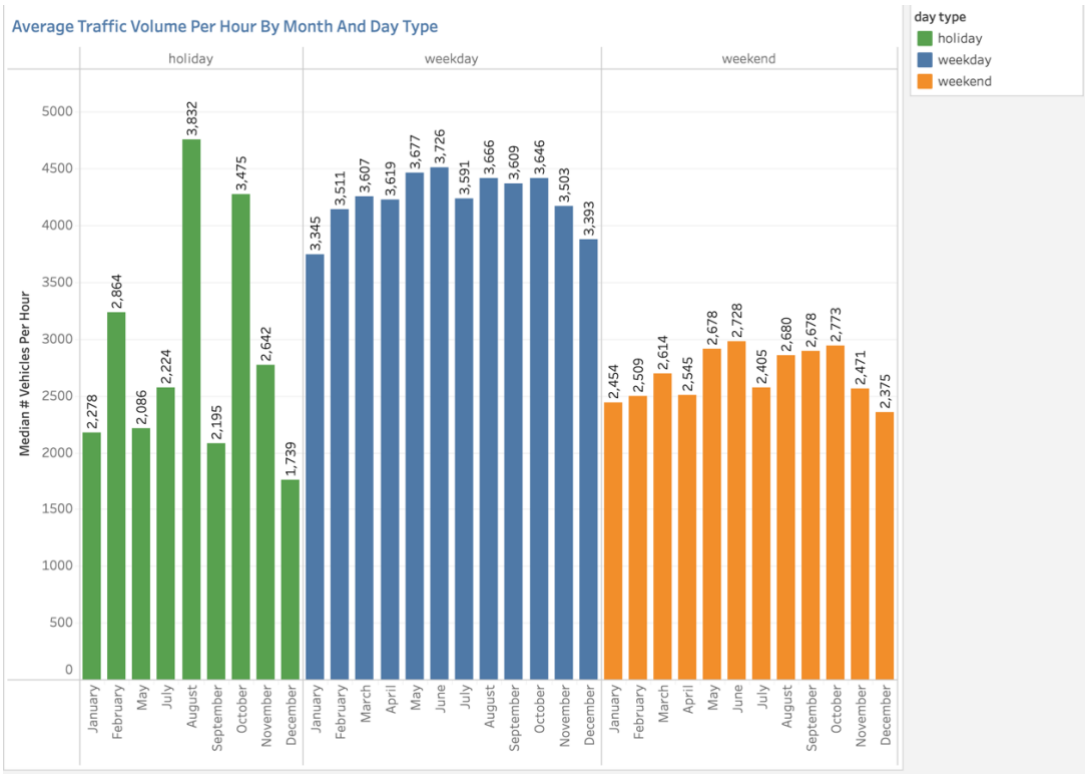
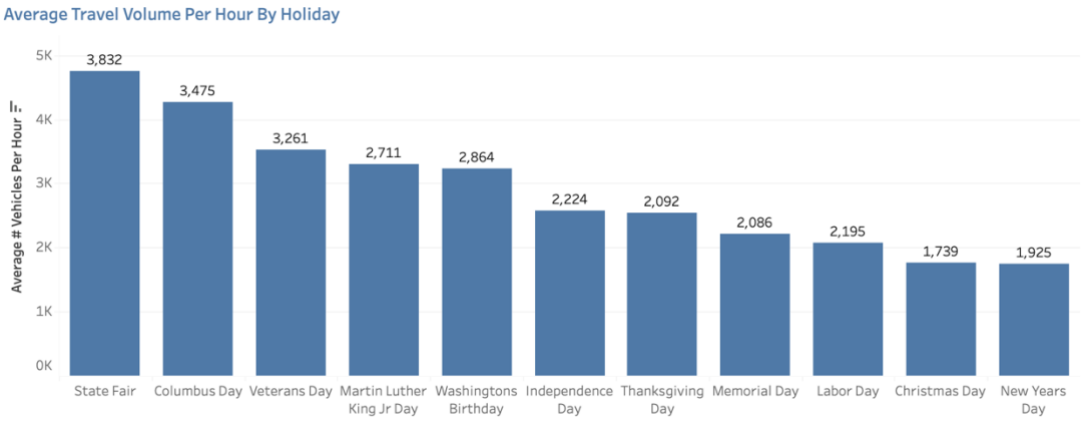


Figure 4.12 Average Traffic Volume Per Hour by Month and Day Type

The analysis of average and median traffic volumes for each holiday in Figure 4.13 provides valuable insights that can aid in traffic volume prediction. By examining the average traffic volumes, we can observe the variations in traffic patterns on different holidays. Certain holidays, such as State Fair and Veterans Day, have significantly higher average traffic volumes compared to others. Incorporating this knowledge into prediction models allows for better anticipation of increased traffic volume during these holidays, enabling more accurate forecasts. Similarly, the examination of median traffic volumes provides information about the central tendency of traffic volumes on holidays. Holidays like Labor Day and Memorial Day have higher median traffic volumes, indicating a concentrated peak in traffic during those periods. This insight can be utilized to adjust prediction models and account for the expected surge in traffic during such holidays.



Median Travel Volume Per Hour By Holiday

holiday (adj)	
Christmas Day	1,758
Columbus Day	4,277
Independence Day	2,571
Labor Day	2,081
Martin Luther King Jr Day	3,307
Memorial Day	2,211
New Years Day	1,738
State Fair	4,756
Thanksgiving Day	2,533
Veterans Day	3,533
Washingtons Birthday	3,239

Figure 4.13 Average and Median traffic volumes for each holiday

By considering the insights from both average and median traffic volumes, prediction models can be enhanced to better capture the variations in traffic patterns

during different holidays. This, in turn, enables more accurate predictions of traffic volume, allowing transportation planners and authorities to optimize their traffic management strategies and effectively allocate resources to ensure smooth traffic flow and minimize congestion during peak holiday periods.

In conclusion, the bivariate analysis of traffic volume based on hours, holidays, and months provides valuable insights into the variations and patterns in traffic volume. Figure 4.14, the graphical representation of the analysis, demonstrates how traffic volume differs across different factors.

The analysis reveals that traffic volume varies significantly by hour, with peak hours exhibiting higher volumes compared to non-peak hours. This information can help in predicting and managing traffic flow during specific times of the day, allowing for efficient allocation of resources and improved traffic management strategies. Furthermore, the examination of traffic volume on holidays highlights the impact of these special days on traffic patterns. Certain holidays, such as State Fair and Veterans Day, exhibit significantly higher traffic volumes, while others show more moderate or lower volumes. This understanding can aid in predicting traffic demand and implementing appropriate measures to handle increased traffic during specific holidays. Moreover, the analysis of traffic volume by month uncovers seasonal variations and patterns. Different months exhibit distinct traffic volume characteristics, influenced by factors such as weather conditions and cultural events. Incorporating this knowledge into prediction models can enhance their accuracy in forecasting traffic volume, enabling better planning and management of transportation resources.

Overall, the bivariate analysis sheds light on the complex relationship between traffic volume and various factors. The insights gained from this analysis contribute to the development of an effective traffic prediction model.

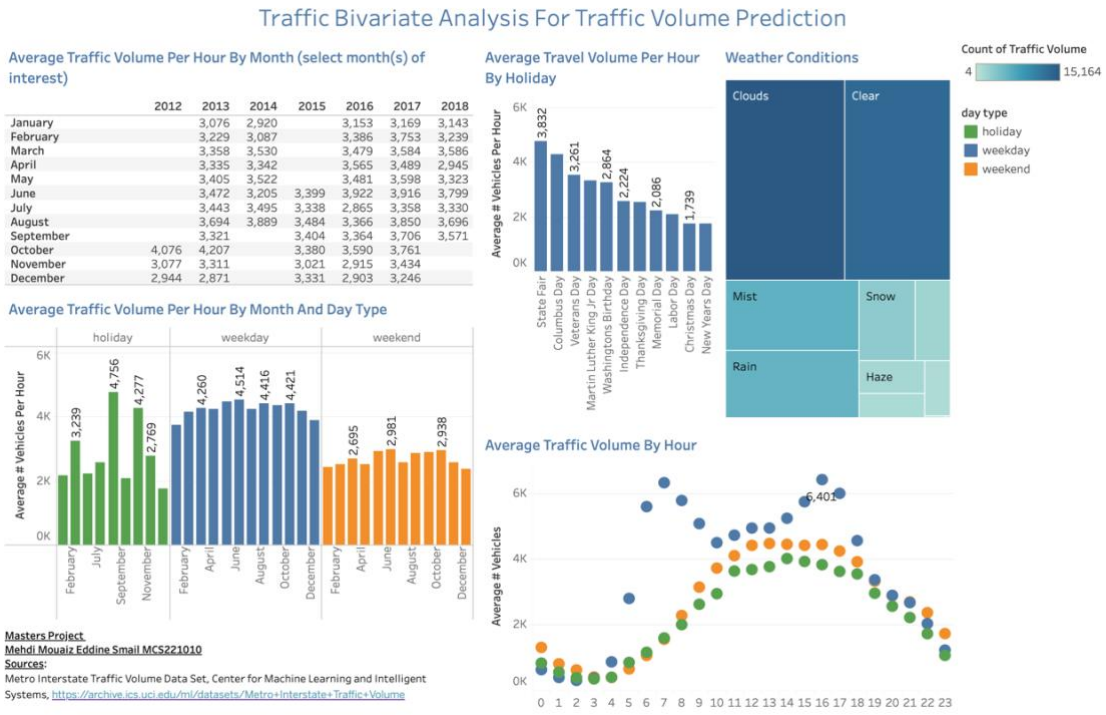


Figure 4.14 Initial Dashboard

4.3 Feature Importance

In this Section, the aim is to prepare the data in a way so that it can be fed into a random forest regressor.

