ADDIS ABABA UNIVERSITY

COLLEGE OF NATURAL AND COMPUTATIONAL SCIENCES

SCHOOL OF INFORMATION SCIENCE

TRAFFIC SPEED PREDICTION USING MACHINE LEARNING: IN CASE OF ADDIS ABABA CITY

A MASTER'S THESIS

By

TEKA MOSISA DARA

October, 2023

ADDIS ABABA, ETHIOPIA

ADDIS ABABA CAR TRAFFIC SPEED PREDICTION USING MACHINE LEARNING: IN CASE OF ADDIS ABABA CITY

By

TEKA MOSISA DARA (ID:- GSR/1053/14)

Advisor: -

Michael Melese(PhD.)

A Thesis Submitted as a Partial Fulfillments for the Award of Degree of Master of Science in Information Science

To

ADDIS ABABA UNIVERSITY

COLLEGE OF NATURAL AND COMPUTATIONAL SCIENCES

SCHOOL OF INFORMATION SCIENCE

November, 2023

Addis Ababa, Ethiopia

Declaration

I, the undersigned, declare that the thesis comprises my own work in compliance with internationally accepted practices; I have fully acknowledged and referred all materials used in this thesis work.

Teka Mosisa

Name Signature Date

This Thesis has been submitted for examination with my approval

Dr. Michael Melese

Name Signature Date

Name and Signature of Members of the Examining Board

Name	**Title**	**Signature**	**Date**
Michael Melese(PhD)	Advisor	_____	_____
_____	Examiner	_____	_____
_____	Examiner	_____	_____

Acknowledgments

First and foremost, "Glory be to God the Almighty," who gave me incredible endurance from the start to the finish of my task. There are many people, besides God, without the help of whom it would not have been possible to complete this thesis. I'm gratitude for the help, direction, and inspiration I received from my advisor, Dr. Michael Melesse. I am grateful for the support and help from my families, especially my wife Betelhem Tefera, and my brother Abdi Mosisa. Along with all of my friends, especially Geleta Tesfaye and Meseret Tenaw, I am grateful to OSTA's top management and staffs. I sincerely appreciate Habtu Reda from Addis Ababa Science and Technology University for providing the dataset of Sheger Bus Public Transport and the necessary data for this thesis work. Last but not least, I would like to express my gratitude to everyone who took part in the research process, as their contributions to the project's success were greatly appreciated.

Table of Contents

List of tables

List of figures

List of acronyms and abbreviations

ML---------- Machine Learning

DL---------- Deep Learning

ITS---------- Intelligent Transportation System

CV---------- Automated Vehicles

GPS---------- Geographical Point System

OD---------- Origin-Destination

OLS---------- Ordinary Least Squares Regression

MARS---------- Multivariate Adaptive Regression Spline

LOESS---------- Locally Estimated Scatterplot Smoothing

KNN---------- K-Nearest Neighbor

LVQ---------- Learning Vector Quantization

SOM---------- Self-Organizing Map

LARS---------- Least-Angle Regression

LASSO---------- Least Absolute Shrinkage and Selection Operator

SVM----------Support Vector Machine

ANN---------- Artificial Neural Network

EML---------- Ensemble Machine Learning

GBDT---------- Gradient Boosting Decision Tree

RF---------- Random Forest

OOB---------- Out of Bag

CNN---------- Convolutional Neural Network

RNN---------- Recurrent Neural network

LSTM---------- Long Short Term Memory

ARIMA---------- Autoregressive Integrated Moving Average-Based Models

GIS---------- Geographic Information System

AASTU---------- Addis Ababa Science and Technology University

TSP---------- Traffic Speed Prediction

RMSE---------- Root Mean Square Error

MAE---------- Mean Absolute Error

MAPE---------- Mean Absolute Percentage Error

GUI---------- Graphical User Interface

MLP---------- Multilayer Perceptron Protocol

Abstract

The demand for mobility and transportation in Addis Ababa is growing along with the city's population expansion. Even though Addis Ababa City is having low motorization rates by global standards, the challenge of traffic congestion is get worse by the city's growing urban and industrial populations, which demand more vehicles. Studies shows that Ethiopia has an unreliable mobility traffic control and management system. Nevertheless, building infrastructure in accordance with international standards is a challenging and long-term solution due to the state of the economy and organizational difficulties.

Analysis of traffic speed and its variables is crucial given the effects of traffic congestion on the economy and other aspects. In order to effectively utilize the limited resources in Addis Ababa City, traffic speed prediction, one of the elements of an intelligent transportation system, will be a solution. The Addis Ababa Traffic Management Agency has only performed manual traffic count flow study of traffic congestion at intersections but actual traffic movement patterns are dynamic and route-based. Various sectors employ machine learning techniques, including online transportation networks and traffic prediction. This study aimed to develop machine learning models that predict the road traffic speed of Addis Ababa City. ML methods use less time and computer power and are considered to be a viable model for the presented dataset.

Sheger Public Bus Transportation traffic dataset which compiled and published by Addis Ababa Science and Technology University was used for this study. Five regression models were used to train and test the data. To find the most effective method of predicting the speed of traffic on a city-wide scale, three scenarios, including model parameters, feature importance identification, and hyper-parameters tuning, were tested. With consistent performance across all scenarios and scoring values Random Forest (RF) achieved best performance and after parameter tuning other ensemble algorithm Gradient Boosting achieved best performance.

Keywords: *Machine learning, Traffic speed prediction, Regression model, Intelligent transport system, random forest*

CHAPTER ONE: INTRODUCTION

1.1. Background of the Study

The demand for mobility and motorized transportation is growing along with increasing rate of urbanization. As a result, transportation systems are now fundamental to human endeavors. Thus, increasing demand on the transportation system makes it challenging to offer effective travel services. Traffic congestion is one of the challenges (Boukerche & Wang, 2020). Intelligent Transportation System (ITS) is a comprehensive transportation management system designed to reduce traffic problems, primarily congestion, then increase traffic efficiency. ITS offers innovative solutions by merging advanced technology and advancements in communication, information systems, sensors, and controllers with the conventional realm of road infrastructure (Wang et al., 2017).

Machine learning is one of the techniques used by ITS for predictive analysis, such as traffic speed prediction problems. Machine learning, according to Boukerche & Wang (2020), has better non-linear features, is more applicable to a variety of real-world scenarios, and requires less prior knowledge about the relationships between various traffic patterns in order to build models.

