Republic of Iraq
Ministry of Higher Education and
Scientific Research
University of Babylon
College of Information Technology
Information Network Department

A Developed Machine Learning Approach to Predict the Highway Traffic Congestion in Vehicular Ad-Hoc Networks

A Dissertation

Submitted to the Council of the College of Information Technology at University of Babylon in Partial Fulfillment of the Requirements for the Degree of Doctorate of Philosophy in Information Technology / Information Network

By

Ahmed Ibrahim Turki Khalaf

Supervised by

Prof. Dr. Saad Talib Hasson

2023 A.C. **1445 A.H.**

بِسْمِ الله الرحمن الرحِيمِ

(قُل إِنَّ صَلاتي وَنُسُكي وَمَحيايَ وَمَماتي لِلَّـهِ رَبِّ العالَمينَ * لا شَريكَ لَهُ وَبِذلِكَ
أُمِرتُ وَأَنا أَوَّلُ المُسلِمينَ)

صدق الله العظيم

سورة الانعام: 162-163

Supervisor Certification

I certify that this dissertation was prepared under my supervision at the Department of Information Network / Collage of Information Technology / Babylon University, by **Ahmed Ibrahim Turki Khalaf** as a partial fulfillment of the requirements for the degree of **Ph.D. in Information Technology**.

Signature:

Name: **Dr. Saad Talib Hasson**

Title: **Professor**

Date: / / 2023

The Head of the Department Certification

In view of the available recommendation, we forward this dissertation for debate by the examining committee.

Signature:

Name: **Dr. Saad Talib Hasson**

Title**: Professor**

Date: / / 2023

Declaration

I hereby declare that this dissertation, **A Developed Machine Learning Approach to Model the Urban Traffic Congestion in Vehicular Ad-Hoc Networks** submitted to University of Babylon in partial fulfillment of requirements for the degree of Doctorate of Philosophy in Information Technology-Information Network has not been submitted as an exercise for a similar degree at any other University. I also certify that this work described here is entirely my own except for reports and summaries whose sources are appropriately cited in the references

Signature:

Name: **Ahmed Ibrahim Turki Al-Jubouri**

Date: /   / 2023

Acknowledgements

Praise be to God who helped me to complete this task successfully.

First, I would like to express my deep appreciation for the wonderful man, my supervisor **Prof. Dr. Saad Talib Hasson Al-Jubouri**, for her invaluable guidance, supervision and untiring efforts during the course of this work. I must express my very profound gratitude to **my father, my mother, my wife, my brothers and my sisters** for providing me with unfailing support and continuous encouragement throughout my years of study and through the process of researching and writing this dissertation. I would like to declare my deep thanks and appreciation to my friends especially (**Muitaz Ibrahim Ali**) for their patience, encouragement and help during the work.

Thanks, and appreciation to the head of department physics, **Asst. Prof. Hussam Abdel Hamid Al Darraji** for the great assistance provided me.

Last but not least, most importantly, I would like to thank all the kind, helpful and lovely people who helped me directly or indirectly to complete this work and apologize to them for not being able to mention them by name here, though they are in my heart.

Ahmed I. Turki

Dedication

To ***my mighty father*** *.... May God give him more health and wellness.*

To ***my loving mother****.... All this gossip and acquaintances are some of your table crumbs*

To ***my wife****.... the highest symbols of sincerity, loyalty, and companion on the path.*

To ***my brothers, sisters****.... who support me in this life.*

To ***my children*** *.... the same as the livers.*

to all the ***brethren****;*

I dedicate to you my PhD dissertation

Abstract

In recent years, Intelligent Transport Systems (ITS) have rapidly evolved, driven by the increasing demand for improved transport network management and advancements in computing. ITS encompass a wide array of applications, necessitating proactive strategies and predictive data fueled by artificial intelligence and big data. This dissertation focuses on developing precise short-term traffic prediction models, particularly for traffic density, with the aim of bolstering urban transport planning and empowering individual travelers, potentially revolutionizing urban traffic control and planning.

The problem at hand revolves around the imperative need for accurate short-term traffic predictions to facilitate proactive ITS applications and support informed decisions by individual travelers. Current research predominantly revolves around the comparison of machine learning methods while often neglecting the integration of technical indicators into traffic density predictions. The central challenge is to enhance prediction accuracy while comprehending the impact of technical indicators on traffic density.

The dissertation's approach begins with a comprehensive review of data prediction methods and delves into various machine learning techniques tailored for short-term traffic prediction. It introduces three distinct models incorporating data normalization to account for technical factors influencing traffic density. The significant breakthrough arises from the integration of technical indicator features, substantially bolstering regression accuracy. These models are rigorously tested using real-world data from the M25 and M60 motorways under diverse traffic conditions. Additionally, the study introduces an algorithm to assess level of service (LOS) on an hourly basis, leveraging vehicle density data from the Motorway Incident Detection and Automatic Signaling (MIDAS) system. This approach combines technical indicators with machine learning models to classify LOS accurately. Ground-truth LOS data is derived from stationary sensors, showcasing the remarkable accuracy enhancement achieved through the integration of technical indicators.

The dissertation's key findings underscore the transformative impact of incorporating technical indicators, significantly improving traffic density prediction accuracy by 86.63% for the M60 highway data and 68.2% for the M25 highway, regardless of the chosen machine learning approach. Furthermore, it demonstrates the enhanced accuracy of LOS estimation (approximately 6.52%), with potential applicability to highways in various geographical locations.

Finally, this research makes substantial contributions to the enhancement of ITS applications and the overall efficiency of transport networks, yielding benefits for both transportation agencies and individual travelers.

Table of Contents

Chapter Five: Conclusions and Future Works

List of Algorithms

List of Figures

List of Tables

List of Abbreviations

Abbreviation	Meaning
AA%	Average Accuracy Percentage
AADF	Annual Average Daily Flow
ANPR	Automatic Number Plate Recognition
ATR	Average True Range
AUC	Area Under the Curve
BTI	Buffer Time Index
CCTV	Closed-Circuit Television.
CNNs	Convolutional neural networks
CPU	Central processing unit
DBN	Deep Belief Networks
DOT	Department of Transportation
EMA	exponential moving average
FFNN	Feedforward neural network
FN	False Negative
FP	False Positive
FPR	False Positive Rate
GPU	Graphics processing unit
GUI	graphical user interface
HCM	Highway Capacity Manual
HETMS	Highways England Traffic Management Systems
IoT	Internet of Things
IoV	Internet of Vehicles
ITS	Intelligent Transportation System

Abbreviation	Meaning
KNN	K-Nearest Neighbors
LOS	levels of service
LPR	License Plate Recognition
MAPE	Mean Absolute Percentage Error
MHMM	Modified Hidden Markov Model
MIDAS	Motorway Incident Detection and Automatic Signaling
MLE	Maximum Likelihood Estimates
MLPNN	Multi-layer perceptron neural networks
MLR	Multiple Linear Regression
MOM	Momentum
MSE	Mean Square Error
NIS	Network Information Services
NTIS	National Technical Information Service
OLS	Ordinary least-squares
OOB	Out-of-bag
PeMS	Performance Measurement System
PTI	Planning Time Index
RF	Random Forest
RMSE	Root Mean Square Error
RNNS	Recurrent neural networks
ROC	Rate of Change
ROC	Receiver Operator Characteristic
RRSE	Root Relative Squared Error
RSI	Relative Strength Index
RSUBoost	Random Under sampling Boost algorithm
RTMS	Remote traffic microwave sensors
SCATS	Sydney Coordinated Adaptive Traffic System
SMA	Simple moving average
SPSS	statistical package for social sciences
SVM	Support Vector Machine
TAME	Traffic Appraisal, Modelling, and Economics
TII	Transportation Infrastructure Ireland
TIs	Technical Indicators
TMU	Traffic Monitoring Units
TN	True Negative
TP	True Positive
TPR	Tue Positive Rate

Abbreviation	Meaning
TTI	Travel Time Index
V/C	volume-to-capacity
VANET	Vehicular Ad-Hoc Networks
VMS	Variable Message Sign
VSL	Variable Speed Limit

Chapter One

General Introduction

1.1 Introduction

This chapter serves as an introduction, providing an overview of the chosen research topic along with a brief background. This section addresses the topic of motivation, outlines the research questions, and highlights the objectives and contributions of the dissertation. According to Ritchie and Roser (2020), the global urban population is projected to reach nearly 7 billion individuals by 2050, posing complex challenges in transportation due to urbanization and widespread automobile use, resulting in congested roadways and deteriorating traffic environments. While increasing road capacity is a common solution, Dechenaux et al. (2014) found that it could lead to more congestion. The concept of a smart city, integrating advanced technologies such as 5G, artificial intelligence, Internet of Things (IoT), and cloud computing (Sun et al., 2020), offers a renowned solution for future urban development. Smart cities can enhance resource management and citizen living environments (Ghaleb et al., 2018), with ITS playing a pivotal role (Younes & Boukerche, 2018). ITS, incorporating communication systems and on-road sensors (Siddiqui et al., 2016), enables real-time road analysis, traffic control efficiency enhancement (Younes & Boukerche, 2013), and fast vehicular cloud services. Implementing ITS contributes to secure, sustainable, and comfortable road environments (Rezende et al., 2012), necessitating precise traffic density prediction systems (Rezende et al., 2014). These systems offer valuable insights for ITS applications like VANET, vehicular cloud, traffic light control, and congestion management.

The dissertation's focus is on developing machine learning models to predict vehicle densities across different days of the week, addressing traffic congestion on urban highways and its implications for vehicular communication. Short-term traffic prediction, referring to estimating near-future traffic conditions (Chikkakrishna et al., 2019), is essential in ITS. Congestion arises when traffic volume exceeds road capacity, making ITS applications crucial. The ITS Handbook (Miles & Chen, 2004) highlights ITS as an effective tool for congestion management through traffic optimization (Cheng et al., 2020).

ITS is integral to assessing traffic conditions, utilizing measurements for various purposes such as traffic operations, roadwork planning, queue evaluation, and congestion management. The United States Highway Capacity Manual (HCM) defines six LOS based on traffic density (Elefteriadou, 2016). Transportation agencies seek real-time or historical hourly traffic data, typically collected via various stationary sensors (Hargrove et al., 2016). LOS assessment primarily relies on the density or volume-to-capacity ratio. This dissertation proposes a novel methodology using vehicle density-based technical indicators to evaluate LOS on urban highways, offering a new approach to analyzing traffic data characteristics for congestion management.

1.2 Problem Statement

In recent years, the Intelligent Transportation System (ITS) has received considerable attention due to higher demands for road safety and efficiency in highly interconnected road networks. As an essential part of ITS, traffic prediction can provide support in many aspects, such as road routing, traffic congestion control, etc. (Boukerche et al., 2020).

Based on the above discussion, Sekuła et al. (2018) attempted to estimate historical hourly traffic volumes, which transportation agencies need to perform annual calculations of various network-level performance metrics. However, this model faces the problem of estimating short-term future volumes.

Previous literature attempts to address the issue of traffic conditions estimation problem, as attempted by Hoseinzadeh et al. (2021), using Crowdsourced data. This study did not take into account the variability and sensitivity of the methodology regarding seasonal influences such as holidays and weekends. Moreover, vehicle density was not taken into account in the Level of Service (LOS) measurement.

1.3 Research Questions

The central inquiry of this dissertation revolves around the development of an integrated model capable of forecasting traffic congestion in VANETs. Based on the inquiry at hand, it is possible to derive a number of subsidiary questions:

1- Can the utilization of machine learning techniques and neural networks yield a viable short-term predictive model?
2- Can the application of technical analysis and its mechanisms result in the development of a short-term predictive model that exhibits superior performance compared to models that rely on traditional input?
3- Is it feasible to develop a predictive model that can accurately assess traffic conditions using the LOS indicator?
4- Can the utilization of technical analysis indicators be employed in the construction of a classification machine learning model that effectively characterizes traffic conditions by utilizing the level of service indicator?
5- Can machine learning algorithms be employed to ascertain instances of unobstructed traffic flow or instances of traffic congestion?

1.4 Dissertation Aims and objectives

The main aim of this dissertation is to construct models that possess the capability to forecast short-term traffic variables accurately, specifically traffic density, on urban highway roads across various traffic scenarios. This is accomplished following various subordinate objectives:

1- Utilizing machine learning techniques to construct a more comprehensive prediction model.
2- Developing resilient traffic prediction models for urban highways that can effectively forecast traffic patterns under diverse conditions and exhibit location transferability for easy implementation.
3- Employing technical indicators that are commonly employed in financial trading management and control to explore their potential applicability in forecasting traffic congestion.
4- Constructing classification models that can accurately estimate the level of service by utilizing technical indicators.

1.5 Dissertation Contributions

The primary objective of these models is to facilitate the identification and implementation of strategies to alleviate or prevent traffic congestion. Each of the proposed models consists of four stages. This dissertation makes several significant contributions in comparison to the existing literature.

1- This dissertation represents the initial research endeavor to predict traffic density utilizing specific technical analysis indicators as input variables for the chosen machine learning and neural network models.

2- It is the first dissertation to use technical analysis indicators as input features for the proposed classification models that determine traffic conditions through the LOS index. The inclusion of these indicators is justified by their strong explanatory capabilities and their ability to predict upward and downward trends accurately.

3- The outcomes of this dissertation holds significant implications for transportation agencies, researchers, and decision-makers. These findings can serve as a reliable basis for making informed decisions aimed at mitigating traffic congestion within the vehicular network.

1.6 Related Work

A comprehensive examination of prior research pertaining to the prediction of traffic congestion was conducted. Based on the concerns elucidated in this dissertation, this section is partitioned into three distinct components.

1.6.1 Machine learning-based traffic prediction approaches

The relevant papers are divided into several groups in terms of relevant datasets, prediction techniques, prediction condition, methodologies, and evaluation criteria. Their contributions to the field of traffic flow prediction are substantial. To provide a more comprehensive representation, Table 1.1 presents a summary of the comparative analysis of various machine learning method.

Rahi (2019) focused on the development of traffic flow prediction models for ITS applications such as Advanced Route Planning and Traffic Control Systems. It reviewed existing time series prediction techniques and evaluated various machine learning models for predicting freeway traffic flow. The research proposed an objective function that significantly improved prediction accuracy by breaking down the traffic network into virtual patches and nodes. Among the tested models, RNN variant LSTMs, combined with neural networks and Deep CNNs, outperformed others. The study highlighted the importance of model structure, data Preprocessing, and error matrices for achieving accurate predictions. This framework showed promise in reducing error rates in congestion predictions and travel time delays in real-time, aligning with the goals of smart transport systems.

Liu (2019) aimed to enhance the reliability of short-term traffic flow prediction, which is typically crucial for various applications. They extended the K-nearest neighbors (K-NN) model to include prediction intervals to account for uncertainty. Recognizing the stochastic nature of traffic, they also tested different time intervals (ranging from 3 minutes to 30 minutes) for traffic flow rate measurements. The results indicated that shorter time intervals (less than 10 minutes) favored K-NN for point prediction accuracy over the benchmark model, suggesting its suitability for short-interval traffic flow prediction.

According to Sun et al. (2020), in recent years, there has been a growing interest in using machine learning models in the automotive industry and academia to support Internet of Vehicles (IoV) applications, particularly in predicting traffic and road conditions. These predictions were crucial for improving safety and enhancing the quality of service for applications like safety and infotainment. While there was a focus on improving prediction accuracy, it remained unclear whether machine learning-based prediction schemes were suitable for real-time traffic forecasting in the IoV. To address this question, previous articles conducted extensive studies to assess the efficiency of various machine learning-based prediction models, considering both prediction accuracy and computational time cost. The goal was to identify factors that might limit the use of these models for real-time services in the IoV environment.

Zheng et al. (2020) developed a deep learning model for short-term traffic flow prediction in ITSs was discussed. The model combined convolutional neural network (CNN) and long short-term memory (LSTM) networks, along with attention mechanisms to extract spatial and short-term temporal features. Additionally, a bidirectional LSTM (Bi-LSTM) module was introduced to capture long-term temporal features. Experimental results demonstrated that this hybrid model outperformed existing approaches in predicting traffic flow accurately.

Chen et al. (2021) aimed to enhance short-term traffic flow prediction accuracy by employing an improved wavelet neural network (WNN) model, providing support for intelligent traffic management. It utilized WNN as the foundational prediction model and optimized it through an enhanced particle swarm optimization (PSO) algorithm. The experimental results of these previous studies demonstrated that this approach was more efficient than using WNN or PSO-WNN in isolation, resulting in a 14.994% reduction in prediction error compared to traditional WNN.

Chen & Chaudhari (2021) introduced MIDAS, a reinforcement learning-based method that aimed to enable autonomous navigation in urban environments. MIDAS was designed to allow an Ego agent to influence the actions of other vehicles. It utilized an attention mechanism to handle multiple agents and incorporated a "driver-type" parameter to enhance its planning capabilities. The method underwent validation through extensive experiments, demonstrating its adaptability to various road scenarios. It showcased the ability to generate adaptive Ego policies, maintain robustness in the face of changes in other agents' behavior, and outperform existing interaction-aware decision-making approaches in terms of safety and efficiency.

Raskar & Nema (2022) proposed an enhanced prediction model for traffic flow was designed using a Modified Hidden Markov Model (MHMM). The input features considered for prediction via MHMM included "ATR, EMA, RSI, and ROC." The modification in HMM was based on the optimal tuning of state numbers using the Mean Fitness-oriented Dragonfly Algorithm (MF-DA). Ultimately, the study compared and demonstrated the improvements achieved with the implemented approach over conventional models.

Rajalakshmi & Ganesh (2022) aimed to forecast future traffic flow using time-series models, with a focus on minimizing prediction errors with real-time data. Hybrid models that combined autoregressive integrated moving average (ARIMA) with multilayer perceptron (MLP) and recurrent neural network (RNN) for traffic prediction using UK Highways data were proposed. The efficacy of these hybrid models was assessed using various metrics, with promising results (e.g., R2 values around 0.94 for peak hours).

Table 1.1: A comparative analysis of recent studies within the field of Machine Learning.

Ref.	Prediction Techniques	Dataset	Prediction condition	Comb. or single use	Accuracy
Rahi (2019)	Support Vector Machine, Random Forest, Etc.	Motorway Incident Detection and Automatic Signaling (MIDAS)	Urban highway	Single and comb	RMSE= 0.179, 0.178
Liu (2019)	K-Nearest Neighbor	MIDAS	Highway	Comb.	RMSE= 30
Sun et al. (2020)	Support Vector Machine, Artificial Neural Netork, Long Short-Term Memory	MIDAS	Traffic Monitoring Unit (TMU) and MIDAS	Single	MAPE= 10.5, 12.4 8.4
Zheng et al. (2020)	Attention-based convolutional - Long Short-Term Memory	Performance Measurement System (PeMS)	Freeway and urban road	Comb	RMSE= 15.56
Chen et al. (2021)	Improved particle swarm optimization - Wavelet neural network	368 time points were recorded as a case study	Urban	Comb	MAE = 16.327
Chen & Chaudhari (2021)	Reinforcement learning-based method	MIDAS	Highway	Single	2.7
Raskar & Nema, (2022)	Hidden Markov Model	Performance Measurement System (PeMS)	Urban	Comb	0.222
Rajalakshmi & Ganesh (2022)	(ARIMA – MLP), (ARIMA – RNN)	MIDAS	Highway	Comb.	RMSE= 0.84, 0.81
Proposed models	Multi Linear Regression, Feed Forward Neural Network, Random Forest, Markov chain	MIDAS	Highway	Single	RMSE= 0.1 0.6 0.8 2.2 (as average)

1.6.2 Estimating LOS based on Machine learning

The relevant papers are divided into several groups in terms of relevant data, classification techniques, index used, and evaluation criteria. Their contributions to the field of LOS estimation are substantial. To provide a more comprehensive representation, Table 1.2 presents a summary of the comparative analysis of various machine learning method.

Aljamal (2019) proposed a model for vehicle count estimation was introduced using an adaptive Kalman filter (AKF) algorithm. The AKF was employed to enhance accuracy compared to the traditional Kalman filter, effectively reducing prediction errors by up to 29%. Furthermore, the study combined the AKF with a neural network (AKFNN) to further improve vehicle count estimates, resulting in a substantial accuracy boost of up to 26% compared to the AKF used in isolation. The research also delved into investigating the sensitivity of the AKF model to initial conditions, emphasizing the importance of selecting appropriate initial parameters. In conclusion, both the AKF and AKFNN approaches outperformed the traditional Kalman filter in the realm of vehicle count estimation.

Kodupuganti et al. (2019) established link-level travel time-based LOS thresholds for urban areas utilizing large-scale travel time data sourced from GPS devices, sensors, smartphones, and various other data-gathering devices. The study integrated posted speed limits with raw travel time data within the context of Charlotte, North Carolina, and computed several travel time metrics, including average travel time, 95th percentile travel time (PT), planning time index (PTI), and buffer time index (BTI), all categorized by posted speed limits. The research focused on examining the relationships between estimated speeds derived from the regional network model and the computed travel time metrics in order to develop LOS thresholds based on posted speed limits.

Pulugurtha & Imran (2020) relied on the commonly used LOS scale, which was based on density, and emphasized speed and travel time as key performance measures. Additionally, travel time reliability was gaining importance in transportation planning and management. In a related study, microscopic simulation was employed to investigate the validity of travel time and travel time reliability indices in quantifying LOS thresholds for freeway sections. The study found that travel time thresholds varied with speed limits, but beyond a saturation point, speed limits had no influence on operational performance. Furthermore, travel time reliability thresholds decreased with lower speed limits.

Wilby et. al. (2020) highlighted a shift from predicting travel time (TT) to LOS as a more practical metric. A Random Under Sampling Boost algorithm (RUSBoost) was employed to address the unbalanced nature of LOS classes. The classifier, trained on 12 months of data from a Bluetooth network, achieved an average recall of 82.8% for prediction horizons up to 15 minutes and 92.5% for congestion prediction. The study also emphasized the importance of considering data from all links and the day of the week to improve accuracy, ultimately offering a promising tool for traffic management practitioners.

Hoseinzadeh et al. (2021) introduced an algorithm that employed big data features to classify LOS using machine learning models, achieving a 10% improvement in accuracy (accuracy = 0.93, Kappa = 0.83) compared to traditional methods. It was noted that this method proved to be adaptable to various freeway locations, providing transportation agencies with a cost-effective tool for LOS assessment.

Özinal & Avşar (2022) applied various deep learning architectures, such as Bidirectional LSTM and Stacked LSTM, were compared with shallow neural networks for short-term traffic condition prediction. It was found that Bidirectional LSTM and Stacked LSTM had outperformed other models in short-term traffic prediction, highlighting the superiority of deep learning over shallow neural networks in their research.

Tišljarić et al. (2022) utilized the connected vehicles as mobile sensors to gather traffic data. A speed transition matrix-based model was introduced for estimating bottleneck probabilities, considering traffic patterns and the center of mass. This method underwent evaluation across various motorway scenarios and demonstrated a 92% accuracy on the validation dataset. These findings suggested its potential application in motorway traffic control systems with high CV penetration.

Vrbanić et al. (2023) investigated Dynamic Speed Transition Matrices - Q-Learning - Variable Speed Limit in various traffic scenarios. It was found that this method outperformed other control strategies, resulting in improvements in traffic parameters such as Total Time Spent (TTS) and Mean Travel Time (MTT).

Table 1.2: A comprehensive overview of various alternative methods for assessing LOS as documented in the existing literature.

Reference	Data	Index Used	classification techniques	Accuracy
Aljamal (2019)	probe vehicles, single-loop detector	-Time-mean speed for probe vehicles -Total number of probe arrivals - Total number of probe departures, …etc.	adaptive Kalman filter with a neural network	96.57
Kodupuganti et al. (2019)	Travel time data provided by North Carolina Department of Transportation	- Planning Time Index - Buffer Time Index - Average travel time	- Travel time reliability threshold - Regression model	-
Pulugurtha & Imran (2020)	Simulation (travel time)	- Planning Time Index - Buffer Time Index	- Travel time reliability threshold - Statistical regression	-
Wilby et. Al. (2020)	Bluetooth vehicle identifiers deployed on the highway SE-30 in Seville	Arrival travel time	RUSBoost	92.5
Hoseinzadeh et al. (2021)	Waze speed/travel time and Waze alert	- Basic Statistical Measures - Travel Time Performance Measures -Crowdsourced Data	-Random Forest -Support Vector Machine, -K-Nearest Neighbor	0.92, 0.90, 0.88 respectively
Özinal & Avşar (2022)	Traffic flow and speed	- temporal features - measurement features	-MLPNN -Support Vector Regression -Gradient	97.02, 96.49, 96.76, 96.08

			boosted decision trees -KNN	
Tišljarić et al. (2022)	Simulation (traffic density, speed)	- Center of Mass Estimation - Fuzzy Inference System	bottleneck probability estimation.	0.92
Vrbanić et al. (2023)	Simulation (speed)	- Variable Speed Limit	Reinforcement Learning	-
Proposed model	Density	Technical indicators (ATR, SMA, EMA, ROC, MOM, RSI)	Random Forest, K-Nearest Neighbors	0.96 as average

1.7 Dissertation Outline

This dissertation is organized as follows:

Chapter Two presents a thorough exposition of the fundamental principles underlying the chosen machine learning models, which are subsequently employed in this dissertation.

Chapter Three provides a comprehensive elucidation of the primary procedures involved in the design of an integrated system for machine learning and neural networks, specifically tailored to address the issue of traffic congestion.

Chapter Four shows and discusses an analysis of the utilization of the proposed system on an authentic historical traffic dataset, along with an examination of the experimental outcomes derived from its implementation.

Chapter Five of concludes the key concepts and offers recommendations for future research endeavors.

Chapter Two

Theoretical Background

2.1 Overview

The structure of this chapter encompasses a literature review that examines studies on predicting traffic coefficients and classifying traffic congestion based on service level measurements. Furthermore, the chapter discusses the experimental setup, which involves the use of machine learning-based algorithm models. Furthermore, detailed descriptions of the dataset utilized for computational analysis in this dissertation are provided. Finally, assessment measures for forecasting traffic density are offered, as well as metrics for LOS estimation.

2.2 Vehicular ad-hoc Network (VANET)

VANETs networks are formed through the convergence and advancements of wireless communication technologies, intelligent transport systems, and automotive construction technologies. Vehicular networks are recognized as a distinct subset within the broader category of Mobile Ad hoc Networks (MANETs), characterized by their unique set of node properties and operational requirements. A VANET refers to a collection of mobile entities (vehicles) and stationary entities (roadside units) that collaborate to exchange crucial information pertaining to road conditions and other vehicles. Figure 2.1 presents several areas of VANET communications (Mchergui et al., 2022).

In the past ten years, a multitude of Vehicular Ad-Hoc Network (VANET) services have been introduced, including infotainment applications, driver assistance systems, and video on-demand services. Contemporary vehicles are equipped with both hardware and software that serve not only safety purposes, such as accident prevention and the dissemination of warning messages, but also provide support for a range of entertainment and comfort applications (Ali et al., 2020). One notable contemporary phenomenon involves the significant increase in enthusiasm surrounding the implementation of Artificial Intelligence methodologies across various fields of application, such as cybersecurity, traffic congestion detection, data analytics, routing, healthcare, robotics, and others (Hajlaoui et al., 2019). The current trend of heightened focus on the application of artificial intelligence techniques, including basic Machine Learning, Deep Learning, and Swarm Intelligence, in emerging VANET solutions to address diverse VANET challenges is to be expected. However, further research is required in order to apply artificial intelligence methods to vehicular communications.

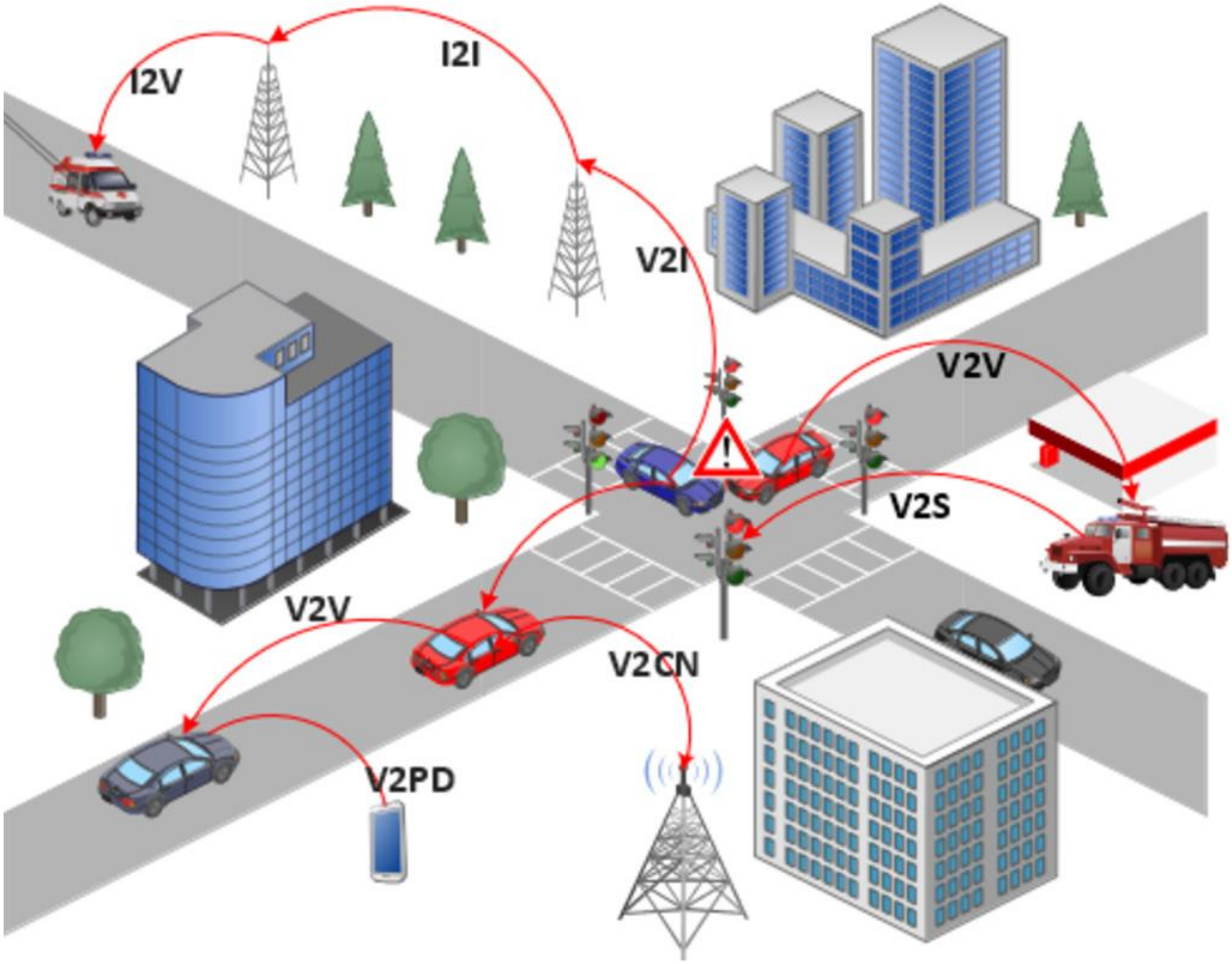


Figure 2.1. Communication in VANET (Mchergui, et al., 2022).

2.3 Factors Influencing Traffic Prediction Models

Numerous factors contribute to the traffic prediction of a model, thereby impacting the process of predicting traffic congestion. Apart from the hyperparameters of the models, several factors include the treatment of input traffic parameters within their respective contexts, the resolution of the input data samples, the number of prediction steps, the interplay between different traffic parameters utilized, and the concealed spatial-based temporal dependencies inherent in the traffic variable data. The prediction performance can be influenced by additional factors such as seasonality and trend present in the time series data. The subsequent subsections provide a comprehensive review of each of these significant factors (Rahi, 2019).

2.3.1 Road Traffic Predictions

Road traffic predictions are employed in diverse contexts, including intelligent transportation systems (ITS), traffic operations and planning, estimation of travel time, modeling of traffic flow, management of incidents, freight and logistics, assessment of environmental impact, planning of public transportation, and research. The aforementioned predictions aim to maximize the efficiency of traffic flow, enhance safety measures, improve the overall effectiveness of transportation systems, facilitate navigation and incident management, optimize freight operations, assess the environmental consequences, and advance the field of transportation engineering. The primary emphasis of ITS applications is typically on highways, freeways, and motorways. However, urban and connecting roads present a higher level of complexity due to the presence of uncontrolled connections and intersections of varying sizes, which are often equipped with limited data acquisition equipment (Boukerche et al., 2020).

2.3.2 Input variables for Traffic Prediction

The selection of variables plays a critical role in traffic flow forecasting models, as it has a direct influence on their performance and efficiency. Indirect approaches, such as employing mutual information derived from entropy theory, have been employed for the purpose of extracting information from unprocessed feature values. Variables commonly taken into account include the volume of traffic flow, the duration of travel, and the speed data obtained from on-site sensors such as loop detectors and laser sensors. Various models incorporate input parameters such as traffic density, speed, incident severity, road delays, and lane blocking duration (Mchergui et al., 2022).

2.3.3 Data Resolution for Traffic Prediction

Assessing the effectiveness of traffic prediction models requires careful consideration of data resolution, which determines the level of detail at which traffic-related data is collected and analyzed. In the context of recent research, data resolution ranged from 30 seconds to 60 minutes, with higher resolution providing more detailed information but also introducing more noise. The dynamic adjustment of data resolution in line with prediction model requirements is crucial to capture the dynamic nature of traffic accurately. Relying solely on fixed measurement instruments may overlook important traffic fluctuations. Therefore, selecting the appropriate data resolution is vital for accurate and effective traffic prediction models (Hu et al., 2022).

2.3.4 Traffic Flow Prediction steps

The temporal divisions or intervals over which the prediction model makes forecasts are commonly known as the prediction step, prediction interval, or prediction horizon. It is widely acknowledged that the accuracy of predictions tends to diminish as the prediction horizon expands. While multi-step predictions are frequently employed in prediction models discussed in the literature, they often entail a compromise in the accuracy of the model's predictions. The objective of this study is to perform predictions for both one-step and multi-step ahead forecasting (Essien et al., 2021).

2.3.5 Seasonal Effects and Spatial-Temporal Patterns in Traffic Flow Prediction

The examination of the spatial and temporal relationship has been extensively examined within the realm of traffic flow and overall traffic forecasting. The objective has consistently been to leverage the temporal data of road traffic in relation to its spatial characteristics. Traffic time series data displays seasonal and periodic patterns when they are examined for trends within the dataset. There exists a significant correlation between free flow ways and motorway roads in terms of their spatiotemporal characteristic (Heshami & Kattan, 2022; Zhang et al., 2023; Belt et al., 2023).

2.4 Study Area

The primary objective of this dissertation is to conduct an analysis of the data pertaining to the M25 highway, specifically the section between junction 13 and junction 14 in a clockwise direction around London, as well as the M60 highway, specifically the section between junction 1 and junction 2 in a clockwise direction around the Manchester area in the United Kingdom. The decision to select the UK traffic road networks as a case study for this dissertation was based on factors such as the type, availability, and format of the data. This section provides a comprehensive overview of the study area and the datasets that are relevant to the research. The following section provides a description of the characteristics of the corridor.

2.4.1 M25 highway

The M25, also known as the London Orbital Motorway, is a significant thoroughfare that surrounds a majority of the Greater London area. The motorway spanning a distance of 117 miles (equivalent to 188 kilometers) holds significant importance within the transportation network of the United Kingdom, being recognized as one of the busiest thoroughfares. According to Kyriacou et al. (2022), a daily average of 196,000 vehicles was observed in the vicinity of Heathrow Airport in 2003, specifically between junctions 13 and 14. The focal point of investigation is the junction situated between junction 13 and junction 14, facilitating the connection between Wraysbury Civil Parish and Stanwell, in close proximity to Heathrow airport. Figure 2.2 displays a map depicting the chosen corridor.

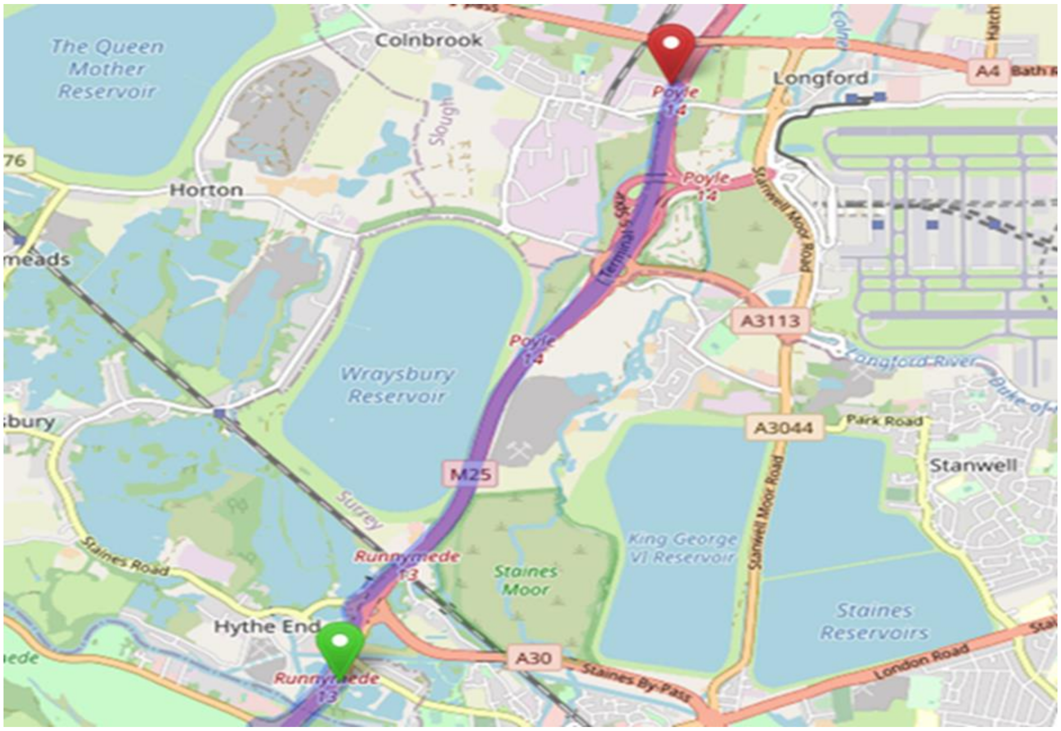


Figure 2.2: The selected test area namely the M25 (junction 13 - junction 14)

2.4.2 M60 highway

The M60 motorway, also known as the Manchester Ring Motorway or Manchester Outer Ring Road, is a circumferential motorway located in the North West region of England. Constructed over a span of four decades, the aforementioned infrastructure traverses the majority of the metropolitan boroughs within Greater Manchester, with the exception of Wigan and Bolton. The majority of Manchester is contained within the motorway, with the exception of the southernmost region comprising Wythenshawe and Manchester Airport, which is accessible via the M56. According to Highways Agency (2023), the M60, which spans a distance of 36.1 miles (58.1 km), underwent a renaming process in 1998. This involved the consolidation of sections from the M62, M66, and the entirety of the M63 into the newly established route. The finalization of this circular route occurred in the year 2000. The junction under investigation is the connection between junction 1 and junction 2, which serves as a link between Stockport and Cheadle. Figure 2.3 displays a map depicting the chosen corridor.