Starting with the 'holiday' column, a simple function is created to categorize the column as 'Holiday' and 'None'. This is achieved by mapping the function over the column, effectively reducing the different holiday categories into a binary distinction. This simplification will allow the random forest to easily identify the impact of holidays on traffic volume.

As for the 'date_time' column, it is initially parsed using the pandas function pd.to_datetime(), turning it into a more manageable datetime format. From this parsed column, new features are derived: 'Year', 'Month', 'Weekday', and 'Hour'. These features capture the temporal elements of the dataset and are expected to provide valuable insights for traffic volume prediction. The 'Hour' column is also further categorized into four segments of the day - 'dawn', 'morning', 'afternoon', and 'night'. This process allows for understanding of the diurnal pattern in the data.

Before these transformations, any duplicate rows in the dataset are removed to ensure the uniqueness of each record, further enhancing the quality of the data fed into the model.

Continuing, the 'weather_description' column is removed due to its redundancy with 'weather_main'. While 'weather_description' offers more detailed information, it closely mirrors the information provided by 'weather_main'. To identify the importance of each feature, this fine-grained level of detail might not be necessary and could potentially introduce noise into the random forest. As such, removing this column allows for a cleaner and more streamlined dataset.

Similarly, the 'rain_1h' and 'snow_1h' columns are also dropped from the dataset. Despite their potential relevance, the distribution of these columns was observed to be heavily skewed towards zero during the exploratory data analysis. This indicates that they have limited variability and might not contribute significantly to predicting traffic volume. The 'weather_main' column already provides a broad understanding of whether it's raining or snowing, thereby reducing the need for these columns. Consequently, the data cleaning process leaves the dataset as can be seen in Figure 4.15 with the 'holiday', 'temp', 'clouds_all', 'weather_main', 'traffic_volume', 'Year', 'Month', 'Weekday', and 'Hour' columns. These streamlined and cleaned data now serve as a robust foundation to determine the most important features in the dataset.

	holiday	temp	clouds_all	weather_main	traffic_volume	Year	Month	Weekday	Hour
date_time									
2012-10-02 09:00:00	None	288.28	40	Clouds	5545	2012	10	1	9
2012-10-02 10:00:00	None	289.36	75	Clouds	4516	2012	10	1	10
2012-10-02 11:00:00	None	289.58	90	Clouds	4767	2012	10	1	11
2012-10-02 12:00:00	None	290.13	90	Clouds	5026	2012	10	1	12
2012-10-02 13:00:00	None	291.14	75	Clouds	4918	2012	10	1	13

Figure 4.15 A display of the final dataset

The Feature Importance section details an analysis to ascertain the relative influence of each feature on the traffic volume outcome. This process commences by separating features from the traffic volume data, which then transforms Label Encoding and One Hot Encoding method for any categorical data present. A Random Forest Regressor, a machine learning algorithm lauded for its robustness and interpretability, is then employed to evaluate the importance of each feature.

The relative importance of each feature is visually represented through a bar graph in Figure 4.16, plotted using the feature_importances_ attribute of the Random Forest model. In the assessments conducted on the Label Encoded dataset, the 'Hour' feature emerges as the most significant determinant of traffic volume. This observation corroborates the common urban phenomenon of increased traffic volumes during peak hours.

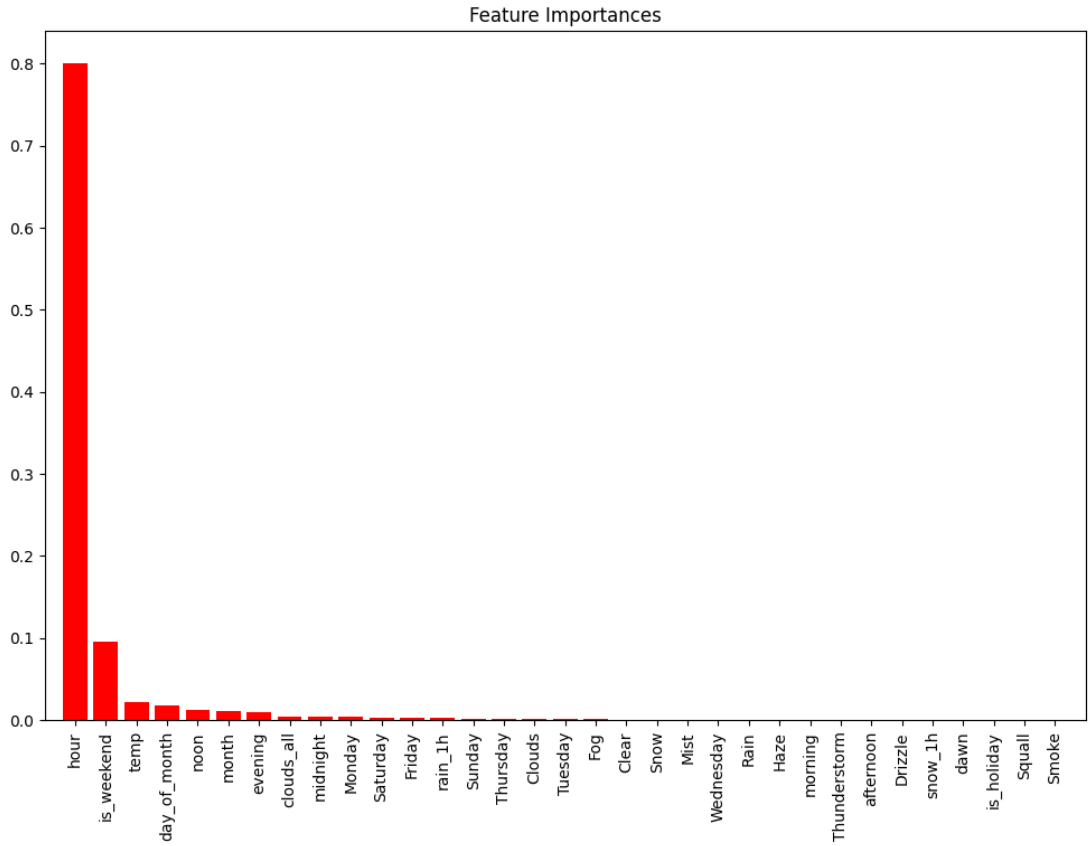


Figure 4.16 Feature Importance

The feature importance analysis derived from the Random Forest Regressor serves as a valuable tool for understanding which variables hold the most sway in

traffic volume predictions within a tree-based framework. When designing a dashboard that provides traffic insights, the prominence of certain features can guide the layout and presentation of information. For example, if 'hour' and 'weekend' emerge as key predictors, the dashboard might highlight data visualizations that focus on these aspects, enabling users to filter and view traffic patterns based on time of day and weekends.

In terms of operational efficiency, such analysis can direct efforts and resources towards the most impactful data collection efforts. If, for instance, temporal features like 'hour' display higher importance than weather conditions, this might suggest an emphasis on ensuring the accuracy and granularity of time-based data over weather data. This focus can assist stakeholders in making informed decisions, potentially leading to better traffic management strategies and urban planning initiatives. For stakeholders and end-users, understanding the primary influencers of traffic volume—such as peak hours and weekend trends—can be crucial. Urban planners and traffic authorities might leverage this knowledge for infrastructure development and congestion management, while drivers could benefit from real-time updates that help them avoid traffic hotspots. However, it's important to recognize the limitations of applying insights from a Random Forest model directly to the development of a Temporal Convolutional Network (TCN). TCNs are adept at handling temporal sequences and can uncover patterns across different scales of time through their convolutional layers. This means that a feature deemed less important by Random Forest could play a pivotal role in a TCN due to its interaction with other features over time. Consequently, while the feature importance from Random Forest can inform the initial stages of TCN development, particularly in high-dimensional datasets, it should not be the sole criterion for feature selection but it’s not the case for the dataset in this project. Furthermore, features that appear to have minor importance in a Random Forest model might participate in significant interactions when viewed through the lens of a TCN, which can capture complex, non-linear relationships and temporal dynamics that tree-based models may overlook. Therefore, it's imperative to evaluate the significance of each feature within the specific modeling context of a TCN.

In summary, while feature importance from Random Forest models can offer preliminary guidance and influence dashboard design and data collection strategies, it is not definitive for TCN model development.

4.4 Conclusion

Chapter 4 has provided a comprehensive overview of the dataset, elucidating the features and their impact on the target variable - traffic volume. The Exploratory Data Analysis (EDA) stage allowed for a deeper understanding of the dataset's structure and composition, leading to the identification of potential outliers and anomalies, and prompting the necessary data cleaning actions. Detailed analysis on variables such as 'holiday', 'temperature', 'rain', 'snow', 'clouds_all', and 'weather_main' were undertaken, and certain modifications such as removing outliers from 'rain_1h' and 'snow_1h' columns and converting 'date_time' column into time-stamp format and that is to be able to index the data into the correct sequence.

Feature importance was determined through the application of a Random Forest Regressor, with 'Hour' emerging as a key determinant of traffic volume. However, it's critical to note that this observation will need further validation within the context of the Temporal Convolutional Network (TCN) model to be utilized in future steps. Furthermore, the identification of feature importance, such as the significance of the 'Hour' variable, provides valuable insights for businesses. By understanding the temporal patterns and their impact on traffic volume, businesses can optimize their operations accordingly. For example, transportation companies can adjust their schedules and routes based on the predicted traffic volume during different hours of the day.

Based on these comprehensive assessments and transformations, the dataset has been prepared in a manner that should be optimal for predicting traffic volume. It comprises a blend of temporal and weather-related features which, in their processed and enhanced form, are expected to feed effectively into the TCN model. The dataset's rich temporal details coupled with diverse weather conditions make it an ideal

candidate for time-series traffic volume prediction. This carefully curated dataset, in tandem with the advanced modeling techniques to be deployed in subsequent chapters, lays a robust foundation for the objective of accurate traffic volume prediction.

CHAPTER 5

MODEL DEVELOPMENT

5.1 Introduction

This chapter outlines the development of a time series model for predicting traffic volume. It begins with the essential Input modeling steps and progresses to constructing a Temporal Convolutional Neural Network (TCN).