According to a world bank analysis of data from the central statistical office, urbanization in Ethiopia will occur at a quicker rate, at roughly 5.4 percent annually. By 2034, the current urban population is projected to have tripled, with 30 percent of the nation's population living in urban areas (Ethiopia Urbanization Review, 2015). Due to its commercial and political hub, Addis Ababa City, is a prime example of Ethiopia's explosive urbanization. Hence, demand for mobility and transportation is growing along with the city's population expansion. However, in developing nations like Ethiopia, meeting this demand with an effective transportation system while using limited resources is difficult. Lack of infrastructure including parking spaces, road lanes, traffic signals, and efficient traffic management inevitably leads to traffic congestion. According to study conducted by Kefyalew Tadele on economic effects of traffic congestion along the Meskel Square to Kality Interchange route: -

> “Total congestion costs including delay costs and wasted fuel costs of passenger and truck vehicles at the road segments were calculated at ETB 42,940,309.52, 24,265,882.55, 56,517,437.03 and 89,244,561.02 in respect of the Riche-Meshualekya,

Global-Meskel Flower, Nifas Silk-Adey Abeba and Akaki-Kality Maseltegna road segments" (Kefyalew, 2017).

Hence, analysis of traffic speed and its variables is crucial given the effects of traffic congestion on the economy and other aspects. Various sectors employ machine learning techniques, including online transportation networks and traffic prediction (Ray, 2019). This study aimed to develop machine learning models that predict the road traffic speed of Addis Ababa City because ML methods use less time and computer power (Fadilah, 2022) and are considered to be a viable model for the presented data.

1.2. Motivation of the Study

Due to the growing population and number of vehicles, Addis Ababa's basic transportation system is facing challenges from traffic congestion. As the nation's capital and the home of several national and international organizations, Addis Ababa is a significant political, social, and economic hub in Africa. According to a study conducted by Kefyalew Tadele (Kefyalew, 2017), 500,000 vehicles were estimated to be in use in Addis Ababa City. According study conducted by Tarikwa two years later, this figure was increased by 20% with estimated number of vehicles 596, 084 (Tarikwa, 2020). Even though this figure shows the city having low motorization rates, the challenge low traffic speed due to low travel time is get worse by the city's growing urban and industrial populations, which demand more vehicles. As a result, drivers and travelers must contend with traffic travel delay or traffic speed, which has a variety of negative effects, including delays, air pollution, and excessive fuel consumption by cars, increased spare part costs, and psychological effects. From the perspective of ITS implementation, there are a number of issues that need to be addressed when using conventional road infrastructure, which in turn, mitigate traffic delay.

One of the essentially important traffic state indicators that has potential use in Intelligent Transportation Systems (ITS) is traffic speed (Kakde, 2022). Consequently, the government of Addis Ababa City will be assisted in making decisions regarding traffic congestion by creating a traffic data-driven road traffic speed prediction model which is one of the indicators used to assess the state of the traffic, utilizing the current road infrastructure. (Fadilah, 2022) In comparison to deep learning techniques, machine learning is a powerful technique that facilitates the predictive analysis of large amounts of data with less computation time and power.

1.3. Problem Statement

Addis Ababa serves as a hub for regional and global trade in goods for commercial and import-export purposes, including agricultural products. Business labor market, customer delivery market, and shopper market locations that may be reached within a finite window of appropriate travel time decreases as a result of traffic congestion (Dekeba & Bekele, 2020). Traffic flow status should be taken into account by decision-makers when planning their transportation with intelligent transportation systems, which are now a crucial part of smart cities.

Addis Ababa, Ethiopia, is experiencing one of the fastest rates of urban growth in the world (Ethiopia Urbanization Review, 2015). In which, growth and relatively car based transport infrastructure have contributed to the growing use of private motor vehicles, leading to increased pollution, congestion and road safety challenges. Addis Ababa has an estimated 70% of the country's vehicles. As a result, these vehicles nearly releases CO2 a half of the city's emissions every year (Addis Ababa City Administration, 2018). Low traffic speed occurs throughout the city in spatial patterns, which are area-wide or location-specific and temporal patterns, which repeat continuously at predictable times or at random times, caused by external factors like weather and accidents (Kefyalew, 2017). Traffic congestion due to low traffic speed in Addis Ababa city has already affected a number of enterprises and government services as the number of vehicles has been growing tremendously (Kefyalew, 2017; Tarikwa, 2020).

Globally, several traffic management techniques have been put to use, demonstrating their potential for resolving issues with traffic, such as accidents, low traffic speed, and other issues. Building infrastructure in accordance with international standards is a challenging and long-term solution due to the state of the economy and organizational difficulties. According to study conducted by Tezazu, Ethiopia has an unreliable mobility traffic control and management system (Tezazu, 2017).

Traffic speed, flow, density, and duration are among the observable characteristics of roadway performance that are taken into account when managing road networks in order to measure traffic congestion. Indicators of traffic speed can be obtained from spatial features like the road network's longitude and latitude as well as temporal variables like link travel time and delay. Average vehicle trip speed and average vehicle speed during peak hours can be used to characterize a speed-based indication of traffic congestion. Thus, one of the recognized

indicators of traffic conditions taken into account in this study is traffic speed, and a traffic speed forecasting model will also be helpful in solving the problem mentioned above. The Addis Ababa Traffic Management Agency has only performed manual traffic count flow study of traffic congestion at intersections. But due to lack of permanent electronic traffic recording devices, traffic count was undertaken manually by observers who visually count and record traffic on a hand-held electronic device. Nevertheless, actual traffic movement patterns are dynamic and route-based.

1.4. Research Questions

This study is examining for answers to the following research questions in order to address the problems with Addis Ababa's road traffic that have been mentioned in problem statement.

Q1: What features are most important in predicting traffic speed?

Q2: Which machine learning algorithm performs best in predicting traffic speed?

Q3: How is city-wide traffic speed during days of the week and months of the year?

1.5. Research Objectives

1.5.1. General Objective

The primary goal of this research is to develop a machine learning model that can predict traffic speed and determine key elements or characteristics of Addis Ababa's road traffic speed prediction.

1.5.2. Specific Objectives

The following specific goals must be met throughout the study in order to fulfill the above general objective.

- Identify key features those precisely predict traffic speed
- Identify the traffic speed status in the days of the week and months of the year
- Select the most accurate traffic speed prediction algorithms by comparing the models.

1.4. Significance of the Study

Realization of traffic congestion forecasting system boost the efficiency of transportation, particularly in cities like Addis Ababa where scarce of transportation system facility is there. To reduce traffic congestion, actions, such as manual traffic count at junction point have been taken by city government traffic management agency even though actual traffic movement patterns are dynamic and route-based. Therefore, this study introduces city-wide traffic speed prediction model which is, one component of intelligent transportation system so that decision makers considers these technologies in their traffic management infrastructures design and services strategic plan.

Route traffic speed prediction will therefore be helpful in learning more about the scenario of city traffic speed, which then, will aid to design response mechanism that could lessen the problems associated with traffic congestion. The majority of studies on traffic prediction were conducted only along a corridor or in close proximity to a sensor location (Mystakidis & Tjortjis, 2020). It is therefore necessary to explore the city-wide scale traffic speed prediction (Wang et al., 2019). As a result, the following statement best captures the contribution of this study:

- The features discussed in section three will be used to predict traffic speed, taking into account months in a year and the days of the week in a month.
- The government will benefit from this study's increased awareness and understanding of the determinants or features that contribute to accurate traffic speed prediction.
- The models will be trained using a variety of ML regression algorithms. After analyzing and contrasting the study's prediction results, scenarios will be developed to find the best approach to forecast traffic speeds throughout the entire city.