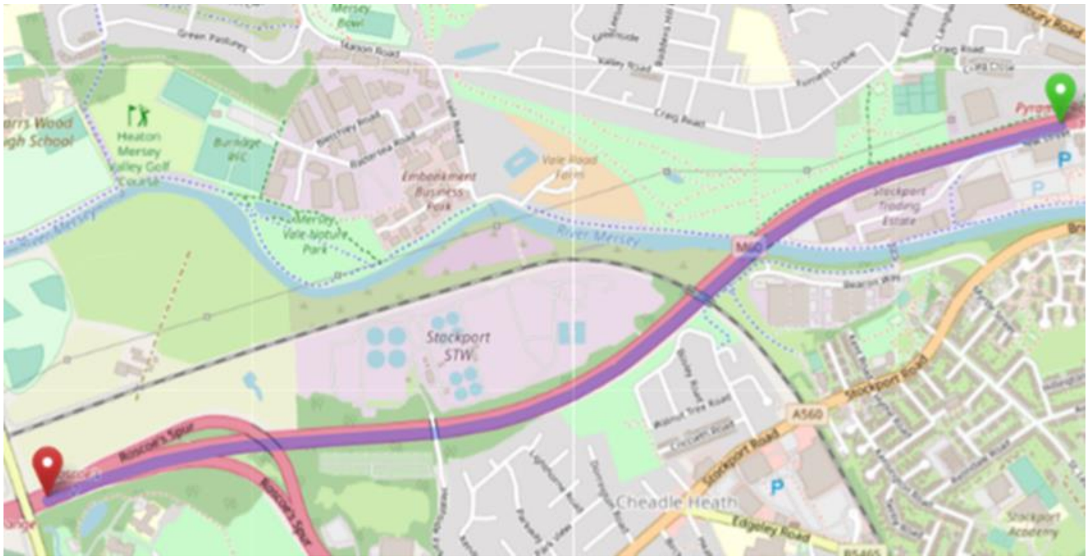


Figure 2.3: The selected test area namely the M60 (junction 1 - junction 2)

2.5 The Research Dataset

Various data sources exhibit variations in the parameters that are documented. Certain parameters are shared among them, such as the timestamp of the log and the vehicle flow. Data is collected by monitoring the activity of sensors at the designated locations. The data collection for this study involved the utilization of loop-based sensors from traffic monitoring units and the inference of journey time using Automatic Number Plate Recognition equipment, specifically for the dataset obtained from Highway England. The road surface's sensor loops were utilized to measure the real-time speeds, vehicle flows, and occupancy. Additionally, the travel times between two specific points were measured through the implementation of ANPR camera recognition. In the event that a loop on the site was identified as defective, it was duly reported. In such cases, the flow values were derived from previous data rather than being based on the vehicle category and speeds. We have selected the following two datasets based on their appropriateness for evaluating and validating our proposed network methodology.

Highway England offers comprehensive data for each quarter-hour interval starting from April 2015, encompassing all motorways and category 'A' roads under its management. These roads collectively form the Strategic Road Network in England. Category A major roads consist of motorways, dual carriageways, and motorways. The Motorway Incident Detection and Automatic Signaling (MIDAS) original gold dataset is recorded at a frequency of one entry per minute. The site implemented specific regulations for logging the collected data, which primarily included the following criteria: publication time, speed (with a threshold of 240 km/h), vehicle flows (with a threshold of 120 veh/min), and reporting of occupancy and headway on a per lane basis. The categorization of vehicle flows is based on the length of each individual vehicle and is determined by the traffic monitoring equipment installed along the roadside. The vehicle flows, which were classified into categories, were converted into volumetric measurements in units of vehicles per minute for each lane.

These measurements were then combined to obtain readings for the entire carriageway. Table 2.1 below presents a comprehensive overview of the significant data fields contained within the MIDAS traffic flow dataset. Monthly files are generated for each model site. The files exclusively consist of logs pertaining to flow, speed, and day type data from the primary highways, junctions, and motorways, as they are under the management of, HE (Units, 2018).

Table 2.1: The MIDAS dataset encompasses various aspects related to traffic flow, including additional field names and description features that are distinct to this dataset. (England, 2015).

MIDAS ID	A distinctive identifier specific to the National Technical Information Service (NTIS) hyperlink.
Legacy MIDAS ID	A distinct identifier specific to the NTIS hyperlink.
Site Name	The following is a depiction of the site.
Local Date	Provide the local date according to British Summer Time.
Local Time	The local time intervals, denoted in 15-minute increments, within British Summer Time region.
Day Type	The following items are considered to be valid: The value "0" represents the first working day of a normal week. • 1 - Normal Tuesday workday; • 2 - Normal Wednesday workday; • 3 - Normal Thursday workday; • 4 - Final workday of a normal week; • 5 - Saturday, excluding days that fall under type 14; • 6 - Sunday, excluding days that fall under type 14;

	• 7 - First day of school holidays; • 9 - Middle of the week - school holidays, excluding days falling under types 12, 13, or 14; • 11 - Last day of the week - school holidays, excluding days that fall under type 12,13, or 14; • 12 - Bank Holidays, including Good Friday, with the exception of days falling under type 14; • 13 - Christmas holiday days between Christmas and the New Year; • Christmas Day and New Year's Day are on December 14.
Total Carriageway Flow	Within the 15-minute time slice, the number of vehicles detected on any lane.
Total Flow vehicles less than 5.2m	The number of vehicles shorter than 5,2 meters that were detected on any lane during the 15-minute time slice.
Total Flow vehicles 5.21m - 6.6m	Count of vehicles between 5.21m and 6.61m detected on any lane during the 15-minute interval.
Total Flow vehicles 6.61m - 11.6m	The number of vehicles between 6.61 million and 11.6 million that were detected on any lane during the 15-minute time slice.
Total Flow vehicles above 11.6m	The number of vehicles with a length greater than 11.6 meters detected on any lane within a 15-minute time slice.
Speed Value	The average speed in kilometers per hour of all vehicles on all lanes as measured by the site over the course of 15 minutes.
Quality Index	The indication of the provided data's quality. The quantity of valid one-minute records that were reported and used to calculate the Total Traffic Flow and speed. No valid records are indicated by a quality index of 0.
Network Link Id	A distinct identifier specific to the National Technical Information Service (NTIS) hyperlink.
NTIS Model Version	The data pertains to the specific iteration of the NTIS model.

2.6 Variable selection

Prior research has extensively investigated a range of indicators and metrics with the aim of developing efficient predictive systems for traffic congestion. The most prominent parameters that have been examined in various studies include speed, flow, travel time, user accidents, jams, and risks. These parameters have been investigated both individually and in combination. Studies have utilized a wide range of data sources, including simulations, camera images and videos, and probe vehicles (Rahi 2019). Furthermore, various datasets have been employed by researchers, encompassing Bus Breakdown and delays, Annual Average Daily Flow, and Annual Traffic Volume, among other sources of information. The MIDAS system, known as Motorway Incident Detection and Automatic Signaling, has been utilized to determine vehicle density in VANET by utilizing speed and flow data. This calculation is based on Equation 2.1, which was introduced by Kulkarni & Rao (2010) and referenced by Abboud & Zhuang (2016). Researchers can improve the accuracy and effectiveness of their predictive models for traffic congestion management by meticulously selecting pertinent features from these extensive datasets.

$$Density(K) = {}^{Flow(q)}\Big/_{Speed\ (v)} \tag{2.1}$$

where:

q = Flow (vehicles/hour)

v = Speed (kilometers/hour)

k = Density (vehicles/kilometers)

Equation 2.1 exhibits several intriguing characteristics. The flow exhibits minimal characteristics when either the speed or density is at a low level. These points can be illustrated by two commonly observed traffic conditions. The initial circumstance to be considered is the occurrence of a traffic jam, characterized by a significant increase in traffic density coupled with a notable decrease in vehicular speed. The amalgamation yields a diminished rate of fluid movement. The second condition occurs when traffic density is significantly reduced, allowing drivers to achieve free flow speed. The exceptionally low density effectively counterbalances the high velocity, thereby yielding a significantly reduced flow rate.

2.7 Data preprocessing

Data preprocessing is a crucial preliminary procedure that must be undertaken prior to inputting the data into the model. Data preprocessing plays a crucial role in machine learning by emphasizing the relevant features that we desire the model to learn. It facilitates faster convergence of the model and prevents it from being influenced by extraneous information. Nevertheless, the selection of an appropriate preprocessing method remains challenging as it necessitates the alignment of the chosen method with both the model and the task at hand. In One example of an inappropriate preprocessing method is when the preprocessing is inadequate, resulting in an excessive amount of noise and outliers that can lead the model astray. Conversely, in the event that the preprocessing stage eliminates a significant number of vibration features within the dataset, the model's capacity to effectively capture the temporal patterns of the time-series data may be compromised (Wang, 2021). A series of Preprocessing steps will be conducted on the features prior to their incorporation into the models.

2.7.1 Data Cleaning

The classification of a data deficiency scenario can be determined by the duration of the missing temporal interval, specifically categorized as either long-term missing or short-term random missing. The historical average is the prevailing method employed for imputing missing data. In instances where a time interval lacks documented data, it will be substituted with the mean value of the recorded data corresponding to the identical time period within the day or week. The historical average method, while straightforward in its implementation, is considered a rudimentary approach that compromises numerous advanced features (Wang, 2021).

2.7.2 Data Smoothing

Data smoothing is a widely employed preprocessing technique in the field of machine learning, aimed at mitigating the presence of noise and variability within datasets. The primary objective is to generate a more refined depiction of the fundamental patterns and tendencies inherent in the data, thereby facilitating the identification and comprehension of significant patterns by machine learning algorithms according to (Rosenfeld et al., 2020; Baykal et al., 2022). The calculation of the exponentially smoothed statistic for a series Y can be performed recursively as shown in Equation 2.2.

$S_0 = Y_0$

For $t > 0, S_t = \alpha * Y_t + (1 - \alpha) + S_{t-1}$ (2.2)

Let α represent the smoothing factor, where α is a value between 0 and 1, exclusive. Increasing the value of α leads to a decrease in the degree of smoothing. When the value of α is set to 1, the smoothed statistic is equivalent to the observed data point. The calculation of the smoothed statistic St becomes feasible once a series of consecutive observations becomes accessible.

2.7.3 Data Normalization

The Normalization method is considered to be the most crucial type of data preprocessing technique, particularly for regression models and neural network models (Fujita & Cimr, 2019). The utilization of an activation function in the neural network model is one of the factors necessitating the implementation of a normalization method. The activation function typically yields a specific range of values. For instance, the Sigmoid function produces outputs within the range of 0 to 1, while the Tanh function yields outputs within the range of -1 to 1. If the dataset surpasses this range, it will significantly diminish the efficacy of employing the activation function, leading to a model that is more challenging to converge. Another rationale for employing a normalization technique is to ensure that the ranges of various data types are standardized. For instance, in cases where both flow and speed data are available, it is observed that the maximum speed limit on a road typically does not surpass 200 units. Conversely, the highest recorded value for traffic flow can easily surpass 300 units, occasionally even reaching 2000 units. Additionally, the model convergence speed will be affected (Kurani et al., 2023).

The Z-score function can be used to normalize input features. The representation of this function is as Equation 2.3:

$$v' = \frac{v-\mu_A}{\sigma_A} \tag{2.3}$$

Let A represent the original dataset, μ denote the mean value of the feature, and σ represent the standard deviation of the feature.

2.8 Technical Indicators Generation

The selection of technical indicators utilized in this dissertation is based on the conventional financial trading indicators commonly employed in the field. Given the focus of traffic agencies on short-term prediction, our interest lies in identifying technical indicators that can enhance the accuracy of such predictions. This section addresses the primary categories of technical indicators, namely trend, momentum, volatility, and volume. The trend indicators primarily utilize moving averages to assess the direction and magnitude of vehicle movement. Conversely, the momentum indicator gauges the velocity of vehicle movement by comparing the present density of vehicles with past densities. volatility indicators utilized in this dissertation assess the velocity of vehicle movement, irrespective of its direction. Additionally, volume indicators are employed to gauge the intensity of a trend, taking into account the density of the volume (Kumbure et al., 2022).

The methodology employed in the computation of each technical indicator for the hourly models is explicated here. In the hourly model, calculations involving a moving average are computed by considering the preceding one-hour period, with a window size of one. The formulas pertaining to TI, necessitate the provision of sequential inputs, specifically in the form of continuous linear time. The MIDAS system generates a sequential dataset by concurrently providing traffic flow and speed data at 15-minute intervals. In a formal context, the h-hour density is employed to compute h-hour TIs. As an illustration, the three-hour SMA with a sample size of two (n = 2) on February 1, 2022, is determined by computing the mean of two two-hour intervals: specifically, the first and second hours of

February 1st. This dissertation employs six widely used and effective technical indicators for short-term prediction.

2.8.1 Average True Range:

According to Soleymani & Paquet (2020), the average true range is a metric utilized to assess market volatility. It achieves this by dissecting the complete range of an asset's price within a specified timeframe. The ATR is a statistical measure that quantifies the range and standard deviation of a given dataset within a specified time period (Turki and Hasson, 2023). ATR calculated as shown in Equations 2.4, 2.5, 2.6, 2.7 and 2.8:

$$ATR = \left(\frac{1}{k}\right) \sum_{i}^{k} TR_i \tag{2.4}$$

$$TR_i = Max\{A_k, B_k, C_k\} \tag{2.5}$$

$$A_k = Highestclose_k - Lowestclose_k \tag{2.6}$$

$$B_k = |Highestclose_k - close_k| \tag{2.7}$$

$$C_k = |Lowestclose_k - close_k| \tag{2.8}$$

where:
TR_i= particular true range
$k =$ no. of periods
Max=Highest value of the three terms
$close_k$=Yesterday’s closing price

2.8.2 Simple Moving Average:

SMA, as described by Muangprathub et al. (2020), is considered a fundamental technical indicator. The utilization of this element is frequently observed as a fundamental component in the computation of various composite indicators, including Bollinger Bands (BBANDs). The SMA is a statistical technique that is employed to identify and analyze trends in data. It achieves this by applying a smoothing process to a given traffic density, utilizing a lag-factor denoted as "n." According to Mak (2021), the utilization of a solitary SMA curve, either independently or in combination with traffic density, holds potential for predicting forthcoming changes in density. The SMA technique has been employed for the purpose of detecting instances of free flow and breakdown, as discussed by (Turki & Hasson, 2023). SMA is calculated as shown in Equation 2.9:

$$SMA = \frac{1}{n}\sum_{i=1}^{n-1} close_i \qquad (2.9)$$

2.8.3 Exponential Moving Average:

This particular technical indicator is widely recognized and frequently utilized. The exponential moving average is a distinct form of averaging that incorporates historical density in a weighted manner. In order to achieve this objective, EMA technique is employed to mitigate the impact of random density fluctuations by calculating the average density over a specific time period. EMA is derived from historical data and, as a result, it functions as a trailing indicator. This implies that EMA lacks the capability to forecast emerging trends, but it can validate the trajectory of an existing trend (Ayala et al., 2021). EMA is calculated as shown in Equation 2.10:

$$EMA = (close_n * k) + (EMA_{n-1} * (1 - k)) \qquad (2.10)$$

Where:

k = the smoothing constant, equal to $\frac{2}{n+1}$

Let *n* represent the quantity of periods in a SMA, which can be reasonably estimated by EMA.

2.8.4 Relative Strength Index (RSI):

The oscillator indicator examines the relative densities of recent free flow and breakdown events. The indicator exhibits oscillation within a range of 0 to 100. A value approaching 100 indicates that the majority of traffic density units during the specified period are classified as density Up, while a value approaching 0 signifies that the majority of traffic density units are classified as density Down. The calculation of density Up and density Down for the hourly models involved determining the average of the preceding s hours, with n being set to 1 in order to capture the hourly trend. The determination of density Up and density Down was achieved using a piecewise function, as described by (Lee, 2022). Specifically, the function evaluated whether the density difference was greater than zero, in which case the value assigned was density Up; otherwise, the value assigned was density Down. RSI is calculated as shown in Equations 2.11, 2.12, 2.13 and 2.14:

$$RSI = 100 - \left[\frac{100}{1-RS}\right] \tag{2.11}$$

where:

$$RS = \frac{average\ gain\ (close\ up)\ over\ n\ periods}{average\ loss\ (close\ down)\ over\ n\ periods} \tag{2.12}$$

$$average\ gain = \frac{(previous\ average\ up\ gain * (n-1) + current\ gain)}{n} \tag{2.13}$$

$$average\ loss = \frac{(previous\ average\ loss * (n-1) + current\ loss)}{n} \tag{2.14}$$

2.8.5 Rate of Change (ROC):

The ROC is a type of oscillator that can be likened to the Momentum (MOM) indicator. It measures the magnitude of a change in a variable as a percentage rather than an absolute value. The ROC serves as a standardized metric for assessing change and can be employed to detect instances of either unrestricted movement or significant disruption that have historically indicated an impending shift in a given trend. It should be noted that when the ROC is above zero, it signifies a general upward trend, whereas when it is below zero, it signifies a downward trend. According to Basak et al. (2019), the ROC does not offer significant predictive value in determining future traffic density patterns. ROC is calculated as shown in Equation 2.15:

$$ROC = \frac{Current\ closing\ price}{closing\ price_{n-periods\ ago}} * 100 \tag{2.15}$$

2.8.6 Momentum (MOM):

The MOM proposed by Demir et al. (2019) is a prominent leading indicator that tracks trends. In further explication, the MOM offers valuable observations regarding the patterns of density, serving as an indicator of smooth traffic flow or congestion when surpassing the positive or negative threshold. In contrast to the SMA, the MOM has the ability to reach its peak or trough prior to the density, thereby offering a prospective (or "leading") trend forecast. According to Nti et al. (2020), the MOM can serve as a predictive indicator by identifying bearish or bullish divergence when it reaches its highest or lowest point and deviates from the primary density trend. MOM is calculated as shown in Equation 2.16:

$$MOM = Current\ closing\ price - \ closing\ price_{n-periods\ ago} \tag{2.16}$$

2.9 Level of service indicator

The determination of traffic conditions based on the available variables presents challenge, thus necessitating the utilization of the level of service indicator for this objective. The Level of Service (LOS) is a commonly employed metric for evaluating the degree of traffic congestion on a specific road segment within vehicular networks (Vrbanić et al., 2022). According to Vrbanić et al. (2023), the Highway Capacity Manual (HCM) has established six distinct categories for freeways and highways in terms of flow, speed, density, and road characteristics. The HCM utilizes traffic density as the principal indicator for LOS assessment on freeway sections as shown in figure 2.4.

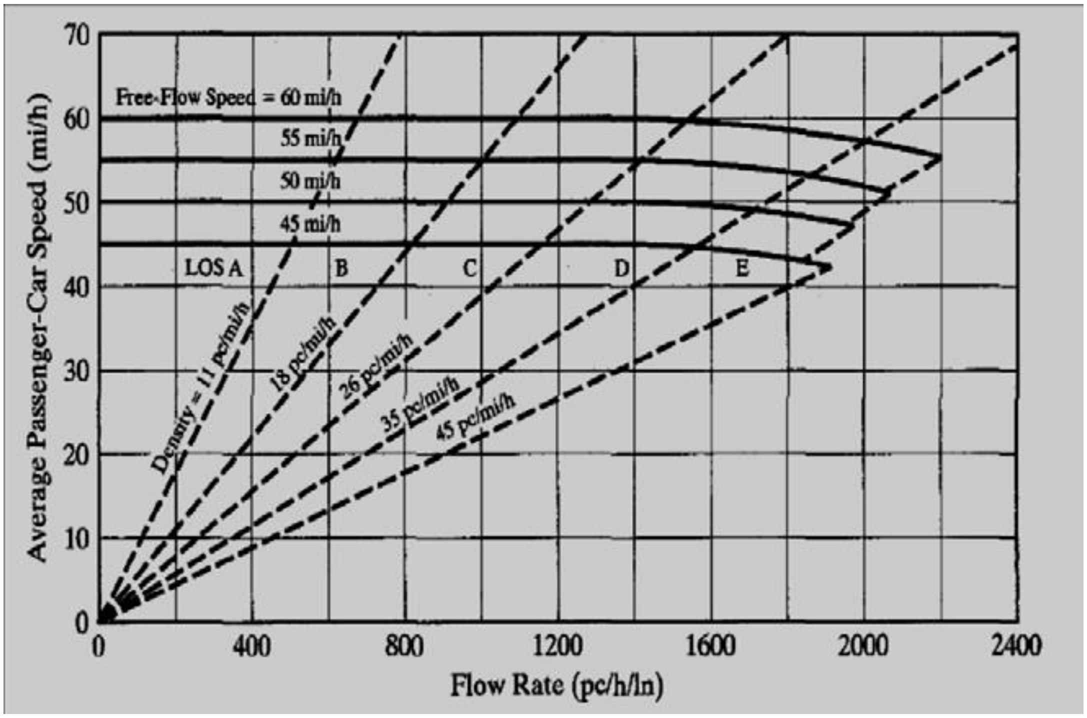


Figure 2.4: Speed-Flow Curves for Multilane Highway Sections (Vrbanić et al. 2023).

The LOS measure is categorized into six distinct classes, denoted as A to F as shown in table 2.2. One notable benefit of the LOS measure is its ability to be easily understood by a wide range of individuals who may not possess technical expertise (Elefteriadou 2016).

Table 2.2: The customary ratings for Highway Level-of-Service (Elefteriadou 2016).

LOS	Description	Density (Veh. /mile)
A	The movement of vehicles on the road occurs at or exceeds the designated speed limit. Motorists possess unrestricted mobility when transitioning between lanes.	0-11
B	The subject exhibits a mild level of congestion, resulting in a slight limitation in maneuverability.	>11-18
C	The capacity to pass or switch lanes is limited. The posted speed limits are being upheld; however, the roads are nearing their maximum capacity. The target level of service (LOS) for the majority of urban highways is as follows.	>18-26

D	The speeds are somewhat decreased, and the maneuverability of the vehicle is constrained. The prevailing conditions observed on urban highways during peak periods.	>26-35
E	The flow of traffic exhibits irregular patterns, with varying speeds that seldom reach the designated limit. This can be classified as a system failure.	>35-45
F	The flow exhibits a coerced nature, characterized by frequent instances of deceleration to an almost negligible velocity of zero miles per hour. The duration of travel is inherently uncertain.	>45

2.10 Models and Architectures

This section provides a comprehensive analysis of the selected models. Subsequently, the models are executed in the experimental phase. The rationale behind selecting these particular models is elaborated upon in section 2.10.1.

2.10.1 The Selected Models Theory

This section provides an explanation of the implemented models. According to researchers, non-parametric models are deemed more suitable for the learning phase of problem-solving in comparison to parametric models. This preference stems from their superior ability to effectively generalize complex data and adapt to its intricate patterns, as exemplified in the context of forecasting traffic data. Parametric tests and methods are predicated on the assumption of underlying statistical distributions within the data. Parametric methodologies are typically favored as the preferred approach due to the presence of noisy input and output traffic variables, as well as the complex and poorly understood nonlinear relationship between them. Pattern recognition-based approaches, which fall under the category of non-parametric approaches, appear to be more suitable due to their efficacy in identifying comparable traffic conditions required for prediction generation. The five models utilized in this dissertation employ a combination of parametric and non-parametric methodologies. These models include Random Forest, Feed Forward Neural Network, Multiple Linear Regression, Markov-chain model, and k-Nearest Neighbors. The rationale behind the selection of the models and a comprehensive mathematical elucidation are provided in the respective sections.

2.10.2 Random Forest (RF)

The initial introduction of the random forests-based approach occurred as a statistical learning technique designed specifically for addressing high-dimensional regression and classification challenges. In this context, classification serves as a means to model categorical variables, while regression is employed to predict continuous variables. RF are a type of ensemble learning method that utilizes trees. These trees are constructed using bootstrap samples and incorporating randomness in the building process (Pavlov, 2019).

The random forests algorithm, which is applicable to both regression and classification tasks, is delineated in the subsequent manner:

1- Take samples from the initial data for the n_{tree} bootstrap.

2- Grow an unpruned regression or classification tree for each of the bootstrap samples, but with the following modification: at each node, instead of selecting the best split among all predictors, randomly sample m_{try} of the predictors and select the best split from these variables. When $m_{try} = p$, the number of predictors, bagging is the special case of random forests that are resulted.

3- By combining the predictions of the n_{tree} trees (average for regression and majority votes for classification,), one can predict new data (Genuer & Poggi, 2020).

For prediction, classification, outlier detection and variable selection, RF is gaining popularity in a number of fields. summarizes the RF method and its uses in the engineering fields. The RF approach has only been applied in Traffic Management and Information Systems to forecast traffic flow under typical traffic conditions, so it has not been widely used in traffic prediction. The study's prediction horizon is 30 minutes, but other variables like prediction step, frequency of data sampling, and traffic conditions are not covered (Hansch, 2020).

The main benefits of RF are that handling over-fits, avoiding feature selection, and only needs a small number of input values. As a result, it may not be effective, especially for time-sensitive applications (like safety applications) where the specified test data is substantial (Mchergui et al., 2022)

2.10.3 Multiple Linear Regression (MLR)

A number of explanatory variables are combined in a statistical process called MLR, also referred to as multiple regression. To model the linear relationship between the explanatory (independent) and response (dependent) variables, multiple linear regression is used. Since multiple linear regression takes into account multiple explanatory variables, it can be considered an extension of ordinary least squares (OLS) regression, which is a commonly used method for calculating coefficients in multiple linear regression (Meerasri & Sothornvit, 2022). It is represented according to Equation 2.17 (Robert et al., 2018).

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_p x_{ip} + U \qquad (2.17)$$

where:

$i = no.of\ observations$

$y_i =\ dependent\ variable$

$x_i = explanatory\ variable$

$\beta_0 = y - intercept\ (constant\ term)$

$\beta_p = slop\ coefficients\ for\ each\ explanatory\ variable$

$U = the\ model\ error\ term\ (also\ known\ as\ the\ residuals)$

The ability to analyze multiple predictor variables and their connections to a response variable is one of MLR's key benefits. Insights into the significance of predictors are provided, confounding variables are controlled for, hypothesis testing is supported, prediction is made easier, and the model's fit and validity are evaluated (Kashyap et al., 2022). However, there are a number of restrictions to take into account when using multiple linear regression on VANET data, including assuming a linear relationship between the predictor variables and the response variable. Multiple linear regression may be used to analyze VANET data; however, the results of this analysis may be adversely affected by outliers or other significant observations (Gonçalves, 2019).

2.10.4 Neural Network

According to Alizadeh (2020), "neural network learning methods provide a robust approach to approximating real-valued, discrete-valued, and vector-valued target functions". In order to recognize or learn patterns in data, neural networks can model complex non-linear relationships between numerous inputs and outputs.

Nodes, connections, layers, and transfer function make up the basic framework of a neural network model. Simple processing units are referred to as nodes or neurons. Weighted connections that reflect the nature of the interaction between the two connected nodes are used to connect them. During the training phase, which is used to calibrate the model using patterns in the data, optimal weights for each connection can be determined. A neural network's layers, which determine nodes1 and connections, are its topology. The state of each neuron is determined by transfer function. Figure 2.5 illustrates the mathematical procedure at a single neuron. There are several synapses on each neuron that are linked to the inputs. They are each distinguished by a weight. Two steps make up this process:

- Creating a linear combination of inputs by calculation.
- Applying an activation function to output the weighted sum.

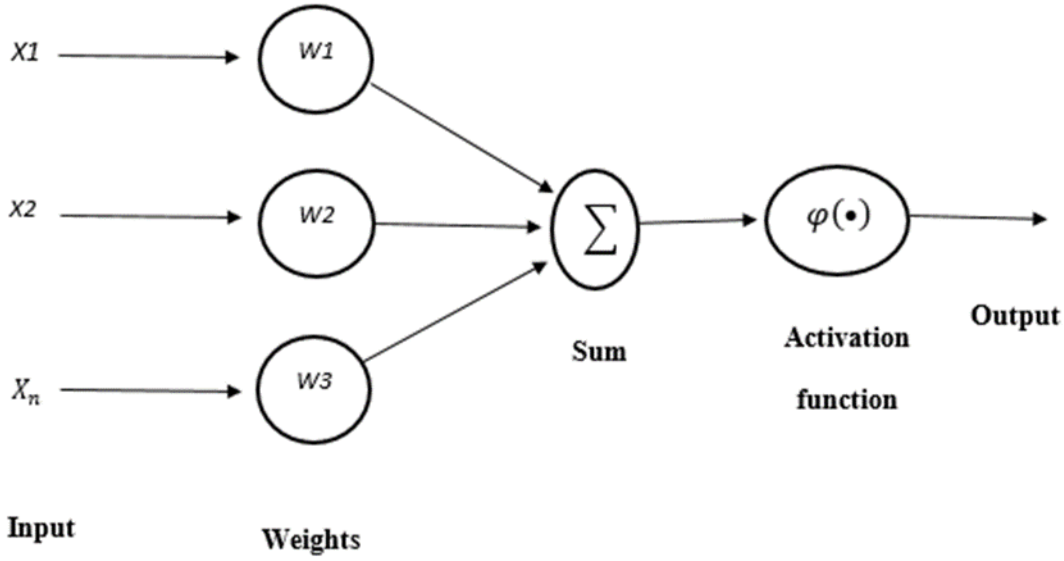


Figure 2.5: A single neuron's process (Rokach, 2010)

Let w_i be the appropriate weight and x_i be the i^{th} input. $Z = \sum w_i x_i$ represents the sum of a linear combination of inputs. The weighted sum is then subjected to a nonlinear activation function, denoted by $\varphi(.)$. This neuron produces the value $y = \varphi(Z)$. Generally speaking, sigmoid, piecewise-linear, and sign functions are the most frequently used activation functions. Feedforward neural network (FFNN), a straightforward and popular network whose structure is depicted in Figure 2.6, is used to predict short-term traffic. This model structure has three layers—an input layer, a few hidden layers, and an output layer—and is devoid of cycles and loops. Each neuron in a layer of the FFNN strictly feeds forward to the output units of the layer above it (Lee, 2021).

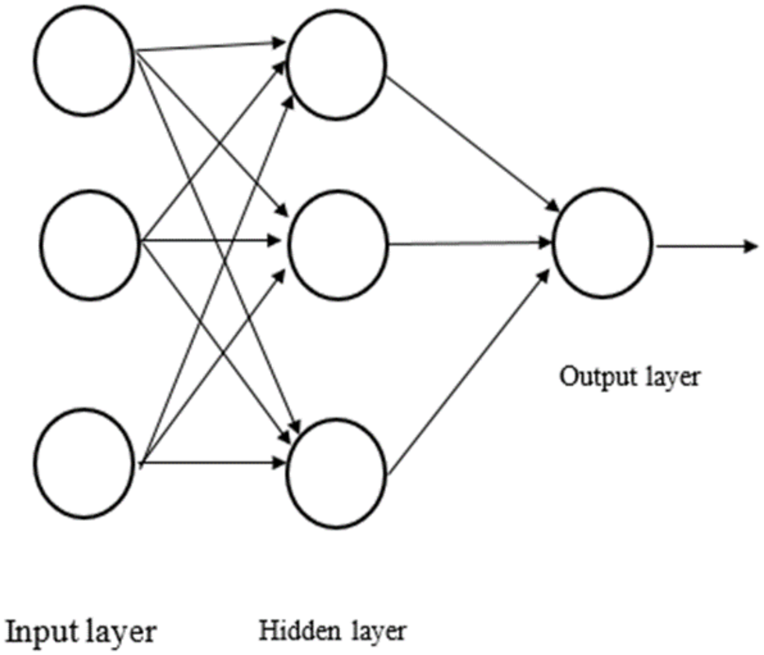


Figure 2.6: Feed-forward networks architectures (Guo, 2013)

One of the primary benefits of neural networks is their ability to learn and model non-linear connections between input and output variables. Additionally, neural networks have the capacity to acquire knowledge directly from data, eliminating the need for explicit understanding or assumptions regarding the underlying data distribution or relationships. neural networks possess the capacity to effectively generalize to unfamiliar data, as they possess the capability to autonomously acquire and extract significant features from unprocessed input data, thereby diminishing the necessity for manual feature engineering. However, neural network encounter certain limitations when applied to VANET. These limitations include difficulties in adapting to dynamic environments and accurately representing the intricate relationships among vehicles within such environments. Neural networks necessitate a substantial quantity of annotated training data in order to acquire and extrapolate patterns with efficacy. Neural networks may encounter scalability issues when confronted with large-scale VANETs, due to the escalating computational complexity and memory demands of the network, which are directly proportional to the number of parameters and connections. Neural networks might not consistently satisfy the rigorous timing requirements of real-time applications as a result of the computational burden associated with training and inference processes. The utilization of neural network models in VANETs frequently necessitates the exchange of information or updates of the models between vehicles. This results in an increase in communication overhead and bandwidth consumption (Naskath et al., 2023; Giovanis, 2010; Di Piazza, 2021).

2.10.5 Markov chains model

Markov chains are a type of stochastic process that can be characterized by estimating the transition probabilities between discrete states based on empirical observations of the systems. The first-order Markov chain is characterized by the property that the probability distribution of each subsequent state is solely determined by the state immediately preceding it. Markov chains of second or higher orders refer to stochastic processes wherein the determination of the subsequent state is contingent upon two or more preceding states (Huang et al., 2022). The probability of transitioning from state S_i to state S_j is denoted as p_{ij} in Equation 2.18. The set of stationary state transition probabilities, denoted as p_{ij}, is structured in this particular way due to the focus of the dissertation on processes where the state transition probabilities do not vary with time.

$$p_{ij} = P\left(S_t = s_j \middle| S_{t-1} = s_i\right) \;,1 \leq i, j \leq N \qquad (2.18)$$

The conditional probability that the Markov chain will transition to state s_j at time *t* given that it was in state s_i at time $t - 1$ (Turki & Hasson, 2022). The transition matrix *P* is used to represent them. For a system with *k* states, the size of the first order transition matrix P is $k \times k$ and it is represented in the following form:

$$P = \begin{bmatrix} p_{1,1} & \cdots & p_{1,k} \\ \vdots & \ddots & \vdots \\ p_{k,1} & \cdots & p_{k,k} \end{bmatrix}$$

The estimation of state probabilities at time t can be derived from the relative frequencies observed in the k states. The Maximum Likelihood Estimates (MLE) of the transition probabilities can be obtained by considering the number of transitions from state i to state j in the sequence of density data, denoted as n_{ij} as shown in Equation 2.19 and Theorem 2.1.

$$p_{ij} = {n_{ij}}/{\sum_j n_{ij}} \qquad (2.19)$$

The transition probabilities of each state exhibit a range between 0 and 1. According to (Candès & Sur, 2020). the total of transition probabilities within a row is equal to one. In a mathematical context, it can be formulated as in Equation 2.20:

$$\sum_{j=1} p_{ij} = 1 \qquad (2.20)$$

Theorem 2.1:
Let X= {X1, X2, X3, …, XM}
represent a sequence of observations of a discrete random variable that draws values from a sample space Ω of finite cardinality K. In other words, xi ∈ Ω for all i = 0, 1, 2, ⋯, M. If X is an observation of a chain from a discrete time Markov model, then the entries of the transition matrix for the model have the following maximum likelihood estimate:

$p_{ij} = \frac{n_{ij}}{\sum_j n_{ij}}$ for all i, j = 1, 2, 3, …, K

Here, (nij) is the transition count for state j to state i. That is, it represents the number of times state j is followed immediately by state i in X.

The Markov chain model offers several notable benefits. Firstly, it is adept at capturing the mobility patterns of vehicles in VANETs. Additionally, it provides a concise and straightforward representation of the system's behavior. Furthermore, the model assumes that the future state of the system solely depends on its current state and is unaffected by past states. This simplifies the modeling process and reduces computational complexity (El Joubari, 2022). Markov chains primarily emphasize the temporal dynamics of states, potentially overlooking the comprehensive consideration of spatial dependencies. Nonetheless, it is important to acknowledge the limitations associated with this approach. The primary emphasis of Markov chains lies in the examination of temporal dynamics of states, potentially overlooking the spatial dependencies within VANET. To ensure precise transition probabilities for the Markov chain model, it may be necessary to undertake extensive data collection and calibration endeavors.

2.10.6 K-Nearest Neighbors (KNN)

K-Nearest Neighbors (KNN) is a popular way to make predictions without building a complex model in advance. Unlike some methods that need a detailed model, KNN doesn't require us to understand all the connections between our data's features and the results we're interested in. Instead, it looks at the data itself. KNN is a flexible and robust tool because it doesn't make strong assumptions about how our data works, making it more adaptable than some other methods, especially for time-related data.

The fundamental premise underlying the KNN method is that instances that are proximate in feature-space are probable to be affiliated with the same class or possess a comparable posterior distribution of their respective classes. KNN method is utilized for data prediction by identifying a set of observations, commonly referred to as nearest neighbors, from a pre-existing dataset. Subsequently, future variables are predicted based on this set of nearest neighbors. According to Ali (2019), the nearest neighbor set has the ability to represent historical traffic data that closely resemble the present traffic conditions in times of congestion. The prediction method based on K-nearest neighbors (KNN) can be decomposed into three essential elements: an observation database, a procedure for searching the neighborhood, and a process for making predictions. The general density of data through the KNN-based prediction method is illustrated in Figure 2.7, as presented by Isnain et al. (2021).