Key objectives include the correct implementation of the TCN architecture, optimization of its hyperparameters, and validation of its forecasting accuracy. The Approach detailed here will lay the groundwork for a comparison with established time series models like LSTM and GRU, aiming to highlight the advantages of using a TCN for traffic volume prediction.

5.2 Input Modeling

Input modeling is a critical phase in the development of any deep learning model, and it plays a pivotal role in ensuring the model's accuracy and consistency. Furthermore, the following are the preprocessing steps that were conducted to shape the input of the TCN model all the steps are a replication of the pre-processing done by (Das, 2023). Moreover, all the features will be used and the dataset has 8 features besides the traffic volume feature.

(a) Data sequencing

- Sorting by date/time: the data was sorted chronologically based on the ‘date_time’ column, ensuring the correct sequence of events

- Setting date/time index: the ‘date_time’ is set as the data frame index, and that’s to anchor each data point to a specific point in time.

(b) Data Scaling (Normalization)

The MinMaxScaler is applied to normalize numerical values to a range between 0 and 1. Furthermore, Normalization is critical for deep learning models like TCN to prevent features with larger scales from dominating the learning process. It ensures a consistent impact from all features.

(c) Time Series Windowing

Time series windowing is a crucial technique for preparing data for temporal models, such as TCNs, LSTMs, or GRUs. It involves taking a sequence of data points and creating a set of input/output pairs that the model can learn from. Each input is a window of consecutive data points, and the corresponding output is the data point immediately following the window.

In this case, the windowing process can be described as follows:

1. Window Size (Past Observation): The window size is set to 12, meaning that 12 consecutive hours of data will be used to predict the next hour's traffic volume. However, during the benchmarking the window size will be set to 6 as well as 24 along with 12 for benchmarking against LSTM and GRU.

2. Number of Windows: The total number of windows depends on the dataset size minus the window size, minus 1 (for the prediction step).

3. Sliding the Window: The window starts from the beginning of the dataset and slides one time step forward for each new input/output pair until it reaches the end of the dataset.

4. Output Data Point: For each window of 12 hours, the following hour's traffic volume is used as the output label for the model to predict.

Here is a simplified representation of the process in Figure 5.1:

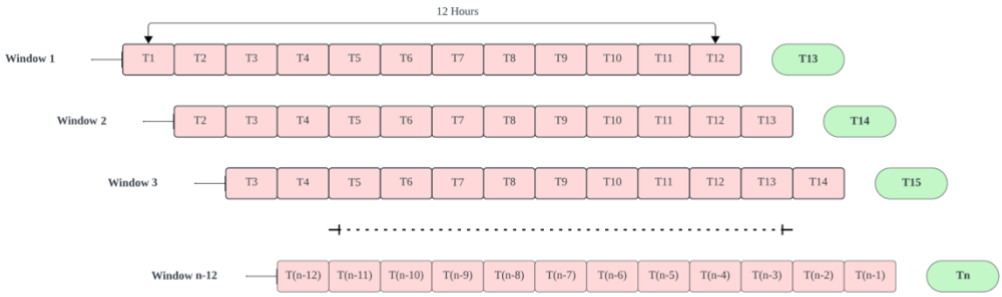


Figure 5.1 Windowing Process

Figure 5.1 represents the windowing technique applied to the dataset, commonly employed in preparing data for predictive modeling with neural networks such as TCNs and, LSTMs.The process involves creating 'windows' of consecutive data points, each window being a fixed size, in this case, 12 hours long. For instance, the first window, 'Window 1,' contains the first 12 data points (T1 to T12), which the model uses to predict the subsequent data point (T13). As the process advances, the window slides forward by one time unit, creating 'Window 2' with data points T2 to T13 to predict T14, and so on, until the end of the series is reached. This sliding window continues, with each step advancing one time unit and the subsequent data point serving as the prediction target for the model. The last window in the sequence uses the data points from T(n-12) to T(n-1) to predict the final point, Tn. This methodological approach ensures that the model can capture temporal dependencies, allowing for an accurate forecast of traffic volume based on historical data patterns.

(d) Data Splitting for Training and Testing:

The dataset encompasses a period from 2012 to 2018. Furthermore, it has been segmented into distinct subsets for training, validation, and testing purposes, with the following details:

1. Training Set:

- Data Period: All records from 2012 up to the end of 2017.

- Usage: This dataset forms the primary basis for training the Temporal Convolutional Network (TCN). It encompasses a broad range of historical data, allowing the model to learn diverse temporal patterns, seasonal trends, and traffic behaviours.

- Proportion: The dataset is subdivided, with 90% of this period's data allocated to actual model training.

2. Validation Set:

- Data Period: The remaining 10% of the data from the same 2012 to 2017 period.

- Usage: This subset is used to validate the model's performance during the training phase. It helps in fine-tuning the model parameters and provides an early indication of how well the model generalizes to new data.

- Proportion: 10% of the pre-'2018-01-01' data, distinct from the training set.

3. Testing Set:

- Data Period: All records from January 1, 2018, onwards in Figure 5.2.

- Usage: This set is crucial for evaluating the model's forecasting performance on completely unseen data. It simulates a real-world scenario where the model is required to make predictions on future data that was not available during the training phase.

- Proportion: 100% of the post '2018-01-01' data.

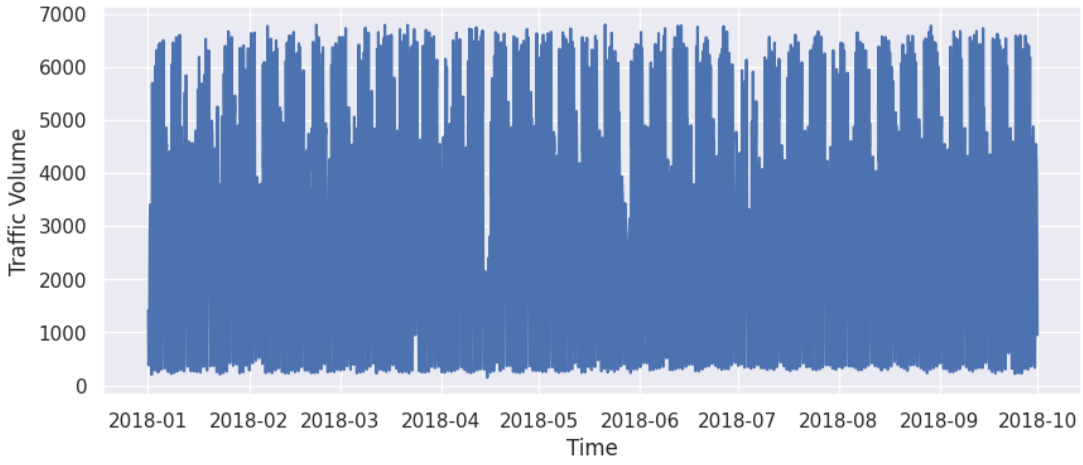


Figure 5.2 Test dataset time frame

By following this data splitting strategy, it ensures that the TCN model is trained on a comprehensive historical dataset, fine-tuned and validated on a relevant subset from the same period, and finally assessed on completely new data to verify its forecasting ability in a real-world context. This structured approach is critical for developing a robust and reliable traffic volume prediction model.

Each of these pre-processing steps is designed Shape the input data that would be fed into the TCN . Moreover, They collectively address issues like missing values, outliers, varying scales, and categorical data representation, ensuring the model receives clean, normalized, and relevant information for effective learning and prediction.

5.3 Experimentation

In the pursuit of developing an effective Temporal Convolutional Network (TCN) for traffic volume prediction, an experiment is conducted with two distinct hyperparameter configurations. This section compares the baseline and adjusted TCN models and explores if there is a necessity for hyperparameter optimization in enhancing model performance.

Baseline model:

Table 5.1 Baseline Model Configuration

Parameter	Value
Filters	64
Kernel size	3
Dilations	[1, 2, 4, 8, 16, 32, 64, 128, 256]
Stacks	1
Skip connection	No
Learning Rate	0.0001
Dropout rate	0.05
Batch size	16
Training Epochs	100

Table 5.1 showcases the Hyperparameter Configuration for the Baseline Model and Values for Each Parameter were Chosen Randomly Since there are No Default Values to Start with.

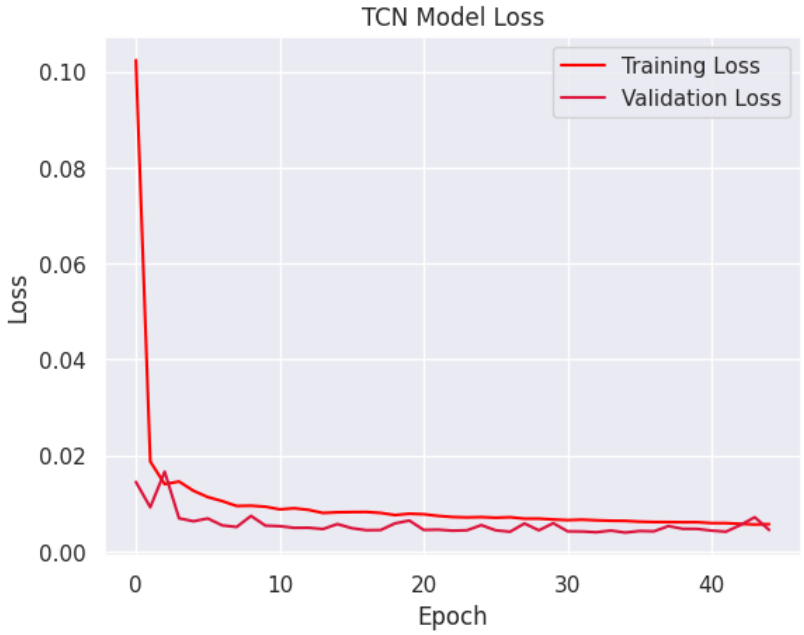


Figure 5.3 Training loss vs validation loss for the baseline model

Figure 5.3 provides a visual representation of the training and validation loss throughout the training process over the specified number of epochs, which shows validation loss spiking above training loss at certain epochs, suggesting that the model's performance on the validation set is inconsistent. This could be due to the model's inability to generalize well or due to the variability in the validation dataset that the model is not accounting for during training.