Lastly, it is expected that the study will provide important results that can be applied as a starting point for additional research aimed at refining the current study's conceptual framework and methodology.

1.5. Scope and Limitations of the Study

The development of traffic speed prediction models for Addis Ababa road traffic using machine learning regression algorithms is the sole focus of this study. Only the daily GPS dataset gathered in 2021 year was utilized for the research. Hence, the research was restricted to predicting traffic speeds on a daily and monthly basis; it did not take into account time intervals or random times that could be influenced by outside variables like weather or accidents. As a result, the model's design for the study is restricted to its analysis of the GPS dataset that was acquired historically.

1.6. Brief about Methodology

This study uses machine learning techniques in an experimental research design. The literature review section includes theoretical ideas and relevant works relating to various machine learning techniques as well as intelligent transportation systems in general and traffic speed prediction in particular. The Trans-link company's GPS data, which was subsequently used by Addis Ababa Science and Technology University (AASTU) to predict bus station arrivals, was used as the dataset for this study. Python 3.9.13 with Sklearn 1.3.0 analysis tools in an Anaconda environment were used to extract relevant details from dataset. The R-Squared (R^2), Root mean squared error (RMSE), mean squared error (MSE), and mean absolute error (MAE) performance evaluation metrics were used to make sure the designed model forecasts accurately.

1.7. Structure of the Thesis

There are five chapters in this study. The study's background, problem statement, objective, significance, scope, and limitations are all covered in the first chapter. In the second chapter, a variety of literatures on intelligent transportation systems, traffic congestion, and associated traffic prediction systems have been reviewed. The study's technique and method the research follow covered in the third chapter. Dataset preprocessing and other literature are provided in the fourth chapter and examined, and ultimately, based on the findings, conclusions and recommendations are presented in the fifth chapter. After chapter five, there is a list of references and annexes.

CHAPTER TWO: LITERATURE REVIEW

The theoretical foundation and understanding as well as related work of traffic speed prediction analysis are explained in this chapter. The first section discusses the challenges and causes of traffic congestion. The chapter goes on to explain how comprehensive technology used in intelligent transportation systems helps to reduce traffic congestion. An explanation of ensemble and classical machine learning used in this study will be provided, along with a summary of the deep learning techniques. Additionally, a summary of traffic speed forecasting methods is given, emphasizing the use of machine learning techniques in traffic speed prediction.

2.2. Traffic Congestion

To understand and create the best possible transportation system with efficient traffic movement and the fewest possible traffic congestion issues, transportation engineers study the interactions between travelers—such as pedestrians, vehicles, and infrastructure—such as highways and traffic control systems (Zhang et al., 2022).

Figure 2.2: *Traffic Congestion*

Traffic congestion is a condition in transport that is characterized by slower speeds, longer trip times, and increased vehicular queue up. When traffic demand is great enough that the interaction between vehicles slows the speed of the traffic stream, this results in some congestion. Nowadays, transportation systems become an essential part of human activities. It was estimated an average of 40% of the population uses transportation system at least an hour each day (Zhang J. et.al.,2011). Due to urbanization and the high use of roadway vehicles the problem of traffic congestion is increasing, which in turn, increase the demand for costs on society as well as the country for spending (Kumar et.al.,2021). As people become more dependent on this system, it become not only with several opportunities but also challenges. Congestion is one of the challenges, which in turn, lead to an increase in fuel consumption, air pollution, accidents. Further, limitation of resources in developing countries, makes it difficult to expand infrastructure as per the demand and implementing plans for additional public vehicles (Verlag S., 2005). But, as currently, the competitiveness of a country, its economic strength, and productivity heavily depend on the performance of its transportation systems, there has to be a traffic congestion mitigation plan.

2.3. Intelligent Transportation System

The Internet of Things, or IoT, is growing in popularity these days, and innovations centered around applying smart technology to make the world smarter, including smart transportation, smart industries, and smart cities, are becoming more and more prevalent. The Intelligent Transportation System (ITS) a component of IoT as an innovator technology for smart transportation. IoT-based smart city implementation, or ITS, uses machine and deep learning among other techniques to address urbanization problems associated with human mobility (Kumar et al., 2019).

The primary goal of ITS is to provide innovative and improved traffic control services along with a range of transportation options so that travelers can travel in a planned, secure, and smart way (Guatam, 2021). Systems and innovations have unquestionably changed to become more structured and organized as a result of technological advancements. Along with enhancing traffic efficiency, cutting down on commuter travel times, and raising passenger comfort and safety standards, ITS aims to reduce traffic issues, primarily congestion. ITS not only guarantees the safe and effective use of roads and infrastructure, but it also offers other advantages like transportation networks that facilitate the mobility and accessibility of travelers within a region.

Additionally, it can provide a variety of practical and reasonably priced transportation options while making the best use of the infrastructure. ITS further oversees incidents and congestions to provide road users and travelers the best level of service (Hassn et.al., 2016).

ITS-developed technology includes connected or automated vehicles (CV). According to Chang et al. (2019), CV technology allows different types of vehicles (such as cars, buses, trucks, etc.) to wirelessly connect with one another and share useful and crucial mobility and safety information. Congestion is a major issue that researchers are still working to address with the development and advancement of CV (Miglani & Kumar, 2019). According to survey study conducted by Razali et.al. on traffic flow prediction techniques and evaluation, in order to create a prediction system for traffic flow and congestion, an extensive amount of work and research have been done, and it is crucial that congestion information be distributed (Razali et.al., 2021).

Currently, the requirements for transportation capability, as most of the road are becoming saturated due to number of vehicles growing exponentially. Efficiency of transportation system contribute a lot for national productivity and international competitiveness of the institution. So, institutions should focus on modern transportation system for sustainability of transportation by transforming its operation (Lin et al.,2017).

Intelligent Transportation Systems, or ITS, use cutting-edge technology, including those aimed to collect data from various sources, such as GPS, sensors (Li et al., 2018). This is enabled by applying new transportation technologies. ITS contribute to both mobility and sustainable transportation which offer innovative services for various modes of transportation management. It improves the efficiency of conventional transportation infrastructure together with information systems, communication devices, sensors etc. This mechanism reduces traffic congestion, which in turn, improve safety and enhance productivity that done through integrating different components of transportation system and its service operation.