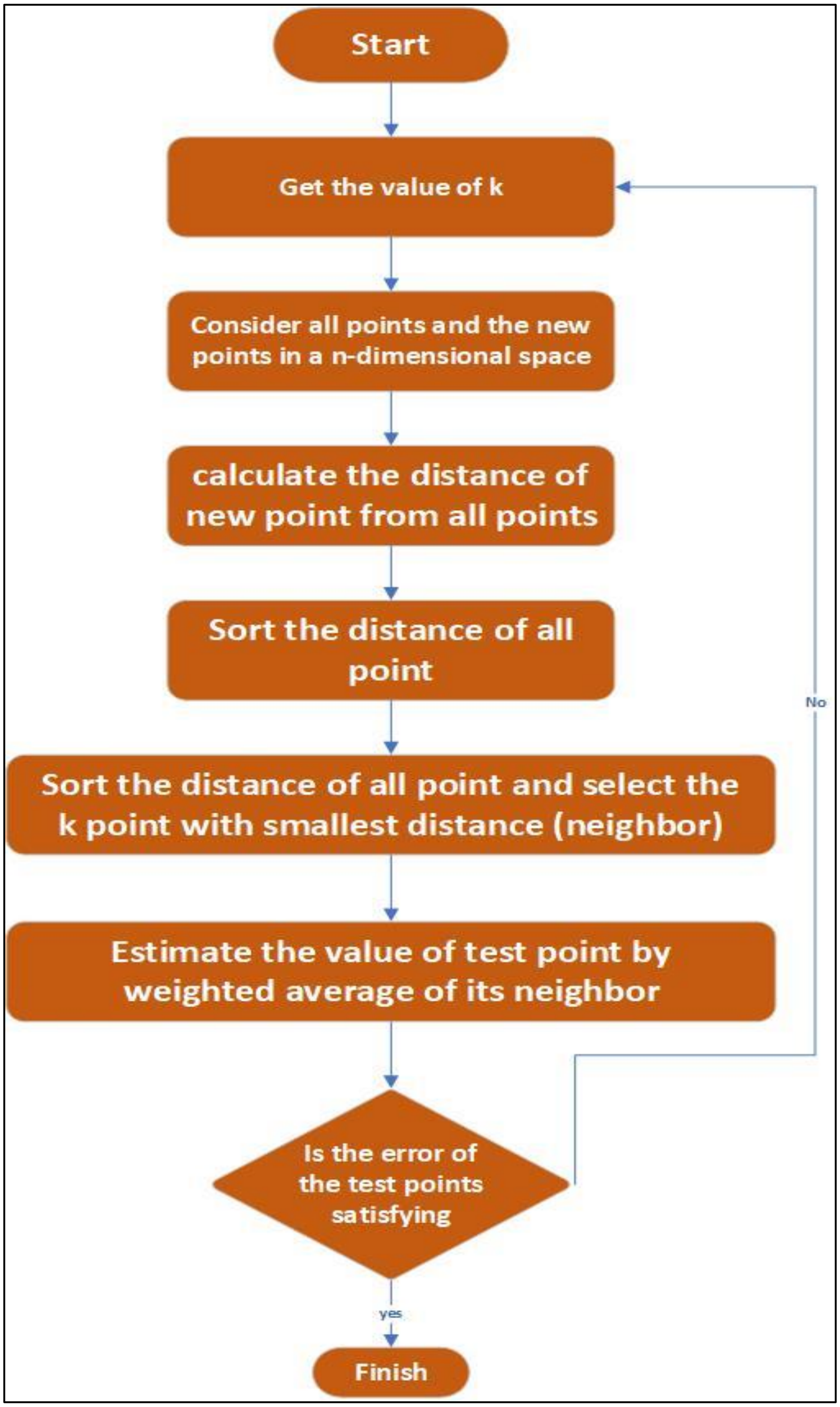

Start
Get the value of k
Consider all points and the new points in a n-dimensional space
calculate the distance of new point from all points
Sort the distance of all point
Sort the distance of all point and select the k point with smallest distance (neighbor)
Estimate the value of test point by weighted average of its neighbor
Is the error of the test points satisfying
No
yes
Finish

Figure 2.7: General structure of the KNN based classification method (Mahdiani et al., 2020).

The search procedure is responsible for identifying the nearest neighbors, which refer to the historical observations that exhibit the highest degree of similarity to the present condition. The closest neighbors are subsequently utilized as the input for the classification process, enabling the calculation of a label for classification. Throughout these three procedures, three crucial design parameters include the establishment of a distance metric for evaluating the proximity between historical data and present circumstances, the determination of an appropriate value for K, and the selection of a classification function based on a set of nearest neighbors.

- The distance metric is employed to ascertain the spatial separation between the present input feature vector and past observations. The metrics that are frequently employed include Euclidean distance, weighted Euclidean distance, the Mahalanobis distance metric, and the Minkowski distance metric. Let $dist_{p,q}$ denote the distance between two feature vectors x_p and x_q, each having a dimension of n (Jiao et al., 2019). The equations representing the three-distance metrics discussed earlier can be found in Table 2.3.

Table 2.3: Equations pertaining to distance metrics (Abu Alfeilat, et al., 2019).

Distance Metric	Equation
Minkowski distance	$dist_{p,q}{}^{2} = \left[\sum_{i=1}^{n} \lvert(x_{pi} - x_{qi})\rvert^{c}\right]^{\frac{1}{c}}$ c: norms of distance metric
Mahalanobis distance	$dist_{p,q}{}^{2} = (x_p - x_q).S^{-1}.(x_p - x_q)^T$ S: the variance covariance matrix
Euclidian distance	$dist_{p,q}{}^{2} = (x_p - x_q).(x_p - x_q)^T$ $c: norms\ of\ distance\ metric$

- The selection of the parameter K determines the number of nearest neighbors that are selected from the historical dataset. For instance, when the value of K is set to 10, the prediction process will involve utilizing the ten historical observations that exhibit the closest distances to the input feature vector (Kelleher et al., 2020).

- The proposed KNN algorithm utilizes the majority vote mechanism. Data is gathered from the training dataset, which is subsequently utilized to generate predictions for novel records. In order to achieve this objective, the proposed neural network algorithm chooses the training instance that is most similar to the given arbitrary instance. Subsequently, the neural network algorithm assigns the class label of the training instance as the predicted class label for the arbitrary instance. KNN algorithm expands upon this procedure by incorporating a predetermined value $k \geq 1$, which represents the number of closest training instances to be considered, as opposed to utilizing just a single instance. The range of typical values typically spans from 1 to several tens (Vidales, 2019).

  One of the primary benefits of KNN algorithm is its inherent simplicity, making it accessible and straightforward to comprehend and apply. KNN algorithm is capable of effectively addressing both binary and multi-class classification problems, rendering it a versatile choice for a wide range of applications in VANETs. KNN algorithm can be employed in the context of VANETs to facilitate localization and positioning tasks. This is achieved by leveraging the positional information of neighboring vehicles. However, it should be noted that the determination of the optimal value of k in VANET is subject to limitations, as the optimal value of k often varies across different datasets. Consequently, the process of identifying the optimal value of k can be challenging and time-consuming (Mchergui et al., 2022).

2.11 Evaluation Measures

Below are the performance metrics commonly used to evaluate the performance of models or algorithms in different fields, especially in the context of machine learning, statistics, and data analysis.

A- Root Mean Square Error (RMSE):

Root Mean Square Error (RMSE) is a commonly employed metric for quantifying the disparities between predicted values (whether from a model or an estimator) and the corresponding observed values, be it a sample or a population. The root mean square error (RMSE) is a statistical measure that quantifies the square root of the second sample moment of the discrepancies between predicted values and observed values, or alternatively, the quadratic mean of these discrepancies. The aforementioned discrepancies are referred to as residuals when the computations are conducted on the data sample utilized for estimation. Conversely, they are labeled as errors (or prediction errors) when computed on data that is not part of the original sample. Mathematically, the calculation can be determined utilizing Equation 2.21:

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{n}(Actual_i - predicted_i)^2} \qquad (2.21)$$

A small RMSE value indicates a close correspondence between the predicted values and the actual values on average. RMSE, as discussed by Saxena (2019), is ideally expected to approach zero (McHugh et al., 2021).

B- Mean Absolute Percentage Error (MAPE):

MAPE is frequently employed as a loss function in regression problems and for evaluating models due to its easily understandable interpretation in relation to relative error. MAPE is a metric used to determine the average magnitude of the absolute difference between predicted and actual values. The accuracy measurement can encompass both positive and negative predictive errors. Mathematically, the calculation can be determined utilizing the following Equation 2.22:

$$MAPE = \frac{1}{N}\sum_{i=1}^{n}\frac{|Actual_i - Predicted_i|}{Actual_i} \qquad (2.22)$$

According to Kumar et al. (2020), a MAPE value below 5% is regarded as an indication that the forecast possesses an acceptable level of accuracy.

C- Average Accuracy Percentage (AA%):

The concept of percentage accuracy pertains to the degree of proximity between a measurement or test and its corresponding true or theoretical value. The aforementioned is an expression denoting the ratio between the discrepancy of the measured value from the true value, divided by the true value itself. The level of accuracy deemed satisfactory varies depending on the specific test being conducted; however, it is generally agreed that a higher accuracy percentage, approaching 100%, is desirable in all instances. Equation 2.23 presented by Faiq (2012) is utilized for the computation of the test's percentage accuracy.

$$AA\% = 100\% - MAPE \tag{2.23}$$

D- Cohen's kappa

Kappa is an additional metric used to evaluate the performance of classification models, which quantifies the degree of agreement between the predicted class labels and the true class labels assigned to instances. The Kappa statistic is employed to mitigate the influence of chance on the accuracy of predictions. The kappa statistic is a valuable tool in situations where the distribution of observations across different categories is imbalanced. It is important to acknowledge that a higher Kappa value corresponds to superior performance of the method. The calculation of the Kappa can be determined by employing the Equation 2.24:

$$k = \frac{(p_o - p_e)}{(1 - p_e)} \tag{2.24}$$

The variable p_o represents the empirical probability of agreement on the label assigned to any sample, also known as the observed agreement ratio. On the other hand, p_e denotes the expected agreement that would occur if both annotators were to assign labels randomly. The estimation of p_e is conducted by employing a per-annotator empirical prior based on the class labels, as discussed in the works of De Raadt et al. (2019) and Hoseinzadeh et al. (2021).

E- Precision

The Precision criterion measures the ability to predict rising behavior accurately based on the parameter FP, as defined by Equation 2.25. However, it's important to note that Precision only assesses the accuracy of predicting upward trends and may not identify instances where increasing behavior is wrongly anticipated as declining, as pointed out by Han et al. (2011). Therefore, an additional criterion is considered to address this limitation.

$$Precision = \frac{TP}{TP + FP} \tag{2.25}$$

F- Recall (True Positive Rate)

The Recall, which measures the True Positive Rate, is employed to identify true positive predictions linked to the FN parameter. However, the Recall metric falls short in its ability to differentiate between forecasts that incorrectly predict a decrease when an increase is expected. This criterion is calculated using Equation 2.26 from the study by Han et al., (2011).

$$Recall = \frac{TP}{TP+FN} \tag{2.26}$$

G- F1 Measure criterion:

Given the statements above, it is feasible to reduce the recall value by elevating the Precision parameter and vice versa. Since both Recall and Precision hold significance in algorithm training, a third metric known as the F1-Measure is utilized. This metric combines the two aforementioned criteria and signifies how effectively the algorithm predicted growth behavior in terms of Precision. The F1-Measure is calculated using Equation 2.27, as outlined by Han et al., (2011).

$$F1 - score = \frac{2*TP}{2*TP+FN+FP} \tag{2.27}$$

H- Accuracy

Accuracy refers to the proportion of accurately classified predictions (specifically, LOS in this particular dissertation) relative to the ground truth data. Equation 2.28 provided by Kumar et al. (2019) can be utilized to determine the accuracy.

$$Accuracy = \frac{Number\ o\ f\ correct\ predictions}{Total\ number\ o\ f\ predictions} = \frac{TP+TN}{TP+FP+TN+FN} \tag{2.28}$$

I- Receiver Operator Characteristic and AUC Score

The Receiver Operator Characteristic curve is a widely used evaluation metric in the context of multiclass classification problems. Receiver Operator Characteristic curve is a graphical representation of the relationship between the true positive rate (recall) and the false positive rate (FPR) at different threshold values as shown in Equation 2.29. Its primary function is to distinguish the relevant information (signal) from the irrelevant information (noise). In alternative terms, it demonstrates the efficacy of a classification model across various thresholds for classification.

$$\mathrm{FPR} = \frac{FP}{FP+TN} \tag{2.29}$$

The Area Under the Curve (AUC) is a metric utilized to assess the discriminative capacity of a multiclass classifier in distinguishing between different classes. It serves as a concise representation of the Receiver Operating Characteristic (ROC) curve. The area under the receiver operating characteristic (ROC) curve, commonly referred to as AUC, is a numerical measure that ranges from 0 to 1. A model that produces predictions with an accuracy of 0% is associated with an Area Under the Curve (AUC) value of 0.0, while a model that produces predictions with an accuracy of 100% is associated with an AUC value of 1.0. According to (Nahm 2022; Basak et al., 2019).

J- Out-of-bag (OOB)

After generating all decision trees within the forest, we proceed to select, for each training sample $Z_i = (X_i, Y_i)$ in the original training set T, all bagged sets T_k that do not include Z_i. The provided set comprises bootstrap datasets that lack a specific training sample from the original training dataset. These sets are commonly referred to as out-of-bag examples. For every *n* data samples in the original training dataset, there exists a corresponding set of *n* such sets. OOB error refers to the average error for each Z_i, which is computed by utilizing predictions from the trees that do not include Z_i in their respective bootstrap sample. OOB error serves as a metric for evaluating the generalization error of a random forest model, quantifying its ability to accurately predict data that it has not been trained on (Basak et al., 2019).

2.12 Software Implementation Details

This dissertation employs Python 3.9.12[1] and the Anaconda distribution for the purpose of research development and experimentation. The selection of Python as the programming language is based on its widespread usage and its appropriateness for real-time processing tasks. The dynamic interpretation capability of the language facilitates efficient and expedited development.

Moreover, the favorable choice of Python is attributed to its seamless integration with established C libraries designed for scientific computing and resource-intensive tasks. The selection of Python is justified by the extensive and growing community support provided by its contributors.

2.12.1 Data Exploration Library

The python pandas library was utilized for conducting data exploration and pre-analysis. Pandas facilitates the process of data exploration by transforming the data into tabular structures and frames that consist of columns. Pandas[2] is a highly efficient scientific tool that facilitates the resampling of time series data, reindexing of data frames, and grouping of data by column headers. These functionalities enhance the comprehension and visualization of data plots directly from the data frames.

2.12.2 Machine Learning Implementation Library

The dissertation employs open-source library packages to implement the model architectures that were previously discussed. The primary machine learning libraries employed are Keras[3] and TensorFlow[4]. Keras, a Python-based high-level machine learning API, is commonly favored over TensorFlow owing to its expedited experimentation capabilities and its user-friendly approach to constructing intricate neural networks from the ground up. Keras provides support for a wide range of neural network architectures and facilitates the distribution of computational tasks across multiple CPU or processor cores.

TensorFlow, an API developed by Google, operates on computational graphs and facilitates the creation of novel architectures using foundational units. The platform provides functionalities such as streamlined model development and support for both central processing unit (CPU) and graphics processing unit (GPU) computations. The utilization of the scikit-learn library[5] is common for the purposes of data Preprocessing, as well as the preparation of training and validation datasets. The tool enables the creation of model flow pipelines and the identification of optimal parameters through the utilization of the grid search functionality.

It is worth noting that the functions of scikit-learn are compatible with the high-level prediction functions of Keras. The optimal model parameters can be located in Appendix A.

[1] https://www.anaconda.com/distribution/

[2] https://pandas.pydata.org/

[3] https://keras.io/

[4] https://www.tensorflow.org/

[5] https://scikit-learn.org/

Chapter Three

The Proposed System

3.1 Overview

This dissertation presents a new system for short-term traffic prediction utilizing machine learning methodologies to forecast traffic variables across various traffic scenarios. Furthermore, the proposed model utilizes a level of service (LOS) indicator to assess traffic congestion and estimate traffic conditions.

3.2 The Proposed System Architecture

As depicted in Figure 3.1, the proposed system comprises four primary stages. Each stage comprises multiple sub-stages that are determined based on the necessary tasks to be accomplished within them. Additional information is necessary to facilitate a comprehensive analysis of the proposed models for predicting and categorizing traffic congestion.

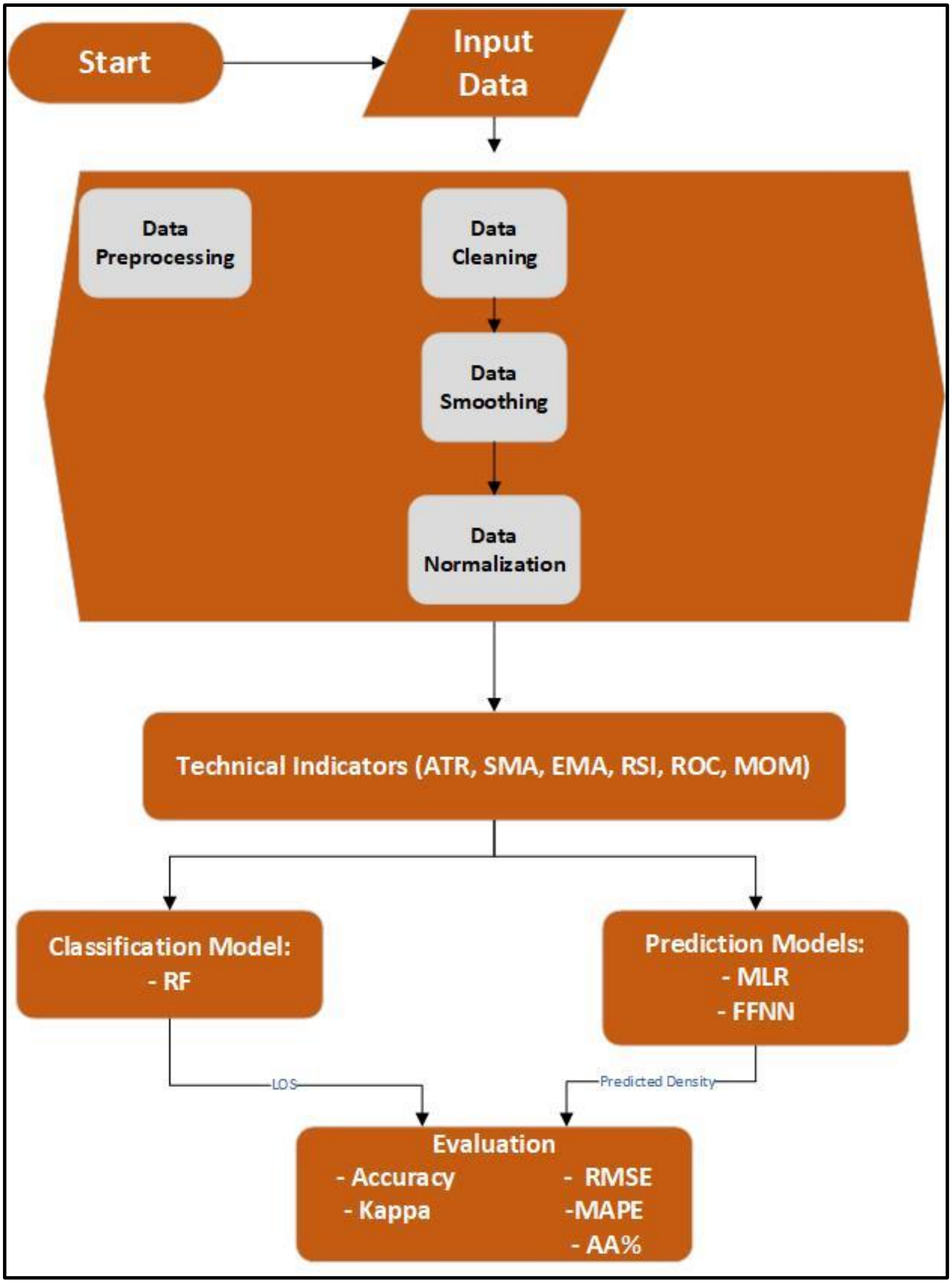


Figure 3.1: The architecture of the proposed system

3.2.1 Input Data

Prior research has made predictions regarding the relationship between speed and flow in order to assess traffic conditions. The present dissertation integrated various parameters, including speed and flow data, to estimate traffic density. This estimation was performed by employing Equation 2.1, as outlined in Section 2.6, utilizing the MIDAS dataset as shown in Table 2.2. The density of the vehicles that have been calculated is utilized as an input for the computation of technical analysis indicators. The following sections will provide an elaboration on each index, namely speed, flow, and density.

- Speed can be defined as the measure of an object's motion, specifically the amount of distance it covers within a given unit of time.
- Flow is a widely recognized and frequently studied traffic parameter. Flow refers to the velocity at which vehicles traverse a specific location on the road, typically quantified as the number of vehicles passing through per hour.
- Density pertains to the quantity of vehicles that are present within a specific distance of roadway. Typically, density is expressed in units of vehicles per kilometer.

3.2.2 Preprocessing Stage

Preprocessing means preparing data for mining in a proper form. The goal of this stage is to build a database for MIDAS that is organized, structured and evaluated as a benchmark. The data is processed and prepared into prediction (classification or regression) because actual dataset like MIDAS might have some unsuitable structure. This stage consists of three main steps, including data cleaning, smoothing and normalization. Algorithm 3.1 illustrates the preprocessing stage.

Algorithm 3.1: Preprocessing MIDAS Dataset

```
Input: An array RD (r), represents the raw MIDAS dataset where r is the historical data size (vehicle density over time).
Output: Array PD (r): Processed MIDAS dataset.

Data Cleaning (Historical Average Method):
Set k_clean ← 5, the number of time periods to calculate historical averages.
For each data point t in RD do
    If ($RDt\ missing$) then
        $RDt$ ← $\mathbf{historical\_avg\_method}(RD$, t, k_clean)
    End if
End For

Data Smoothing (Exponential Smoothing Method):
Set alpha ← 0.2 (smoothing parameter).
For each data point t in RD do
    If ($RDt\ smoothing$) then
        $RDt$ ← $\mathbf{exponential\_smooth}(RD$, t, alpha)
    End if
End For

Data Normalization (Z-score Method):
Calculate the mean ($\mu$) and standard deviation ($\sigma$) of the smoothed dataset RD.
For each data point t in RD do
    If ($RDt\ normalizing$) then
        $RDt$ ← $\mathbf{z\_score\_normalize}(RD$, t, $\mu$, $\sigma$)
    End if
End For

Combine the Preprocessed Features (if necessary):
If you need to store the preprocessed data separately, set PD ← RD after cleaning, smoothing, and normalizing.
Otherwise, PD ← RD.
End.
```

According to Algorithm 3.1, the first step is to clean the data from missing values using the historical mean method. The dataset containing the raw links showed an approximate missing rate of 0.5% for its values, as the proportion of these missing values is relatively insignificant compared to the overall data collected. Data smoothing aims to improve the quality of traffic data by reducing noise and helping machine learning identify basic traffic patterns. This is done in (Step 2) by means of exponential smoothing, similar to a low-pass filter, which works to enhance the importance of recent data while reducing the importance of previous observations using Equation 2.2. In the final step, the smoothed input feature records are converted to a specified range using data normalization. In our thesis, the Z-score function was used to standardize data on vehicular traffic density using Equation 2.3. These three steps are all explained in more detail in Section 2.6.

3.2.3 The Technical Indictors Generation Stage

The active stock traders and technical analysts commonly use TIs to analyze short-term and long-term price movements and to identify entry and exit points. Technical indicators can be useful while predicting the future density of vehicles so they can be integrated into traffic management systems. As a result, this stage aims to enhance the accuracy of prediction by taking the standard features (six features) as an input for the models. The outputs from this stage are the six TIs features.

As delineated in Section 2.8, the primary classifications of technical indicators encompass trend, oscillator, volume, and momentum. The trend indicators primarily utilize moving averages to assess fluctuations in traffic density, determining whether it is on the rise or decline. On the other hand, the oscillator indicators are employed to identify recurring patterns in traffic density. Lastly, the momentum indicator serves to gauge the strength and anticipated level of traffic density. Volatility indicators are utilized to assess the rate at which vehicles move, irrespective of their direction. On the other hand, volume indicators are employed to gauge the strength of a trend by considering the density of trading volume.

The success of individual TIs is contingent upon the specific domain and time period. Density-based trend-following indicators can be advantageous during periods characterized by high levels of autocorrelation. We exercise caution in the process of selecting a list of technical indicators for examination. Consequently, we opt for TIs that meet the following two criteria.

- The calculation of the Technical Indicator which solely necessitates the consideration of traffic density.
- The inclusion of Traffic Intensity is expected to enhance the predictive performance by emphasizing the presence of oscillations or patterns in traffic densities.

The methodologies employed to derive each technical indicator for the hourly models are explicated in this section. Calculations involving a moving average are computed by considering the preceding one-hour time frame, with a window size of one. Notably, the initial 15-minute value is excluded from the calculation. In the following section, we present a compilation of selected Technical Indicators and offer the corresponding mathematical expressions for their computation and the procedures for its implementation.

Other details of the TIs generating stage are presented in Algorithm 3.2. This algorithm is constructed using Equations 2.4 to 2.16. Each Equation was modified according to the input (vehicle density) so that the proposed new Equations are 3.1 to 3.13.

Algorithm 3.2: The TIs Generation stage for MIDAS Dataset

***Input**: An array **S ($n \times m \times r$),** represents MIDAS dataset where **n** is the number of data points, **m** is the number of standard technical features, and **r** is historical data size.*

***Output**: Array **O1** ($n \times j \times r$), where **j** is the number of Technical Indicator features.*

```
Begin
Let k represent the period sliding window.
Initialize $k \leftarrow 4$.
Initialize an empty array O1 of size ($n \times j \times r$) to store the Technical Indicator features.
For each data point t in S do
    For $i \leftarrow 0$ to m do
        Execute ATR(Density[t], High[i], Low[i], Density, t, i, k) in parallel and store the result in ATR(i).
        Execute SMA(Density, t, i, r) in parallel and store the result in SMA(i).
        Execute EMA(Density, t, i, r) in parallel and store the result in EMA(i).
        Execute RSI(Density, t, i, r) in parallel and store the result in RSI(i).
        Execute ROC(Density, t, i, r) in parallel and store the result in ROC(i).
        Execute MOM(Density, t, i, r) in parallel and store the result in MOM(i).
    End For
End For
End For
Merge array S of size ($n \times m \times r$) with array O1 of size ($n \times j \times r$) to create Input2 of size ($n \times |S+O1| \times r$).
End.
```

The input to Algorithm 3.2 is n arrays with size (m×r), which represents MIDAS dataset where m is the no. of standard technical features, and r is historical data size. It implements six algorithms (ATR, SMA, EMA, RSI, ROC, MOM), these are also executed in parallel to find a TIs features.

A- Average True Range (ATR):

The proposed indicator is designed to assess the level of volatility in traffic density, thereby capturing the prevailing trend. During a one-hour time frame, three distinct calculations are performed: (a) the difference between the highest and lowest traffic densities, (b) the difference between the highest traffic density and a previously recorded traffic density denoted as $Density_n$, and (c) the difference between the lowest traffic density and a previously recorded traffic density $Density_n$. The highest value among the three options is chosen for each hour of traffic, and then averaged over a continuous one-hour period (k=1h). The ATR indicator can be calculated using the Equations 3.1, 3.2, 3.3, 3.4 and 3.5.

$$ATR = \text{ average (TR, k)} \tag{3.1}$$

Where:

$$TR = Max\{A_k, B_k, C_k\} \tag{3.2}$$

$$A_k = HighestDensity_k - LowestDensity_k \tag{3.3}$$

$$B_k = |HighestDensity_k - Density_k| \tag{3.4}$$

$$C_k = |LowestDensity_k - Density_k| \tag{3.5}$$

To compute TR, take the largest value between $A_k, B_k and\ C_k$.

The algorithm 3.3 to calculate the ATR indicator as follows:

Algorithm 3.3: Average True Range (High, Low, Density, i, k)

```
Input:
High: Array of high values.
Low: Array of low values.
Density: Array of density values.
i: Current index.
k: Period sliding window.
Output:
ATR: Average True Range.
Begin
1.  Define array TR with size k to store True Range values.
2.  For t ← i down to (i - k + 1) do:
    a.  Calculate True Range (TR[t]):
        •  Set HighLowDiff = High[t] - Low[t]
        •  Set HighDensityDiff = abs (High[t] - Density [t - 1])
        •  Set LowDensityDiff = abs (Low[t] - Density [t - 1])
        •  TR[t] = max (HighLowDiff, HighDensityDiff, LowDensityDiff)
//True Range is the greatest of the three terms above, representing the maximum distance between the current high and low densities, the current high and the previous density, and the current low and the previous density.//
3.  Calculate ATR as the average of the True Range values:
    a.  Set ATRSum = 0
    b.  For j ← 0 to (k - 1) do:
            i. ATRSum = ATRSum + TR[j]
    c. ATR = ATRSum / k
4.  Return ATR
End.
```

where:

Density is an array that contains the vehicles density

High is an array that contains the highest densities.

Low is an array that contains the lowest densities.

i is the index of the current data point to calculate the ATR.

k is the chosen period for the ATR.

B- Simple Moving Average (SMA):

This is used to identify trends and reversals, as well as to set up free-flow and breakdown levels. The SMA indicator can be calculated using the Equation 3.6.

$$SMA = \frac{1}{n}\sum_{i=1}^{n-1} Density_i \tag{3.6}$$

To calculate the SMA indicator, Algorithm 3.4 can be used:

```
Algorithm 3.4: Simple Moving Average (Density, i, n)

Input:
Array Density of vehicle density values.
Integer i representing the current index.
Integer n representing the chosen period.
Output:
Single value representing the SMA.
________________________________________________

Begin
Initialize sum to 0.
For t from i down to (i - n + 1) do:
        a. Add Density[t] to sum: sum ← sum + Density[t]
Calculate SMA as the sum divided by the chosen period "n":
a. SMA ← sum / n
Return SMA as the result.
End.
```

In this algorithm:
Density is an array that contains of the vehicle densities.
i is the index of the current data point for which you want to calculate the SMA.
n is the chosen period for the moving average

C- Exponential Moving Average (EMA):

EMA is a distinct variant of the moving average method that employs an exponential weighting scheme to calculate the average of historical densities. By assigning weights, the EMA has the ability to assign higher importance to recent density trends. The inclusion of weighting in EMA sets it apart from SMA by enabling the EMA to promptly respond to fluctuations in density. In instances characterized by heightened volatility, assigning greater significance to recent density fluctuations can confer a strategic advantage. The EMA indicator can be calculated using the Equation 3.7:

$$EMA = (Density_n EMA_{t-1}) * k + EMA_{n-1} \tag{3.7}$$

where k = the smoothing constant, equal to $\frac{2}{n+1}$
Let n represent the quantity of periods in a SMA, which can be reasonably estimated by EMA.

To compute the EMA indicator, Algorithm 3.3 can be used:

Algorithm 3.5: Exponential Moving Average (Density, i, n)

Input:

Array Density of vehicle density values over time.

*Integer **i** representing the current index.*

*Integer **n** representing the period.*

Output:

EMA[i]: Exponential Moving Average at the current index.

```
Begin
    1. If i is equal to 0, then
        a. Set $EMA[i] = Density[i]$. //The first EMA value is the same as the
           corresponding Density of vehicle. //
    2. If i is greater than 0, then
        a. Set $k = 2 / (n + 1)$. (The smoothing constant for the EMA)
        b. Set $EMA[i] = (Density[i] - EMA[i-1]) * k + EMA[i-1]$.
           //The EMA at the current index is calculated by taking the difference
           between the current Density of vehicle and the previous EMA value, then
           multiplying it by the smoothing constant k, and adding it to the previous
           EMA value.//
    3. Return EMA[i].
    End.
```

where:

- ***Density*** is an array that contains the closing prices of the financial instrument.
- ***i*** is the index of the current data point for which you want to calculate the EMA.
- ***n*** is the chosen period for the EMA.

D-Relative Strength Index (RSI):

RSI is a metric that evaluates the current strength of traffic, the rate at which the trend is changing, and the extent of the movement. Equations 3.8, 3.9, 3.10, and 3.11 utilized to calculate RSI is as follows:

$$RSI = 100 - \left[\frac{100}{1-RS}\right] \qquad (3.8)$$

where:

$$RS = \frac{average\ up\ density\ over\ n\ periods}{average\ down\ density\ over\ n\ periods} \qquad (3.9)$$

$$average\ up\ density = \frac{(previous\ average\ up\ density * (n-1) + current\ up\ density)}{n} \qquad (3.10)$$

$$average\ down\ density = \frac{(previous\ average\ down\ density * (n-1) + current\ down\ density)}{n} \qquad (3.11)$$

To compute the RSI indicator, Algorithm 3.6 can be used:

Algorithm 3.6: Relative Strength Index (Density, i, n)

Input*:*

- *An array Density of size n representing vehicle density data.*
- *An integer i representing the current index.*
- *An integer n representing the period for RSI calculation.*

Output*:*

- *RSI, the Relative Strength Index for the current index i.*

```
Begin
    1. Initialize AvgUpDensity[i] to 0.0 (Initialize the average up density for the
       current index to 0).
    2. Initialize AvgDownDensity[i] to 0.0 (Initialize the average down density
       for the current index to 0).
    3. If $i = 0$, return 0. (RSI is not defined for the first data point).
    4. For t from 1 to i do the following:
       a. Calculate the density change: $Diff[t] = Density[t] - Density[t-1]$.
       b. If $Diff[t] > 0$, add Diff[t] to AvgUpDensity[i].
       c. If $Diff[t] < 0$, subtract Diff[t] from AvgDownDensity[i].
    5. Calculate the average up density for the current index:
       a. Divide AvgUpDensity[i] by n.
    6. Calculate the average down density for the current index:
       a. Divide AvgDownDensity[i] by n.
    7. Calculate the relative strength (RS) for the current index:
       a. If AvgDownDensity[i] is zero, set RS to a large value (e.g., infinity) to
       avoid division by zero.
       b. Otherwise, set $RS = AvgUpDensity[i] / AvgDownDensity[i]$.
    8. Calculate the RSI for the current index:
       a. Set $RSI = 100 - (100 / (1 + RS))$.
    9. Return RSI as the result of RSI calculation for the current index i.
End.
```

E- Rate of Change (ROC):

The ROC indicator serves to highlight the observed increases in volume. These occurrences typically manifest predominantly at points of high density, low density, or instances of abrupt change. The ROC indicator can be calculated using the Equation 3.12:

$$ROC = \frac{Current\ density - density_{n-period\ ago}}{density_{n-periods\ ago}} * 100 \qquad (3.12)$$

To compute the ROC indicator, Algorithm 3.7 can be used:

```
        Algorithm 3.7: Rate of Change (Density, i, n)

Input:

        Array Density of size r, representing the density of vehicles at different time periods.
        Integer i, the current index for which ROC is to be calculated.
        Integer n, the number of periods ago to compare with.

Output:

        Float ROC, representing the rate of change of density.
        ________________________________________________________________

Begin

        1.  If i < n, return "Not enough data points to calculate ROC."
        //ROC requires at least n data points; if the current index is less than n, there won't be
        enough data.//
        2.  Set CurrentDensity = Density[i] (The density at the current index).
        3.  Set DensityNPeriodsAgo = Density [i - n] (The density n periods ago).
        4.  Compute the rate of change (ROC) using the formula:
        ROC = ((CurrentDensity - DensityNPeriodsAgo) / DensityNPeriodsAgo) * 100.
        5.  Return ROC.

End.
```

F- Momentum (MOM):

The MOM is a leading indicator that tracks trends. In further explication, MOM offers valuable observations regarding density patterns, serving as an indicator of smooth traffic movement or congestion when surpassing or falling below the zero threshold. In contrast to SMA, MOM has the ability to reach its peak or trough prior to the density, thereby offering a proactive (or "leading") indication of the trend. MOM serves as a prospective indicator, indicating potential bearish or bullish divergence when it reaches its highest or lowest point and deviates from the primary density trend. The ATR indicator can be calculated using Equation 3.13.

$$MOM = Current\ density - \ density_{n-periods\ ago} \tag{3.13}$$

To compute the MOM indicator, algorithm 3.8 can be used:

Algorithm 3.8: Momentum (Density, n, i)

***Input**:*

***Density**: An array representing the density of vehicles at different time periods.*

***n**: The number of periods ago to compare.*

***i**: The current time period.*

***Output**:*

***MOM**: The calculated momentum value.*

```
Begin
    1. Initialize MOM to 0.
    2. Calculate MOM as the difference between the current density and the density
       "n" periods ago:
        a. MOM = Density[i] - Density [i - n].
    3. Return MOM.
End.
```

3.2.4 The Prediction Stage

The prediction of short-term traffic is a multifaceted and intricate dynamic problem, necessitating the utilization of machine learning techniques to effectively address the dynamic nature of the process. Researchers often apply statistical methods to historical traffic data, which is an exhaustive task and may produce incorrect predictions. The machine learning coupled with fundamental and / or technical analysis which can yield satisfactory results for traffic density prediction. In this work, an effort is made to predict the density and density trend of vehicles by building a new prediction model. It is implemented by one from two models: FFNN model to predict the future traffic density in different scenarios and RF classifier model to know traffic conditions, whether the flow is free or congested. The approach proposed in this work is capable to identify hidden relationships and underlying dynamics in the historical traffic data.

A- Feed Forward Neural Network (FFNN)

The main objective of the FFNN model is to predict the future density of vehicles. This is achieved by building a certified and a perfect regression model for the traffic density. This model is based on solving the problem completely from the input until predicting the density. The implementation of the topology of FFNN is executed. The implementation of neural networks involves two distinct stages: training and prediction.

Input is the first layer and it is compatible with input variables (ATR, SMA, EMA, RSI, ROC, MOM) of a problem with a node for each input variable (6 neurons). The hidden layer is the second layer, which is used to capture nonlinear relationships between variables using the tanh activation function. Three neurons are determined in hidden layer in addition to bias. The third layer represents the output layer, which is used to provide the expected values using the linear activation function. One neuron is selected in output layer, which represents the desired output (density in continuous form). Equation 3.15 describes the relationship between the input x_t(ATR, EMA, SMA, RSI, ROC and MOM) and the output Y_t $(Density)$:

$$Y_t = F\left(w_0 + \sum_{a=1}^{h} W_a * F\left[w_{0,a} + \sum_{b=1}^{m} w_{b,a} * x_t\right]\right) \quad (3.15)$$

Where $w_{b,a}$ (a=1, 2, …, h; b=1, 2, …, m), W_a (a=0,1, 2, …, h), $w_{0,a}$,and w_0 are the weights of the network, m: input neurons, h: hidden neurons. $w_{b,a}$ represents the weights between input and hidden layer, W_a represents the weights between hidden and output layer, $w_{0,a}$ represents the weight of input bias neuron, w_0 represents the weight of hidden bias neuron, and finally F() represents activation function.