Table 5.2 Baseline Model Evaluation

Data	MAE	MSE	RMSE
Training	0.05213	0.005692	0.07545
Validation	0.04242	0.0037940	0.071596

The baseline model's performance, as demonstrated in Table 5.2.2, indicates a discrepancy between training and validation results. Specifically, the training Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) are higher than those for the validation dataset. This suggests that the model may not be fully capturing the complexity of the training data, hinting at a potential underfitting issue.

Given that the validation loss occasionally spikes above the training loss, it can also be inferred that the model's current hyperparameter setting may not be robust enough to handle the variability in the validation set. This fluctuation in validation loss could lead to less reliable predictions when the model is exposed to new, unseen data. Moreover, The need to adjust the parameters arises from the desire to improve the model's ability to learn from the training data without losing its generalization capabilities.

Adjusted model:

Table 5.3 Adjusted Model Configuration

Parameter	Value
Filters	256
Kernel size	7
Dilations	[1, 2, 4, 8]
Stacks	2
Skip connection	Enabled
Learning Rate	0.001
Dropout rate	0.3
Batch size	64
Training Epochs	100

Adjustments to the baseline (TCN) model's hyperparameters were undertaken to improve performance. Key parameters including filter counts, kernel sizes, and dilation rates were re-evaluated in Table 5.3. The goal was to rectify potential underfitting from the baseline model and to improve the model’s ability to generalize by increasing the model complexity. The modified, adjusted model aimed to achieve better error metrics and a steadier alignment of training and validation loss.

Figure 5.4 Training loss vs validation loss for the adjusted model

In Figure 5.4, the adjusted TCN model shows an initial spike in validation loss which quickly aligns with the training loss, indicating a brief period of learning adjustment. Subsequently, the model demonstrates stable learning with the validation loss closely tracking the training loss, suggesting effective generalization without overfitting.

Table 5.4 Adjusted Model Valuation

Data	MAE	MSE	RMSE
Training	0.04722	0.005145	0.065457
Validation	0.04480	0.004854	0.071083

The adjusted model's metrics in Table 5.4 exhibit a higher RMSE on the validation data compared to the training data, which usually suggests that the model is slightly overfitting to the training data. This is because the error on the validation set is larger than the error on the training set. However, the differences are not substantial, indicating that the model generalizes reasonably well.

The marginal improvements and inconsistencies observed between the two models raise an important point regarding the optimization of hyperparameters. The existing hyperparameters, while effective to a degree, may not be the optimal set for the best possible performance. The disparity between the training and validation metrics, especially notable in the baseline model, points to the potential benefits of further tuning. Systematic optimization techniques, like Bayesian optimization, could be instrumental in navigating the complex hyperparameter space. Such optimization aims to strike a delicate balance between various parameters to enhance the model's accuracy and ensure its effective generalization to new datasets. This is crucial in creating a model that not only fits the training data well but also maintains robust performance when faced with unseen data.

CHAPTER 6

MODEL OPTIMIZATION AND RESULTS DISCUSSION

6.1 Introduction

This chapter serves as an in-depth exploration of the optimization and performance evaluation of the Temporal Convolutional Network (TCN) model developed for traffic volume prediction. This chapter will detail the rigorous process of hyperparameter optimization, employing Bayesian techniques to fine-tune the TCN model to achieve the lowest possible validation loss. It will delve into the specifics of the optimization process, elucidate on the results obtained, and present a comparative analysis of the optimized TCN model against established models like LSTM and GRU. Furthermore, this chapter will showcase the final dashboard developed from the model's findings, illustrating the practical application and potential integration of the TCN model into real-world traffic management systems. The chapter aims to provide a comprehensive analysis of the TCN model's performance, concluding with reflections on the implications of these findings for enhancing traffic volume prediction and the strategic computational approaches employed throughout the project.

6.2 Hyperparameters Optimization

(a) Optimization Process:

An objective function was crafted to compile, train the TCN model with a set of hyperparameters, and output the validation loss. The aim was to minimize this loss across trials, each trial being an execution of the function with a particular set of hyperparameters suggested by the Bayesian optimization algorithm.

The search space for hyperparameters included in Figure 6.1:

- nb_filters: Quantities of convolutional filters were selected from [16, 32, 64, 128], providing a range of model capacities, from a more straightforward model to a highly complex one that could capture more nuanced patterns in the data.

- kernel_size: The size of the convolutional kernel is defined as an integer between 2 and 9, determining the temporal scope each filter covers, or in other words how many past time steps each filter would consider simultaneously.

- nb_stacks: The number of residual blocks stacked, chosen from [1, 2, 3], allowed exploration of model depth.

- dilations: Exponential growth rates for dilated convolutions were set from 1 to 8, influencing the model's receptive field growth as the network deepens.

- dropout_rate: Varied uniformly between 0.0 and 0.5 to identify optimal regularization.

- learning_rate: The learning rate for the Adam optimizerVaried log-uniformly between 0.0001 and 0.01 to find a learning rate that balances speed and stability of learning.

- batch_size: Batch sizes [16, 32, 64, 128, 256] were tested to see the effect on the optimization landscape, and to test various degrees of stochasticity and convergence rates.

- use_skip_connections: This boolean parameter was included to assess the impact of skip connections on the learning process.

```
space = {
    'nb_filters': hp.choice('nb_filters', [16, 32, 64, 128]),
    'kernel_size': hp.quniform('kernel_size', 2, 9, 1),
    'nb_stacks': hp.choice('nb_stacks', [1, 2, 3]),
    'dilations': hp.quniform('dilations', 1, 8, 1),
    'dropout_rate': hp.uniform('dropout_rate', 0.0, 0.5),
    'learning_rate': hp.loguniform('learning_rate', np.log(0.0001), np.log(0.01)),
    'batch_size': hp.choice('batch_size', [16, 32, 64, 128, 256]),
    'use_skip_connections': hp.choice('use_skip_connections', [True, False])
}
```

Figure 6.1 Search space for the hyperparameters

To manage the optimization trials, the Trials object was utilized from 'hyperopt', which records all the experiments. Checkpointing this object to a file (trials_checkpoint.pkl) after every iteration ensured that in case of any interruptions, the optimization could resume from the last saved state without losing progress.

(b) Results of the optimization

After an extensive hyperparameter optimization process involving Bayesian techniques, the Temporal Convolutional Network (TCN) model has been fine-tuned over 70 trials, which took over 48 hours due to computational constraints. The final set of hyperparameters obtained from these trials, which yielded the lowest validation loss in Figure 6.2, are as follows:

```
Best hyperparameters:
    batch_size: 16
    dilations: 32
    dropout_rate: 0.15774489618820273
    kernel_size: 7.0
    learning_rate: 0.0008364026012348232
    nb_filters: 128
    nb_stacks: 1
    use_skip_connections: True
```

Figure 6.2 Optimal Hyperparameters

These parameters indicate a configuration that strikes a balance between the model's complexity and its ability to generalize from the training data. Specifically:

1. The selected batch size implies that the model benefits from more frequent updates, possibly enhancing its ability to converge towards a more accurate solution.

2. A dilation rate of 32 allows the network to capture long-term dependencies effectively, which is critical for time series forecasting.

3. The dropout rate of approximately 15.8% helps in preventing overfitting, ensuring that the network maintains a degree of robustness.

4. A kernel size of 7 allows the model to capture a broad context of temporal information in each layer.

5. The learning rate suggests an optimal speed of convergence, avoiding overshooting the minimum.

6. A higher number of filters (128) provides the model with ample capacity to process and represent the nuances in the data.

7. A single stack of residual blocks indicates that increasing the network's depth may not necessarily contribute to better performance for this particular task.

8. The use of skip connections confirms their effectiveness in improving model training and performance for this dataset.

These insights not only affirm the effectiveness of Bayesian optimization over more traditional hyperparameter tuning methods like grid search but also highlight the critical role each hyperparameter plays in the model's forecasting performance. The impressive loss value obtained serves as evidence of the TCN model's attuned forecasting abilities, tailored to the specific dynamics and intricacies of the given time series data.