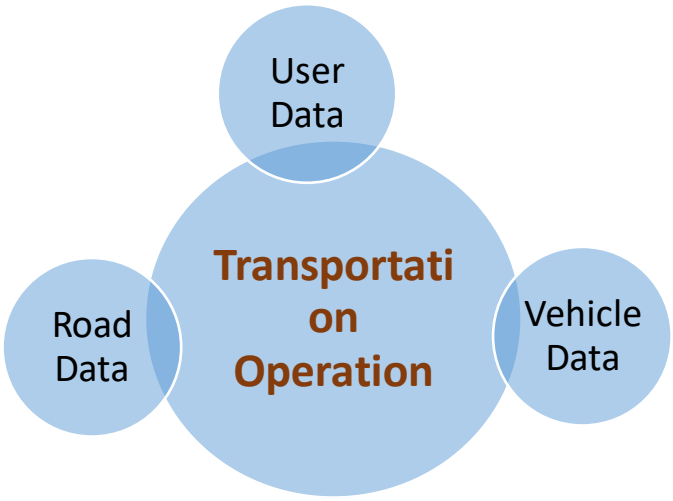


Figure 2.3: *Interconnection and Operations in ITS*

2.3.1. Data-Driven Analytics in Intelligent Transportation System

ITS data can be obtained from different sources; effective data analytical methods should be applied to evaluate collected data as per its goal. ITS can handle the data this way to provide information to traffic management process. Hence, real-time and historical huge traffic data can improve the transportation system safety level through infrastructure asset problem detection and traffic accident occurrence and real-time traffic speed prediction (Li et.al., 2018).

According to a survey conducted by Li et.al. on data analytics in ITS, ITS is categorized into data collection, data analytics and application layer. Data collection layer is the basis for ITS architecture that data collected through video surveillance, remote sensing, GPS, etc. Data analytics layer takes in data from the layer of data collection and apply different approaches such as machine learning methods to manage the data. Application layer applies the data process results from analytics layer in different transportation circumstances, for instance, traffic speed prediction.

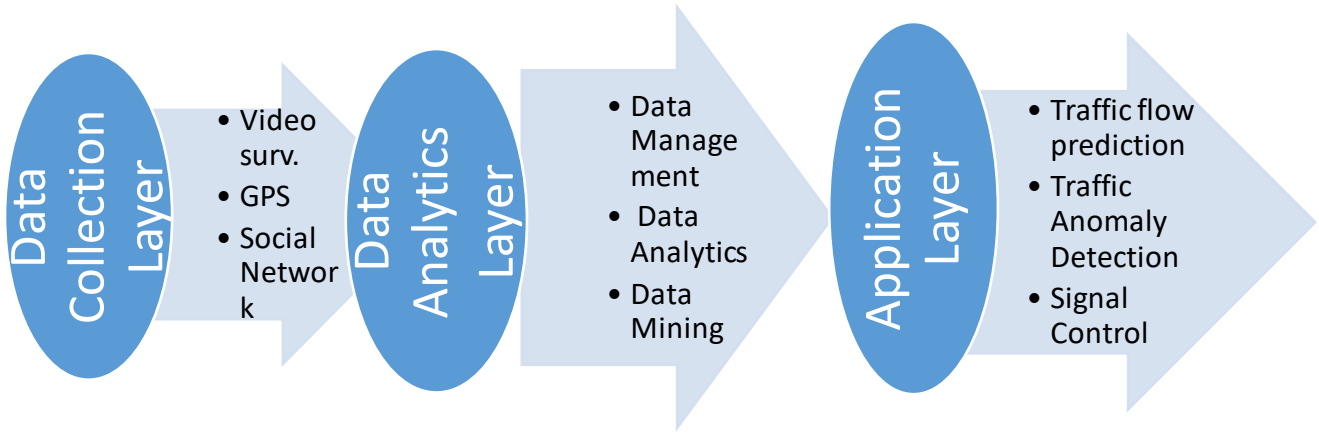


Figure 2.4: *ITS Data Analytics Architecture*

2.3.1.1 Sources of ITS Data Collection

Amount of data created and collected from ITS components such as; vehicle, people movement, road capacity is diverse and complex. Accordingly, ITS data can be categorized in to the following types.

Data from Smart Cards is a data source in which electronic readers will capture passenger details as they touch their smart cards. It is important component of public transportation to analyze comprehensive spatial-temporal information (Z. Li et al., 2018).

Data from GPS is the data collection method in which traffic data can be collected with location tracking. Combined with geographic information system(GIS) or other relevant map displaying technologies, data collected via this technology can be used for addressing many traffic issues, for example, traffic flow prediction.

Data from Passive Collection are those data not collected through active collection which can be used for future study. Mobile data, internet access data, social media data are examples of passive data.

2.4. Traffic Speed

The average speed of cars moving via a given area unit in a defined certain time is known as traffic speed, and it is one of the most important traffic state indicators that could be used in intelligent transportation systems. Traffic condition of the road segment determines the traffic speed distribution (Zefreh, 2019). The speed value on an urban road may be a good indicator of how congested the road is. For instance, Google Maps displays this degree of crowdsourcing crowdedness using information gathered from different portable devices and in-vehicle sensors. Applications for route navigation and estimation-of-arrival can both benefit from more accurate traffic speed predictions. A vehicle's speed is determined by a number of variables, including its geometry, the flow of traffic, the location, the time, the driver, and the surroundings. Traffic speed can be classified into spot, average, journey and running speed (Kakde,2022). Spot speed is a rapid speed of vehicle at certain instant of time or segment of road. Based on the spot speeds of numerous vehicles, the average speed of traffic is calculated. Vt = (V1 + V2…+Vn)/n. And harmonic mean of spot speed of all vehicle is calculated as n/(1/v1+1/v2…1/vn)

Concerns about traffic speed exist for both freeways and urban roads. However, these two distinct situations present different difficulties. Compared to the urban case, the prediction is simpler on freeways because there are more traffic lights. Additionally, the complex temporal dependency is the main source of the difficulty. The connection patterns and sudden alterations on urban roads are more complex in the traffic networks, which are more complex. The permitted vehicle types and the speed limits, for instance, may vary between different road segments. It becomes more difficult to model the region dependency in addition to the complex temporal dependency.

2.5. Machine Learning

The field of machine learning (ML) focuses on developing computer algorithms that can simulate human intelligence. It incorporates concepts from varies disciplines, such as computer science, information theory, probability and statistics and applied in varies industries or sectors including. Prediction analysis one of the application of machine learning. The idea of teaching computers to carry out tasks intelligently without explicit programming by observing and understanding the surroundings through repeated examples is shared by many definitions of machine learning provided by different machine learning scholars (Martin & Issam, 2015).

According to review study by Martin & Issam, machine learning can be divided into different categories based on the type of data., from a concept learning and probabilistic perspectives and learning by accommodating a feedback system.

ML classifies records into supervised, unsupervised, and semi-supervised categories based on their nature. With supervised learning, one can consider (input, output) samples and use them to estimate an unknown (input, output) mapping. For example, one can use regression and classification to categorize the output. In unsupervised learning, the learning machine is only provided with input samples for the purposes of clustering and probability density feature estimation.