In order to attain an optimized network performance, it is necessary to train neural networks by iteratively adjusting the weight values and minimizing network bias. Training performance was evaluated using standard mean square error (MSE). This criterion calculates the average squared error between the network outputs and the target outputs. The equation that provides the definition of MSE is expressed as Equation 3.16.

$$MSE = \frac{1}{n} \sum_{i}^{n} (output_i - target_i)^2 \quad (3.16)$$

Let n represent the sample size, $output_i$ denote the predicted results for i, and $target_i$ represent the actual value for i. The training process will be terminated once the mean squared error (MSE) reaches a threshold value that has been predetermined. During the prediction phase, the model that has been developed and trained in the preceding training phase can be employed to compute the network output for new input data during the testing process. Figure 3.2 illustrates the sequential steps involved in the training and prediction processes when employing a Feedforward Neural Network (FFNN) methodology.

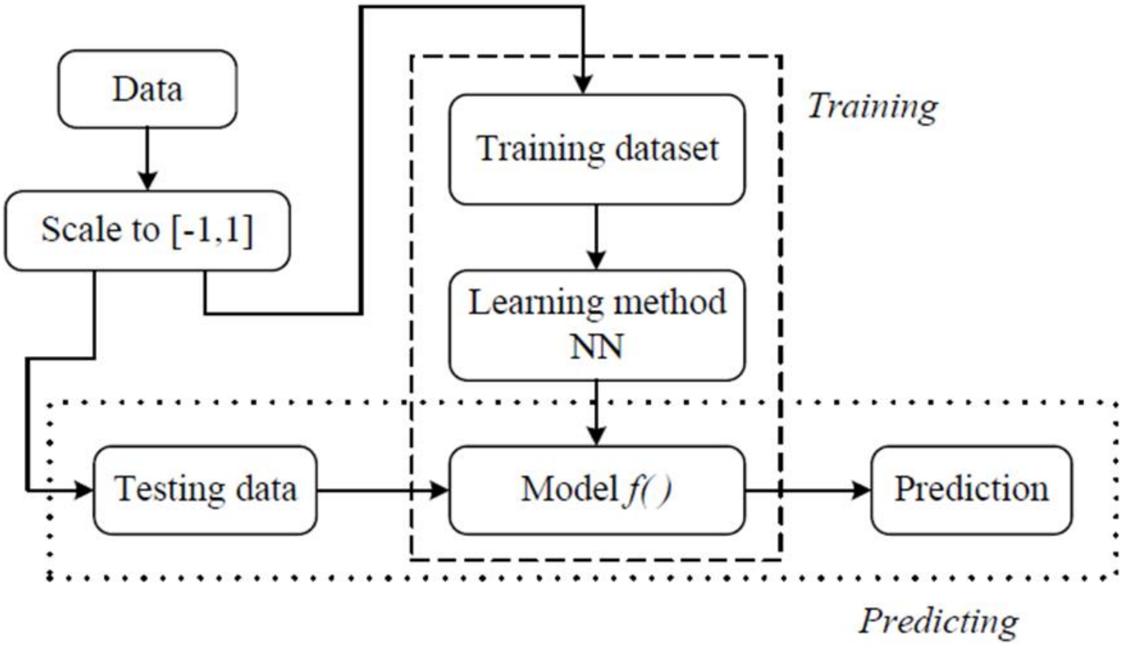


Figure 3.2: The methodology employed for prediction problems involves the utilization of a FFNN based approach.

The research employed a methodology for constructing the artificial neural network model, which involved the creation of several sub-models. These sub-models encompassed the Input Model, Output Model, Data Division Model, Neural Network Architecture Selection Model, Adjusting Weight Model, and Learning Rate Model as mention in appendix D. The artificial neural network was constructed using the statistical package for social sciences (SPSS) software. The utilization of this program has been employed to illustrate the process of constructing sub-models based on an existing model. The input terminates with the output model, as depicted in Figure 3.3, illustrating the architecture of the proposed neural network.

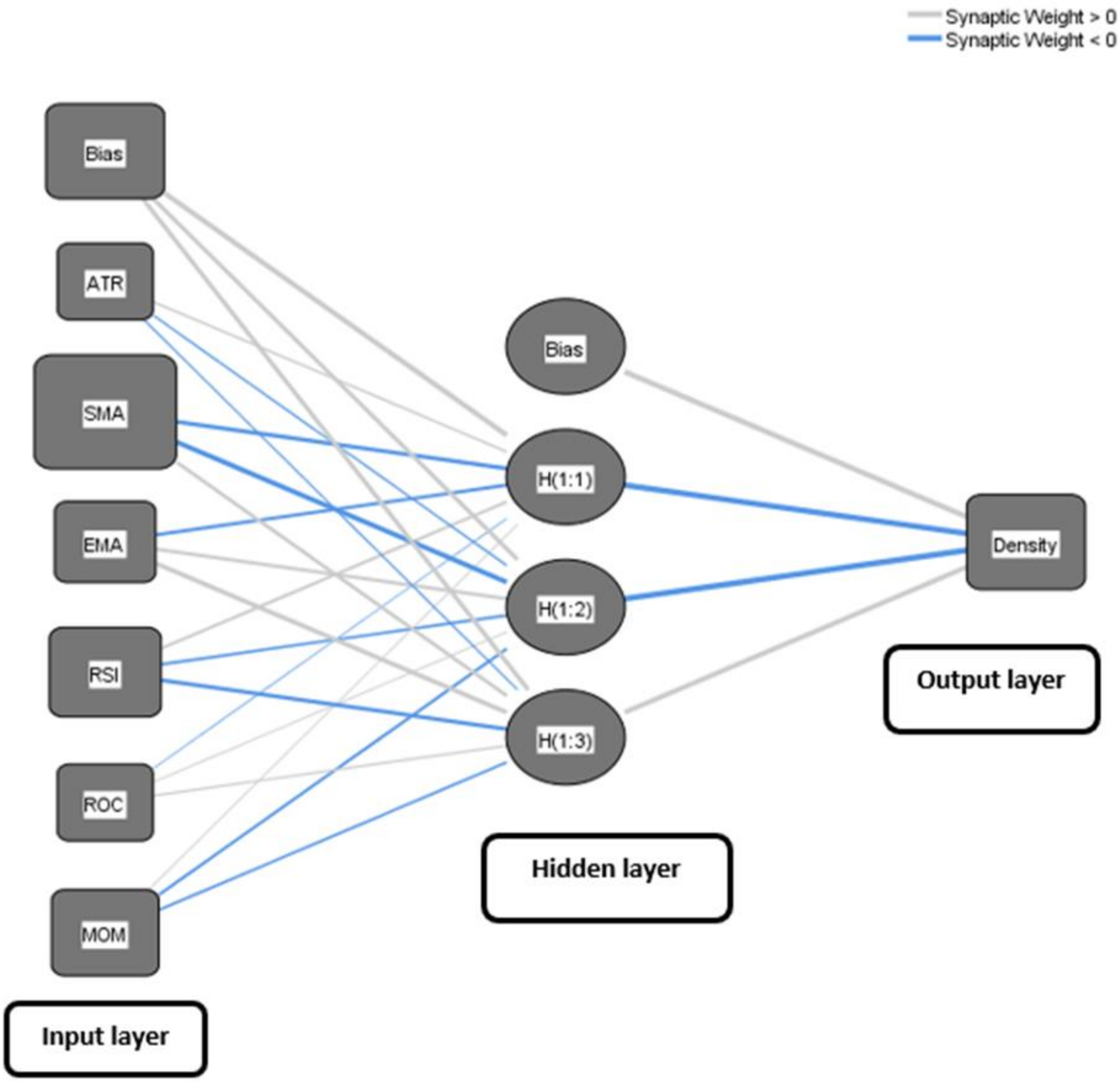


Figure 3.3: The structure of the proposed FFNN

And by using the weights (Wi) and threshold (θ_1, θ_2) shown in the tables (A.6, A.7, A.8), the vehicle density can be predicted through the given equations that were derived below for different traffic conditions:

1- Equation 3.17 provided allows for the prediction of vehicle density on normal working days.

$$\boldsymbol{Scale\ Density = \ Linear}[(\boldsymbol{H1 - i * W19}) + (\boldsymbol{H2 - i * W20}) + (\boldsymbol{H3 - i * W21}) + \boldsymbol{Bais\ out}] \quad \textbf{(3.17)}$$

And by using the weights and the threshold bias (θ_2) shown in the table (A.6), Equation 3.18 is as follows:

$$\boldsymbol{Scale\ Density} = \ \boldsymbol{Linear}[(\boldsymbol{H1} * -\boldsymbol{1.1}) + (\boldsymbol{H2} * -\boldsymbol{1.3}) + (\boldsymbol{H3} * \boldsymbol{0.6}) + \boldsymbol{1.1}] \qquad \textit{(3.18)}$$

As for the variables (H1, H2, H3), they can be found through Equations 319, 3.20 and 3.21:

$$\boldsymbol{H1} = \boldsymbol{Tanh}[(\boldsymbol{ATR} * \boldsymbol{N1}) + (\boldsymbol{SMA} * \boldsymbol{N2}) + (\boldsymbol{EMA} * \boldsymbol{N3}) + (\boldsymbol{RSI} * \boldsymbol{N4}) + (\boldsymbol{ROC} * \boldsymbol{N5}) + (\boldsymbol{MOM} * \boldsymbol{N6}) + \boldsymbol{Bias\ N1}] \qquad \textit{(3.19)}$$

$$\boldsymbol{H2} = \ \boldsymbol{Tanh}[(\boldsymbol{ATR} * \boldsymbol{N7}) + (\boldsymbol{SMA} * \boldsymbol{N8}) + (\boldsymbol{EMA} * \boldsymbol{N9}) + (\boldsymbol{RSI} * \boldsymbol{N10}) +) + (\boldsymbol{ROC} * \boldsymbol{N11}) + (\boldsymbol{MOM} * \boldsymbol{N12}) + \boldsymbol{Bias\ N2}] \qquad \textit{(3.20)}$$

$$\boldsymbol{H3} = \ \boldsymbol{Tanh}[(\boldsymbol{ATR} * \boldsymbol{N13}) + (\boldsymbol{SMA} * \boldsymbol{N14}) + (\boldsymbol{EMA} * \boldsymbol{N15}) + (\boldsymbol{RSI} * \boldsymbol{N16}) +) + (\boldsymbol{ROC} * \boldsymbol{N17}) + (\boldsymbol{MOM} * \boldsymbol{N182}) + \boldsymbol{Bias\ N3}] \qquad \textit{(3.21)}$$

And by using the weights and the threshold (θ_1) shown in the table (A.6), the hidden layer Equations 3.22, 3.23 and 3.24 are as follows:

$$\boldsymbol{H1} = \boldsymbol{Tanh}[(\boldsymbol{ATR} * \boldsymbol{0.0032}) + (\boldsymbol{SMA} * -\boldsymbol{0.01}) + (\boldsymbol{EMA} * -\boldsymbol{.007}) + (\boldsymbol{RSI} * -\boldsymbol{0.00003}) + (\boldsymbol{ROC} * -\boldsymbol{0.0001}) + (\boldsymbol{MOM} * \boldsymbol{0009}) + \boldsymbol{1.4445}] \qquad \textit{(3.22)}$$

$$\boldsymbol{H2} = \ \boldsymbol{Tanh}[(\boldsymbol{ATR} * -\boldsymbol{0.002}) + (\boldsymbol{SMA} * -\boldsymbol{0.019}) + (\boldsymbol{EMA} * \boldsymbol{0.0085}) + (\boldsymbol{RSI} * -\boldsymbol{0.00002}) + (\boldsymbol{ROC} * \boldsymbol{0.0002}) + (\boldsymbol{MOM} * -\boldsymbol{0.006}) + \boldsymbol{0.791}] \qquad \textit{(3.23)}$$

$$\boldsymbol{H3} = \ \boldsymbol{Tanh}[(\boldsymbol{ATR} * -\boldsymbol{0.002}) + (\boldsymbol{SMA} * \boldsymbol{0.0089}) + (\boldsymbol{EMA} * \boldsymbol{0.0189}) + (\boldsymbol{RSI} * -\boldsymbol{0.00004}) +) + (\boldsymbol{ROC} * \boldsymbol{0.0004}) + (\boldsymbol{MOM} * -\boldsymbol{0.005}) - \boldsymbol{0.356}] \qquad \textit{(3.24)}$$

It is important to acknowledge that during the training period, all input features (ATR, SMA, EMA, RSI, ROC and MOM) have been transformed into standardized values within the range of [-1, +1]. Consequently, the determination of the measured density using equation (3.18) can be performed. The values will range from negative one to positive one. To acquire accurate density values, it is necessary to modify the weights using equation (3.25) provided below, in order to restore the values to their true state.

$$\boldsymbol{Scaled\ value} = {}^{(\boldsymbol{X} - \boldsymbol{X_{min}})}\Big/_{(\boldsymbol{X_{max}} - \boldsymbol{X_{min}})} \qquad \textit{(3.25)}$$

Finally, the final form of the density equation is as follows:

$$\boldsymbol{Unscale\ Density} = [\boldsymbol{Scale\ Density} * \boldsymbol{D}] + \boldsymbol{A} \qquad \textit{(3.26)}$$

Where:

$$\boldsymbol{D} = (\boldsymbol{True\ output_{max}} - \boldsymbol{True\ output_{min}})/\boldsymbol{2} \qquad \textit{(3.27)}$$

$$\boldsymbol{A} = \boldsymbol{True\ output_{max}} - \boldsymbol{D} \qquad \textit{(3.28)}$$

2- Regarding the vehicular density during holidays, the same methodology employed for regular working days, as described above, can be utilized. However, it is necessary to consider the weights and thresholds specified in Table (A.7). Thus, Equations for the predictive model can be derived in the following manner:

$$\boldsymbol{Scale\ Density} = \boldsymbol{Linear}[(\boldsymbol{H1} * \boldsymbol{W19}) + (\boldsymbol{H2} * \boldsymbol{W20}) + (\boldsymbol{H3} * \boldsymbol{W21}) + \boldsymbol{Bais\ out}] \ \textit{(3.29)}$$

And by using the weights and the threshold (θ_2) shown in the table (A.7), Equation 3.29 is as follows:

$$\boldsymbol{Scale\ Density} = \boldsymbol{Linear}[(\boldsymbol{H1} * -\boldsymbol{1.1}) + (\boldsymbol{H2} * \boldsymbol{0.3}) + (\boldsymbol{H3} * -\boldsymbol{1.3}) + \boldsymbol{0.6}] \qquad \textit{(3.30)}$$

To find the values of the three neurons (H1, H2, H3) in the hidden layer, the same Equations 3.19, 3.20 and 3.21, above can be used, but using the weights and thresholds in Table (A.7).

$$\boldsymbol{H1} = \boldsymbol{Tanh}[(\boldsymbol{ATR} * \boldsymbol{0.003}) + (\boldsymbol{SMA} * -\boldsymbol{0.018}) + (\boldsymbol{EMA} * -\boldsymbol{0.003}) + (\boldsymbol{RSI} * \\ -\boldsymbol{0.00004}) + (\boldsymbol{ROC} * -\boldsymbol{0.0002}) + (\boldsymbol{MOM} * \boldsymbol{0.0018}) + \boldsymbol{0.3947}] \qquad \textit{(3.31)}$$

$$\boldsymbol{H2} = \boldsymbol{Tanh}[(\boldsymbol{ATR} * \boldsymbol{0.0101}) + (\boldsymbol{SMA} * -\boldsymbol{0.004}) + (\boldsymbol{EMA} * \boldsymbol{0.0166}) + (\boldsymbol{RSI} * \\ -\boldsymbol{0.00005}) +) + (\boldsymbol{ROC} * -\boldsymbol{0.001}) + (\boldsymbol{MOM} * \boldsymbol{0.014}) - \boldsymbol{0.409}] \qquad \textit{(3.32)}$$

$$\boldsymbol{H3} = \boldsymbol{Tanh}[(\boldsymbol{ATR} * \boldsymbol{0.0006}) + (\boldsymbol{SMA} * -\boldsymbol{0.023}) + (\boldsymbol{EMA} * \boldsymbol{0.0047}) + (\boldsymbol{RSI} * \\ \boldsymbol{0.00003}) + (\boldsymbol{ROC} * -\boldsymbol{0.0002}) + (\boldsymbol{MOM} * -\boldsymbol{0.004}) + \boldsymbol{1.2678}] \qquad \textit{(3.33)}$$

Finally, the predicted density can be found by returning its scale values to the real values using Equation 3.34.

$$\boldsymbol{Unscale\ Density} = [\boldsymbol{Scale\ Density} * (\mathbf{27.5})] + \mathbf{28.2} \qquad \textbf{\textit{(3.34)}}$$

Where:

$\boldsymbol{D} = (\boldsymbol{True\ output}_{\boldsymbol{max}} - \boldsymbol{True\ output}_{\boldsymbol{min}})/\mathbf{2} = \mathbf{27.5}$

$\boldsymbol{A} = \boldsymbol{True\ output}_{\boldsymbol{max}} - \boldsymbol{D} = \mathbf{28.2}$

3- Finally, in the predictive model for measuring vehicle density on *weekend days*, the same equations derived above can be used in the two previous models, but the weights and thresholds in Table (A.8) are taken to produce Equation 3.35.

$$\boldsymbol{Scale\ Density} = \boldsymbol{Linear}[(\boldsymbol{H}\mathbf{1} * \mathbf{1.4}) + (\boldsymbol{H}\mathbf{2} * \mathbf{1.0}) + (\boldsymbol{H}\mathbf{3} * -\mathbf{0.1}) + \mathbf{0.7}] \qquad \textbf{\textit{(3.35)}}$$

The neural nodes in the hidden layer (H1, H2, H3) can be derived through the three Equations 3.36, 3.37 and 3.38 and based on the information (weights and thresholds) obtained through experiments conducted on the models that were explained in the previous steps.

$$\boldsymbol{H}\mathbf{1} = \boldsymbol{Tanh}[(\boldsymbol{ATR} * -\mathbf{0.0002}) + (\boldsymbol{SMA} * \mathbf{0.0162}) + (\boldsymbol{EMA} * -\mathbf{0.001}) + (\boldsymbol{RSI} * \mathbf{0.000007}) + (\boldsymbol{ROC} * -\mathbf{0.0004}) + (\boldsymbol{MOM} * \mathbf{0.0021}) - \mathbf{1.027}] \qquad \textbf{\textit{(3.36)}}$$

$$\boldsymbol{H}\mathbf{2} = \boldsymbol{Tanh}[(\boldsymbol{ATR} * \mathbf{0.0019}) + (\boldsymbol{SMA} * \mathbf{0.0219}) + (\boldsymbol{EMA} * -\mathbf{0.003}) + (\boldsymbol{RSI} * \mathbf{0.000006}) + (\boldsymbol{ROC} * \mathbf{0.00009}) + (\boldsymbol{MOM} * \mathbf{0.0031}) - \mathbf{0.571}] \qquad \textbf{\textit{(3.37)}}$$

$$\boldsymbol{H}\mathbf{3} = \boldsymbol{Tanh}[(\boldsymbol{ATR} * \mathbf{0.0103}) + (\boldsymbol{SMA} * \mathbf{0.0007}) + (\boldsymbol{EMA} * -\mathbf{0.02}) + (\boldsymbol{RSI} * -\mathbf{0.000002})+) + (\boldsymbol{ROC} * -\mathbf{0.0007}) + (\boldsymbol{MOM} * -\mathbf{0.012}) + \mathbf{0.6861}] \qquad \textbf{\textit{(3.38)}}$$

Finally, the final Equation 3.39 of the vehicle density equation after un-scaling the data is as follows:

$$\boldsymbol{Unscale\ Density} = [\boldsymbol{Scale\ Density} * \mathbf{30.7}] + \mathbf{31.5} \qquad \textbf{\textit{(3.39)}}$$

Where:

$\boldsymbol{D} = (\boldsymbol{True\ output}_{\boldsymbol{max}} - \boldsymbol{True\ output}_{\boldsymbol{min}})/\mathbf{2} = \textbf{\textit{30.7}}$

$\boldsymbol{A} = \boldsymbol{True\ output}_{\boldsymbol{max}} - \boldsymbol{D} = \textbf{\textit{31.5}}$

By practical application of the final density equations (3.26, 3.34, 3.39) we can predict the vehicle density for highways for any day types based on the historical data provided by the MIDAS system.

B- Multiple Linear Regression (MLR)

In this section, the multiple linear regression model described in section 2.10.3 was employed. It is assumed that there exist p explanatory variables (ATRi1, SMAi2, EMAi3, RSIi4, ROCi5 and MOMi6) that are to be examined in relation to a dependent variable $\boldsymbol{y_i}$ (Density). The data matrix is hypothesized to be obtained from a randomly selected sample of *n* observations (ATRi1, SMAi2, EMAi3, RSIi4, ROCi5 and MOMi6, $Density_i$), i= 1,2, ..., n., where *i* ranges from *1* to *n as shown in Equation 3.40*. The random variables are postulated to conform to the linear model as described by Equation (2.1):

$$\boldsymbol{y_i = \beta_0 + \beta_1 ATR_{i1} + \beta_2 SMA_{i2} + \beta_3 EMA_{i3} + \beta_4 RSI_{i4} + \beta_5 ROC_{i5} + \beta_6 MOM_{i6} + u_i,\quad i=1,2,\ \dots,n} \tag{3.40}$$

- The variables u_i, where *i* ranges from *1* to *n*, represents the values of an unobserved error term *U*. These variables are assumed to be mutually independent and identically distributed. It is also assumed that the expected value of each u_i is *0*, and the variance of each u_i is $E[u_i] = 0; V[u_i] = \sigma_u^2$.

- Mutually Independent: The values of u_i are not related to each other.
- Identically Distributed: They follow the same probability distribution.
- Zero Mean: The expected value (average) of each u_i is assumed to be 0.
- Constant Variance: The variance of each u_i is assumed to be constant and equal to σ_u^2.

- The distribution of the error term *U* is independent of the joint distribution of $XI, X\ 2, \dots, Xp$ and hence the regression function $E[Y|ATR_1, SMA_2, \dots, MOM_6] = \beta_0 + \beta_1 ATR_1 + \beta_2 SMA_2 + \cdots + \beta_6 MOM_6$; and $V[Y|ATR_1, SMA_2, \dots, MOM_6] = \sigma^2_{density.ATR_1,SMA_2,\dots,MOM_6} = \sigma_u^2$.
- The parameters $\beta_0, \beta_1, \beta_2, \dots, \beta_p$ are constant and unknown.

The variables to be utilized in our model are enumerated in section (3.2.3). The provided data will be utilized to estimate a linear relationship between the employed traffic density and the remaining six variables, namely ATR, SMA, EMA, RSI, ROC and MOM. Three distinct modeling periods were selected to ensure a precise estimation: normal working days, holidays, and weekend days. The normal working days for (M25 highway) are determined and represented by the linear Equation 3.41:

$$\boldsymbol{Density_i = -0.1 + (-0.008 * ATR_{i1}) + (-0.7 * SMA_{i2}) + (1.7 * EMA_{i3}) + (0.003 * RSI_{i4}) + (-0.003 * ROC_{i5)} + (0.2 * MOM_{i6}) + U} \tag{3.41}$$

The vehicle density on holidays is predicted by proposed the linear Equation 3.42:

$$\boldsymbol{Density_i = -0.2 + (-0.007 * ATR_{i1}) + (-0.7 * SMA_{i2)} + (1.7 * EMA_{i3}) + (0.004 * RSI_{i4}) + (-0.004 * ROC_{i5)} + (0.19 * MOM_{i6}) + U} \tag{3.42}$$

The vehicle density on weekend days is predicted according to the proposed the linear mathematical equation 3.43:

$$\boldsymbol{Density_i = -0.1 + (-0.001 * ATR_{i1}) + (-0.7 * SMA_{i2)} + (1.7 * EMA_{i3}) + (0.002 * RSI_{i4}) + (-0.003 * ROC_{i5)} + (0.2 * MOM_{i6}) + U} \tag{3.43}$$

The multiple linear regression model for the M60 highway is implemented through Equations 3.44, 3.45 and 3.46 for different conditions:

The normal working days for (M60 highway) are determined and represented by the linear Equation 3.44:

$$\boldsymbol{Density_i = -0.08 + (0.001 * ATR_{i1}) + (-0.6 * SMA_{i2)} + (1.6 * EMA_{i3}) + (0.001 * RSI_{i4}) + (-0.0008 * ROC_{i5)} + (0.2 * MOM_{i6}) + (0.0002 * CCI_{i7}) + U} \tag{3.44}$$

The vehicle density on holidays is predicted by proposed the linear Equation 3.45:

$$\boldsymbol{Density_i = -0.09 + (0.002 * ATR_{i1}) + (-0.7 * SMA_{i2)} + (1.7 * EMA_{i3}) + (0.001 * RSI_{i4}) + (-0.000003 * ROC_{i5)} + (0.18 * MOM_{i6}) + (0.0002 * CCI_{i7}) + U} \tag{3.45}$$

Finally, the vehicle density on weekend days is predicted according to the proposed linear mathematical Equation 3.46:

$$\boldsymbol{Density_i = -0.04 + (-0.006 * ATR_{i1}) + (-0.7 * SMA_{i2)} + (1.7 * EMA_{i3}) + (0.0008 * RSI_{i4}) + (-0.0001 * ROC_{i5)} + (0.1 * MOM_{i6}) + (0.0001 * CCI_{i7}) + U} \tag{3.46}$$

Each equation represents a specific situation (normal working days, holidays, or weekend days) and provides coefficients for the predictor variables. In Equation 3.41 (Normal Working Days) Coefficients ($\beta_1, \beta_2, \dots, \beta_6$) indicate the impact of each technical indicator on density. As example a one-unit increase in ATRi1 is associated with a decrease of 0.008 in vehicle density, while a one-unit increase in EMAi3 is associated with an increase of 1.7 in vehicle density. While, β_0 value represents the intercept or constant term in the multiple linear regression equation. It is the estimated or predicted value of the dependent variable (Density) when all the predictor variables (ATRi1, SMAi2, EMAi3, RSIi4, ROCi5 and MOMi6) are set to zero.

The weights (coefficients) of the independent variables (predicted variables) were calculated using the ordinary least squares (OLS). Minimization of Residuals: The OLS method seeks to find the values of $\beta_0, \beta_1, \dots, \beta_6$ that minimize the sum of squared residuals (SSE or RSS). The residuals are the differences between the actual observed values of $\boldsymbol{y_i}\,(\boldsymbol{Density})$ and the values predicted by the model. Mathematically, the goal is to minimize as shown in Equation 3.47.

$$SSE = \sum \left(actual_{\,y} - predicted_y\right)^2 \tag{3.47}$$

It is important to acknowledge that during the training period, all inputs, which are technical indicators, were transformed into standardized values within the range of -1 to +1. Hence, by employing equations (3.41, 3.42, 3.43, 3.44, 3.45 and 3.46) to determine the measured density, the resulting values will fall within the range of [-1, +1]. To accurately determine the true density values, it is imperative to employ the equations (3.26, 3.34, 3.39) previously discussed, which allow for the correction of the values to their actual magnitude.

3.2.5 The Classification Stage

Historically, the utilization of LOS has been widely adopted as a prominent indicator for assessing traffic congestion in VANET. The concept of LOS, as it is defined and utilized in the HCM, encompasses a spectrum of operational circumstances. LOS of a facility is established through an analysis of traffic flow characteristics, including 1) vehicle density, 2) average speed, 3) traffic flow, and

4) intersection delay. These factors vary depending on the type of facility, as outlined in section 2.9. In order to achieve the dissertation objectives and estimate the hourly LOS (as a label) using traffic density data as mention in Table 2.2, machine learning classification methods were employed for the selected highways. The RF classification method was chosen based on several factors:

- RF mitigate the risk of overfitting.
- RF provides a measure of feature importance.
- Non-linearity Handling
- Handling Imbalanced Data.
- Ensemble nature helps in reducing variance and improving stability in predictions.
- RF provides an OOB error estimate during training, which serves as a reliable indicator of model performance.
- RF is relatively easy to implement and tune.

• Random Forest classifier

Decision trees and random forests are widely used machine learning techniques for tackling a variety of classification problems. Decision trees operate by recursively partitioning the feature space using a tree structure. In this process, each child node is divided further until pure nodes are reached, meaning nodes containing samples of a single class. The division of nodes is guided by a criterion that aims to maximize the purity of child nodes relative to their parent nodes. Once pure nodes are achieved, they become leaf nodes, and no further splitting occurs. When classifying a test sample using a decision tree, the tree is traversed to a leaf node, and the test sample is assigned the class label associated with the training samples of that leaf node.

Random forests, on the other hand, employ an ensemble of multiple decision trees to mitigate the risk of overfitting. In a random forest, each tree is constructed using a random subset of the feature space. Typically, if a dataset has M features, m (where m < M) features are randomly selected for growing each tree. Random forests are favored over individual decision trees because they incorporate a significant number of voting-based decisions. They implement a bootstrap aggregation (bagging) technique using a large number of decorrelated decision trees to classify a test sample. This approach is well-suited for handling stock data classification, as it systematically explores the feature space to make robust class predictions. To evaluate the quality of a split at each node, Gini impurity is employed as a measure. The Gini impurity at node N is calculated based on the proportion of the population with each class label, denoted as P_i as shown in Equation 3.48.

$$G(N) = 1 - (P_1)^2 - (P_{-1})^2 \qquad (3.48)$$

Where P_i is the proportion of the population with class label i. The ideal splitting decision at a node is the one that maximally reduces impurity or, equivalently, yields the highest information gain and impurity reduction. It's worth noting that random forests are non-metric classifiers, distinguishing them from gradient-based and Bayesian methods. This characteristic eliminates the need to fine-tune learning parameters or make prior distribution assumptions, contributing to their popularity in various classification tasks. RF classifier training and classification algorithm routines are 3.9 and 3.10 below respectively.

Algorithm 3.9: Random Forest Classifier Training

```
    Input:

    Training samples (X) with features (ATR, SMA, EMA, RSI, ROC, MOM) and LOS
    indicators

    Number of trees (n_trees)

    Number of features to consider at each split (k)

    Maximum number of leaf splits (max_leaf_splits)

    Output:

    Trained Random Forest Model
    ______________________________________________________________________

Begin

    1- Initialize an empty list to store the decision trees: ‘forest = []’
    2- For each tree in the range of ‘n_trees’:
       a. Randomly select k features from the set (ATR, SMA, EMA, RSI, ROC, MOM).
       b. Create the root node of the decision tree: ‘root = create_node(X,
       selected_features, criterion="Gini")’
           • ‘create_node’ function should find the best split point based on the Gini
             criterion for the selected features.
      c. Initialize a queue for node expansion: ‘queue = [root]’
      d. While the queue is not empty:
         i.      Pop the front node from the queue: ‘current_node = queue.pop(0)’
         ii.     If the node meets the stopping criteria (e.g., maximum depth or impurity
                 threshold), mark it as a leaf node.
         iii.    Otherwise, find the best split for the current node based on the Gini
                 criterion using the selected features.
         iv.     Split the node into daughter nodes based on the best split.
         v.      Add daughter nodes to the queue for further expansion:
                 ‘queue.extend(daughter_nodes)’
        e. Add the trained decision tree to the forest: ‘forest.append(root)’ Return the
           trained Random Forest Model ‘forest’.

End
```

Algorithm 3.10: Random Forest Classifier Classification

```
Input:
        Test samples (Y) with features (ATR, SMA, EMA, RSI, ROC, MOM)
        Trained Random Forest Model ('forest')
Output:
        Predicted LOS indicators for the test samples
Begin
        1- Initialize an empty list to store the predictions for each tree: 'tree_predictions = []'
        2- For each tree in the 'forest':
            a. Initialize an empty list to store predictions for the current tree:
               'current_tree_predictions = []'
            b. For each sample in 'Y':
                i.  Traverse the tree to a leaf node using the features (ATR, SMA, EMA, RSI,
                    ROC, MOM) of the current sample.
                ii. Assign the majority class (LOS indicator) of the training samples in the leaf
                    node to the current sample:
                    'current_tree_predictions.append(majority_class_of_leaf_node)'
            c. Add 'current_tree_predictions to tree_predictions'.
        3- Calculate the final predictions by considering the most voted predicted outcome
           or the average of closely related outcomes from all trees in the forest:
            a. For each sample in Y, calculate the mode (most frequent class) of predictions
               from 'tree_predictions' or consider the average of closely related outcomes.
        4- Return the final predicted LOS indicators for the test samples.
End
```

3.2.6 The Evaluation Model Stage

In this stage, the proposed system is evaluated based on testing dataset. RMSE, MAPE and AA% measures are used to evaluate FFNN and MLR models. On the other hand, Accuracy, Recall, F1-Measure, Precision, Cohen Kappa, OOB, Receiver Operator Characteristic and AUC are used to evaluate the RF model.

Chapter four
Results and Discussion

4.1 Overview

The effectiveness of the proposed system illustrated in the previous chapter has been tested with different parameters values, and the results are presented and discussed in this chapter. A real global dataset has been applied as an employment case study to determine the behavior of the proposed system. Furthermore, the experimental results stages of the proposed system are described and shown in this chapter.

Before starting to analyze the results of the proposed system, it is important to present the general characteristics of the proposed system:

1- It deals with real, numeric, complex, and big dataset of MIDAS.
2- It deals with decisions as discrete and continuous (classification, regression).
3- It depends on machine learning principals.
4- It generates TIs to increase the accuracy of the prediction.

4.2 Evaluating the Regression Models

This section provides additional evidence of the efficacy of the proposed short-term traffic prediction frameworks through the utilization of authentic traffic data. The examination of the model's robustness involves the application of the proposed models to traffic conditions on normal work days, holidays, and weekend days. Reliable evaluation of data mining techniques is performed on test data that had not been seen before during the training phase. The metrics employed for the quantitative assessment of accuracy encompass RMSE, MAPE, and AA%.

4.2.1 The Results of Normal working days

This subsection exclusively examines traffic density data obtained from the MIDAS system under normal work conditions. The dataset contains traffic density data for the entire month of February 2022, which has been partitioned into two distinct sets: training data and test data. Given that the primary emphasis is on weekdays, the data pertaining to weekends and holidays is excluded. The accuracy of traffic density prediction is evaluated by comparing the RMSE, MAPE, AA% metrics, as presented in Tables 4.1 and 4.2.

Table 4.1: The evaluation results of different models under normal work days conditions (M25)

	RMSE	**MAPE (%)**	**AA%**
MLR	0.2	0.02	99.98
FFNN	0.2	1.8	98.2
RFR	0.8	0.02	99.98
Markov chain	2.51	0.15	99.95

Table 4.2: The evaluation results of different models under normal work days conditions (M60)

	RMSE	**MAPE (%)**	**AA%**
MLR	0.09	0.02	99.98
FFNN	0.96	4.5	95.5
RFR	0.4	0.02	99.98
Markov chain	3.32	0.257	99.75

The section provides an analysis of the results of the effectiveness of different machine learning methodologies in predicting traffic density. The excellent accuracy of the machine learning techniques used in prediction can be attributed to the linear nature of the data and using technical indicators. The research findings indicate that the utilization of data normalization techniques can enhance the accuracy of predictions, particularly in the case of the FFNN technique.

The findings presented in Tables 4.1 and 4.2 indicate that the MLR-based method exhibits the highest level of prediction accuracy. Nevertheless, owing to the robust performance exhibited by FFNN in forecasting traffic congestion, it is recommended as a fundamental proposed framework for predicting traffic density, as it demonstrates the lowest margin of error and a notable level of precision. The scatter plots depicted in Figures 4.1 and 4.2 illustrate the comparison between the predicted and observed traffic density data. Additionally, these figures display the error auto-correlation plot of the predictions and the histogram representing the distribution of errors within the proposed FFNN prediction framework. The scatter plot reveals that the prediction model exhibited a slight underestimation of the observed traffic density in cases of high traffic density under highly congested conditions.

The error distribution histogram reveals that the prediction errors follow an almost normal distribution, which indicates that the FFNN model shows satisfactory performance in predicting traffic density. The errors are distributed around zero, which means that the model generates accurate predictions and any differences are due to random fluctuation. An autocorrelation plot of prediction errors indicates a correlation between errors observed at different time points, but it decays with time and the length of the training period.

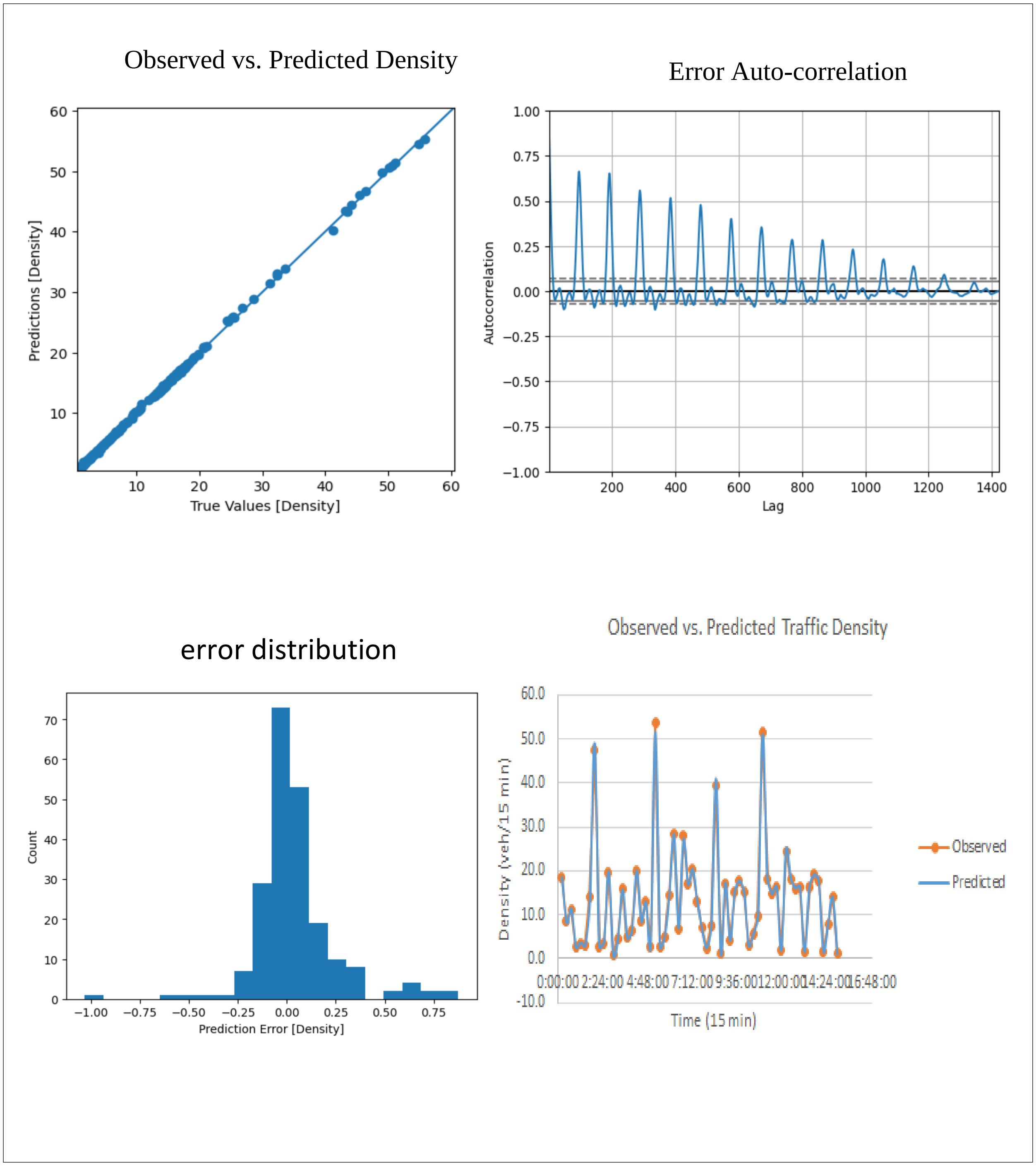


Figure 4.1: Traffic density prediction performance using FFNN framework on the M25 highway under normal work traffic conditions.