(c) The optimized model metrics in comparison to the baseline and adjusted models:

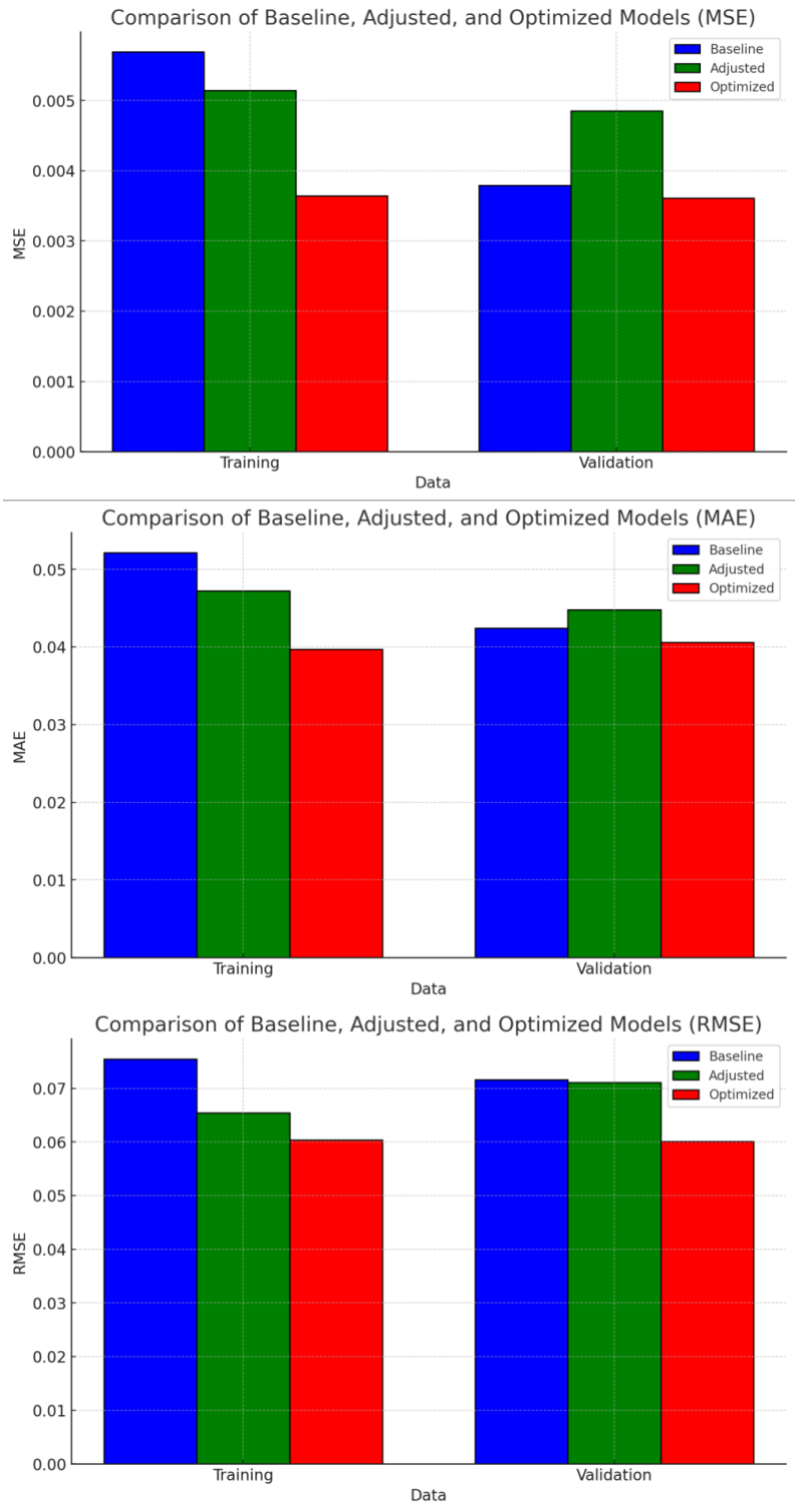


Figure 6.3 Optimal Hyperparameters

The comparative plots in Figure 6.3 illustrate the performance metrics (MAE, MSE, RMSE) for the baseline, adjusted, and optimized Temporal Convolutional Network (TCN) models. These visualizations succinctly compare how each model version fares in terms of predictive accuracy on both training and validation datasets.

The baseline model, with its initial hyperparameter settings, serves as the starting point. It exhibits higher error metrics compared to the other models, which suggests there is considerable room for improvement, especially considering the inconsistency observed between training and validation losses. The adjusted model, with its re-evaluated hyperparameters, shows improved performance over the baseline. The adjustments made appear to have reduced the errors slightly, indicating better learning from the training data and a degree of enhanced generalization. Despite this, there remains a small gap in RMSE between training and validation, hinting at possible overfitting.

Finally, the optimized model, after extensive hyperparameter tuning using Bayesian optimization techniques, achieves the lowest error metrics, demonstrating the efficacy of systematic hyperparameter optimization. This model's errors are relatively balanced between training and validation, suggesting a model that has learned the complexities of the dataset well and can generalize effectively to unseen data.

6.3 Final Model Results

The implementation of optimized hyperparameters in the Temporal Convolutional Network (TCN) has yielded the following results:

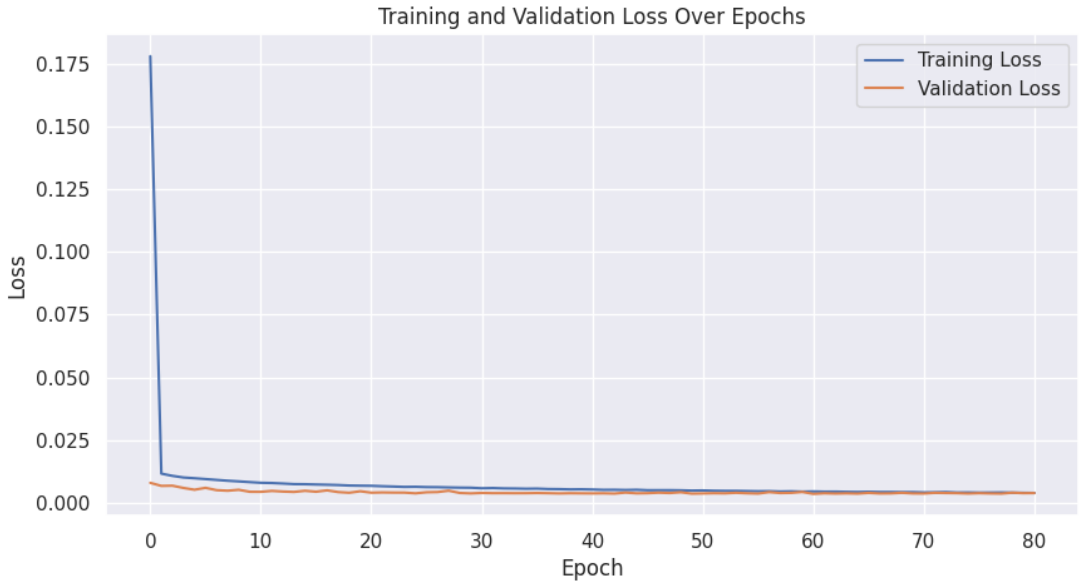


Figure 6.4 Training Loss vs Validation Loss for the Optimized Model

Figure 6.4 presents the training and validation loss curves for the optimized model, essential in assessing the model's learning and generalization capabilities. Throughout the epochs, both curves exhibit a pronounced decline, indicative of the model's ability to learn from the training data. Notably, the validation loss swiftly aligns with the training loss, suggesting the model is not overfitting. The convergence of the two curves with minimal gap demonstrates the model's aptitude to generalize well to unseen data, a desirable trait in predictive modeling. This graph assures that the model is well-calibrated, striking a balance between learning the training data patterns and maintaining flexibility for validation data.

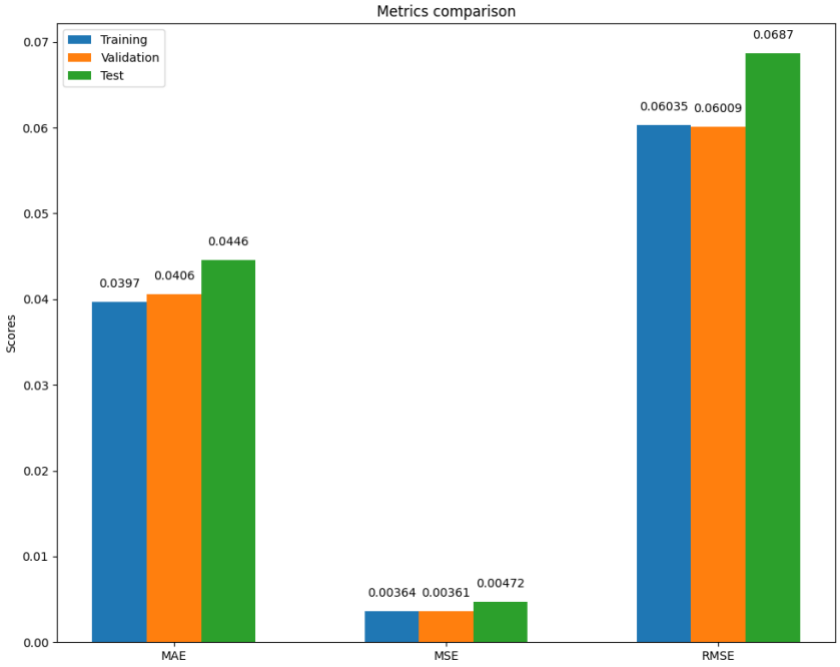


Figure 6.5 Performance Metrics of the Optimized Model

Figure 6.5 illustrates the model's performance metrics across training, validation, and test datasets. The Training Mean Absolute Error (MAE) of 0.0397 signifies the model's predictions are within a 3.97% margin from the actual training data on average. The Training Mean Squared Error (MSE) at 0.00364, and the Root Mean Squared Error (RMSE) at 0.06035, both affirm the model's high accuracy, with RMSE providing insight into the magnitude of error. For the unseen validation data, the Validation MAE at 0.0406 closely mirrors the training MAE, implying effective model generalization. Similarly, the Validation MSE and RMSE at 0.00361 and 0.06009 respectively, maintain near-identical values to their training counterparts, further highlighting the model's consistent performance.

On the test dataset, however, the Test MAE increases slightly to 0.0446, suggesting the model encounters new data patterns not learned during training. This is further evidenced by a Test MSE of 0.00472 and an RMSE of 0.0687, which are higher than those for the training and validation sets but not to the extent that would raise a concern as it is only normal because test data is new to the model.