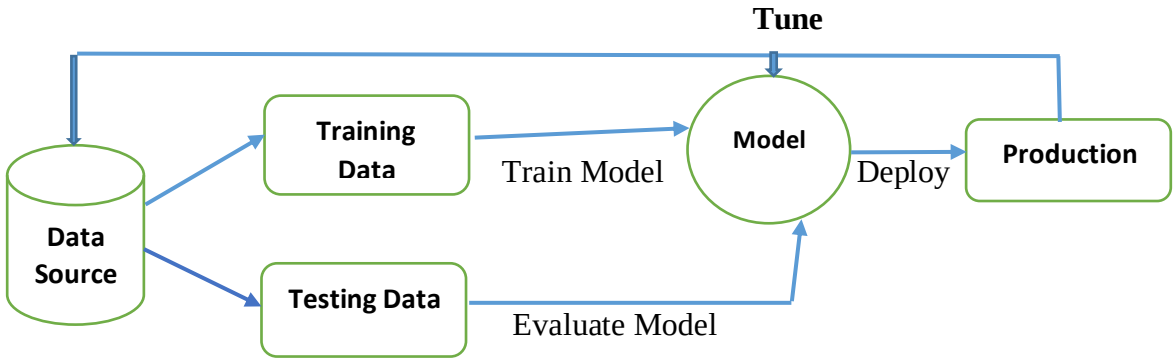


Figure 2.5: *Supervised Learning Workflow*

(***Source,*** Martin & Issam, 2015)

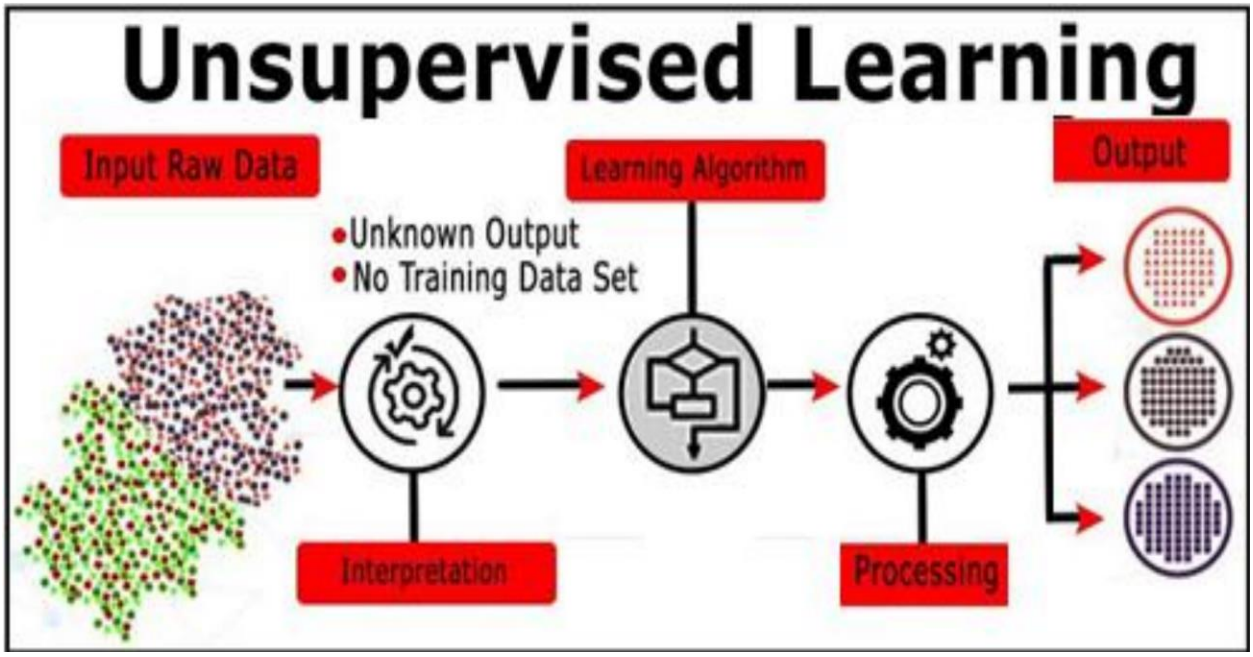


Figure 2.6: *Unsupervised Learning Workflow*

*(***Source:** *Mahesh, 2019)*

Transductive and inductive learning are the two categories into which machine learning can be divided from the standpoint of learning concepts. Despite having characteristics in common with supervised learning, transductive learning does not produce an explicit classifier. Based on the training data, training label, and test data, it attempts to predict the outcome (Rao et.al., 2017). On the other hand, inductive learning targets to predict outputs from inputs that the learner has not encountered earlier than (Vapnik, 1998).

From a probabilistic standpoint, machine learning algorithms fall into one of two categories: generative or discriminant models. A discriminant model determines the conditional probability of an output given typically deterministic inputs, like those from neural networks or support vector machines. A generative model employs graph modeling techniques like Bayesian networks and is completely probabilistic. Though it shares similar traits with supervise learning, it does not develop an explicit classifier. It attempts to predict the output based on training data, training label, and testing data.

Learning via accommodating a reward system is reinforcement learning, which is a class of learning algorithms that an agent tries to take a chain of actions that could maximize a cumulative reward. This form of technique is especially useful for online gaining knowledge of learning packages (Saton and Bartlo, 1998).

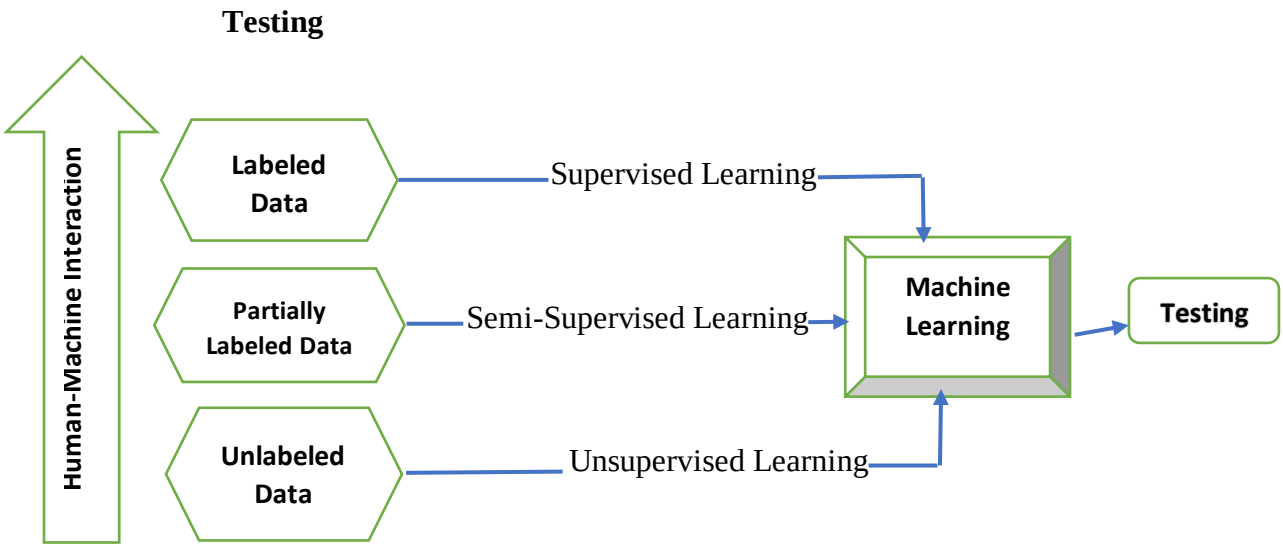


Figure 2.7: *Categories of machine learning algorithms according to training data nature*

2.6. Machine Learning Algorithms

Scientific research on machine learning focuses on statistical models and algorithms that let computer systems perform specific tasks without explicit programming (Matta, 2019). Numerous commonly utilized applications, including data mining, image processing, and predictive analytics, integrate learning algorithms, etc.