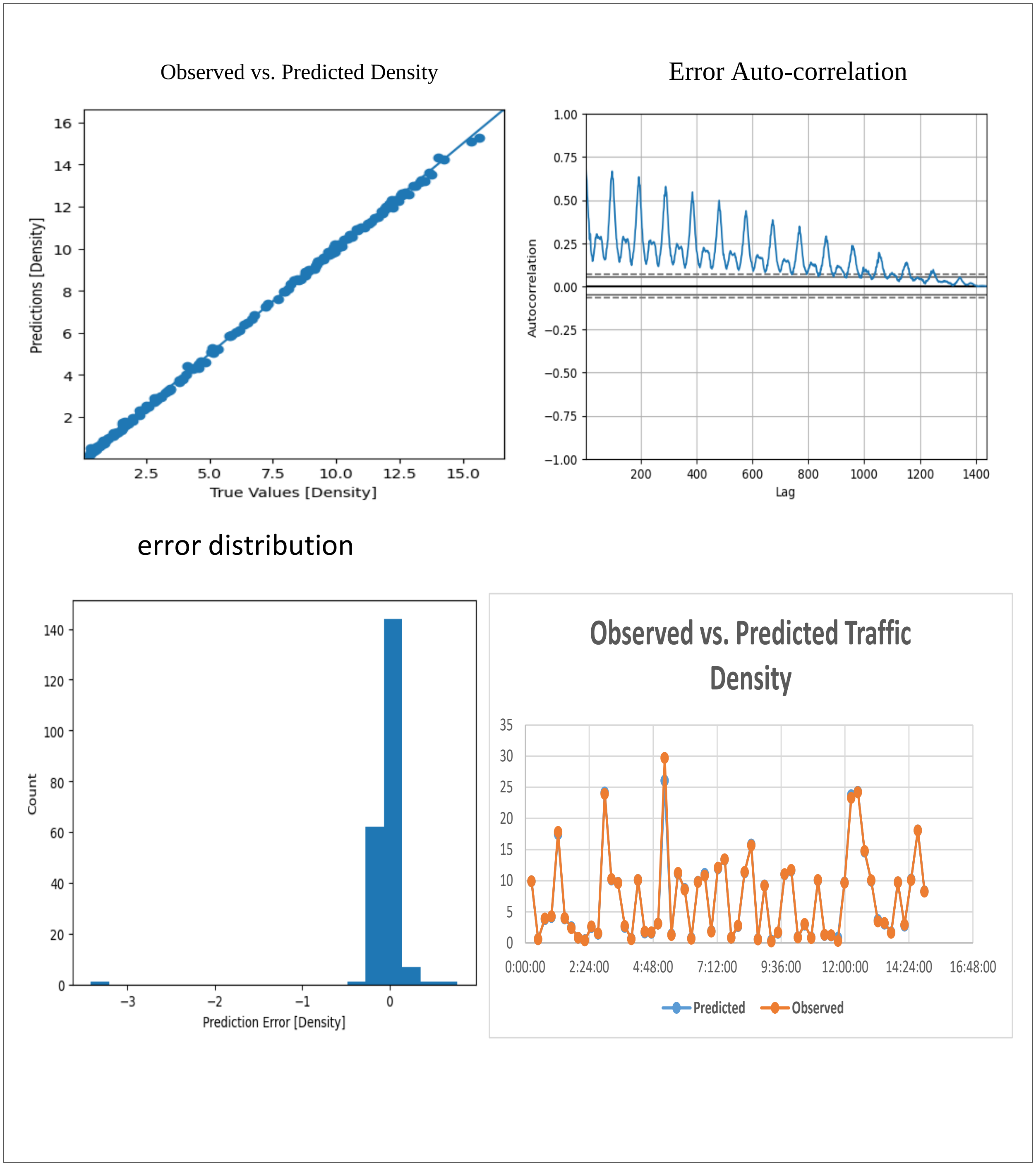


Figure 4.2: Traffic density prediction performance using FFNN framework on the M60 road under normal work traffic conditions

4.2.2 The Results of Holiday days

The traffic flow data for this subsection has been collected from selected highways in the vicinity of London and Manchester. The 15-minute traffic data that has been compiled is sourced from the MIDAS system. For instance, in the dataset pertaining to M25 highways, a total of 302 records were chosen from a five-day period encompassing holidays in February 2022. The test data were chosen from the same month and consisted of 156 records. The dataset for the month under investigation was subjected to filtering, specifically isolating the data pertaining to regular weekdays and weekends, while exclusively considering holidays. This approach was adopted due to the observed variations in vehicle density, which necessitated a separate analysis. The training and testing datasets consist of traffic data that exclusively represents normal traffic conditions, excluding any incidents or other abnormal events.

The accuracy of three traffic prediction frameworks for forecasting traffic conditions on a selected corridor during normal traffic conditions on holidays is presented in Tables 4.3 and 4.4. The neural networks model is selected based on the findings presented in the table, as it demonstrates strong predictive capabilities for both linear and nonlinear data. Furthermore, it exhibits a notable prediction accuracy of 92.27% in table 4.2. Consistent with the findings presented in Section 4.2.1, the utilization of the data normalizing leads to enhanced prediction accuracy under typical traffic conditions.

Table 4.3: The evaluation results of different models under holidays days conditions (M25)

	RMSE	**MAPE (%)**	**AA%**
MLR	0.27	0.03	99.97
FFNN	1.02	5.7	94.3
RFR	1.5	0.04	99.96
Markov chain	1.3	0.27	99.73

Table 4.4: The evaluation results of different models under holiday days conditions (M60)

	RMSE	MAPE (%)	AA%
MLR	0.07	0.02	99.98
FFNN	0.6	11.7	88.3
RFR	0.68	0.07	99.93
Markov chain	1.32	0.107	99.89

Figures 4.3 and 4.4 present a scatter plot depicting the relationship between predicted and observed traffic density data, an error auto-correlation plot illustrating the correlation of predictions, a histogram displaying the distribution of errors, and a sample time-series plot showcasing the relationship between predicted and observed traffic density within the FFNN framework. These plots have the same interpretation as the previous plots in the section 3.2.1.

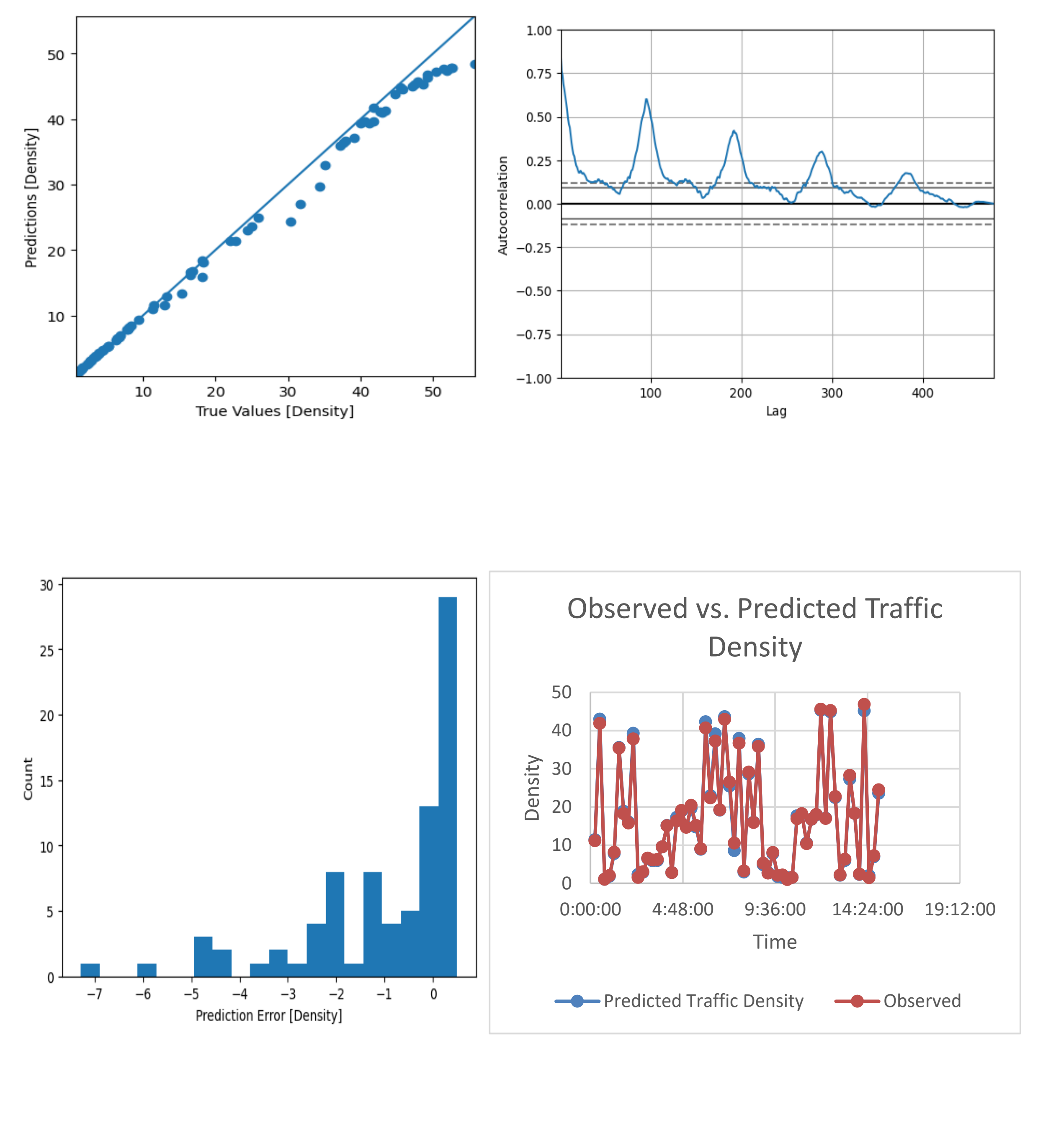


Figure 4.3: Traffic density prediction performance using FFNN framework on the M25 highway in holidays

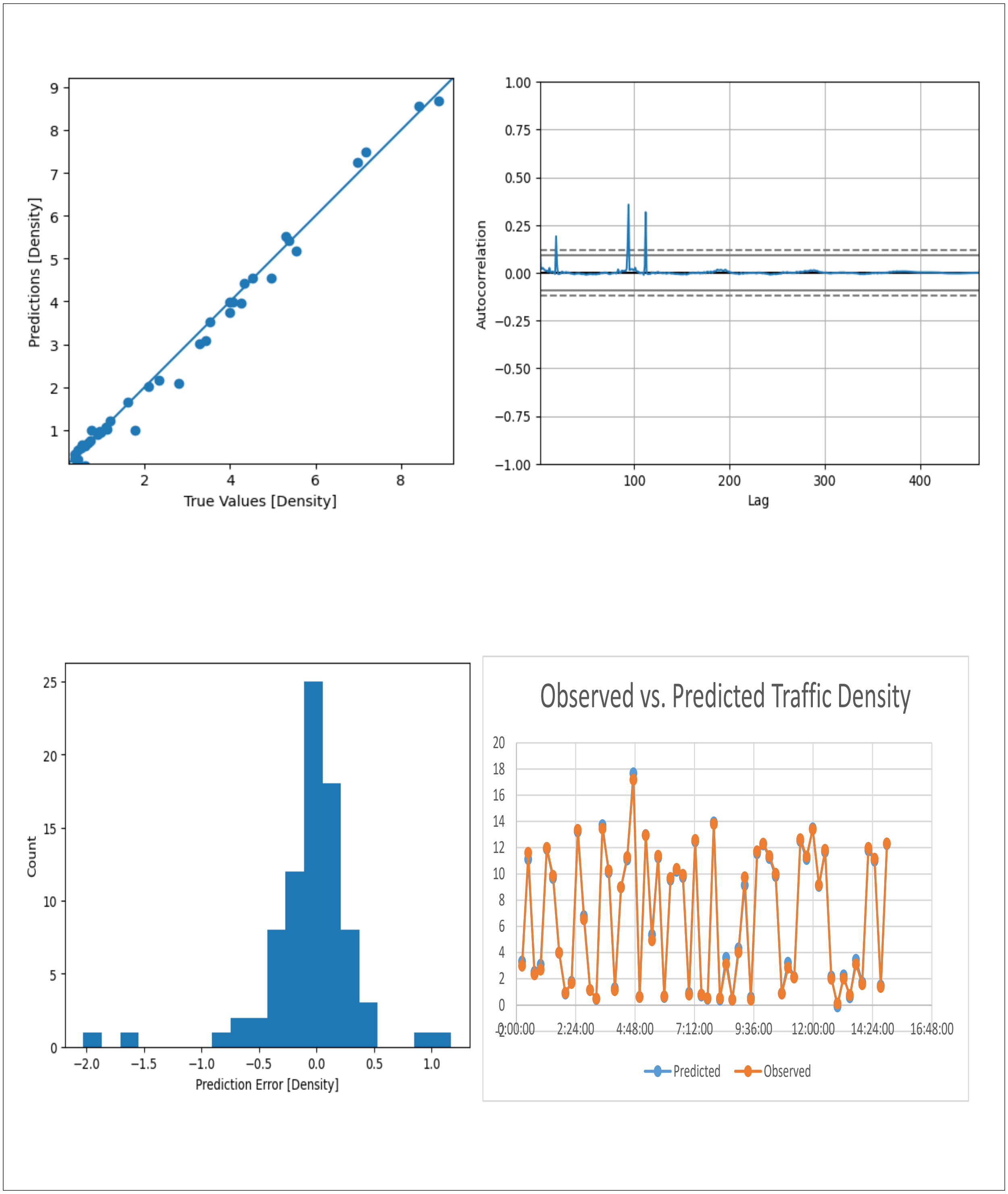


Figure 4.4: Traffic density prediction performance using FFNN framework on the M60 highway in holidays

4.2.3 The Results of Weekend days

The dataset comprises 15-minute traffic flow and occupancy data from selected MIDAS system corridors, subjected to rigorous preprocessing to eliminate missing values. It focuses on normal, accident-free traffic conditions during weekends in February 2022. 72% of the data is allocated for training, and 28% for testing. The study evaluated three traffic prediction models as shown in Tables 4.5 and 4.6, with multiple linear regression outperforming others in terms of MAPE and RMSE due to the data's linear characteristics. While random forest is effective, it may be inefficient for large test datasets. Therefore, a neural network-based model is chosen for its robustness and superior predictive performance, particularly in time-sensitive applications.

Table 4.5: The evaluation results of different models under weekend days conditions (M25)

	RMSE	MAPE (%)	AA%
MLR	0.27	0.01	99.99
FFNN	0.5	3.8	96.2
RFR	1.21	0.03	99.97
Markov chain	8.75	0.4	99.96

Table 4.6: The evaluation results of different models under weekend days conditions (M60)

	RMSE	MAPE (%)	AA%
MLR	0.03	0.03	99.97
FFNN	0.4	4.05	95.95
RFR	0.15	0.06	99.94
Markov chain	2.48	0.24	99.76

Figures 4.5 and 4.6 depict the scatter plot illustrating the relationship between predicted and observed travel time data, as well as the error auto-correlation plot of predictions. Additionally, the histogram of error distribution and the sample time-series plot between predicted and observed traffic density are presented within the FFNN prediction framework.

In the proposed prediction models, the error distribution pertains to the observed pattern of discrepancies between the actual values and the predicted values, as visually depicted in Figures 4.5 and 4.6. The presence of a normal distribution can be inferred from the distribution of errors between the observed and expected values. The errors exhibit a random distribution centered around a mean of zero. This implies that the proposed model is generating precise forecasts, and any discrepancies are attributable to stochastic variability.

To summarize, the performance measures for short-term prediction can be found in the Tables 4.1 to 4.16. It is evident that the majority of machine learning techniques exhibit relatively similar performance outcomes in terms of both MAPE and RMSE. In each case, it is typically observed that MAPE exhibits a lower value compared to its counterpart, RMSE. The reason for this is that MAPE calculates the absolute percentage error rates and then computes their average across the ensemble results space. In contrast, RMSE is a relative error measure that takes into account the relative error deviation between the results ensemble space. It is evident that all models exhibit satisfactory performance in learning the data, which can be attributed to the linear characteristics of the data.

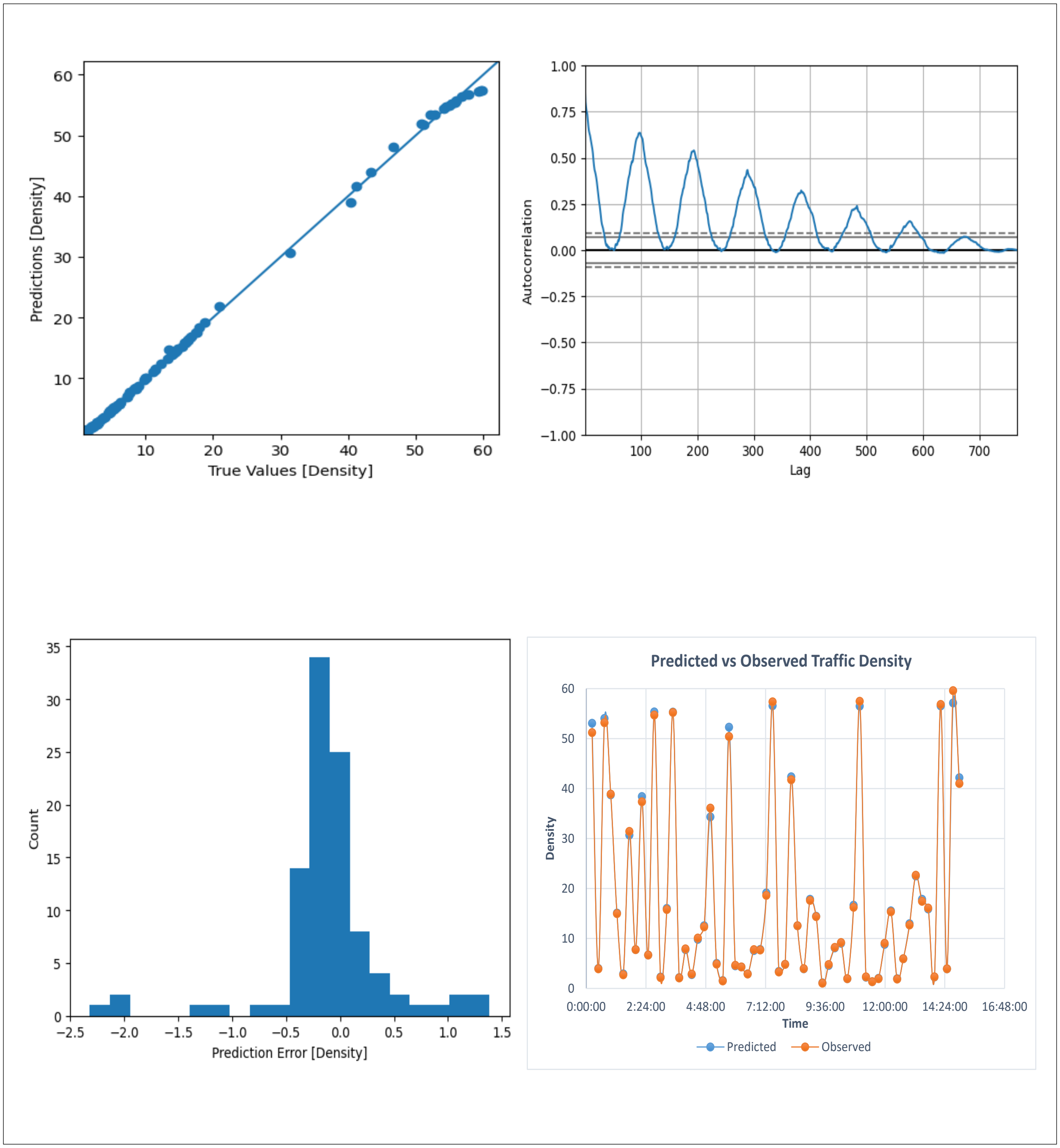


Figure 4.5: Traffic density prediction performance using FFNN framework on the M25 highway in weekend days

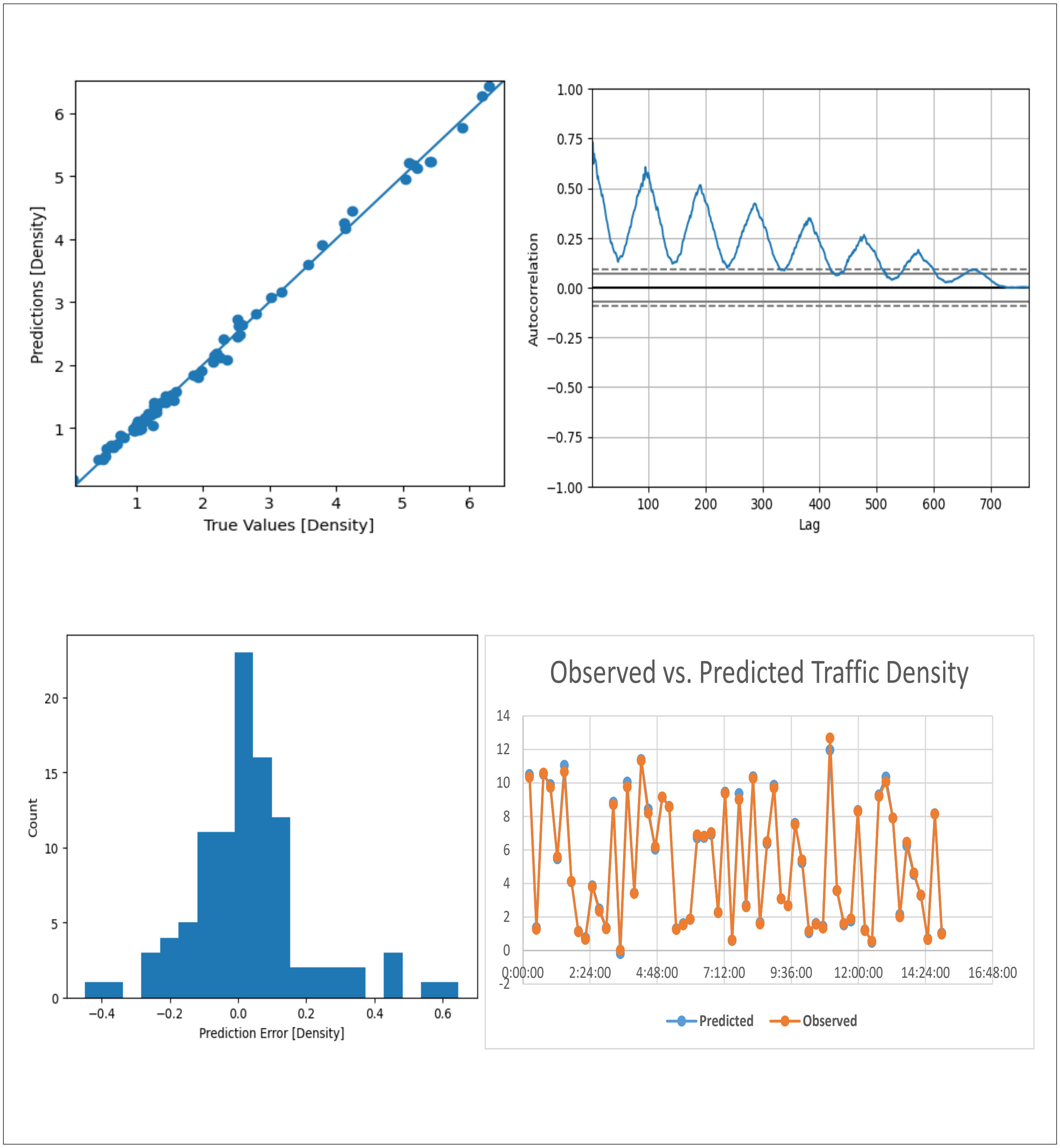


Figure 4.6: Traffic density prediction performance using FFNN framework on the M60 highway in weekend days

4.2.4 *Sensitivity Analysis*

A sensitivity analysis was conducted on the proposed FFNN models in order to investigate the importance of each technical indicator. Sensitivity analysis is a method used to assess the resilience and effectiveness of a model by identifying the input parameters that exert the greatest influence. The analysis consisted of systematically eliminating individual technical indicators from the model inputs and assessing the model's performance using the three-summary metrics.

The technique employed in this dissertation is referred to as parametric bootstrap, wherein the factors are systematically removed and the model is subsequently reevaluated following each replacement. The summary results pertaining to the removal of each technical indicator can be observed in Tables 4.7 and 4.8.

Table 4.7: FFNN models sensitivity analysis for M25 highway.

Week days	Indicators	RMSE	MAPE	AA%
Normal working days	None	0.2	1.8	98.2
	ATR	0.4	2.6	97.4
	SMA	0.78	3.1	96.9
	EMA	0.97	5.4	94.6
	RSI	1.58	6.6	93.4
	ROC	0.36	2.07	97.93
	MOM	0.49	3.5	96.5
Holiday days	None	1.02	5.7	94.3
	ATR	1.2	4.6	95.4
	SMA	1.2	3.5	96.5
	EMA	1.91	11.7	88.3
	RSI	0.9	3.7	96.3
	ROC	0.98	5.05	94.95
	MOM	0.99	6.2	93.8
Weekend days	None	0.5	3.9	96.1
	ATR	1.2	5.2	96.8
	SMA	1.3	14.2	85.8
	EMA	1.4	8	92
	RSI	1.5	11.6	88.4
	ROC	0.54	2.5	97.5
	MOM	0.93	2.7	97.3

Table 4.8: FFNN models sensitivity analysis for M60 highway.

Week days	Indicators	RMSE	MAPE	AA%
Normal working days	None	0.96	4.5	95.5
	ATR	0.96	2.6	97.4
	SMA	0.68	5	95
	EMA	0.6	6.5	93.5
	RSI	0.21	5.5	94.5
	ROC	0.26	2.6	97.4
	MOM	0.54	4.9	95.1
Holiday days	None	0.6	11.7	88.3
	ATR	0.6	10.3	89.7
	SMA	0.4	62.2	37.8
	EMA	0.62	23.9	76.1
	RSI	0.6	9.9	91.1
	ROC	0.7	8.03	91.97
	MOM	0.6	48.8	51.2
Weekend days	None	0.4	4.05	95.95
	ATR	0.15	12.19	87.91
	SMA	0.13	24.2	75.8
	EMA	0.21	5.3	94.7
	RSI	0.17	7.5	92.5
	ROC	1.16	15.01	84.99
	MOM	0.5	55.2	44.8

Based on the findings presented in Tables 4.7 and 4.8, it is evident that SMA and EMA emerged as the most influential technical indicators in the Table 4.7. This is substantiated by the notable decrease in model accuracy observed when the SMA and EMA was excluded. The removal of the ATR parameter had a minimal impact on the model accuracy, with the resulting values being comparable to the original model's accuracy. This observation was specifically noted during holiday days. Also, the deletion of the ROC and MOM indicators had an impact on the prediction ability in the case of the M60 road, as shown in the Table 4.8. Despite an improvement in RMSE, the overall outcome of the analysis does not match the level of excellence achieved when incorporating all the technical indicators. The results of this dissertation indicate that SMA, MOM, ROC, and EMA exhibit a robust predictive capability for forecasting vehicle density.

4.3 Evaluating the Classification Model

The density of vehicles in the dataset is represented as a continuous value. Class labeling is used as a method to more easily illustrate LOS. The classification process is improved using this representation. Thus, before evaluating the classification models, class feature should be created with six values (LOS A, LOS B, ..., LOS F) based on Table 2.2 for all data points in dataset. It is important to acknowledge that the Python programming language, specifically version 3.9.12, was utilized for all data analyses and visualizations presented within this section. It is important to acknowledge that the datasets contained missing values, which accounted for less than 1% of the overall population. Consequently, these missing values were estimated. The evaluation of the robustness of a multiclass classifier involves the utilization of various performance measures. These measures include the accuracy, precision, recall (also referred to as sensitivity), f-score, Cohen kappa score, OOB and AUC of ROC curve.

4.3.1 Correlation Between Attributes

Tables 4.9 and 4.10 display the correlation among attributes for the input data. Correlation pertains to the association between two variables and the extent to which they exhibit concurrent changes or lack thereof. Pearson's Correlation Coefficient is the prevailing approach for computing correlation, which assumes a normal distribution of the variables under consideration. A correlation coefficient of -1 or 1 indicates a complete negative or positive correlation, respectively. In contrast, a value of 0 indicates a complete absence of correlation.

The matrices presented in Tables 4.9 and 4.10 display a comprehensive arrangement of attributes both horizontally and vertically. These matrices provide a correlation analysis for all possible pairs of attributes, accounting for the symmetrical nature of the matrix. The presence of a diagonal line in the matrix, extending from the top left to the bottom right corners, indicates a state of perfect correlation between each attribute and itself.

Table 4.9: Correlation Between Attributes of week-days data (M25 highway).

	Normal working	**Holiday**	**Weekend**
	Density	**Density**	**Density**
ATR	***0.60***	***0.61***	***0.52***
RSI	0.29	0.34	0.12
SMA	***0.99***	***0.98***	***0.99***
EMA	***0.99***	***0.99***	***0.99***
ROC	0.16	0.16	0.13
MOM	0.20	0.21	0.16

Table 4.10: Correlation Between Attributes of weekdays data (M60 highway).

	Normal working	**Holiday**	**Weekend**
	Density	**Density**	**Density**
ATR	***0.63***	**0.51**	***0.69***
RSI	0.21	0.19	0.13
SMA	***0.99***	***0.99***	***0.99***
EMA	***0.99***	***0.99***	***0.99***
ROC	0.11	-0.08	-0.04
MOM	0.22	0.19	0.11

Based on the aforementioned Tables 4.9 and 4.10, a notable correlation is observed between vehicle density and indicators (ATR, SMA, EMA) during weekday hours. This finding implies that the data can be utilized to formulate effective strategies for traffic management during peak periods. Potential approaches may involve promoting remote work arrangements, expanding public transportation alternatives, or fostering the adoption of carpooling services. Additionally, it is worth mentioning that the remaining three indicators exhibit a limited correlation with traffic density, thereby rendering them unreliable for the purpose of formulating efficient traffic management strategies.

4.3.2 The results of estimating the level of service in normal working days

As previously stated, a one-month dataset from the year 2022 was gathered for the purpose of testing the methodology. The month of February was chosen due to a partial recovery in traffic congestion, with approximately 95% of normal traffic levels being observed. The RF model and KNN model were assessed by utilizing the test data. Tables 4.11 and 4.12 demonstrate that the test outcomes closely align with the training datasets for both of the models proposed.

Table 4.11: Summary of testing the proposed classification methods in normal working days (M25 highway).

Model	LOS	Precision	Recall	F-Score	Support	Accuracy	kappa
RF	A	1.00	0.99	1.00	130	0.97	0.95
	B	0.95	0.99	0.97	91		
	C	0.93	0.87	0.90	30		
	D	1.00	0.77	0.87	13		
	E	0.88	1.00	0.93	14		
	F	1.00	1.00	1.00	10		
KNN	A	0.99	0.98	0.99	133	0.92	0.88
	B	0.89	0.98	0.93	88		
	C	0.79	0.66	0.72	29		
	D	0.67	0.40	0.50	10		
	E	0.83	0.83	0.83	12		
	F	0.94	1.00	0.97	16		

Table 4.12. Summary of testing the proposed classification methods in normal working days (M60 highway).

Model	LOS	Precision	Recall	F-Score	Support	Accuracy	kappa
RF	A	0.98	1.00	0.99	224	0.97	0.91
	B	0.96	0.90	0.93	50		
	C	0.82	0.82	0.82	11		
	D	0.67	0.67	0.67	3		
KNN	A	0.97	0.99	0.98	224	0.95	0.85
	B	0.89	0.84	0.87	50		
	C	0.69	0.82	0.75	11		
	D	1.00	0.33	0.50	3		

The assessment of the proposed models typically involves the consideration of accuracy. When the accuracy of our model is high, it indicates that the classification of items is done correctly. The random forest model outperforms the KNN model in terms of accuracy. For instance, in the M25 highway test data presented in table 4.11, the random forest model achieved an accuracy of 97%. Additionally, the random forest model demonstrated high precision and recall, which is evident from the F-score values. The lowest F-score value obtained was 87%. The utilization of the Harmonic Mean as a substitute for the Arithmetic Mean entails a greater penalization of extreme values. An additional noteworthy observation is that the F-score exhibits a positive correlation with the increase in support value. The findings pertaining to classification outcomes, as well as the observed patterns in the fluctuation of classification accuracy and other metrics in relation to the augmentation of the support value in KNN, exhibit similarities to the patterns observed in random forests. Nevertheless, the results obtained are comparatively inferior when compared to the random forest model.

Tables 4.11 and 4.12 present the kappa values, which demonstrate a high level of agreement between the observed and predicted values for the random forest model. This indicates a strong performance of the model. Simultaneously, KNN model also attained a highly favorable outcome. Hence, based on the Kappa statistic obtained from the proposed models, it can be concluded that these models exhibit reliability in terms of classification.

4.3.3 The results of estimating the level of service in holiday days

The density data for the compounds utilized in the trials consists of quarter-hourly records spanning from 1 February 2022 to 28 February 2022. These records were obtained from the MIDAS system, which specifically focused on normal working days as discussed in the preceding section. This section will examine the vehicle density during holidays occurring within the same month. The technical indicators that have been proposed, as described in Section 3.2.3, are employed as inputs for the classification algorithms in order to estimate the level of service (LOS).

The proposed models undergo training using 80% of the available data, while the evaluation is conducted on the remaining 20% of the data using technical indicators. In order to assess the efficacy of the technical indicators in enhancing the proposed rating models, a training/testing split is employed over a period of 5 days. Tables 4.13 and 4.14 demonstrate that the test outcome closely aligns with the training datasets. The method under consideration was implemented during regular business days using the same selected approach, and the outcome demonstrated comparable levels of accuracy. The proposed method demonstrates potential applicability across various time periods.

Table 4.13. Summary of testing the proposed classification methods in holidays days (M25 highway).

Model	LOS	Precision	Recall	F-Score	Support	Accuracy	kappa
RF	A	0.98	1.00	0.99	40	0.92	0.88
	B	0.96	0.92	0.94	24		
	C	0.77	0.91	0.83	11		
	D	1.00	0.40	0.57	5		
	E	0.86	0.75	0.80	8		
	F	0.80	1.00	0.89	8		
KNN	A	0.98	1.00	0.99	40	0.85	0.80
	B	0.88	0.88	0.88	24		
	C	0.58	0.64	0.61	11		
	D	0.50	0.20	0.29	5		
	E	0.83	0.62	0.71	8		
	F	0.73	1.00	0.84	8		

Table 4.14. Summary of testing proposed classification methods in holidays days (M60 highway).

Model	LOS	Precision	Recall	F-Score	Support	Accuracy	kappa
RF	A	0.97	0.97	0.97	67	0.95	0.88
	B	0.92	0.92	0.92	25		
	C	0.67	1.00	0.80	2		
	D	1.00	0.50	0.67	2		
KNN	A	0.96	0.96	0.96	67	0.93	0.83
	B	0.88	0.88	0.88	25		
	C	0.67	1.00	0.80	2		
	D	1.00	0.50	0.67	2		

The findings displayed in Tables 4.13 and 4.14 indicate that the RF-based approach exhibits the highest level of precision in its predictions. The confirmation of accuracy and kappa values indicates that the model exhibits a high level of classification quality and demonstrates strong reliability.

4.3.4 The results of estimating the level of service in weekend days

The 15-minute traffic data that has been compiled is sourced from the MIDAS system. A total of 614 records from the weekends in February 2022 were chosen for the training dataset. The testing dataset which consists of 153 records was selected from the weekend period in February 2022. The training and testing datasets consist of traffic data that exclusively represents normal traffic conditions, excluding any incidents or other abnormal events. The tables 4.15 and 4.16 present the outcomes that demonstrate the precision of two traffic classification frameworks when subjected to regular traffic conditions. This augmentation of explanatory power in the input features contributes to the improved accuracy under normal traffic conditions.

Table 4.15: Summary of testing the proposed classification methods in weekend days (M25 highway).

Model	LOS	Precision	Recall	F-Score	Support	Accuracy	kappa
RF	A	1.00	0.96	0.98	91	0.95	0.91
	B	0.88	1.00	0.93	35		
	C	1.00	0.33	0.50	3		
	D	0.50	1.00	0.67	3		
	E	1.00	0.67	0.80	6		
	F	1.00	1.00	1.00	16		
KNN	A	0.98	0.95	0.96	91	0.90	0.82
	B	0.82	0.94	0.88	35		
	C	0.00	0.00	0.00	3		
	D	0.38	1.00	0.55	3		
	E	0.50	0.17	0.25	6		
	F	0.94	0.94	0.94	16		

Table 4.16: Summary of testing the proposed classification methods in weekend days (M60 highway).

Model	LOS	Precision	Recall	F-Score	Support	Accuracy	kappa
RF	A	0.98	1.00	0.99	167	0.98	0.92
	B	1.00	0.88	0.94	25		
KNN	A	0.94	1.00	0.97	135	0.94	0.64
	B	0.83	0.42	0.73	12		
	F	0.80	0.57	0.67	7		

The findings displayed in Tables 4.15 and 4.16 indicate that the RF-based approach exhibits the highest level of precision in prediction. The confirmation of accuracy and kappa values indicates that the model exhibits a high level of classification quality and demonstrates strong reliability.

4.3.5 Out-of-bag (OOB)

As stated in section 2.11, OOB score is computed using a portion of data that remains unused during the model's analysis, while the validation set is a subset of data that is deliberately chosen for evaluation purposes. The OOB sample exhibits a slightly higher degree of randomness compared to the validation set. Hence, it is possible that OOB sample, which serves as the basis for calculating the OOB score, may exhibit a higher level of difficulty compared to the validation set. OOB score may exhibit a relatively lower accuracy score as a result. The OOB error rate was calculated for the random forest classifier using the MIDAS dataset.