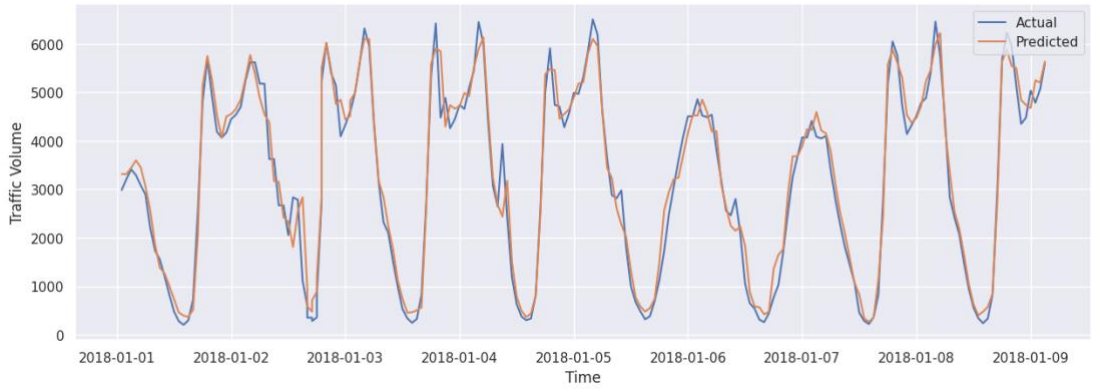


Figure 6.6 Predicted Versus Actual Volume by the Optimized Model

Figure 6.6 illustrates the predicted versus actual traffic volumes, offering a visual assessment of the model's predictive performance. A close match between the actual and predicted lines would signify a model that can accurately forecast traffic volumes, while significant deviations might highlight areas where model performance could be improved.

Overall, the results suggest that the optimization process has contributed to a well-performing model, with close correspondence between training, validation, and test metrics. The slight increase in test errors compared to training and validation warrants further investigation but does not diminish the model's overall predictive power. The optimization process appears to have identified a set of hyperparameters that enable the TCN to generalize effectively across different datasets.

6.4 Benchmarking

The performance of the Temporal Convolutional Network (TCN) model is assessed against two other prevalent models in time-series forecasting: Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU). Across various observation windows—6, 12, and 24 hours—our TCN model exhibits commendable stability in its predictive accuracy. For instance, in Figure 6.7 with a 6-hour window, the TCN's Mean Absolute Error (MAE) is at 0.04397 and it marginally increases to 0.0453 for the 24-

hour predictions. The Mean Squared Error (MSE) follows a similar pattern, reflecting a small rise from 0.00440 to 0.0043 across the time frames.

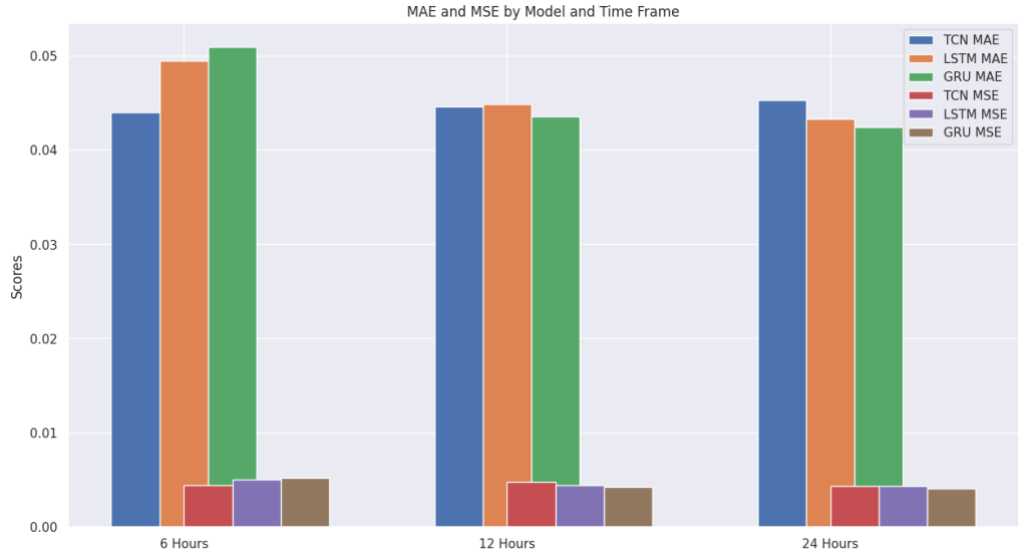


Figure 6.7 Benchmarks for All the Models Across Different Time Windows

In comparison, the LSTM model starts with a higher MAE of 0.0494 at the 6-hour interval but shows improvement over time, lowering to 0.0433 for the 24-hour window. The MSE mirrors this trend with a starting value of 0.005 that decreases to 0.0043. The GRU model, on the other hand, presents the highest initial errors—MAE of 0.0509 and MSE of 0.0052 at 6 hours—but its performance improves significantly at the 24-hour mark, showcasing the lowest errors among the models with an MAE of 0.0424 and an MSE of 0.0041.

These observations suggest that while the TCN model maintains a consistent performance across varying time frames, the LSTM and GRU models demonstrate a capacity for improvement and adaptability as they process more extensive historical data. The gradual decrease in error metrics for LSTM and GRU indicates their potential effectiveness for longer-term forecasts. However, the relatively steady error rates of the TCN model across all time frames suggest a robustness that may be preferable in scenarios where consistent performance is required regardless of the

observation window. Furthermore, the metrics for training and validation for the LSTM and GRU are not provided which makes the analysis.

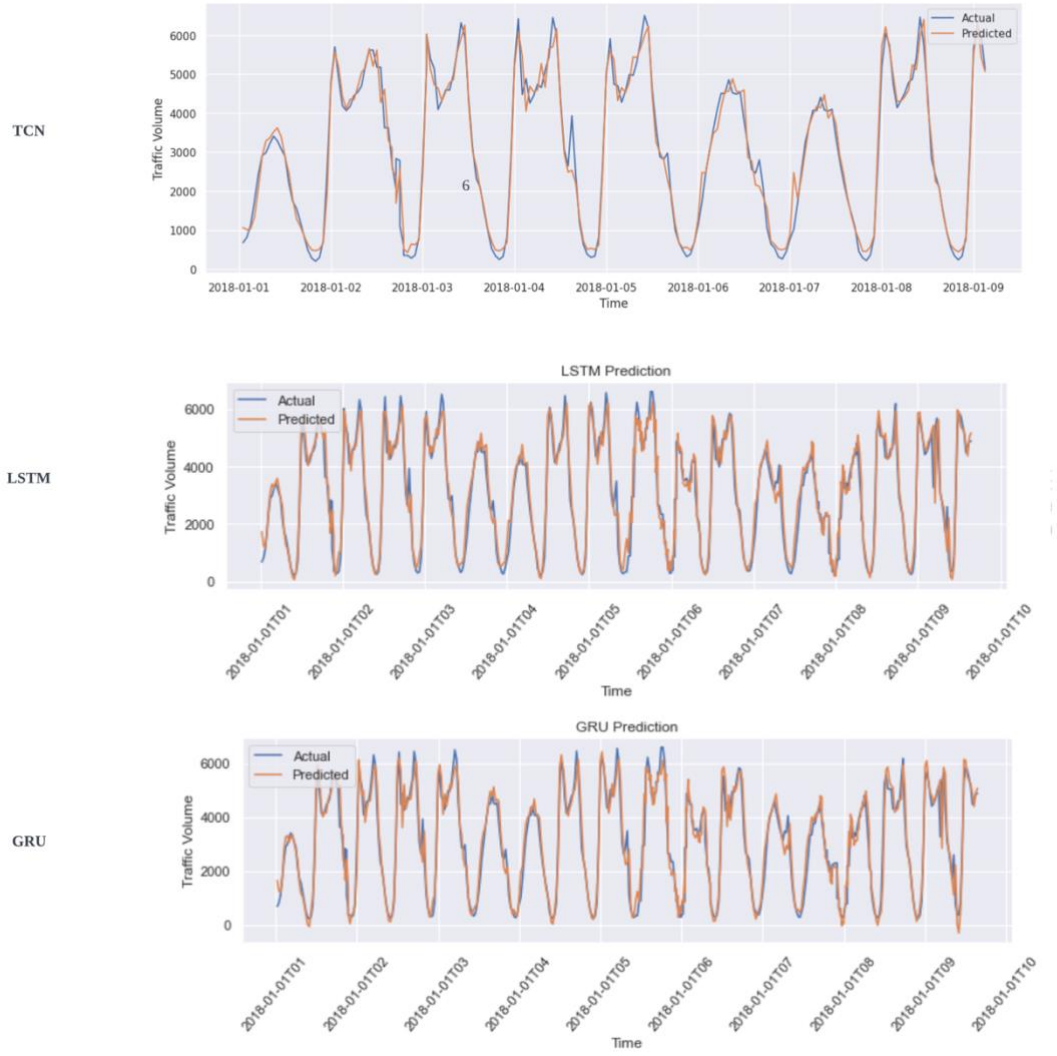


Figure 6.8 TCN vs LSTM and GRU (Das, 2023) Actual vs Predicted (6-hours)