2.6.1. Regression Algorithms

Regression is a supervised learning method that enables us to predict the continuous output variable based on one or more predictor variables and aids in determining the correlation between variables. Predictive analytics includes regression analysis, which takes advantage of the correlation between dependent (goal) and independent variables (Rao et.al.,2017).

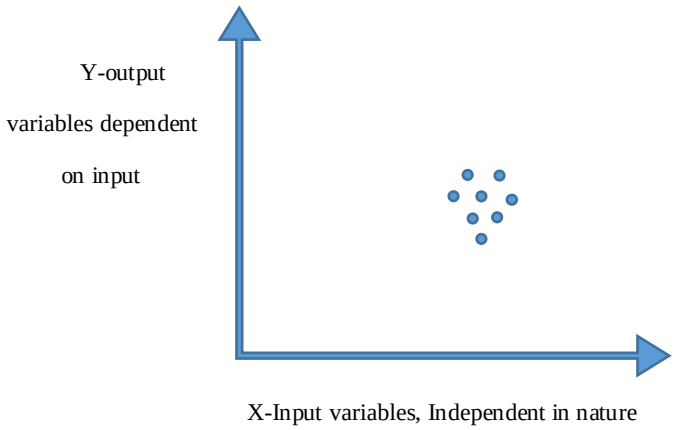


Figure. 2.8: *Regression Analysis*

2.6.1.1. K-Nearest Neighbors (KNN)

According to Zhang (2013), the KNN model represents the most basic machine learning algorithm. By looking at the k closest data points in the training dataset, it generates predictions—hence the term nearest neighbors. When the prediction selects the k closest data points, it will compute the average or the most relevant neighbors based on the k value (Fadilah, 2022). This model, however, has certain shortcomings, including a slower processing time and a lack of ability to handle large amounts of data.

2.6.1.2. Support Vector Regression (SVR)

The Support Vector Machine (SVM) decision function is, more accurately, an ideal hyperplane that is used to distinguish between observations that belong to one class and those that do not, or to classify observations based on feature-based informational patterns. The most likely label for unobserved data can then be determined using that hyperplane (Pisner & Schner, 2020). A set of data points with various labels are categorized using a Support Vector Machine (SVM), which uses a separating hyperplane or decision plane to demarcate decision boundaries. This approach for classification is strictly supervised (Rao et.al.,2017). In other words, using input or training data, the algorithm creates an ideal hyperplane, and this decision plane then sorts additional samples into categories. SVM may do both linear and nonlinear classification depending on the kernel being used.

Support Vector Machine is a machine learning algorithm that is capable of performing both regression and classification. SVMs are powerful because they can learn data classification patterns with a balance between consistency and accuracy (Schnyer & Derek, 2020). However, as the quantity of training data increases, consequently will the requirement for computing and storing it.

2.6.1.3. Instance-Based Algorithms

Instead of performing explicit generalization, these algorithms compare brand-new problem situations to previously viewed training instances that have been kept in memory (Sagi, 2019). Instead of creating a precise specification of the target function, instance-based or memory-based learning models retain instances of the training data. In order to ascertain or forecast the target function value, each new problem or example is compared to the previously stored cases. If a new instance is a better fit than a stored one, it can easily replace the old one. K-Nearest Neighbor (KNN) is one example of instance based algorithm.

2.6.1.4. Regularization Algorithms

Regularization is a method used in machine learning to prevent overfitting. A model performs badly on fresh data when it learns the training set of data too thoroughly. By introducing limitations to the model-building process, regularization helps in the reduction of overfitting (Kumar, 2022). In this method, finding a set of weights that minimizes both the training mistake and the penalty is the goal. Ensemble approaches, which incorporate several hypotheses, are another type of regularization that lowers generalization error. To further prevent overfitting, regularization can also be used in concert with other methods like cross-validation. (Kumar, 2022).

Regularization is the process of reducing outliers or combating overfitting. It is merely a straightforward yet effective alteration that is added to other current ML models, most frequently Regressive Models. By criticizing every bend in the curve that tries to match the outliers, it calms down the regression line. Examples include Ridge Regression, Least-Angle Regression (LARS), Elastic Net, and Least Absolute Shrinkage and Selection Operator (LASSO), etc.

2.6.1.5. Decision Tree Algorithms

Based on a set of restrictions, a decision tree creates a tree-like structure with potential answers to a problem. It gets its name because it starts out as a single straightforward choice or root and then diverges into a number of branches before coming to a conclusion or making a forecast, forming a tree (Rao, 2017). They are preferred because of their capacity to formalize the process of identifying viable solutions to the problem at hand, which enables them to do so faster and more accurately than other methods. A supervised machine learning technique called decision trees is used to solve problems with regression and classification by repeatedly segmenting data according to a given parameter. The nodes split the data, and decisions are made in the leaves. A classification tree's decision variable is categorical (the result is expressed as a Yes/No), whereas the decision variable in a regression tree is continuous. The decision tree has many advantages, such as being appropriate for regression and classification problems, being easy to interpret, handling quantitative and categorical values, being able to replace missing values in attributes with the most likely value, and having high performance because of the tree traversal algorithm's efficiency. But, Decision Tree may encounter the over-fitting issue (Ray, 2019).

2.6.1.6. Clustering Algorithms

In order to identify and label the data appropriately, clustering uses established patterns in datasets (Rao et al., 2017). Clustering algorithms can detect patterns in data automatically, allowing for analysis of obtained data without labeling. A set of samples are to be clustered in the appropriate clusters (or groups) according to similarity. It makes sense that samples that are part of the same right cluster are more comparable to one another than they are to samples from other clusters. Clustering is a crucial step in exploratory data analysis. Thus, clustering actually refers to the techniques for classifying unlabeled data (Lei, 2017). The goal of clustering is to organize a set of unlabeled patterns into useful clusters. Although labels are essentially linked to clusters, they are just derived from the data.

Clustering can be done by algorithms with vastly different ideas of what constitutes a cluster and how the clusters are divided. According to Berkhin (2006), typical conceptions of clusters include groupings where members are close to one another, dense regions of the data space, intervals, or certain statistical distributions. There are various methods those used for clustering, involving, K-means, K nearest neighbor, and so on. For instance, the easiest machine learning algorithm is the KNN model. It is called nearest neighbors because it makes predictions by

examining the k nearest data points in the training dataset. In this case, the user has specified an integer value k. The prediction will compute the most relevant neighbors or the average after choosing the k nearest data points (Fadilah, 2022).