Tables 4.17 and 4.18 reveal an interesting pattern: when the number of estimators (trees) in a random forest model increases, each tree trains on a different subset of the data. This distinction arises due to the utilization of bootstrap sampling, a method where training data is randomly sampled with replacement. This implies that each tree in the model is exposed to a slightly distinct set of samples. As the number of estimators increases, the probability of each tree encountering a greater diversity of training samples also increases. Consequently, there is a reduction in the correlation between the trees, resulting in increased diversity in terms of the patterns they extract from the data.

The enhanced variety of tree species contributes to the enhancement of the model's capacity for generalization. While individual trees may exhibit overfitting to a certain degree, the ensemble technique employed by random forests allows for their combination, resulting in improved predictive accuracy. As a result, the augmentation of the number of estimators in a random forest model is associated with a decrease in OOB error of the model. This phenomenon occurs due to the ensemble's ability to leverage enhanced diversity, thereby enabling it to generate more resilient predictions when confronted with previously unseen data. This finding also provides an explanation for the lack of overfitting in random forests as the ensemble size increases. The decision-making process typically involves a trade-off between the allocation of computational resources and the resulting performance of the model.

Table 4.17: OOB error rate vs Number of estimators (M25 highway).

	50-estimators.	**100-estimators.**	**250-estimators.**	**500-estimators.**	**1000-estimators.**
OOB error rate for normal working days	0.955	0.957	0.958	0.96	0.962
OOB error rate for holiday days	0.905	0.903	0.905	0.905	0.908
OOB error rate for weekend days	0.973	0.967	0.972	0.97	0.972

Table 4.18: OOB error rate vs Number of estimators (M60 highway).

	50-estimators.	**100-estimators.**	**250-estimators.**	**500-estimators.**	**1000-estimators.**
OOB error rate for normal working days	0.971	0.973	0.973	0.973	0.973
OOB error rate for holiday days	0.968	0.968	0.96	0.965	0.971
OOB error rate for weekend days	0.993	0.994	0.994	0.994	0.994

4.3.6 Receiver operating characteristic curves (ROC)

As stated in section 2.11, ROC is a graphical technique used to assess the effectiveness of a multiclass classifier. A graphical representation is created by plotting the True Positive Rate (also known as sensitivity) against the False Positive Rate (specificity subtracted from one) at different threshold values. The ROC curve illustrates the balance between sensitivity and specificity. When the curve approaches the leftmost and uppermost boundaries of ROC space, it signifies a higher level of accuracy for the test.

The test's accuracy increases as the curve approaches the upper and left boundaries. When the curve ROC space closely approximates the 45-degree diagonal line, it indicates that the test's accuracy is low. ROC curves are a valuable tool in the process of model selection, as they enable the identification and elimination of suboptimal models, ultimately leading to the selection of the most optimal model. The ROC of our model is depicted in Figures 4.7 and 4.8.

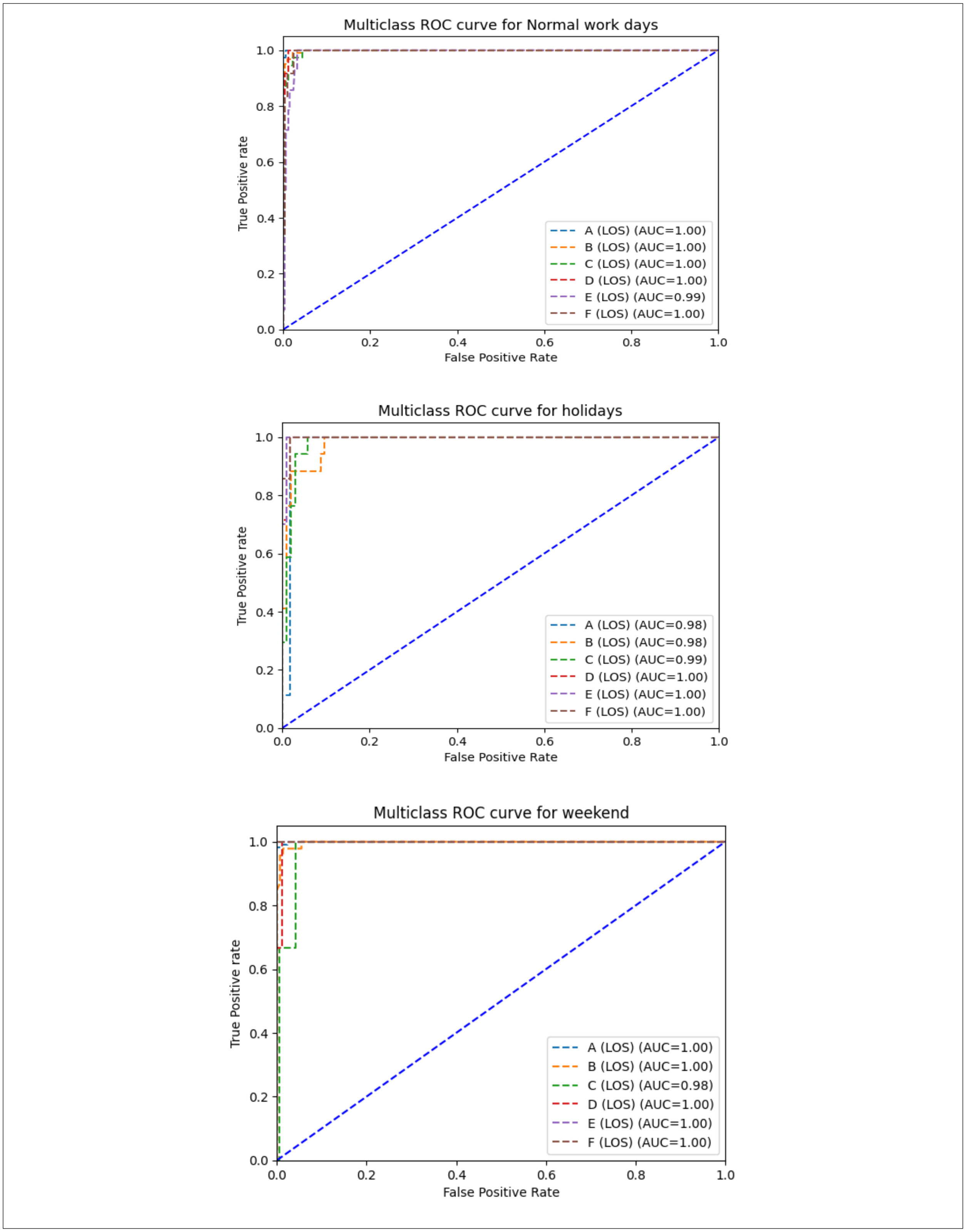


Figure. 4.7: ROC curves plotted for random forests (M25 highway).

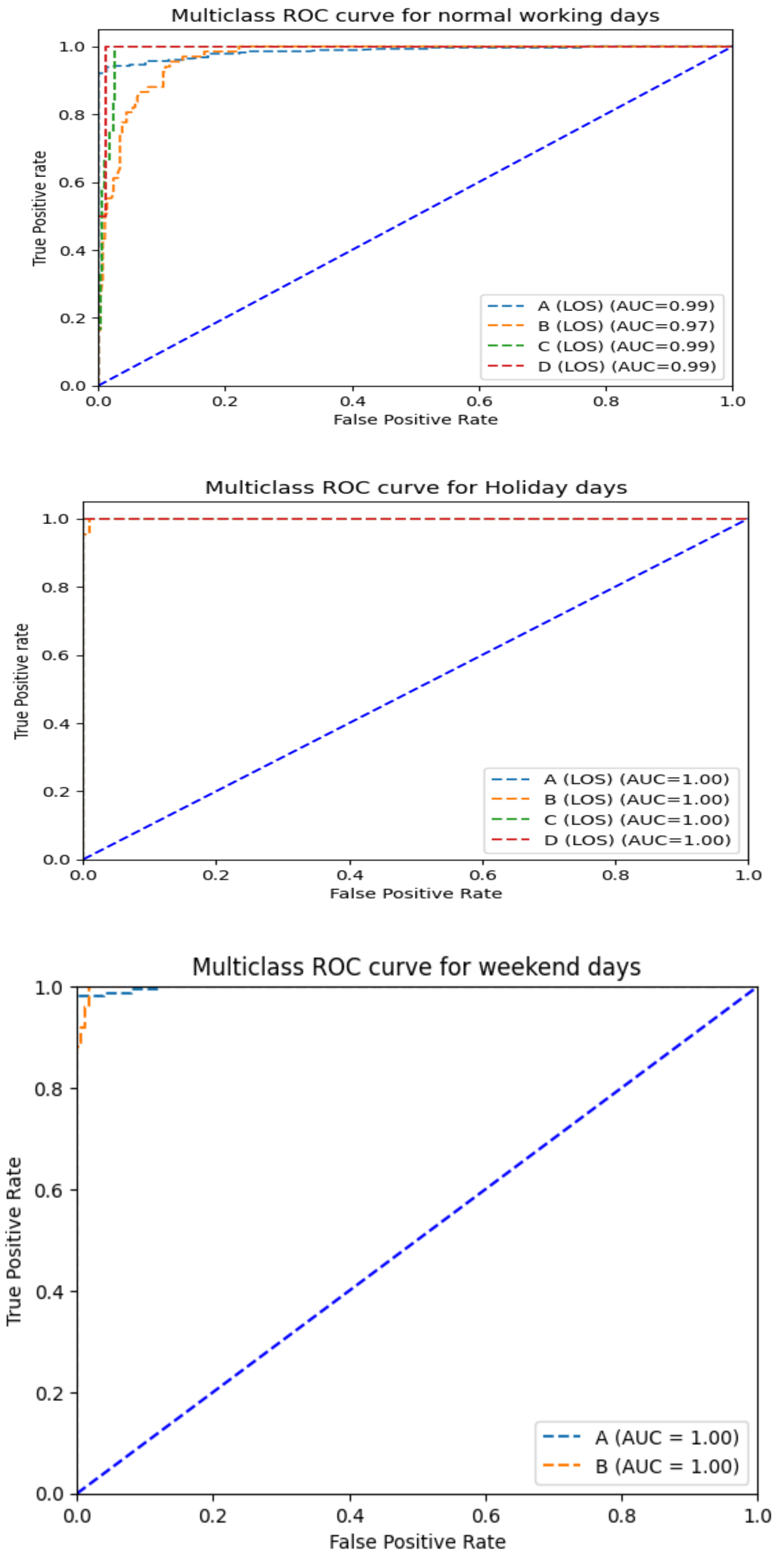


Figure. 4.8: ROC curves plotted for random forests (M60 highway).

One rationale behind the enhancement of accuracy in our proposed models with an increment in the value of t is that the technical indicators have the capacity to capture a greater amount of information pertaining to the fluctuations in densities over an extended temporal span. This implies that the densities of vehicles may not necessarily exhibit a consistent trend of increase or decrease, but rather that a reliable prediction can be made with sufficient information. This observation is further supported by the notable AUC values, which serve as an indicator of the multiclass classifier's capacity to differentiate between various classes.

4.3.7 Feature’s importance

In the field of machine learning, features refer to the variables or attributes employed to characterize the data upon which the algorithm is being trained. The careful consideration and significance of features play a pivotal role in determining the efficacy and triumph of a machine learning model. The enhancement of prediction accuracy in the model is achieved through the careful selection of pertinent and informative features. By employing a process of feature selection, it is possible to streamline the model by reducing the number of variables and enhancing the model's interpretability. The training duration of the model increases proportionally with the number of features incorporated. By employing a strategy of feature selection, it is possible to decrease the duration of the training process by focusing solely on the most significant features. The significance of features may vary in accordance with alterations in the dataset's size.

By strategically choosing the most significant features, the model can be enhanced in its ability to withstand variations in the size of the dataset. This dissertation posits that the utilization of technical indicators data has the potential to enhance the accuracy of LOS classification. Consequently, a variable importance analysis was conducted using the selected random forest model. The calculation of the average reduction in the Gini index was performed using the Random Forest (RF) model. A greater magnitude of this index signifies a greater significance of the variable, as stated in tables 4.19 and 4.20.

Table 4.19: Feature’s importance: of M25 highway sample dataset.

Week days	Indicators	Importance of Features (in %)
Normal working days	SMA	40.3
	EMA	40.6
	ATR	6.6
	MOM	4.7
	RSI	2.9
	ROC	4.5
Holiday days	SMA	36.3
	EMA	36.4
	ATR	10.3
	MOM	5.6
	RSI	5.7
	ROC	5.4
Weekend days	EMA	36.8
	SMA	40.5
	ATR	12
	MOM	4.4
	ROC	3.5
	RSI	2.5

Table 4.20: Feature’s importance: of M60 highway sample dataset.

Week days	Indicators	Importance of Features (in %)
Normal working days	EMA	44.6
	SMA	34.9
	ATR	7.5
	MOM	6.5
	ROC	4.09
	RSI	2.3
Holiday days	EMA	43.5
	SMA	37.5
	ATR	7.2
	ROC	3.6
	MOM	6.04
	RSI	1.9

Weekend days	EMA	42.8
	SMA	42.7
	ATR	3.4
	ROC	2.5
	MOM	6.1
	RSI	2.3

Based on the findings presented in Tables 4.19 and 4.22, it is evident that the SMA, EMA, ATR exhibit significant influence in the determination of the LOS variable, in their respective order of importance. It was observed across all weekdays examined in our experiments.

4.4 Discussion

In the context of regression models, it's noteworthy that the test data displayed consistent validation performance across various models, indicating the comparability of their results. However, when incorporating technical analysis indicators, which involve the application of statistical and computational methods, distinct outcomes emerged for each model. Among these models, MLR demonstrated the most effective performance in predicting short-term traffic density. This superiority can be attributed to MLR's suitability for traffic density data, given its assumption of a linear association between predictors and the response variable. The simplicity of MLR methodology played a significant role in its enhanced performance, enabling a deeper understanding of the relationship between predictors, such as historical traffic data and time of day, and traffic density.

On the other hand, FFNNs showcased promising predictive capabilities compared to Random Forests and Markov chain models when forecasting vehicle density in VANET networks. FFNNs excelled in capturing non-linear relationships, engaging in representation learning, accommodating complex data, and effectively handling temporal dependencies.

When evaluating LOS of specific highways, the proposed classification models exhibited exceptional and closely correlated results. This phenomenon can be attributed to the high explanatory capacity of technical analysis indicators used as input features for classification models. Notably, the Random Forest model outperformed KNN model due to several reasons. Random Forests excelled in capturing intricate non-linear associations between input features and the target variable, especially regarding traffic volume, speed, and occupancy, which are associated with the LOS indicator. In contrast, KNN assumed linear relationships within local neighborhoods. Moreover, Random Forests provided feature importance measures, shedding light on the relative significance of input features in vehicle density classification. The ensemble nature of Random Forests also helped mitigate the influence of outliers and noisy data points, bolstering the accuracy of classification.

Furthermore, when comparing the proposed machine learning methods to previous research findings, we observed significant improvements. Under typical traffic conditions during regular working days, the three machine learning methods showcased an average RMSE score improvement of 86.63% for the M60 motorway data compared to the findings of Sun et al. (2020) using the same MIDAS dataset. Similarly, for the M25 highway data, the proposed models demonstrated a mean RMSE score improvement of 68.2% across the three machine learning methods when compared to the results obtained by Chen & Chaudhari (2021) for the same dataset. These enhancements can be attributed to the utilization of technical analysis indicators that prioritize the examination of data and its patterns within relatively brief time periods. Additionally, the Markov chain model yielded favorable outcomes for the designated highways and corresponding datasets. Overall, the three machine learning methods and the Markov chain exhibited a comparable level of accuracy in prediction. However, the MLR-based approach displayed superior predictive capabilities for traffic variables due to the inherent linearity of the data.

In summary, historical data on traffic flow and speed proved instrumental in forecasting vehicle concentration, facilitating traffic condition evaluation, congestion anticipation, and LOS severity assessment.

4.5 Summary

In this chapter, experiments conducted to predict traffic density are discussed. With the initial correlation analysis of the input data and the breakdown of the dataset, it is discussed in detail. Finally, experimental results for different scenarios were presented.

Chapter Five
Conclusions and Future Works

5.1 Conclusions

The objective of this dissertation was to devise techniques for the short-term forecasting of traffic state variables on urban highway roads across various scenarios, while also identifying the optimal model that can yield the most accurate outcomes. Thus, it expands upon existing scholarly works pertaining to machine learning-based traffic prediction. The proposed topology incorporates machine learning techniques in two distinct phases. The initial phase involves the prediction of traffic density, while the subsequent phase focuses on the determination of level of services (LOSs). The feature values in the initial layer are standardized within a specific range to ensure that the outcomes remain consistent across different features. In the second stage, the data underwent a smoothing process in order to mitigate variance. The subsequent two stages were employed to generate vehicular traffic and estimate LOS. The models proposed in this dissertation, utilizing the MIDAS dataset, demonstrated notable levels of prediction accuracy, as assessed through the MAPE, RMSE, and Accuracy metrics.

The primary findings of this dissertation, derived from the development and execution of the suggested system, are as follows:

1- The experimental findings suggest that the proposed frameworks have the potential to be applied in various geographical contexts. The proposed frameworks were subsequently evaluated by utilizing traffic data from the M60 motorway as a secondary case study, requiring only minimal calibration.

2- The results obtained from the dissertation demonstrate that the proposed models exhibit a high level of effectiveness in comparison to the previous models in situations where there are variations in traffic patterns during holidays and weekends.

3- Classification machine learning estimate LOS hourly using data from stationary MIDAS sensors. By analyzing traffic density and applying HCM density thresholds, the study successfully estimated real-world conditions. Incorporating technical analysis indicators as input improved model accuracy by approximately

6.52%, especially for the M60 highway. The method's applicability was confirmed when tested on data from a different highway (M25).

4- The findings indicate that the accuracy of LOS estimation remained relatively consistent across various times of day and LOS categories in this particular method. The results of the sensitivity analysis provide confirmation that the accuracy of this methodology remains consistent regardless of whether it is applied during peak or non-peak hours. The findings further suggest that the indicators, namely ATR, SMA, and EMA, play a crucial role in determining the LOS.

5- The proposed models offer a versatile way to assess highway traffic conditions, free from reliance on fixed sensors, specific timeframes, or particular days of the week. This approach has broad applicability across freeway segments, reducing implementation and maintenance costs. Transportation agencies and DOTs can use it for traffic operations, even in non-urban areas lacking stationary sensors. Density data can be employed across different time intervals, including hourly, daily, and peak traffic hours.

5.2 Limitations

Although the dissertation’s outcomes are promising, it is important to acknowledge its limitations, which invite further investigation.

1- Methods for reducing model complexity:

The evaluation of model complexity often involves an assessment of two key factors: structural complexity and computational complexity. Typically, as the complexity of the model structure increases, there is a corresponding increase in computational complexity. In contrast to statistics-based models, machine learning-based models typically exhibit a more intricate model structure. This is primarily attributed to the presence of numerous hyperparameters within the model, such as the connection weights between different pairs of neurons and the threshold value of each functional neuron in FFNN, among others. Similarly, the computational expense of the machine learning model is also elevated. Hence, the challenge of effectively reducing model complexity is a significant consideration when employing machine learning-based models in practical applications.

2- The restriction of the data source:

Machine Learning-based models face a significant obstacle in the form of a restricted pool of accessible data sources. As is widely recognized, the training procedure of machine learning models necessitates a substantial dataset, and the efficacy of these datasets can significantly impact the training outcome. Therefore, the deployment of machine learning prediction methods in real-world scenarios is hindered by the insufficient availability of datasets that accurately represent the complex traffic conditions in actual traffic networks. At present, the available data sources primarily consist of traffic data collected on highways. These sources include Highways England (UK government data, 2023), Caltrans Performance Measurement System (PeMs) (Caltrans, 2023), Maryland 511 (MD) (Maryland Department of Transportation, 2023), and Waze (Waze, 2023). The features of the subject in question are enumerated as follows:

- Highway England provides datasets on the speed and volume of traffic for a 15-minute interval. These datasets pertain to a significant portion, approximately one-third, of the total motor vehicle traffic in England (UK government data, 2023).
- The PeMs system offers real-time traffic volume information as well as a comprehensive dataset spanning over ten years. This dataset covers a 5-minute interval and includes data from more than 39,000 probers located in both urban and suburban areas of California. Additionally, the system provides supplementary information on road conditions, including incidents and lane closures (Caltrans, 2023).
- Maryland 511 (MD) offers real-time road condition images and provides traffic flow data at 15-minute intervals. In addition, this system also provides information on weather conditions (Maryland Department of Transportation, 2023).
- The company Waze utilizes user location data within its application to generate analyses on speed and travel time. In addition to its primary navigation services, Waze also offers various event reports encompassing congestion, incidents, severe weather, and road construction (Waze, 2023).

Nevertheless, there remains a deficiency in the available dataset pertaining to traffic flow in urban areas. In order to enhance the efficacy of Intelligent

Transportation Systems (ITS), it is imperative for traffic flow prediction models developed in recent years to encompass not only suburban areas but also urban environments. However, the limited availability of data sources poses a significant constraint on the enhancement of model adaptability. In addition, it should be noted that the temporal resolution of the data sources mentioned is typically limited to 15-minute intervals. This temporal constraint poses a limitation for studies requiring shorter intervals of analysis.

5.3 Future Works

Below are several potential future works that can be considered:

1- Studying external indicators (special events (e.g., sports games), weather conditions, road construction, public transportation, etc.) is crucial in evaluating their impact on forecast accuracy.

2- Implementing the proposed model on additional datasets, such as PeMS and Waze, to further evaluate its effectiveness and generalizability.

3- Taking into account potential fluctuations and the sensitivity of the methodology with regard to meteorological factors.

4- The methodology used in this dissertation as based on density data as a means of estimating LOS. However, spatial variation can serve as an input parameter for LOS evaluation. LOS estimation can also take into account the variation in velocity between upstream and downstream sectors.

References

Abboud, K., & Zhuang, W. (2015). Mobility Modeling for Vehicular Communication Networks. Springerbriefs in Electrical and Computer Engineering. https://doi.org/10.1007/978-3-319-25507-1

Arafat Abu Alfeilat, H., Hassanat, A. B. A., Lasassmeh, O., Tarawneh, A. S., Alhasanat, M. B., Salman, H. S. E., & Surya Prasath, V. B. (2019). Effects of distance measure choice on K-Nearest Neighbor Classifier Performance: A Review. Big Data, 7, 221-248. https://doi:10.1089/big.2018.0175

Ali, I., & Li, F. (2020). An efficient conditional privacy-preserving authentication scheme for Vehicle-To-Infrastructure communication in VANETs. Vehicular Communications, 22, 100228. https://doi.org/10.1016/j.vehcom.2019.100228.

Ali, I., Lawrence, T., Omala, A. A., & Li, F. (2020). An Efficient Hybrid Signcryption Scheme with Conditional Privacy-Preservation for Heterogeneous Vehicular Communication in VANETs. IEEE Transactions on Vehicular Technology, 69(10), 11266-11280. https://doi.org/10.1109/TVT.2020.3008781.

Alizadeh, M., Beheshti, M. T. H., Ramezani, A., & Saadatinezhad, H. (2020). Network Traffic Forecasting Based on Fixed Telecommunication Data Using Deep Learning. In 2020 6th Iranian Conference on Signal Processing and Intelligent Systems (ICSPIS) (pp. 1-7). Mashhad, Iran. https://doi.org/10.1109/ICSPIS51611.2020.9349573.

Aljamal, M. A., Abdelghaffar, H. M., & Rakha, H. A. (2019). Developing a Neural–Kalman Filtering Approach for Estimating Traffic Stream Density Using Probe Vehicle Data. Sensors, 19(19), 4325. https://doi.org/10.3390/s19194325.

Aljeri, N., & Boukerche, A. (2020). Fog-enabled vehicular networks: A new challenge for mobility management. Internet Technology Letters, 3, e141. https://doi.org/10.1002/itl2.141

Aljeri, N., & Boukerche, A. (2020, November). An adaptive traffic-flow-based controller deployment scheme for software-defined vehicular networks. In Proceedings of the 23rd International ACM Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems (pp. 191-198). https://doi.org/10.1145/3416010.3423237

Altintasi, O., Tuydes-Yaman, H., & Tuncay, K. (2017). Detection of urban traffic patterns from Floating Car Data (FCD). Transportation Research Procedia, 22, 382-391. https://doi.org/10.1016/j.trpro.2017.03.057.

Arif, M., Wang, G., Bhuiyan, M. Z. A., Wang, T., & Chen, J. (2019). A survey on security attacks in VANETs: Communication, applications and challenges. Vehicular Communications, 19, 100179. https://doi.org/10.1016/j.vehcom.2019.100179.

Ayala, J., García-Torres, M., Vázquez Noguera, J. L., Gómez-Vela, F., & Divina, F. (2021). Technical analysis strategy optimization using a machine learning approach in stock market indices. Knowledge-Based Systems, 225, 107119. https://doi.org/10.1016/j.knosys.2021.107119.

Basak, S., Kar, S., Saha, S., Khaidem, L., & Roy Dey, S. (2019). Predicting the direction of stock market prices using tree-based classifiers. The North American Journal of Economics and Finance, 47, 552-567. https://doi.org/10.1016/j.najef.2018.06.013.

Memisoglu Baykal, T., Colak, H. E., & Kılınc, C. (2022). Forecasting future climate boundary maps (2021–2060) using exponential smoothing method and GIS. Science of The Total Environment, 848, 157633. https://doi.org/10.1016/j.scitotenv.2022.157633.

Belt, E. A., Koch, T., & Dugundji, E. R. (2023). Hourly forecasting of traffic flow rates using spatial temporal graph neural networks. Procedia Computer Science, 220, 102-109. https://doi.org/10.1016/j.procs.2023.03.016.

Elefteriadou, L. (2016). The Highway Capacity Manual 6th Edition: A Guide for Multimodal Mobility Analysis. ITE Journal, 86(4), 14-18.

Boukerche, A., Tao, Y., & Sun, P. (2020). Artificial intelligence-based vehicular traffic flow prediction methods for supporting intelligent transportation systems. Computer Networks, 182, 107484. https://doi.org/10.1016/j.comnet.2020.107484

California Department of Transportation (Caltrans). (2009). PeMS data source. Retrieved from http://pems.dot.ca.gov

Candès, E. J., & Sur, P. (2020). The phase transition for the existence of the maximum likelihood estimate in high-dimensional logistic regression. Annals of Statistics, 48(1), 27-42. https://doi.org/10.1214/18-AOS1789

Chen, Q., Song, Y., & Zhao, J. (2021). Short-term traffic flow prediction based on improved wavelet neural network. Neural Computing and Applications, 33, 8181–8190. https://doi.org/10.1007/s00521-020-04932-5.

Chen, X., & Chaudhari, P. (2021). MIDAS: Multi-agent Interaction-aware Decision-making with Adaptive Strategies for Urban Autonomous Navigation. In 2021 IEEE International Conference on Robotics and Automation (ICRA) (pp. 7980-7986). Xi'an, China. https://doi:10.1109/ICRA48506.2021.9561148.

Cheng, Z. (Aaron), Pang, M.-S., & Pavlou, P. A. (2020). Mitigating Traffic Congestion: The Role of Intelligent Transportation Systems. Information Systems Research, 31(3), 653-674. https://doi.org/10.1287/isre.2019.0894

Chikkakrishna, N. K., Hardik, C., Deepika, K., & Sparsha, N. (2019). Short-Term Traffic Prediction Using SARIMA and FbPROPHET. In 2019 IEEE 16th

India Council International Conference (INDICON) (pp. 1-4). Rajkot, India. https://doi:10.1109/INDICON47234.2019.9028937.

Chollet, F. (2015). Keras. [Website]. Retrieved May 29, 2019, from https://keras.io

De Raadt, A., Warrens, M. J., Bosker, R. J., & Kiers, H. A. L. (2019). Kappa Coefficients for Missing Data. Educational and Psychological Measurement, 79(3), 558-576. https://doi.org/10.1177/0013164418823249

Dechenaux, E., Mago, S. D., & Razzolini, L. (2014). Traffic congestion: An experimental study of the Downs-Thomson paradox. Experimental Economics, 17, 461–487. https://doi.org/10.1007/s10683-013-9378-4

Demir, S., Mincev, K., Kok, K., & Paterakis, N. G. (2020). Introducing Technical Indicators to Electricity Price Forecasting: A Feature Engineering Study for Linear, Ensemble, and Deep Machine Learning Models. Applied Sciences, 10(1), 255. https://doi.org/10.3390/app10010255

Department for Transport (DFT). (2023, January 1). Road Traffic Statistics. https://www.gov.uk/government/collections/road-traffic-statistics

Department for Transport (DFT). (2014). Road traffic statistics. Retrieved from https://www.dft.gov.uk/traffic-counts/download.php

Di Piazza, A., Di Piazza, M. C., La Tona, G., & Luna, M. (2021). An artificial neural network-based forecasting model of energy-related time series for electrical grid management. Mathematics and Computers in Simulation, 184, 294-305. https://doi.org/10.1016/j.matcom.2020.05.010.

El Joubari, O. (2022). Mobility & traffic models for VANETs (Doctoral dissertation, Université Paris-Saclay). [cs.NI]. https://tel.archives-ouvertes.fr/tel-03627931

England, H. (2015). Highways England–Data.gov.uk–Journey Time and Traffic Flow Data April 2015 onwards–User Guide (No. April, pp. 1-14).

Essien, A., Petrounias, I., Sampaio, P., et al. (2021). A deep-learning model for urban traffic flow prediction with traffic events mined from Twitter. World Wide Web, 24, 1345-1368. https://doi:10.1007/s11280-020-00800-3

Al-Zuwaini, M. S. (2012). The Use of Artificial Neural Networks for Productivity Estimation of Finishing Stone Works for Building Projects. Journal of Engineering and Sustainable Development, 16(2), Ar–42. URL: https://jeasd.uomustansiriyah.edu.iq/index.php/jeasd/article/view/1197

Fang, W., Zhuo, W., Song, Y., Yan, J., Zhou, T., & Qin, J. (2023). Δfree-LSTM: An error distribution free deep learning for short-term traffic flow forecasting. Neurocomputing, 526, 180-190. https://doi.org/10.1016/j.neucom.2023.01.009.

Fujita, H., and Cimr, D. (2019). Computer-aided detection for fibrillations and flutters using deep convolutional neural network. Information Sciences, 486, 231-239. https://doi.org/10.1016/j.ins.2019.02.065.

Genuer, R., & Poggi, J.-M. (2020). Random Forests with R (1st ed., Use R!). Springer. https://doi.org/10.1007/978-3-030-56485-8

Ghaleb, B., et al. (2019). A Survey of Limitations and Enhancements of the IPv6 Routing Protocol for Low-Power and Lossy Networks: A Focus on Core Operations. IEEE Communications Surveys and Tutorials, 21(2), 1607-1635. https://doi:10.1109/COMST.2018.2874356

Giovanis, E. (2010, August 28). Applications of Feed-Forward Neural Networks with Error Backpropagation Algorithm and Non-Linear Methods in MATLAB. SSRN. https://ssrn.com/abstract=1667438. http://dx.doi.org/10.2139/ssrn.1667438

Gonçalves, F., et al. (2019). A Systematic Review on Intelligent Intrusion Detection Systems for VANETs. In 2019 11th International Congress on Ultra-Modern Telecommunications and Control Systems and Workshops

(ICUMT) (pp. 1-10). Dublin, Ireland. doi: 10.1109/ICUMT48472.2019.8970942.

Guo, F. (2013). Short-term traffic prediction under normal and abnormal conditions (Doctoral dissertation). Imperial College London.

Habibzadeh, H., Soyata, T., Kantarci, B., Boukerche, A., & Kaptan, C. (2018). Sensing, communication, and security planes: A new challenge for a smart city system design. Computer Networks, 144, 163-200. https://doi.org/10.1016/j.comnet.2018.08.001.

Hajlaoui, R., Alsolami, E., Moulahi, T., & Guyennet, H. (2019). An adjusted K-medoids clustering algorithm for effective stability in vehicular ad hoc networks. International Journal of Communication Systems, 32, e3995. https://doi.org/10.1002/dac.3995

Hansch, R. (2018). Handbook of Random Forests: Theory and Applications for Remote Sensing. World Scientific Publishing Co Pte Ltd.

Hargrove, S. R., Lim, H., Han, L. D., & Freeze, P. B. (2016). Empirical Evaluation of the Accuracy of Technologies for Measuring Average Speed in Real Time. Transportation Research Record, 2594(1), 73-82. https://doi.org/10.3141/2594-11

Seiran Heshami & Lina Kattan (2023) A stochastic microscopic based freeway traffic state and spatial-temporal pattern prediction in a connected vehicle environment, Journal of Intelligent Transportation Systems, DOI: 10.1080/15472450.2022.2130291

Highways Agency. (2009). M60 Motorway. Archived from the original on November 15, 2009. Retrieved March 18, 2023, from [URL: https://webarchive.nationalarchives.gov.uk/ukgwa/20091115001925/http://www.highways.gov.uk/roads/23612.aspx].

Highways England. (n.d.). Flow. Retrieved from https://m.highwaysengland.co.uk/#flow

Hoseinzadeh, N., Gu, Y., Han, L. D., Brakewood, C., and Freeze, P. B. (2021). Estimating Freeway Level-of-Service Using Crowdsourced Data. Informatics, 8(1), 17. https://doi.org/10.3390/informatics8010017

Hu, Y., Li, Y., Huang, H., Lee, J., Yuan, C., & Zou, G. (2022). A high-resolution trajectory data-driven method for real-time evaluation of traffic safety. Accident Analysis and Prevention, 165, 106503. https://doi.org/10.1016/j.aap.2021.106503.

Huang, W., An, Y., Pan, Y., Li, J., & Chen, C. (2022). Predicting transient particle transport in periodic ventilation using Markov chain model with pre-stored transition probabilities. Building and Environment, 211, 108730. https://doi.org/10.1016/j.buildenv.2021.108730.

Isnain, A. R., Supriyanto, J., & Kharisma, M. P. (2021). Implementation of K-Nearest Neighbor (K-NN) Algorithm for Public Sentiment Analysis of Online Learning. IJCCS (Indonesian Journal of Computing and Cybernetics Systems), 15(2), 121-130. https://doi.org/10.22146/ijccs.65176

Jiao, L., Geng, X., & Pan, Q. (2019). BPkNN: k-Nearest Neighbor Classifier with Pairwise Distance Metrics and Belief Function Theory. IEEE Access, 7, 48935-48947. https://doi.org/10.1109/ACCESS.2019.2909752.

Kashyap, A. A., Raviraj, S., Devarakonda, A., Nayak K, S. R., K V, S., Bhat, S. J., & Galatioto, F. (2022). Traffic flow prediction models – A review of deep learning techniques. Cogent Engineering, 9(1). DOI: 10.1080/23311916.2021.2010510.

Kaushik, P. (2023). Congestion Articulation Control Using Machine Learning Technique. Amity Journal of Professional Practices, 3(01).
doi:10.55054/ajpp. v3i01.631.

Kelleher, J. D., Mac Namee, B., & D'Arcy, A. (2020). Fundamentals of Machine Learning for Predictive Data Analytics, Second Edition: Algorithms, Worked Examples, and Case Studies. MIT Press. ISBN-13: 978-0262361101

Khan, S. M., Dey, K. C., & Chowdhury, M. (2017). Real-Time Traffic State Estimation with Connected Vehicles. IEEE Transactions on Intelligent Transportation Systems, 18(7), 1687-1699. https://doi.org/10.1109/TITS.2017.2658664.

Kodupuganti, S. R., & Pulugurtha, S. S. (2019). Link-level travel time measures-based level of service thresholds by the posted speed limit. Transportation Research Interdisciplinary Perspectives, 3, 100068. https://doi.org/10.1016/j.trip.2019.100068.

Kulkarni, S., & Rao, G. (2010). Mobility and Traffic Model Analysis for Vehicular Ad-hoc Networks. In M. Watfa (Ed.), Advances in Vehicular Ad-Hoc Networks: Developments and Challenges (pp. 214-232). IGI Global. https://doi.org/10.4018/978-1-61520-913-2.ch011

Kumar, M., Gupta, S., Kumar, K., & Sachdeva, M. (2020). Spreading of COVID-19 in India, Italy, Japan, Spain, UK, US: A Prediction Using ARIMA and LSTM Model. Association for Computing Machinery, 1(4). https://doi.org/10.1145/3411760

Kumar, S. R., Gayathri, N., Muthuramalingam, S., Balamurugan, B., Ramesh, C., & Nallakaruppan, M. K. (2019). Medical Big Data Mining and Processing in e-Healthcare. In V. E. Balas, L. H. Son, S. Jha, M. Khari, and R. Kumar (Eds.), Internet of Things in Biomedical Engineering (pp. 323-339). Academic Press. ISBN 978-0128173565. https://doi.org/10.1016/B978-0-12-817356-5.00016-4.

Kurani, A., Doshi, P., Vakharia, A., & others. (2023). A Comprehensive Comparative Study of Artificial Neural Network (ANN) and Support

Vector Machines (SVM) on Stock Forecasting. Annals of Data Science, 10, 183–208. https://doi.org/10.1007/s40745-021-00344-x

Kyriacou, V., Englezou, Y., Panayiotou, C. G., & Timotheou, S. (2023). Bayesian Traffic State Estimation Using Extended Floating Car Data. IEEE Transactions on Intelligent Transportation Systems, 24(2), 1518-1532. doi:10.1109/TITS.2022.3225057.

Lee, M. C. (2022). Research on the Feasibility of Applying GRU and Attention Mechanism Combined with Technical Indicators in Stock Trading Strategies. Applied Sciences, 12(3), 1007. https://doi.org/10.3390/app12031007

Liu, Z., Qin, X., Huang, W., Zhu, X., Wei, Y., Cao, J., & Guo, J. (2019). Effect of Time Intervals on K-nearest Neighbors Model for Short-term Traffic Flow Prediction. Promet - Traffic and Transportation, 31(2), 129-139. https://doi.org/10.7307/ptt.v31i2.2811

Mak, D. K. (2021). Simple Moving Average. In D. K. Mak (Ed.), Trading Tactics in the Financial Market: Mathematical Methods to Improve Performance (pp. 29-55).

Mahdiani, M. R., Khamehchi, E., Hajirezaie, S., et al. (2020). Modeling viscosity of crude oil using k-nearest neighbor algorithm. Advances in Geo-Energy Research, 4(4), 435-447. https://doi.org/10.46690/ager.2020.04.08

Maryland Department of Transportation. (2013). Maryland 511. https://chart.maryland.gov (Accessed 27 May 2023).