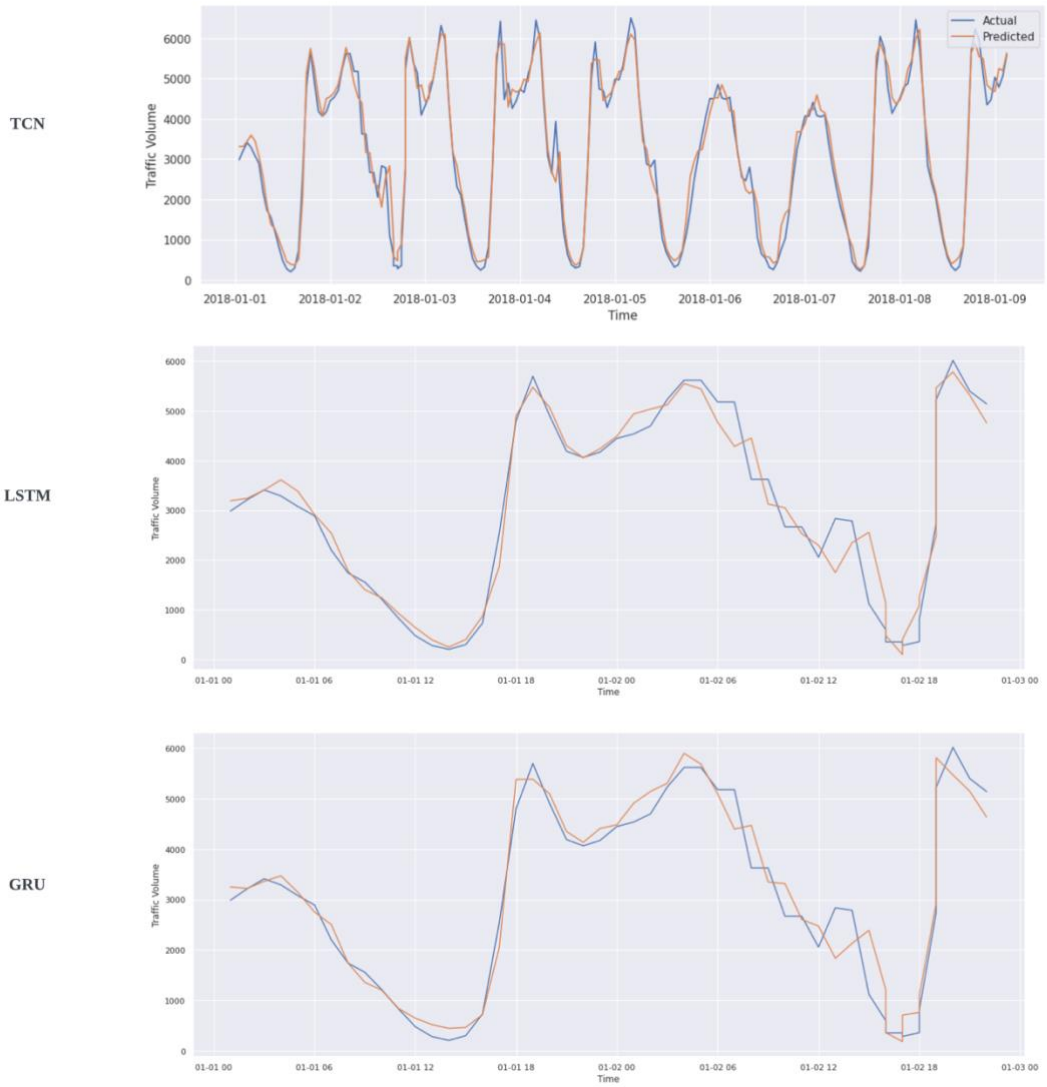


Figure 6.9 TCN vs LSTM and GRU (Das, 2023) Actual vs Predicted (12-hours)

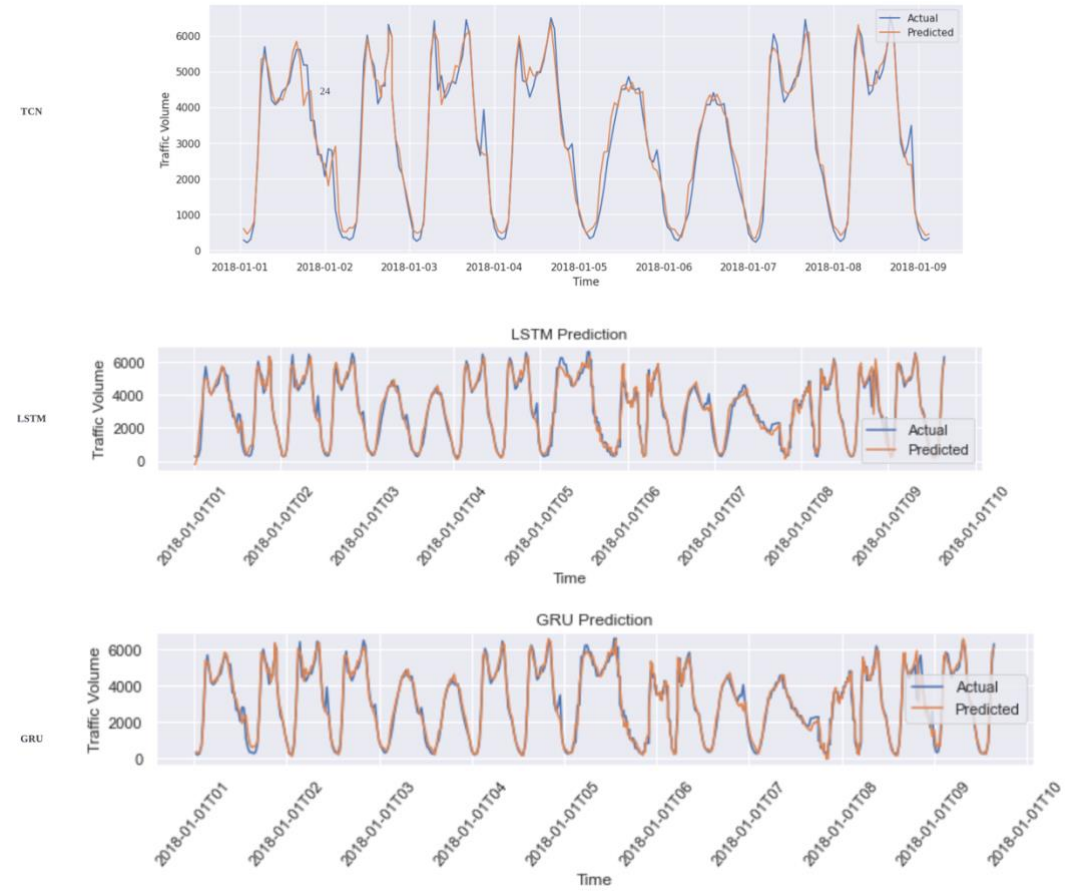


Figure 6.10 TCN vs LSTM and GRU (Das, 2023) Actual vs Predicted (24 -hours)

The TCN model, as depicted in Figure 6.8, shows a remarkable fidelity in tracing the actual traffic patterns across the 6-hour prediction window. This level of precision in the TCN model's predictions highlights its robust temporal understanding and its ability to effectively leverage the sequential nature of the data. The congruence between the predicted and actual values is indicative of the model's strength in capturing not only the trends but also the subtle nuances within the traffic data. Comparatively, as showcased in Figure 6.9, the LSTM and GRU models also demonstrate competent predictive performances within the 12-hour window. While the GRU model slightly surpasses the LSTM in aligning with the actual traffic flow, both models exhibit a commendable grasp of the temporal dynamics within the datasets. It is noteworthy, however, that their predictions are not as tightly coupled with the actual data as those of the TCN model.

As we extend the forecast to a 24-hour horizon in Figure 6.10, the TCN model consistently demonstrates a superior ability to generalize from the training data to unseen test data. Although the differences in accuracy metrics are subtle, they are consistent, underscoring the reliability and stability of the TCN model. Even in the 24-hour prediction window, the TCN model maintains its accuracy comparably with the LSTM and GRU models, suggesting that its architectural advantages may be as influential in scenarios that require capturing longer-term dependencies.

In summary, the comparative analysis of the TCN, LSTM, and GRU models reveals that while all models are competitive in their forecasting abilities, the TCN model exhibits a slight yet consistent edge in prediction accuracy. This comprehensive analysis demonstrates that the TCN model is not only competitive but also slightly more adept at short-term traffic volume prediction tasks, making it a valuable asset in traffic management systems that demand high precision for both short-term and medium-term forecasting. These insights pave the way for future explorations into how TCN models can be further optimized and potentially integrated into broader intelligent traffic systems to enhance predictive accuracy and contribute to more efficient urban mobility solutions.

6.5 Final Dashboard

The culmination of this project is encapsulated in the conceptualization of a comprehensive dashboard, illustrated in Figure 6.11, which serves as a prototype for the application of the traffic volume prediction model in a real-world scenario. It's important to note that while the dashboard is grounded in the findings of the model, it is an illustrative representation and not a real-time traffic management system.

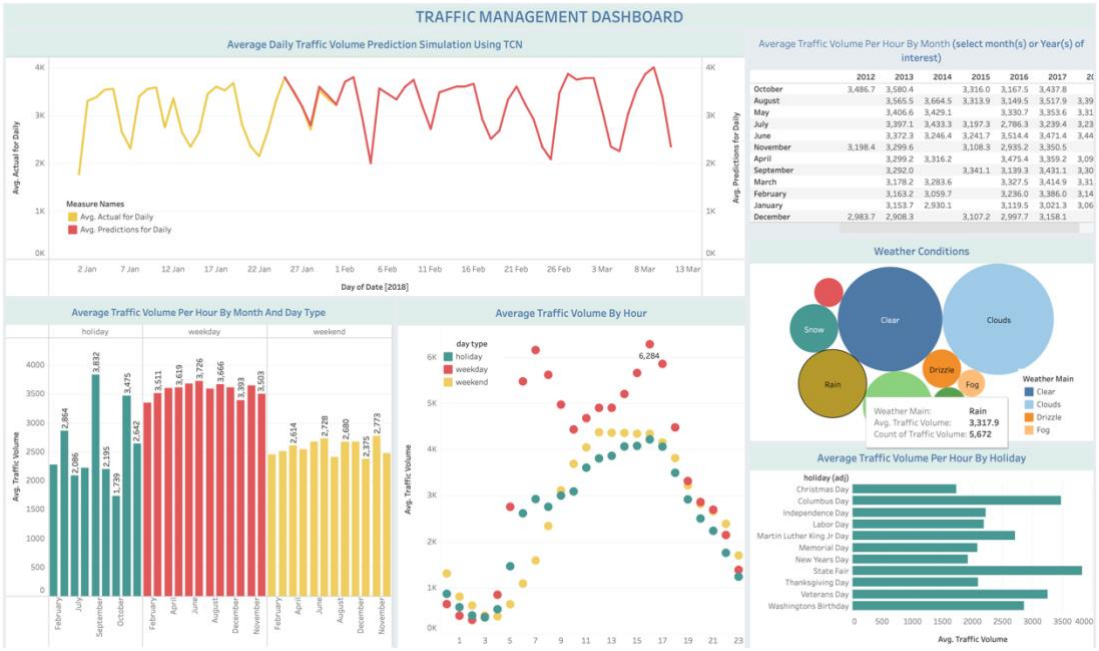


Figure 6.11 Final Dashboard

This dashboard is designed to provide a multi-faceted of traffic dynamics, integrating various data points to offer a rich, interactive user experience. The dashboard presents the average traffic volume per hour segmented by month, type of day, and hour, providing an intuitive understanding of traffic patterns at a glance. Weather conditions, a critical factor influencing traffic flow, are also incorporated, displaying the correlation between different weather scenarios and traffic volume .