2.6.1.7. Associate Rule Algorithms

Association rule learning is a type of unsupervised learning technique, which determines if one data item depends on another. Association rules are if/then statements used in databases, relational databases, and other information repositories to help connect seemingly unrelated pieces of data. Association rules are used to find the relationships between objects that are frequently used together. Applications of association rules include catalog design, basket data analysis, cross-marketing, and clustering.

For instance, if a consumer purchases bread, he may also purchase butter. The buyer may purchase a memory card in addition to a laptop. Association rules rely on two primary criteria: support and confidence. It finds the relationships and rules that arise from looking for recurring if/then patterns in data. Generally, association rules have to satisfy both a user-specified minimum confidence and a user-specified minimum support at the same time (Kumhare & Chobe, 2014). These algorithms are now frequently employed by e-commerce websites to forecast client behavior and potential demands in order to present him with tempting products. Examples include the Apriori and Eclat algorithms.

2.6.1.8. Neural Network (NN)

Neural networks (NN) are made up of interconnected nodes (neurons) that are arranged in two or more layers and are intended to work similarly to the human brain (Zurada, 1992). The neural structure of an artificial neural network is made up of hundreds of single units, or artificial neurons, connected by coefficients (weights).

There are many different kinds of neural networks that were created for various uses. Arteficial Neural Network (ANN), Convolutional neural networks(CNNs), recurrent neural networks (RNNs), Long short term memory(LSTM) and Gated recurrent unit(GRU) are some of algorithms used in traffic analysis and prediction.

ANNs are viewed as non-linear models since they attempt to identify intricate relationships between input and output data. However, it only looks at a sample of the data rather than the complete set, saving time and money (Rao, 2017).

The main applications of convolutional neural networks (CNNs) are in image analysis and recognition. Congestion detection using images from on-road surveillance cameras is one of their frequent applications to transportation issues. Regarding traffic forecasts, attempts to develop CNN-based models for predicting transportation network speed had a mediocre degree of success. To achieve this, researchers transformed traffic flow time and space data into a 2-dimensional image matrix (Kurniawan et al., 2018).

In contrast to CNNs, recurrent neural networks (RNNs) are designed to process time-series data or observations gathered over specific time periods. A good example of such observations is traffic patterns. Although RNN models have a short term memory due to the vanishing gradient issue, which causes some of the data from earlier layers to get lost, research has shown that they are highly accurate in predicting the evolution of congestion when used. Model training is more challenging and time-consuming as a result of this poor memory.

RNN variations include long short term memory (LSTM) and gated recurrent units (GRU). These models deal with the issue of vanishing gradients. A comparison of the models' performances revealed that the Cho et al.-originally proposed GRU model is more accurate in predicting traffic flow and is simpler to train (Hussian et al., 2021).

Multilayer Perceptions (MLPs) uses Feed-forward neural network (NN) to process the given data. A couple of benefits of using the MLP Regression model are its real-time capability and ability to learn non-linear models. Hence, this model has the potential to produce an extraordinarily complex model (Fadilah, 2022). Nevertheless, weight initialization must be used in order to improve validation accuracy because MLP has a nonconvex loss function.

.

2.6.1.9. Ensemble Algorithm

Algorithms that merge the forecasts from two or more models are called ensemble learning algorithms. While there is practically no limit to the ways in which this can be accomplished, boosting, bagging, and stacking are the three classes of ensemble learning techniques that are most frequently discussed and applied in practice (Brownee, 2021). These methods' widespread use can be mostly attributed to their success on a variety of predictive modeling tasks and their ease of application.

Boosting ensemble algorithm is used for reducing bias and variance. Boosting prioritizes minimizing bias over minimizing variation (Oza & Tumer, 2008). As a result, it improves base learners that have a high bias and low variance. Gradient Boosting Algorithms of boosting techniques. Bagging places more of an emphasis on lowering variance among ensemble members than bias. Therefore, the ensemble members with low bias and high variance perform bagging at its best. Bagging takes less time to compute than most ML techniques for large datasets because it trains the model with a small sample size (Alelyani, 2021). The bagging method is frequently used in random forest implementations. Stacking creates models that are distinct and take a different approach to the challenge of predictive modeling by using various base algorithms using the same dataset. In addition, stacking uses a single model to figure out how to integrate the base learners' (Mienye and Sun, 2017).

2.6.1.9.1. Gradient Boosting (GB) Regression

Gradient Boosting uses the boosting method to build robust ensembles. Gradient boosted decision tree (GBDT) is another name for this technique, which primarily uses decision trees as the basis learner to create reliable ensemble classifier (Mienye & Sun, 2022). GB is a component of an ensemble model that can be applied to tasks involving regression or classification and incorporates several decision trees. GB constructs trees in a serial fashion (Wei et al., 2020). It will thus fix the errors made by the earlier trees. It frequently employs extremely shallow trees, ranging in depth from one to five, which reduces the memory requirements of the model and speeds up processing. Each of the shallow trees in this combination will offer a reliable prediction based on the data. The model's performance will eventually improve after this shallow tree is added iteratively (Cheng et al., 2019).

The number of trees, the learning rate, and the pre-pruning of the trees are three parameters that can be changed to optimize the GB model. The model's complexity will rise with more trees, and

it may be possible to fix errors on the training sets. A more complex model will also be possible with a higher learning rate since each tree will be able to make more significant corrections (Fadilah, 2022). Nevertheless, overfitting could also result from increasing the model's complexity.

2.6.1.9.2. Random Forest

Random forest is an ensemble algorithm that builds multiple decision trees from bootstrapped samples using the bagging technique. Using the input data as a starting point, the bagging technique generates replacement random samples, which are then used to train the decision trees (Cohen, 2021). The random forest method has been widely used for a variety of tasks because it is easy to develop, quick, and produces high performance (Mienye & Sun, 2017). Two key elements of the random forest algorithm are the construction of multiple decision trees during training and the combining of their predictions via majority voting. Voting reduces the likelihood of overfitting in the random forest, which decision trees are prone to (Janitza, 2018). This method also uses bagging and feature randomness to build a forest of uncorrelated decision trees. For each decision tree's training in the forest, the random subspace technique makes sure that the features are selected at random (Ksantini, 2022). The decision trees that make up the forest have less correlation because they were trained using a random feature subset rather than the entire feature set.

A random sample of observations—either a bootstrap sample or a subsample of the original data—is used to build each tree in a random forest. Observations classified as out-of-bag (OOB) are those that are not included in the bootstrap sample or subsample. One advantage of the OOB error is that the entire original sample is used for error estimation as well as for constructing the RF classifier (Janitza, 2018). Since only part of the samples is used in the development of the RF, the performance of the RF classifiers generated by cross-validation and related data splitting processes for error estimation is worse. There is also a computational speed benefit to using the OOB error. Unlike other data splitting techniques like cross-validation, only one RF needs to be created, whereas k RF are required for k-fold cross-validation (JS, 2010). Using the OOB error can reduce calculation time and memory requirements when working with large data dimensions, where generating a single RF may take days or even weeks.