Mchergui, A., Moulahi, T., & Zeadally, S. (2022). Survey on Artificial Intelligence (AI) techniques for Vehicular Ad-hoc Networks (VANETs). Vehicular Communications, 34, 100403. https://doi.org/10.1016/j.vehcom.2021.100403.

McHugh, C., Coleman, S., Kerr, D. (2021). Technical indicators for energy market trading. Machine Learning with Applications, 6, 100182. ISSN 2666-8270. https://doi.org/10.1016/j.mlwa.2021.100182.

Muangprathub, J., Intarasit, A., Boongasame, L., & [et al.]. (2020). Portfolio Risk and Return with a New Simple Moving Average of Price Change Ratio. Wireless Personal Communications, 115(3), 3137-3153. doi:10.1007/s11277-020-07374-3.

Naskath, J., Sivakamasundari, G., & Begum, A. A. S. (2023). A study on different deep learning algorithms used in deep neural nets: MLP, SOM, and DBN. Wireless Personal Communications, 128, 2913–2936. https://doi.org/10.1007/s11277-022-10079-4.

Nti, I. K., Adekoya, A. F., & Weyori, B. A. (2020). A systematic review of fundamental and technical analysis of stock market predictions. Artificial Intelligence Review, 53, 3007–3057. doi:10.1007/s10462-019-09754-z

Olivieri, B., & Endler, M. (2017). DADCA: An Efficient Distributed Algorithm for Aerial Data Collection from Wireless Sensor Networks by UAVs. Association for Computing Machinery. https://doi.org/10.1145/3127540.3127553

Özinal Avşar, Y., & Avşar, E. (2022). Short-term traffic state estimation using breakpoint flow calculation and machine learning methods. Zeszyty Naukowe. Transport/Politechnika Śląska, (115), 121-134.

Pavlov, Y. L. (2019). Random Forests. Vsp.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... Duchesnay, É. (2011). Scikit-learn: Machine learning in Python. Journal of Machine Learning Research, 12, 2825-2830.

Pulugurtha, S. S., & Imran, M. S. (2020). Modeling basic freeway section level-of-service based on travel time and reliability. Case Studies on Transport Policy, 8(1), 127-134. https://doi.org/10.1016/j.cstp.2017.08.002.

Rahi, A. A. (2019). Machine Learning Approaches for Traffic Flow Forecasting (Doctoral dissertation). https://doi.org/10.18745/th.22590

Rajalakshmi, V. & Ganesh Vaidyanathan, S. (2022). Hybrid Time-Series Forecasting Models for Traffic Flow Prediction. Promet - TrafficandTransportation, 34 (4), 537-549. https://doi.org/10.7307/ptt.v34i4.3998

Raskar, C., & Nema, S. (2022). Metaheuristic enabled modified hidden Markov model for traffic flow prediction. Computer Networks, 206, 108780. https://doi.org/10.1016/j.comnet.2022.108780.

Rezende, C., Boukerche, A., Ramos, H. S., & Loureiro, A. A. F. (2015). A Reactive and Scalable Unicast Solution for Video Streaming over VANETs. IEEE Transactions on Computers, 64(3), 614-626. doi:10.1109/TC.2014.2308176.

Rezende, C., Ramos, H. S., Pazzi, R. W., Boukerche, A., Frery, A. C., & Loureiro, A. A. F. (2012). VIRTUS: A resilient location-aware video unicast scheme for vehicular networks. In 2012 IEEE International Conference on Communications (ICC) (pp. 698-702). Ottawa, ON, Canada: IEEE. doi: 10.1109/ICC.2012.6364470.

Ritchie, H., & Roser, M. (2020). Urbanization. In Our World in Data. https://ourworldindata.org/urbanization

Robert, N., Gary, M., & Ken, Y. (2018). Numerical Prediction. In Editor(s) of the book (Ed.), Handbook of Statistical Analysis and Data Mining Applications (2nd ed., pp. 187-213). Academic Press.

Rokach, L. (2010). Pattern Classification Using Ensemble Methods (Volume 75 of Series in Machine Perception and Artificial Intelligence). World Scientific. ISBN: 978-9814271073.

Rosenfeld, E., Winston, E., Ravikumar, P., & Kolter, Z. (2020). Certified Robustness to Label-Flipping Attacks via Randomized Smoothing. Proceedings of the 37th International Conference on Machine Learning, Proceedings of Machine Learning Research, 119, 8230-8241. URL: https://proceedings.mlr.press/v119/rosenfeld20b.html.

Saxena, S. (2019, Month Day). What's the difference between RMSE and RMSLE? Analytics Vidhya. Retrieved from https://medium.com/analytics-vidhya/root-mean-square-log-error-rmse-vs-rmlse-935c6cc1802a

Sekuła, P., Marković, N., Vander Laan, Z., & Farokhi Sadabadi, K. (2018). Estimating historical hourly traffic volumes via machine learning and vehicle probe data: A Maryland case study. Transportation Research Part C: Emerging Technologies, 97, 147-158. https://doi.org/10.1016/j.trc.2018.10.012.

Siddiqui, A. J., Mammeri, A., & Boukerche, A. (2016). Real-Time Vehicle Make and Model Recognition Based on a Bag of SURF Features. IEEE Transactions on Intelligent Transportation Systems, 17(11), 3205-3219. https://doi.org/10.1109/TITS.2016.2545640.

Singh, V., Gore, N., Chepuri, A., Arkatkar, S., Joshi, G., and Pulugurtha, S. (2019). Examining travel time variability and reliability on an urban arterial road using Wi-Fi detections - A case study. Journal of the Eastern Asia Society for Transportation Studies, 13, 2390-2411. https://doi.org/10.11175/easts.13.2390

Soleymani, F., & Paquet, E. (2020). Financial portfolio optimization with online deep reinforcement learning and restricted stacked autoencoder—DeepBreath. Expert Systems with Applications, 156, 113456. https://doi.org/10.1016/j.eswa.2020.113456.

Sun, P., AlJeri, N., & Boukerche, A. (2020). DACON: A Novel Traffic Prediction and Data-Highway-Assisted Content Delivery Protocol for Intelligent Vehicular Networks. IEEE Transactions on Sustainable Computing, 5(4), 501-513. doi:10.1109/TSUSC.2020.2971628.

Sun, P., Aljeri, N., & Boukerche, A. (2020). Machine Learning-Based Models for Real-time Traffic Flow Prediction in Vehicular Networks. IEEE Network, 34(3), 178-185. doi: 10.1109/MNET.011.1900338.

Tišljarić, L., Vrbanić, F., Ivanjko, E., & Carić, T. (2022). Motorway Bottleneck Probability Estimation in Connected Vehicles Environment Using Speed Transition Matrices. Sensors, 22(7), 2807. https://doi.org/10.3390/s22072807

Turki, A. I., & Hasson, S. T. (2022). A Markov Chain Approach to Model Vehicle Traffic Behavior. In 2022 International Conference of Science and Information Technology in Smart Administration (ICSINTESA) (pp. 117-122). Denpasar, Bali, Indonesia. doi:10.1109/ICSINTESA56431.2022.10041552.

Turki, A. I., & Hasson, S. T. (2023). Study estimating hourly traffic flow using artificial neural network: A M25 motorway case. Samarra Journal of Pure and Applied Science, 5(1), 47-59. https://doi.org/10.54153/sjpas.2023.v5i1.448

Highways England. (2023, January 1). Highways England Network Journey Time and Traffic Flow Data. data.gov.uk. https://data.gov.uk/dataset/highways-england-network-journey-time-and-traffic-flow-data

UK Department for Transport. (2023). Road Accidents Safety Data. Data.gov.uk. https://data.gov.uk/dataset/road-accidents-safety-data/resource/91789e37-03e5-48cf-9720-2d13639c32b9

UK Government. (2022). Road traffic accidents. Retrieved October 1, 2022, from https://data.gov.uk/dataset/road-traffic-accidents

Vidales, A. (2019). Machine Learning with Matlab. Supervised Learning: Knn Classifiers, Ensemble Learning, Random Forest, Boosting and Bagging. Amazon Digital Services LLC - KDP Print US. ISBN: 1796495697, 9781796495690.

Vrbanić, F., Tišljarić, L., Majstorović, Ž., & Ivanjko, E. (2022). Reinforcement Learning Based Variable Speed Limit Control for Mixed Traffic Flows Using Speed Transition Matrices for State Estimation. In 2022 30th Mediterranean Conference on Control and Automation (MED) (pp. 1093-1098). Vouliagmeni, Greece. doi: 10.1109/MED54222.2022.9837279.

Vrbanić, F., Tišljarić, L., Majstorović, Ž., & Ivanjko, E. (2023). Reinforcement Learning-Based Dynamic Zone Placement Variable Speed Limit Control for Mixed Traffic Flows Using Speed Transition Matrices for State Estimation. Machines, 11(4), 479. https://doi.org/10.3390/machines11040479

Wang, Y., Wu, B., & Li, L. (2022). Urban Redevelopment and Traffic Congestion Management Strategies. Urban Sustainability Series. Springer Singapore. https://doi.org/10.1007/978-981-19-1727-1

Waze. (2023, May 27). About Waze. [Website]. https://www.waze.com/about

Wilby, M. R., González, A. B. R., Pozo, R. F., & Vinagre Díaz, J. J. (2022). Short-Term Prediction of Level of Service in Highways Based on Bluetooth Identification. IEEE Transactions on Intelligent Transportation Systems, 23(1), 142-151. https://doi.org/10.1109/TITS.2020.3008408.

Younes, M. B., & Boukerche, A. (2013). Efficient traffic congestion detection protocol for next-generation VANETs. In 2013 IEEE International Conference on Communications (ICC) (pp. 3764-3768). doi: 10.1109/ICC.2013.6655141.

Younes, M. B., & Boukerche, A. (2018). An efficient dynamic traffic light scheduling algorithm considering emergency vehicles for intelligent transportation systems. Wireless Networks, 24, 2451–2463. https://doi.org/10.1007/s11276-017-1482-5

Zhang, Q., Chang, W., Yin, C., Xiao, P., Li, K., & Tan, M. (2023). Attention-Based Spatial–Temporal Convolution Gated Recurrent Unit for Traffic Flow Forecasting. Entropy, 25(6), 938. https://doi.org/10.3390/e25060938

Zheng, H., Lin, F., Feng, X., & Chen, Y. (2021). A Hybrid Deep Learning Model with Attention-Based Conv-LSTM Networks for Short-Term Traffic Flow Prediction. IEEE Transactions on Intelligent Transportation Systems, 22(11), 6910-6920. https://doi.org/10.1109/TITS.2020.2997352.

Units, T. M. (2018). National Traffic Information Service DATEX II Service.

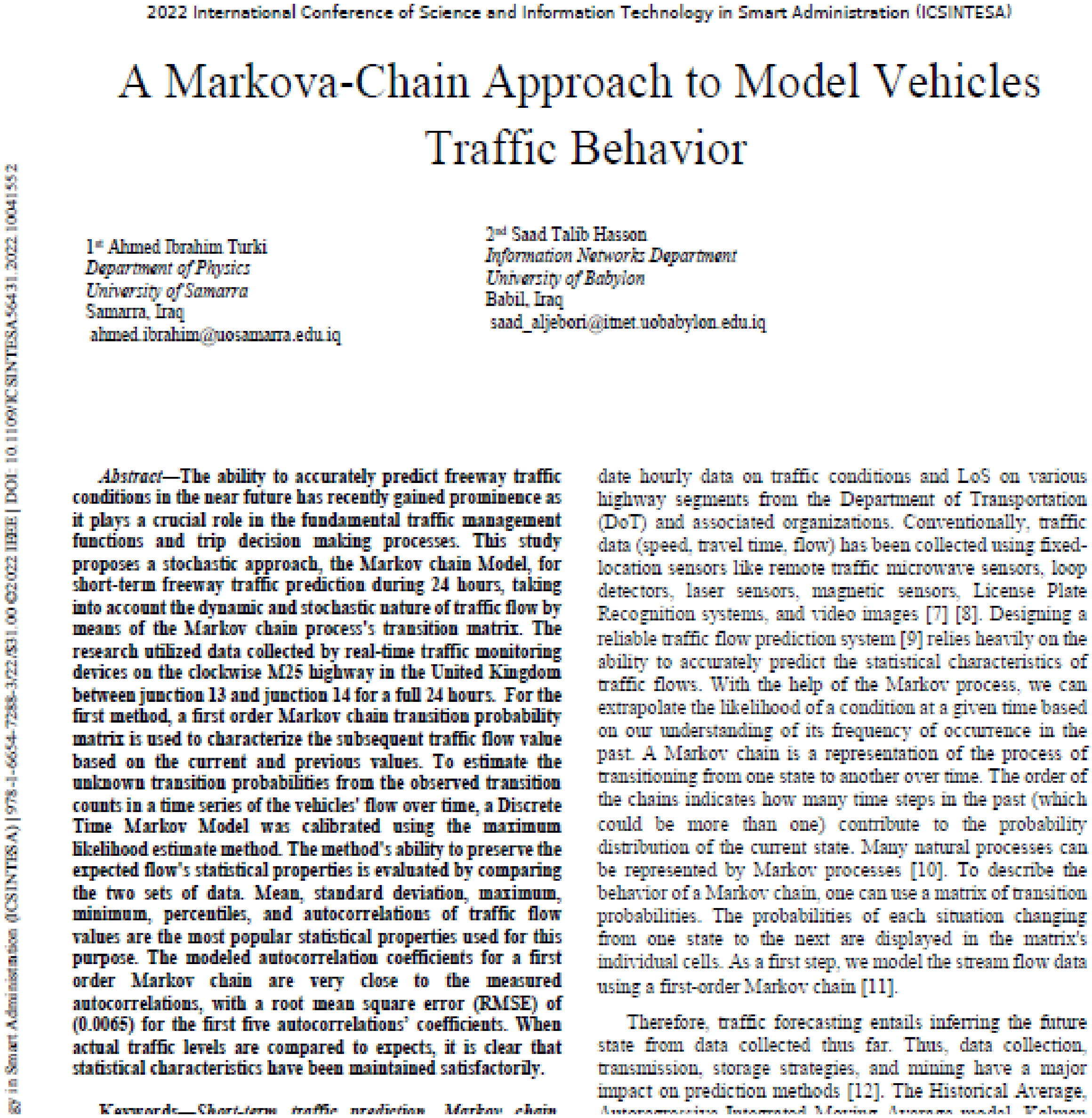
Appendix A: Hyperparameters Tuning

Table A1 displays a compilation of carefully chosen model hyperparameters that have been identified as optimal. It should be noted that scikit-learn (Pedregosa et al., 2011) is utilized for executing linear and ensemble models, while Keras (Chollet, 2015) is employed for running deep models. Consequently, unless explicitly indicated in Table A1, the default model hyperparameters of either scikit or Keras are employed.

Table A1: Presents the hyperparameters that have been selected for the model. The optimization of benchmark model performance is focused on selecting these variables. Interested readers are referred to (Pedregosa, et al., 2011), (Chollet, 2015) for a comprehensive specification and description of the hyperparameters of the models.

Model	**Hyperparameters**
MLR	-
RFR	n_estimators: 100, criterion: squared_error
FFNN	neuron1: 6, neuron2: 3, learning rate: 0.001, loss:MSE, optimizer: SGD, metrics: ['mae', 'mse', 'mape'], Hidden layer activation function: tanh, Output activation function: linear, EPOCHS : 300, batch_size = 32,
RF Classifier	n_estimators: 100, max_depth: None, oob_score: True, criterion: gini, random_state: 42
KNN	n_neighbors: 5, metric: 'minkowski', p: 2

Appendix B: Published Work



2022 International Conference of Science and Information Technology in Smart Administration (ICSINTESA)

A Markova-Chain Approach to Model Vehicles Traffic Behavior

1st Ahmed Ibrahim Turki
Department of Physics
University of Samarra
Samarra, Iraq
ahmed.ibrahim@uosamarra.edu.iq

2nd Saad Talib Hasson
Information Networks Department
University of Babylon
Babil, Iraq
saad_aljebori@itnet.uobabylon.edu.iq

gy in Smart Administration (ICSINTESA) | 978-1-6654-7288-3/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICSINTESA56431.2022.10041552

***Abstract*—The ability to accurately predict freeway traffic conditions in the near future has recently gained prominence as it plays a crucial role in the fundamental traffic management functions and trip decision making processes. This study proposes a stochastic approach, the Markov chain Model, for short-term freeway traffic prediction during 24 hours, taking into account the dynamic and stochastic nature of traffic flow by means of the Markov chain process's transition matrix. The research utilized data collected by real-time traffic monitoring devices on the clockwise M25 highway in the United Kingdom between junction 13 and junction 14 for a full 24 hours. For the first method, a first order Markov chain transition probability matrix is used to characterize the subsequent traffic flow value based on the current and previous values. To estimate the unknown transition probabilities from the observed transition counts in a time series of the vehicles' flow over time, a Discrete Time Markov Model was calibrated using the maximum likelihood estimate method. The method's ability to preserve the expected flow's statistical properties is evaluated by comparing the two sets of data. Mean, standard deviation, maximum, minimum, percentiles, and autocorrelations of traffic flow values are the most popular statistical properties used for this purpose. The modeled autocorrelation coefficients for a first order Markov chain are very close to the measured autocorrelations, with a root mean square error (RMSE) of (0.0065) for the first five autocorrelations' coefficients. When actual traffic levels are compared to expects, it is clear that statistical characteristics have been maintained satisfactorily.**

Keywords—*Short-term traffic prediction, Markov chain*

date hourly data on traffic conditions and LoS on various highway segments from the Department of Transportation (DoT) and associated organizations. Conventionally, traffic data (speed, travel time, flow) has been collected using fixed-location sensors like remote traffic microwave sensors, loop detectors, laser sensors, magnetic sensors, License Plate Recognition systems, and video images [7] [8]. Designing a reliable traffic flow prediction system [9] relies heavily on the ability to accurately predict the statistical characteristics of traffic flows. With the help of the Markov process, we can extrapolate the likelihood of a condition at a given time based on our understanding of its frequency of occurrence in the past. A Markov chain is a representation of the process of transitioning from one state to another over time. The order of the chains indicates how many time steps in the past (which could be more than one) contribute to the probability distribution of the current state. Many natural processes can be represented by Markov processes [10]. To describe the behavior of a Markov chain, one can use a matrix of transition probabilities. The probabilities of each situation changing from one state to the next are displayed in the matrix's individual cells. As a first step, we model the stream flow data using a first-order Markov chain [11].

Therefore, traffic forecasting entails inferring the future state from data collected thus far. Thus, data collection, transmission, storage strategies, and mining have a major impact on prediction methods [12]. The Historical Average, Autoregressive Integrated Moving Average model, Kalman

Tikrit Journal of Pure Science Vol. 28 (3) 2023

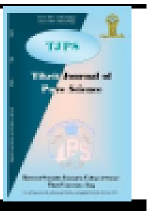
TJPS

https://doi.org/10.25130/tjps.v28i3.1428

Tikrit Journal of Pure Science

ISSN: 1813 – 1662 (Print) --- E-ISSN: 2415 – 1726 (Online)

Journal Homepage: http://tjps.tu.edu.iq/index.php/j



Using a new algorithm in Machine learning Approaches to estimate level-of-service in hourly traffic flow data in vehicular ad hoc networks

Ahmed Ibrahim Turki[1], Saad Talib Hasson[2]

[1]*Department of Physics, College of Education, University of Samarra, Samarra, Iraq*

[2]*Information Networks Department, University of Babylon, Babylon, Iraq*

ARTICLE INFO.

Article history:

-Received:	13 / 12 / 2022
-Received in revised form:	22 / 2 / 2023
-Accepted:	14 / 3 / 2023
-Final Proofreading:	15 / 5 / 2023
-Available online:	25 / 6 / 2023

Keywords: Traffic flow data; Level-of-Service; Random Forest; SVM, KNN, VANET

Corresponding Author:

Name: Ahmed Ibrahim Turki

E-mail:

ahmed.ibrahim@uosamarra.edu.iq

saad_aljebori@itnet.uobabylon.edu.iq

Tel:

ABSTRACT

The primary goals of transportation agencies and researchers studying traffic operations are to ease traffic and increase road safety through the use of vehicular ad hoc networks. Agencies can't achieve their goals without reliable and consistent data on the current traffic situation. The Level-of-Service (LOS) index is a helpful measure of freeway traffic operations. Conventional fixed-location cameras and sensors are impractical and expensive for gathering reliable traffic density data on every road in large networks. Flow data is a new, low-cost option that has the potential to boost safety and operations. This study proposes an algorithm for hourly LOS assessment by incorporating flow data provided by the MIDAS (Motorway Incident Detection and Automatic Signaling) system. The proposed algorithm uses machine learning techniques to classify LOS data based on the flow of traffic. The input features that are subject to prediction are a group of technical indicators. The real-world LOS was determined by analyzing data from stationary sensors. The outcomes demonstrate that technical indicators can be utilized to enhance the accuracy of LOS estimation (Random Forest= 93.1, k-nearest neighbors = 92.5, and Support Vector Machine = 91.4). The current work introduces a novel approach to the selection of technical indicators and their use as features, which allows for highly accurate short-term prediction of LOS estimation.

Appendix C: Technical Indicators

As a case study, one day data is collected and represented by the technical indicator's equations in Section 3.2.3 to show their utility and their implementation graphically.

- ## *Average True Range (ATR):*

  The trend indicator is utilized to quantify the level of volatility within a given density. Figure A1 illustrates those densities with high volatility exhibit a corresponding increase in the ATR, while the opposite holds true for the time period under consideration (one day).

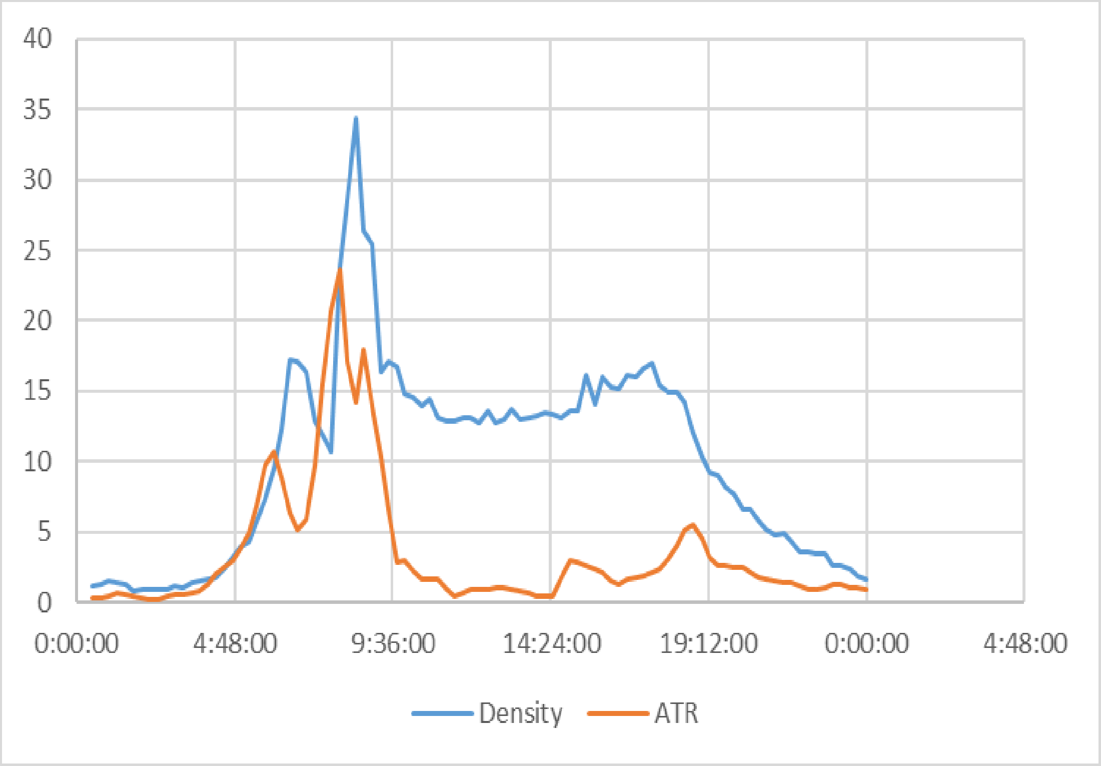


Figure A1: ATR for one day period

- ## *Simple Moving Average (SMA):*

  This method is employed to ascertain the direction of vehicle density, whether it is increasing or decreasing. When the SMA exhibits an ascending trend, it signifies that the density is moving in an upward direction, and conversely, when the SMA is in a descending state, it suggests that the density is moving in a downward direction. The occurrence of a bullish signal is observed when the density of compounds surpasses the SMA, and conversely, a bearish signal is identified when the density falls below the SMA. SMA offers valuable insights into the fluctuations of traffic density, particularly in the short term. This is due to its heightened responsiveness to short-term changes, as depicted in Figure A2.

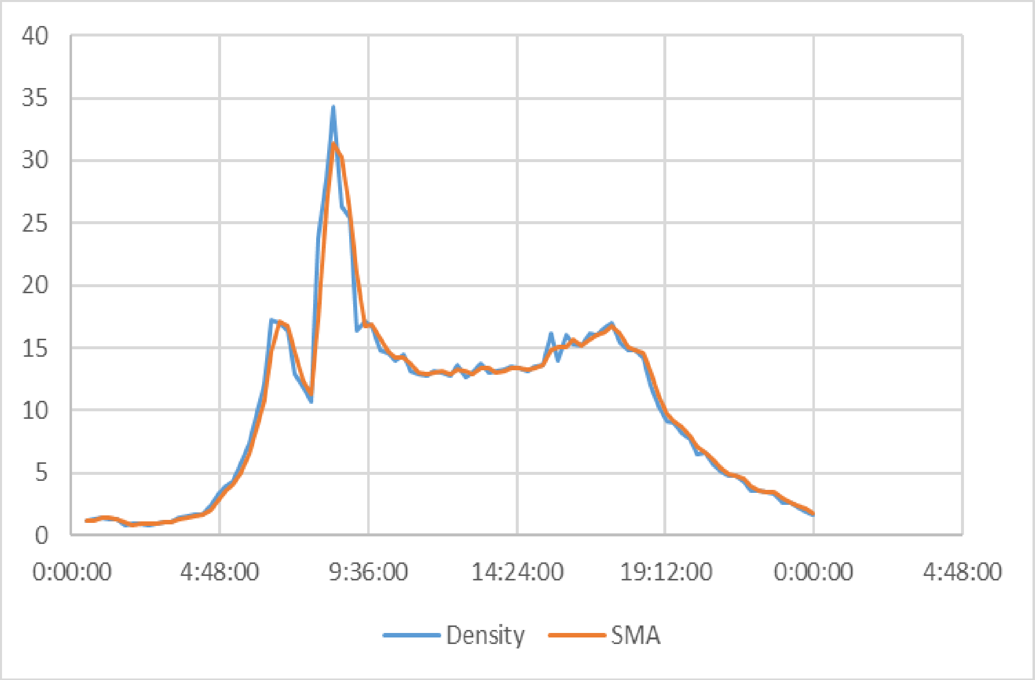


Figure A2: SMA for one day period

- ***Exponential Moving Average (EMA):***

The EMA technique is employed to enhance the smoothness of density measurements by mitigating the impact of random density fluctuations. This is achieved by calculating the average density over a specific time interval. This approach bears resemblance to the SMA, yet places greater emphasis on more recent observations. The response time to changes in density is observed to be faster in the present dissertation, as depicted in Figure A3, when compared to SMA.

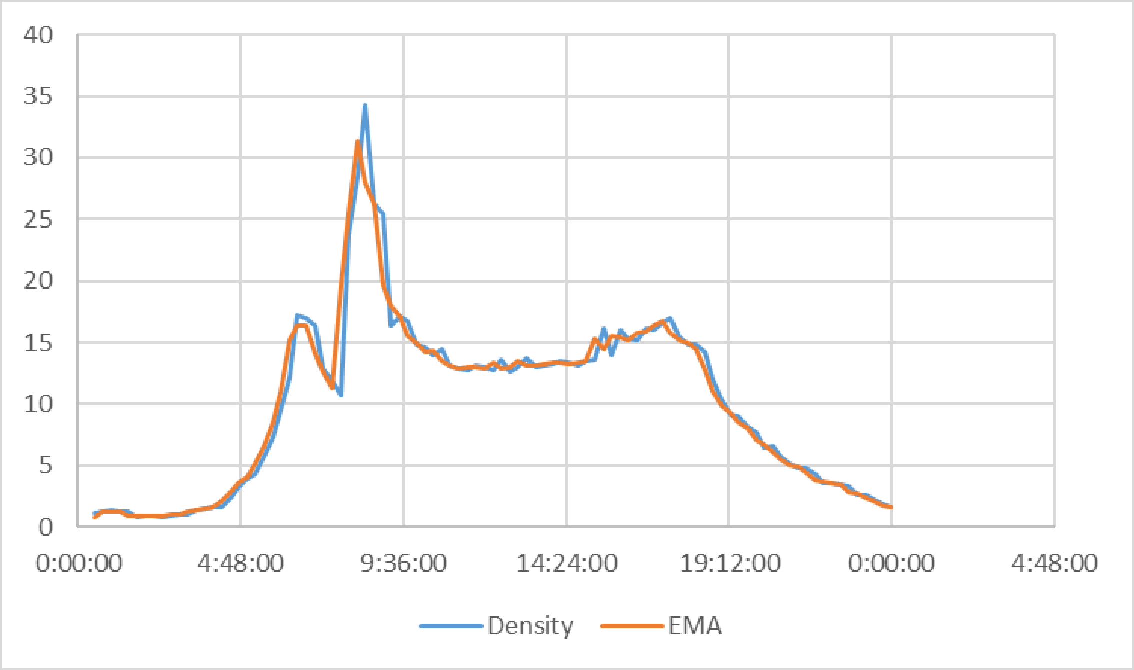


Figure A3: EMA for one day period

- ***Relative Strength Index (RSI):***

The RSI is a widely utilized momentum oscillator within the field of technical analysis, employed to assess the velocity and magnitude of price fluctuations. Differences between the RSI and the density action can also indicate potential changes in momentum, as depicted in Figure A4.

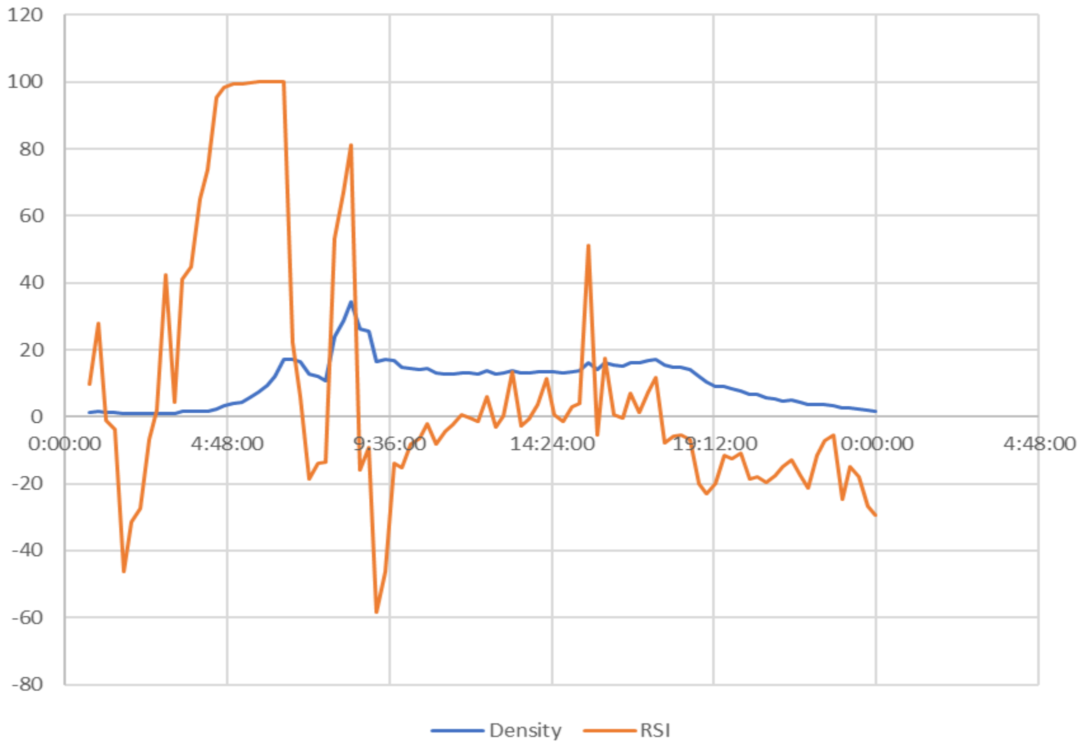


Figure A4: RSI for one day period

- ***Rate of change (ROC):***

The ROC is an oscillator, comparable to the MOM indicator, that expresses change as a percentage instead of an absolute value. It is opposite to the zero line that distinguishes positive and negative values. Positive values indicate an upward trend for the vehicle's density. Zero-line intersections can be used to signal changes in the trend of density up or down as shown in figure A5.

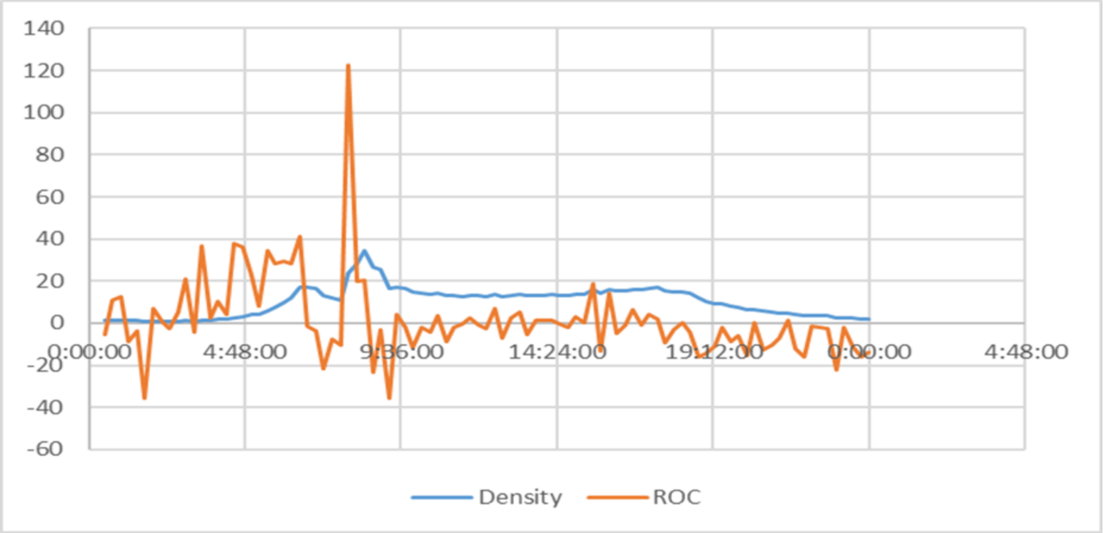


Figure A5: ROC for one day period

- ***Momentum (MOM):***

The instrument was employed to quantify the rate of variation in vehicle density within a brief timeframe. This analysis offers valuable insights into the fluctuations in vehicle density, whether it is on the rise or decline. The momentum line exhibits an upward trajectory above the zero line, signifying an increase in intensity, and conversely, a downward trend is observed, as depicted in Figure A6.

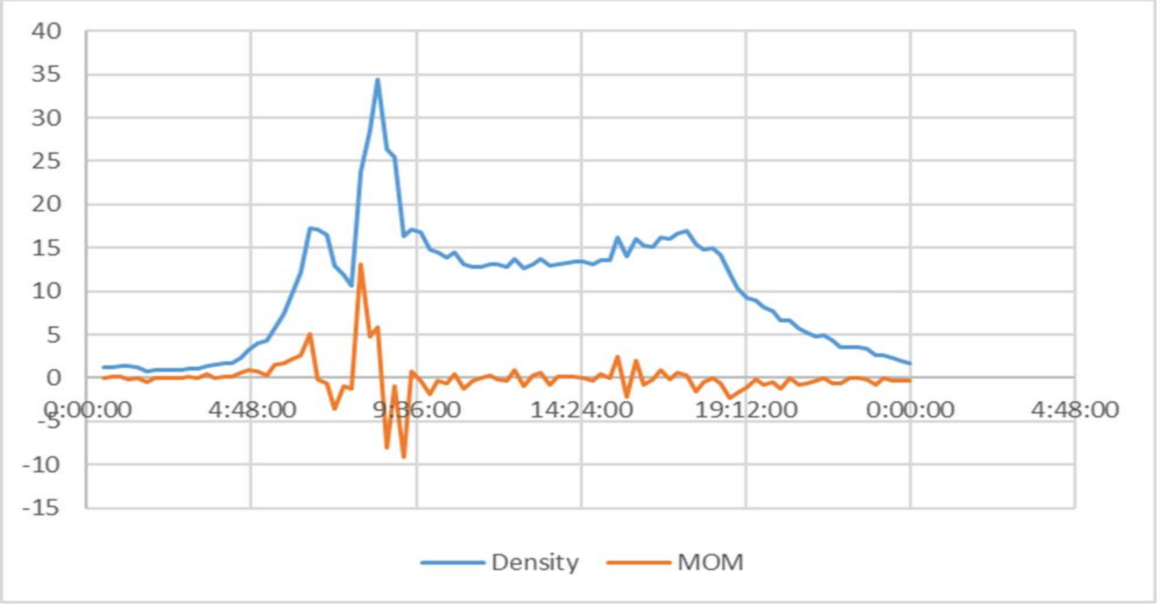


Figure A6: MOM for one day period

Appendix D: Sub-models of FFNN

The construction of the artificial neural network model developed in this thesis includes the development of a set of sub models:

- **Model Inputs and Outputs**

The selection of variables in the input and output models holds significant significance as it plays a crucial role in enhancing the performance of neural networks. The inclusion of additional input and output variables substantially impacts the size of the neural network, resulting in a slower learning process and reduced efficiency of the network. The selection of the number of variables in the input and output models was determined using the Method of Priori Knowledge, as employed in this dissertation. The utilization of this approach is prevalent in scholarly investigations and empirical inquiries. It is particularly applicable in situations where there is a lack of prior knowledge regarding the input variables and their influence on the output variables. Consequently, the input model incorporates independent variables such as ATR, SMA, EMA, RSI, ROC and MOM while the output model encompasses the dependent variable, Density. These two models serve as mechanisms for the storage and documentation of data pertaining to prior studies, with the added capability of being regularly updated.

- **Data Division Model**

The data in a neural network can be categorized as either continuous variables or discrete variables, and it is further classified into three primary groups:

1- A dataset utilized for constructing a neural network model.
2- The evaluation dataset for the neural network model.
3- A validation set, which is independent, is used to estimate the performance of the model in the relevant environment.