The visualization includes a bivariate analysis, juxtaposing traffic volume against time-based variables, such as hours of the day and days of the week, which allows for a nuanced analysis of traffic trends. A separate module presents the count of traffic volume by weather conditions, illustrating how environmental factors can affect traffic density. Furthermore, the dashboard interface is crafted with the end-user in mind, featuring interactive elements that enable users to filter and drill down into specific data ranges or conditions. This interactivity facilitates targeted analysis, aiding in decision-making processes such as traffic management, and infrastructure planning. Moreover, while the model and its findings are based on historical data, the dashboard serves as a conceptual model illustrating how traffic volume prediction can be

dynamically represented. It demonstrates the potential for future integration with live data sources to provide real-time analytics and forecasts.

In its current form, the dashboard stands as a testament to the practical application of the Temporal Convolutional Network model developed in this research. It shows complex data and sophisticated models can be translated into accessible and actionable insights, underscoring the model's relevance and adaptability to real-world management scenarios.

6.6 Conclusion

The project journey culminated in the development of a Temporal Convolutional Network (TCN) model capable of predicting traffic volume with high accuracy. The model's robustness was evident in its comparative performance against established models like LSTM and GRU. The subsequent integration into a conceptual dashboard, demonstrated the model's potential for practical application, offering businesses a powerful tool for traffic management and planning.

This project also navigated through significant computational challenges. The constraints of the Google Colab runtime environment introduced a risk of job interruptions, which was mitigated by the strategic use of the checkpoints. This safeguard ensured progress, a crucial strategy given the limited computational resources. Furthermore, the optimization task was notably demanding, with Bayesian optimization presenting itself as a superior alternative to grid search. This approach allowed the project to overcome the computational barriers to some extent, leading to the successful identification of a set of effective hyperparameters.

The project findings underscore the importance of computational power in deep learning. While the TCN model achieved commendable results, the limitations posed by the computational platform highlighted the potential for even more significant advancements with access to more robust computational resources. Future work could expand upon the current project by exploring a broader hyperparameter

space, potentially uncovering more refined models that could further enhance prediction accuracy.

In conclusion, this project not only contributes to the field of traffic volume prediction through the application of TCNs but also the strategic maneuvering of computational constraints. It sheds light on the reality of conducting deep learning models with limited resources and emphasizes the importance of intelligent optimization techniques. The insights and methodologies presented in this project lay a foundation for future research, driving forward the field of traffic management and urban planning.

REFERENCES

Alajali, W., Zhou, W., Wen, S., & Wang, Y. (2018). Intersection traffic prediction using decision tree models. *Symmetry*, *10*(9), 386.

Bai, S., Kolter, J. Z., & Koltun, V. (2018). An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. arXiv preprint arXiv:1803.01271.

Cao, M., Li, V. O., & Chan, V. W. (2020, May). A CNN-LSTM model for traffic speed prediction. In *2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring)* (pp. 1-5). IEEE.

Chen, C. (2002). *Freeway performance measurement system (PeMS)*. University of California, Berkeley.

Chen, Y., Yang, B., & Dong, J. (2006). Time-series prediction using a local linear wavelet neural network. *Neurocomputing*, *69*(4-6), 449-465.

Das, L. C. (2023). Traffic Volume Prediction using Memory-Based Recurrent Neural Networks: A comparative analysis of LSTM and GRU. *arXiv preprint arXiv:2303.12643*.

Fei, Y., Hu, M., Wei, X., & Chen, M. (2022, December). Orthogonal Spatial-Temporal Graph Convolutional Networks for Traffic Flow Forecasting. In *2022 IEEE Symposium Series on Computational Intelligence (SSCI)* (pp. 71-76). IEEE.

Forest Parameter Optimisation to Include Stability and Cost. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *10536 LNAI*, 102–113.

Hewage, P., Behera, A., Trovati, M., Pereira, E., Ghahremani, M., Palmieri, F., & Liu, Y. (2020). Temporal convolutional neural (TCN) network for an effective

weather forecasting using time-series data from the local weather station. *Soft Computing*, *24*, 16453-16482.

Khiewwan, K., Weeraphan, P., Tantisontisom, K., & Ongate, J. (2020). Application of Data Mining Techniques for Classification of Traffic Affecting Environments. Journal of Renewable Energy and Smart Grid Technology, 15(1).

Kumar, S. V., & Vanajakshi, L. (2015). Short-term traffic flow prediction using seasonal ARIMA model with limited input data. *European Transport Research Review*, *7*(3), 1-9.

Liaw, A., & Wiener, M. (2002). Classification and regression by randomForest. *R news*, *2*(3), 18-22.

Lin, L., Li, Y., & Sadek, A. (2013). A k nearest neighbor based local linear wavelet neural network model for on-line short-term traffic volume prediction. *Procedia-Social and Behavioral Sciences*, *96*, 2066-2077.

Liu, C. H. B., Chamberlain, B. P., Little, D. A., & Cardoso, Â. (2017). Generalising Random.

Liu, S., Borovykh, A., Grzelak, L. A., & Oosterlee, C. W. (2019). A neural network-based framework for financial model calibration. *Journal of Mathematics in Industry*, *9*, 1-28.

Lv, Y., Duan, Y., Kang, W., Li, Z., & Wang, F. Y. (2014). Traffic flow prediction with big data: A deep learning approach. *IEEE Transactions on Intelligent Transportation Systems*, *16*(2), 865-873.

Park, S. H., Kim, B., Kang, C. M., Chung, C. C., & Choi, J. W. (2018, June). Sequence-to-sequence prediction of vehicle trajectory via LSTM encoder-decoder architecture. In *2018 IEEE intelligent vehicles symposium (IV)* (pp. 1672-1678). IEEE.

Polson, N. G., & Sokolov, V. O. (2017). Deep learning for short-term traffic flow prediction. *Transportation Research Part C: Emerging Technologies*, *79*, 1-17.

Rojo, J., Rivero, R., Romero-Morte, J., Fernández-González, F., & Pérez-Badia, R. (2017). Modeling pollen time series using seasonal-trend decomposition procedure based on LOESS smoothing. *International journal of biometeorology*, *61*, 335-348.

Shahriari, B., Swersky, K., Wang, Z., Adams, R. P., & de Freitas, N. (2016). Taking the human out of the loop: A review of Bayesian optimization. *Proceedings of the IEEE, 104(1)*, 148-175.

Shi, H., Xu, M., & Li, R. (2017). Deep learning for household load forecasting—A novel pooling deep RNN. *IEEE Transactions on Smart Grid*, *9*(5), 5271-5280.
Smola, A. J., & Schölkopf, B. (2004). A tutorial on support vector regression. *Statistics and computing*, *14*, 199-222.

Snoek, J., Larochelle, H., & Adams, R. P. (2012). Practical Bayesian Optimization of Machine Learning Algorithms. *In Advances in Neural Information Processing Systems 25 (NIPS 2012)*.

Vlahogianni, E. I., Karlaftis, M. G., & Golias, J. C. (2014). Short-term traffic forecasting: Where we are and where we're going. *Transportation Research Part C: Emerging Technologies*, *43*, 3-19.

Zhang, J., Zheng, Y., & Qi, D. (2017, February). Deep spatio-temporal residual networks for citywide crowd flows prediction. In *Proceedings of the AAAI conference on artificial intelligence* (Vol. 31, No. 1).

Zhao, Ling & Song, Yujiao & Zhang, Chao & Liu, Yu & Wang, Pu & Lin, Tao & Deng, Min & Li, Haifeng. (2019). T-GCN: A Temporal Graph Convolutional Network for Traffic Prediction. IEEE Transactions on Intelligent Transportation Systems. PP. 1-11. 10.1109/TITS.2019.2935152.

Zhao, S., Zhuang, Z., Ran, J., Lin, J., Yang, G., Yang, L., & He, D. (2020). The association between domestic train transportation and novel coronavirus (2019-nCoV) outbreak in China from 2019 to 2020: a data-driven correlational report. *Travel medicine and infectious disease*, *33*, 101568.

Appendix A TCN Pseudo Code

Import necessary modules and classes.

Define function is_power_of_two that checks if a number is a power of two.
Define function adjust_dilations that adjusts dilation rates to be powers of two if not already.

Define class ResidualBlock with methods:
__init__: Initialize a residual block with various parameters like dilation rate, number of filters, etc.
_build_layer: Helper function to build a keras layer.
build: Construct the layers within the residual block.
call: Apply the layers to an input tensor and return the output tensor.
compute_output_shape: Calculate the output shape of the residual block.

Define class TCN with methods:
__init__: Initialize the TCN layer with parameters like number of filters, kernel size, dilations, etc.
build: Construct the TCN layer based on the input shape.
compute_output_shape: Calculate the output shape of the TCN layer.
call: Apply the TCN layer to an input tensor and return the output tensor.
get_config: Get the configuration of the TCN layer for model saving/loading.

Define function compiled_tcn that creates and compiles a TCN model with given specifications.
Adjust dilations.
Define the input layer.
Create a TCN layer and apply it to the input layer.
Define the output layer based on regression or classification.
Compile the model with appropriate optimizer and loss function.
Print model summary.

Define function tcn_full_summary that prints a full summary of the TCN model.

Handle compatibility with different TensorFlow versions.

Iterate through the layers and print their details.

Restore original layers after printing.

End of pseudocode.