The training set is utilized for establishing the interconnected weights within the neural network. The utilization of a test group is employed to assess the efficacy of the network across different educational stages, with the cessation of training occurring upon the observation of an increase in error within the test group. The purpose of the validation set is to assess the model's performance after it has undergone the training process for the neural network. Hence, the process of categorizing the data into the aforementioned three groups holds significant importance in the context of neural network modeling. The present dissertation employed the Statistically Consistent Method to partition the data into three distinct groups, namely the training set, the test set, and the validation set. This approach guarantees a statistical alignment of the data for each group, thereby eliminating any potential bias in stratifying the data within each set through the utilization of the T-test. Through The utilization of statistical measures, namely the arithmetic mean, standard deviation, and range. One of the advantages of this approach is its utilization of the Trial-and-Error Method in order to achieve optimal data partitioning.

Upon examination of Tables A.2, A.3, and A.4, which pertain to normal working days, weekends, and holidays, respectively, it becomes evident that the data division percentages for the training, examination, and investigation groups were determined through the utilization of the trial-and-error method. The researcher employed varying proportions of data allocation for these groups in an endeavor to achieve optimal performance of the neural network, as indicated by attaining the highest coefficient correlation. This correlation serves as an indicator of the strength of the association between the neural network's output (predicted density) and the measured density (ground truth). Simultaneously, the researcher aimed to minimize the testing error rate. The selection of the optimal data partition is based on the utilization of these two criteria in the present dissertation.

Based on the findings presented in Table A.2, it is evident that the optimal allocation of data for regular working days entails assigning 76% to the training set, 15% to the test set, and 9% to the validation set. This allocation is determined based on the criteria of minimizing the test error rate (1%) and maximizing the correlation coefficient (80%). Regarding the weekend days, the optimal data allocation can be observed in Table A.3, wherein the training set comprises 72%, the test set comprises 13%, and the validation group comprises 15% of the total data. This allocation is determined based on the criteria of minimizing the test error rate (1%) and maximizing the correlation coefficient (90%). The distribution (68%, 15%, 17%) of the data was not selected as the optimal distribution, despite the fact that the correlation coefficient is higher and stands at (81%). Nevertheless, the disparity in the correlation coefficients' values is deemed negligible or inconspicuous. Conversely, the test error rate for the division (68%, 15%, 17%) pertaining to a data volume of (7%) is regarded as higher when compared to the division selected as the optimal division for the data. Finally, during holiday periods, the optimal data division can be observed in Table A.4, wherein the training set comprises 62% of the data, the test set comprises 21%, and the validation set comprises 17%. This division is determined based on the criteria of achieving the lowest test error percentage (1%) and the highest correlation coefficient (91%). The chosen division of the data, which accounts for 71%, 17%, and 11% respectively, was not deemed optimal despite a higher correlation coefficient of 97%. However, the disparity between the correlation coefficient values is relatively minor. Conversely, the test error percentage for the chosen division, amounting to 12% of the data, is significantly lower compared to the alternative division.

Table A.2: The impact of data partitioning on the efficacy of neural network models during normal working days.

Data Division %			**Coefficient correlation(r)%**	**Testing error %**
Training %	Testing %	Validation %		
80	5	15	70	3
75	10	15	75	2
72	14	14	73	2
68	15	17	68	1
64	22	14	71.2	2
60	23	17	60	2
60	20	20	66	1
68	12	20	76	1
68	20	12	63	2
71	11	17	73	2
71	17	11	71	1
76	15	9	80	1

Table A.3: The impact of data partitioning on the efficacy of neural network models during weekend days.

Data Division %			**Coefficient correlation(r)%**	**Testing error %**
Training %	Testing %	Validation %		
80	5	15	67	1
75	10	15	77	7
72	14	14	84	1
68	15	17	81	7
64	22	14	74	2
60	23	17	70	1
60	20	20	85	1
68	12	20	66	1
68	20	12	60	4
72	11	15	90	1
71	17	11	83	2
76	15	9	87	1

Table A.4: The impact of data partitioning on the efficacy of neural network models during holiday days.

Data Division %			**Coefficient correlation(r)%**	**Testing error %**
Training %	Testing %	Validation %		
80	5	15	77	16
75	10	15	66	13
72	14	14	79	6
68	15	17	64	3
64	22	14	78	4
62	21	17	91	1
60	20	20	77	1
68	12	20	78	3
68	20	12	68	1
71	11	17	80	5
71	17	11	97	12
76	15	9	72	1

- **FFNN Architecture Model**

The architecture of artificial neural networks refers to the arrangement and connectivity patterns of neurons, which collectively form a network. The selection of an optimal number of neural nodes in the intermediate layer of a neural network is widely acknowledged as a crucial determinant of the network's efficacy. It is worth noting that the number of nodes in the input layer corresponds to the number of factors influencing the density calculation, specifically indicators: ATR, SMA, EMA, RSI, ROC and MOM. The final layer in a neural network, known as the output layer, is responsible for producing the desired output or prediction based on the structure consists of a single neuron node, specifically representing the measured density. There exist numerous techniques for determining the ideal number of neural nodes in neural networks. The most effective approach involves employing equation (A.1), which entails initially selecting a single node in the middle layer and subsequently incrementally increasing the number of nodes until optimal network performance is attained. The maximum number of nodes was determined to be (1 + 2I) as stated in Equation A.1.

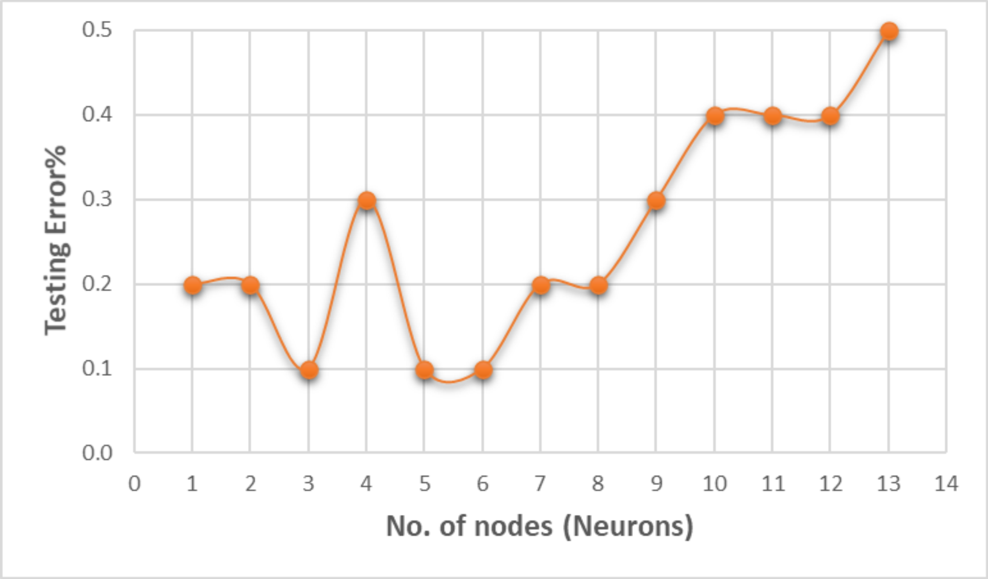
$$Max.no.of\ node = 1 + 2 * Input\ factors \qquad \text{(A.1)}$$

The default parameters employed in this dissertation's program include a learning rate set at 0.4. The transfer function for the output layer is linear, while the hidden middle layer utilizes the hyperbolic tangent (tanh) function. The activation functions under consideration in this dissertation are presented in Equations A.2 and A.3, as shown below.

$$\Phi(x) = x \qquad \text{(A.2)}$$

$$\Phi_x = \frac{1-e^{-2x}}{1+e^{-2x}}\ , \Phi(x) \in [-1, +1] \qquad \text{(A.3)}$$

Based on the analysis of figures A.8, A.9, and A.10, it is evident that there exists a distinct variation in the error rate observed in the test set. Notably, the optimal performance of the neural network is consistently achieved when the number of nodes in the hidden layer is set to three across all models. This is attributed to its possession of the highest correlation coefficient (90%) and the lowest test error rate (0.1%) during regular business days. During holiday periods, the observed correlation coefficient reaches a maximum value of 83%, while the corresponding test error rate is as low as 0.1%. These results are obtained when the neural network model consists of three nodes in the hidden layer. During the weekend model, the three nodes exhibited the highest correlation coefficient of 85% and the lowest test error rate of 0.1%. The network architecture developed in this dissertation consists of three neural layers, namely the input layer, hidden layer, and output layer. The process involves the transmission of information from the input layer to the hidden layer, followed by the transmission from the hidden layer to the output layer. The primary locations for data processing are the hidden layer and the output layer.



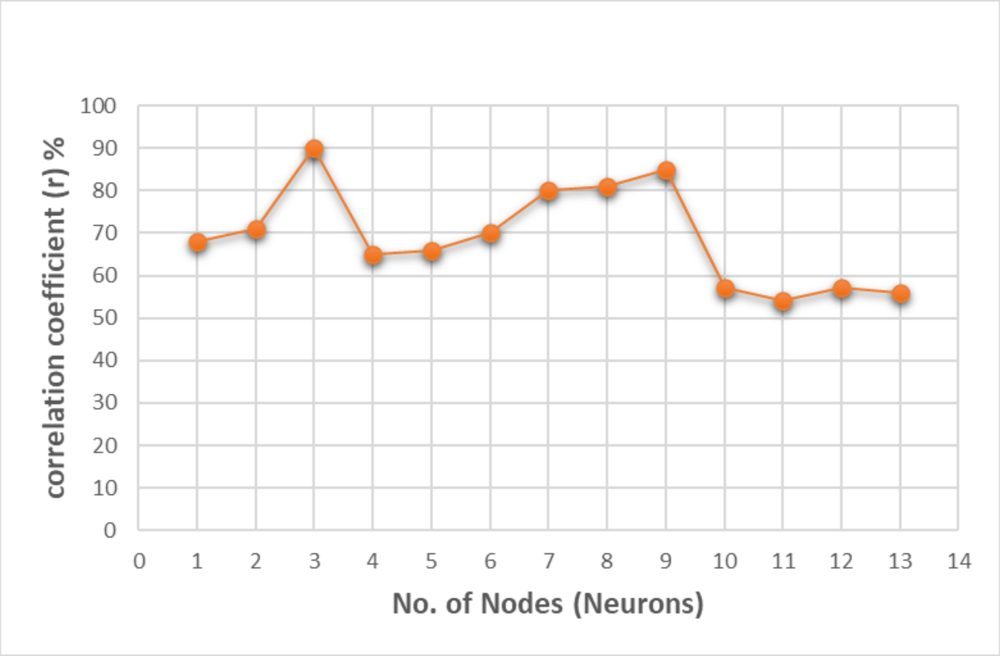
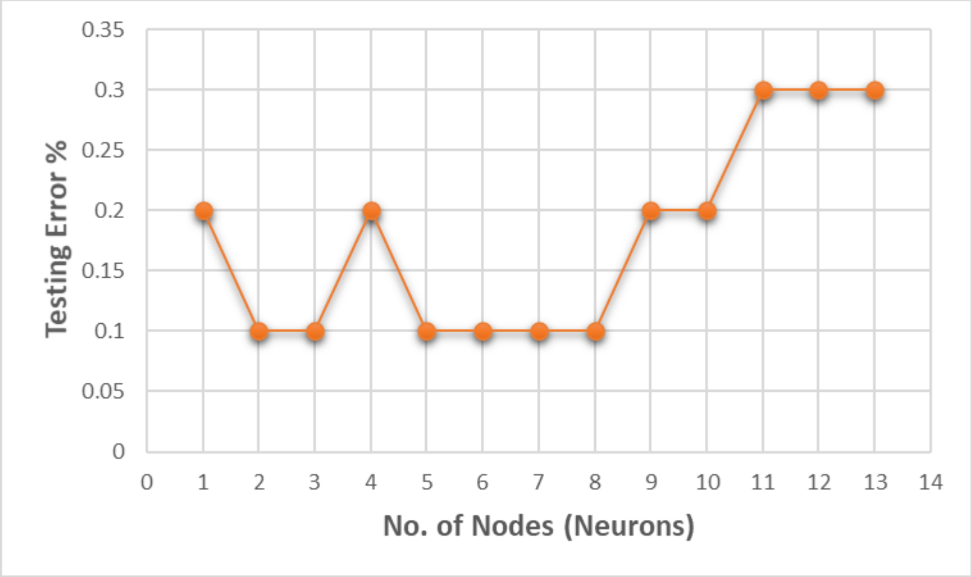


Figure A.7: The performance of neural network model was evaluated by varying the number of nodes (neurons) in the hidden layer specifically for normal working days.



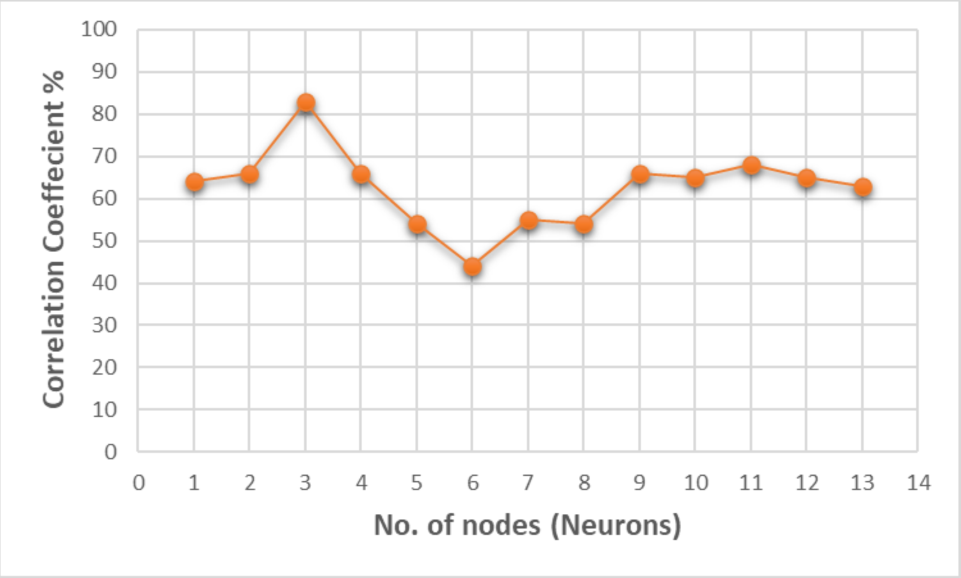
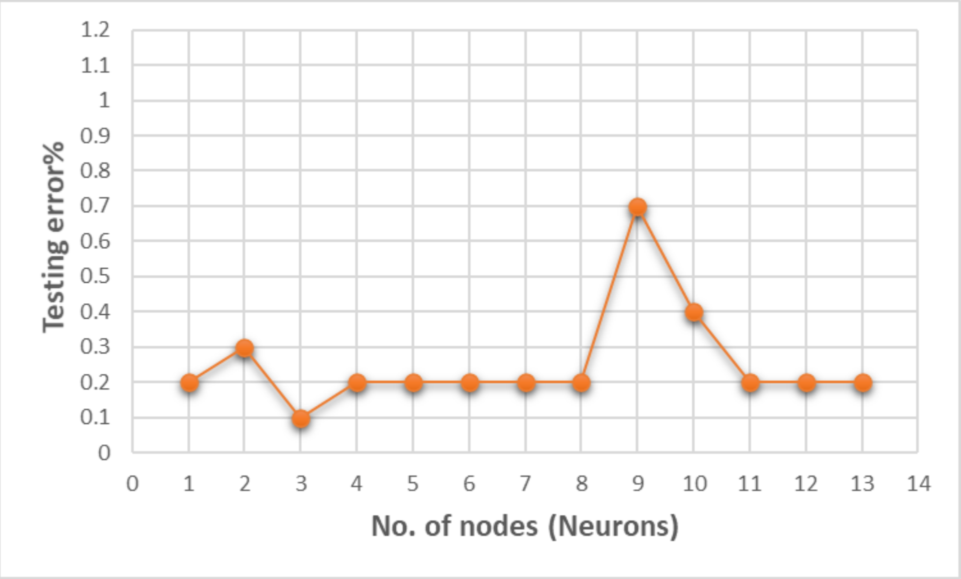


Figure A.8: The performance of neural network model was evaluated by varying the number of nodes (neurons) in the hidden layer specifically for holiday days.



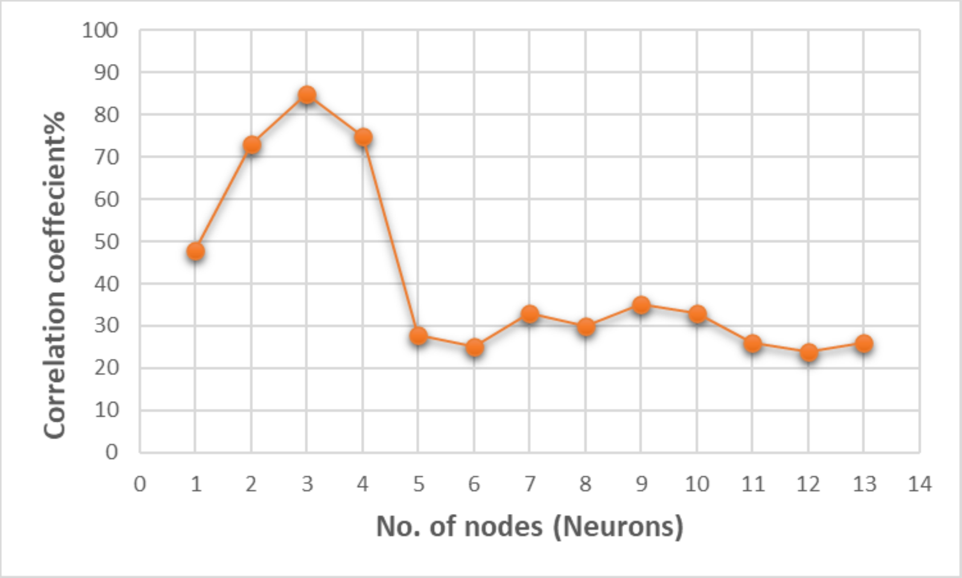


Figure A.9: The performance of neural network model was evaluated by varying the number of nodes (neurons) in the hidden layer specifically for weekend days.

- **Learning Rate Model**

During the training phase, the neural network has the ability to optimize the bias and link values for each direction in order to compute a more precise output, utilizing the learning rate. Increasing the learning rate can lead to network instability, such as oscillatory behavior. The learning rate parameter is commonly assigned a small positive value that is less than 1. In order to assess the impact of the learning rate on the model's performance. A series of experiments were conducted to validate the impact of the learning rate. The figures A.11, A.12 and A.13 demonstrate that the optimal learning rate, determined to be 0.001, exhibits the lowest error rates for the test set (0.1%, 0.1%, 0.1%) and the highest correlation coefficients (80%, 82%, 70%) for regular working days, holidays, and weekend days, respectively. These findings indicate that the network's performance improves as the learning rate approaches its minimum value of 0.001.

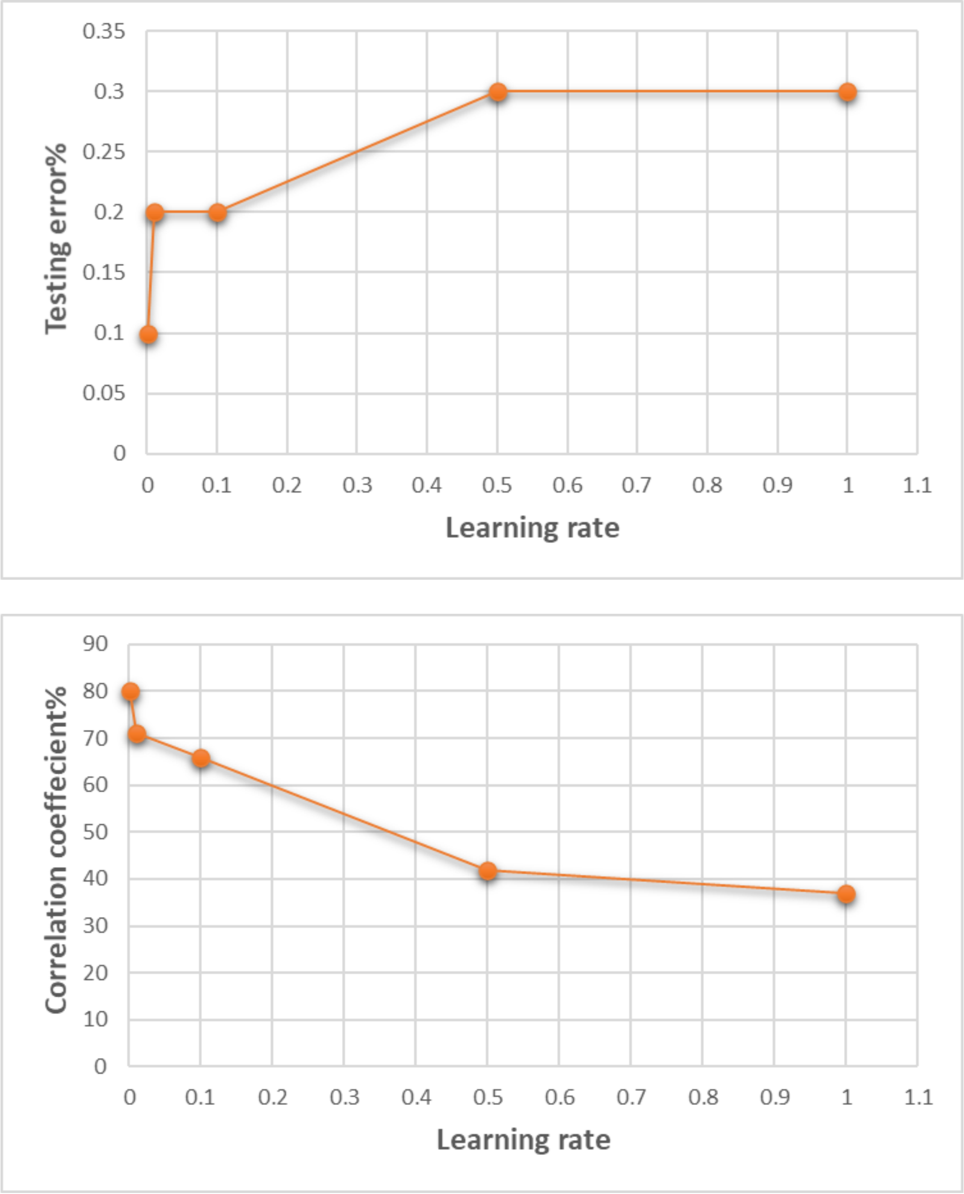


Figure A.10: The effect of variation in learning rate on the performance of the neural network model during normal working days.

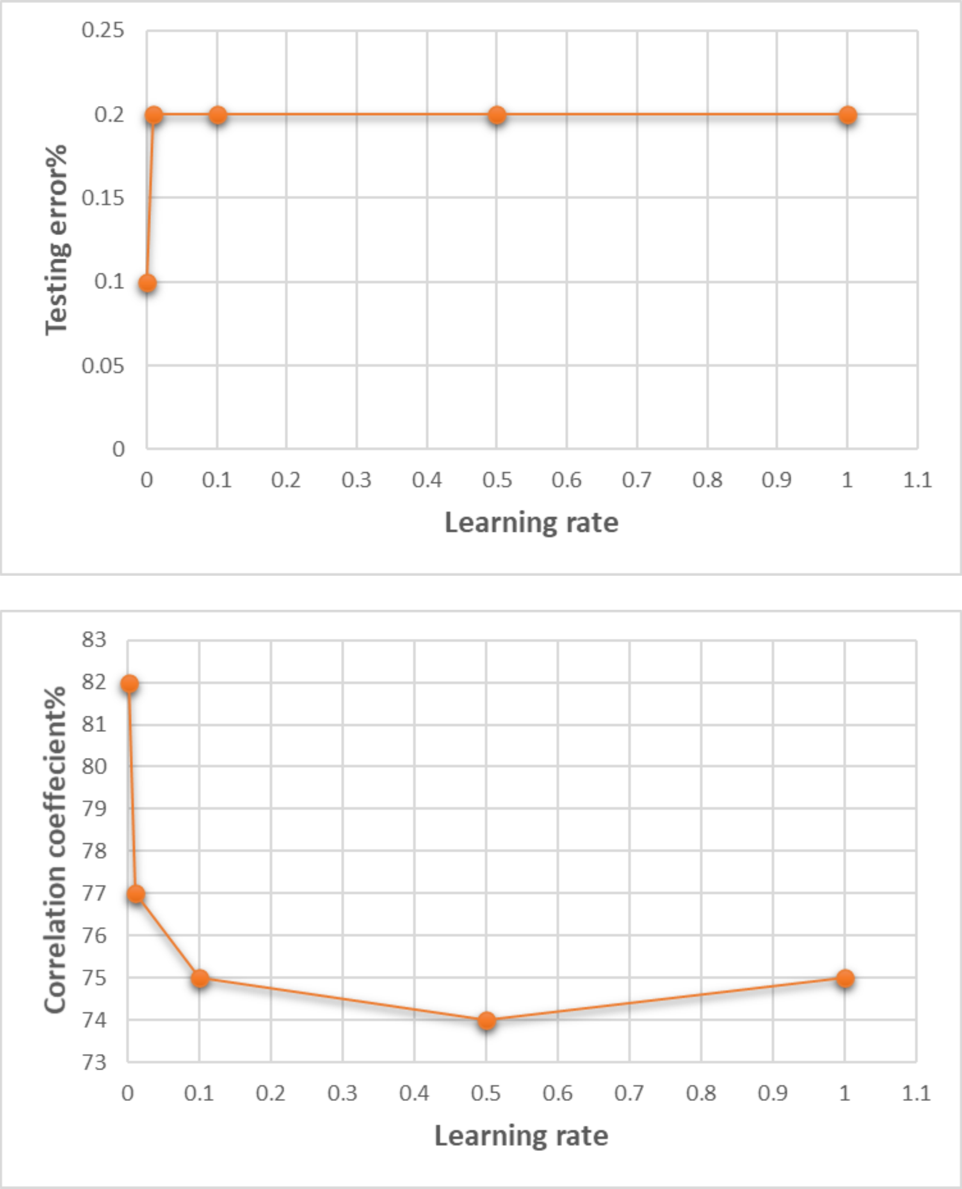


Figure A.11: The effect of variation in learning rate on the performance of the neural network model during holiday days.

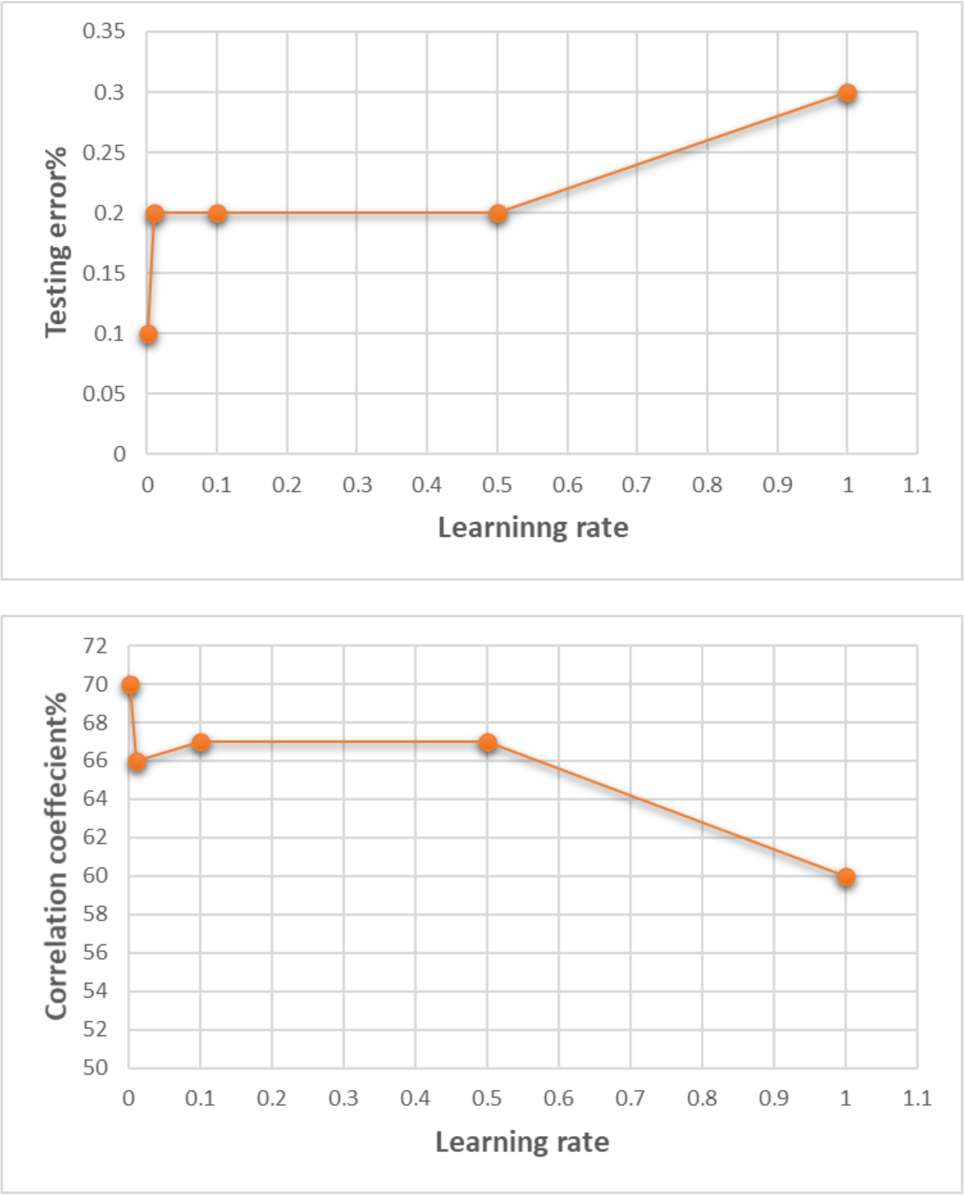


Figure A.12: The effect of variation in learning rate on the performance of the neural network model during weekend days.

Six tests were conducted to examine the effect of the transfer function, as shown in Table (A.5) below. (0.3%, 0.2%, 0.5%), and as a result, we conclude that the best performance of the network model was obtained by using the functional function (tanh) for the hidden layer, whereas the best performance of the output layer was obtained by using the activation function (Linear), due to the linear nature of the data, so distinct results were obtained.

Table A.5: The impact of the transfer function, also known as the activation function, on the efficacy of neural network models.

neural network Layers		**Normal work days**		**Holidays**		**Weekend days**	
Hidden layer	**Output layer**	**Testing error %**	**Correlation coefficient (r) %**	**Testing error%**	**Correlation coefficient (r) %**	**Testing error%**	**Correlation coefficient (r) %**
Sigmoid	Sigmoid	0.4	21	0.6	26	4.6	27
Sigmoid	Tanh.	1	50.8	1.4	44	5.2	31
Sigmoid	Linear	2.4	39.2	0.7	30	3.2	43
Tanh.	Sigmoid	2	27.2	2.7	35	1.7	28
Tanh.	Tanh.	0.5	30	1.5	45	5.2	35
Tanh.	Linear	0.3	80	0.2	82	0.5	85

- ## Weight adjustment Model

In neural networks, the relationship between individual neurons is defined by a parameter known as weight. This weight signifies the significance of the connection between two neurons. When a neuron receives input values from neurons in the preceding layer, it multiplies each input value by the corresponding weight associated with the connection. The neuron then accumulates the products of these multiplications and applies a transfer function, also known as an activation function, to the resulting sum. The output of the transfer function is contingent upon the specific type of neuron, and it represents the output of said neuron that is subsequently transmitted to the neurons situated in the subsequent layer.

Following the optimization of hyperparameters in the previous models, the network was trained. This resulted in the determination of weight values for the connections between the input layer (first layer) and the middle or hidden layer (second layer), as well as the weights between the second layer and the output layer (third layer). These weight values are presented in Tables A.6, A.7, and A.8 below.

Table A.6: Modifying the interlayer weights within the neural network pertaining to the model designed for normal working days.

Predictor		Predicted			
		Hidden Layer			Output Layer
		H (1:1)	H (1:2)	H (1:3)	Density
Input Layer	Bias (θ_1)	1.4445	0.791	-0.356	
	ATR	0.0032	-0.002	-0.002	
	SMA	-0.01	-0.019	0.0089	
	EMA	-.007	0.0085	0.0189	
	RSI	-0.00003	-0.00002	-0.00004	
	ROC	-0.0001	0.0002	0.0004	
	MOM	0. 0009	-0.006	-0.005	
Hidden Layer 1	Bias (θ_2)				1.1
	H (1:1)				-1.1
	H (1:2)				-1.3
	H (1:3)				0.6

Table A.7: Modifying the interlayer weights within the neural network pertaining to the model designed for holiday days.

Predictor		Predicted			
		Hidden Layer			Output Layer
		H (1:1)	H (1:2)	H (1:3)	Density
Input Layer	Bias (θ_1)	0.3947	-0.409	1.2678	
	ATR	0.003	0.0101	0.0006	
	SMA	-0.018	-0.004	-0.023	
	EMA	-0.003	0.0166	0.0047	
	RSI	-0.00004	-0.00005	0.00003	
	ROC	-0.0002	-0.001	-0.0002	
	MOM	0.0018	0.014	-0.004	
Hidden Layer	Bias (θ_2)				0.6
	H (1:1)				-1.1
	H (1:2)				0.3
	H (1:3)				-1.3

Table A.8: Modifying the interlayer weights within the neural network pertaining to the model designed for weekend days.

Predictor		Predicted			
		Hidden Layer			Output Layer
		H (1:1)	H (1:2)	H (1:3)	Density
Input Layer	Bias (θ_1)	-1.027	-0.571	0.6861	
	ATR	-0.0002	0.0019	0.0103	
	SMA	0.0162	0.0219	0.0007	
	EMA	-0.001	-0.003	-0.02	
	RSI	0.000007	0.000006	-0.000002	
	ROC	-0.0004	0.00009	-0.0007	
	MOM	0.0021	0.0031	−0.012	
Hidden Layer	Bias (θ_2)				0.7
	H (1:1)				1.4
	H (1:2)				1.0
	H (1:3)				-0.1

The ultimate configuration is achieved for the three artificial neural networks corresponding to different types of days (normal work days, holidays, and weekend days). These networks consist of three layers, as depicted in Figure 3.3, and are characterized by distinct weights and hyperparameters across the various models.

الملخص

في السنوات الأخيرة، تطورت أنظمة النقل الذكية (ITS) بسرعة، مدفوعة بالطلب المتزايد على تحسين إدارة شبكات النقل والتقدم في مجال الحوسبة. تشمل أنظمة النقل الذكية (ITS) مجموعة واسعة من التطبيقات التي تتطلب استراتيجيات استباقية وبيانات تنبؤية مدعومة بالذكاء الاصطناعي والبيانات الضخمة. ينصب التركيز الأساسي لهذه الأطروحة على تطوير نماذج دقيقة للتنبؤ بحركة المرور على المدى القصير، خاصة فيما يتعلق بكثافة حركة المرور، بهدف تعزيز تخطيط النقل الحضري وتمكين المسافرين الأفراد، مما قد يحدث ثورة في التحكم والتخطيط لحركة المرور في المناطق الحضرية.

تدور المشكلة المطروحة حول الحاجة الملحة إلى تنبؤات دقيقة لحركة المرور على المدى القصير لتسهيل تطبيقات أنظمة النقل الذكية الاستباقية ودعم القرارات المستنيرة من قبل المسافرين الأفراد. تدور الأبحاث الحالية في الغالب حول مقارنة طرق التعلم الآلي مع إهمال غالبًا دمج المؤشرات الفنية في تنبؤات كثافة حركة المرور. ويتمثل التحدي الرئيسي في تعزيز دقة التنبؤ مع فهم تأثير المؤشرات الفنية على كثافة حركة المرور.

يبدأ منهج الأطروحة بمراجعة شاملة لطرق التنبؤ بالبيانات ويتعمق في تقنيات التعلم الآلي المختلفة المصممة للتنبؤ بحركة المرور على المدى القصير. يقدم ثلاثة نماذج متميزة تتضمن تطبيع البيانات لمراعاة العوامل الفنية التي تؤثر على كثافة حركة المرور. ينشأ الاختراق الكبير من تكامل ميزات المؤشرات الفنية، مما يعزز دقة الانحدار بشكل كبير. يتم اختبار هذه النماذج بدقة باستخدام بيانات واقعية من الطريقين السريعين M25 و M60في ظل ظروف مرورية متنوعة. بالإضافة إلى ذلك، تقدم الدراسة خوارزمية لتقييم مستوى الخدمة(LOS) على مدار الساعة، مع الاستفادة من بيانات كثافة المركبات من نظام الكشف عن حوادث الطرق السريعة والإشارة التلقائية. يجمع هذا النهج المبتكر بين المؤشرات الفنية ونماذج التعلم الآلي لتصنيف LOS بدقة. يتم استخلاص بيانات LOS الحقيقية الأرضية من أجهزة استشعار ثابتة، مما يعرض التحسين الملحوظ للدقة الذي تم تحقيقه من خلال تكامل المؤشرات الفنية.

تؤكد النتائج الرئيسية للأطروحة على التأثير التحويلي لدمج المؤشرات الفنية، مما يحسن بشكل كبير دقة التنبؤ بكثافة حركة المرور بنسبة 86.63% لبيانات الطريق السريع M60 و68.2% للطريق السريع M25، بغض النظر عن نهج التعلم الآلي المختار. علاوة على ذلك، فهو يوضح الدقة المحسنة لتقدير LOS (حوالي 6.52%)، مع إمكانية تطبيقه على الطرق السريعة في مواقع جغرافية مختلفة.

وأخيرا، يقدم هذا البحث مساهمات كبيرة في تعزيز تطبيقات أنظمة النقل الذكية والكفاءة الشاملة لشبكات النقل، مما يحقق فوائد لكل من وكالات النقل والمسافرين الأفراد.

جمهورية العراق
وزارة التعليم العالي والبحث العلمي
جامعة بابل
كلية تكنولوجيا المعلومات
قسم شبكات المعلومات

نهج التعلم الآلي المطور للتنبؤ بالازدحام المروري على الطرق السريعة في شبكات المركبات المخصصة

اطروحة مقدمة إلى

مجلس كلية تكنولوجيا المعلومات - جامعة بابل كجزء من متطلبات

نيل درجة الدكتوراه فلسفة في تكنولوجيا المعلومات / شبكات المعلومات

من قبل

احمد ابراهيم تركي خلف

بإشراف
أ.د. سعد طالب حسون الجبوري

1445 هـ **2023 م**