2.7. Traffic Speed Forecasting Methods

According to survey study conducted by Luo and Jiang (Luo &Jiang, 2022), different methods of traffic flow prediction have been extensively used, not only in traffic forecasting but also in intelligent transportation system (Fan et al. 2020). In order to identify traffic patterns at various times, such as during the day, on various days of the week, during events, etc., statistical approaches have been used (Lebedeva & Poltavskaya, 2020). Compared to machine learning, these techniques are simpler, quicker, and less expensive to implement. These models are categorized into parametric models in which structured expression is predetermined making the assumptions about the future. In case of traffic prediction, knowledge of these parameters can be utilized in this model, which could aid in the understanding different behaviors of traffic speed (Attilas & Vilmos, 2018). But, due to the non-linearity and variations in traffic data, these models do not predict effectively (Hussian et al., 2021).

By incorporating massive amounts of heterogeneous data from various sources, machine learning (ML) enables the development of predictive models. ML are classified into non-parametric models that they use parameters grow in numbers as the model learns from more data to mitigate issues of parametric models (Hussian et al., 2021). The use of ML algorithms to predict the speed of road traffic has been the subject of numerous studies. Several decision trees are built using the random forest algorithm, and their data is then combined to produce precise predictions. Given enough training data, it is fairly quick and can produce useful results (Wu & Liu, 2017). The k-nearest neighbors (KNN) algorithm uses feature similarity as a basis for forecasting future values. Studies shows that this algorithm is accurate for short-term traffic flow prediction (Zhang et al., 2013). Support vector machine, ANN are some of methods used in machine learning (Ramchandra & Rajabhushanam, 2021). Support vector machine, ANN are other methods used in machine learning

Compared to machine learning (ML) or statistical methods, deep learning (DL) methods have shown to be significantly more effective at forecasting traffic. In contrast to traditional ML models, DL use multi-layer nonlinear structures to describe distributed and hierarchical features for more complex traffic flow data (Hussian et al., 2021). Neural networks are the foundation for deep learning algorithms. However, it has been found that there are situations in which linear models and machine learning techniques with less computational complexity are preferable, and

that deep learning isn't always the best modeling technique in real-world applications (Manibardo et al., 2021).

2.8. Traffic Speed Prediction Using Machine Learning

Methods for forecasting traffic are categorized as parametric, non-parametric, or hybrid (a combination of both). Smoothing methods, Kalman filters, and Autoregressive Integrated Moving Average-based models (ARIMA) are examples of parametric methods. Non-parametric regression and other kinds of neural networks are examples of non-parametric approaches (Gmira et.al., 2017). Because they require a lot of data, non-parametric approaches are frequently referred to as data-driven methods.

Regression models are one of the many techniques for making short-term predictions that have been introduced in various literatures (Morik et al., 2017), the nearest neighbor method (k-NN) (Smith et.al.,2002), the Autoregressive Integrated Moving Average (ARIMA) (Zhang et.al.,2011) and the discretization modeling approach as easier solutions to the complicated nonlinear models (Li, 2011), in addition to machine learning algorithms such as Support Vector Regression (SVR) (Jeong, 2011) Random Forest (RF) for regression, and Neural Networks (Salanova et.al.,2019).

According to study conducted by Sanders et.al., a range of models from the literature, both parametric and non-parametric, were contrasted. Among the most popular non-parametric models are Random Forest, SVR, and Neural Networks; however, no single optimal technique for precise traffic prediction in any situation has been put forth (Bratsas, 2019). It is established that the SVR approach performs better than simple approaches like Historical Averages. The linear regression model has been demonstrated to perform better than alternative models, including k-NN and kernel smoothing techniques, in the analysis of traffic speed data in other

studies, including Sun et al.

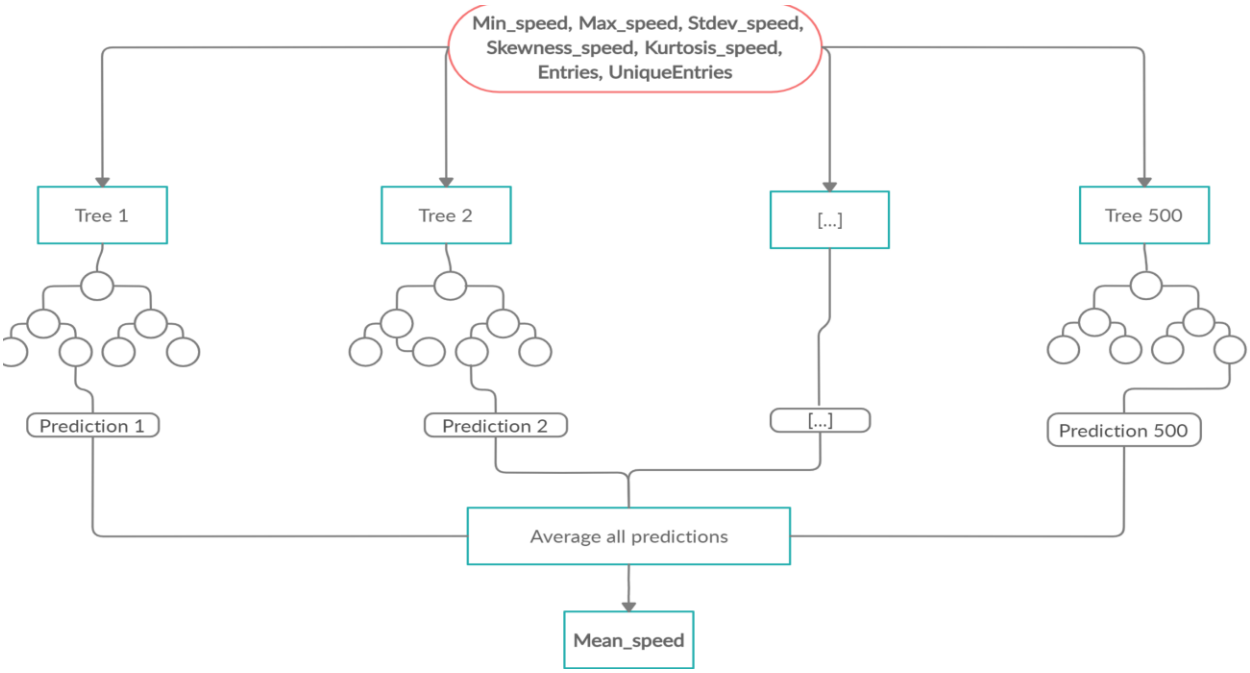


Figure. 2.9: *Example of Random Forest Model for Predictive Analysis*

2.9. Related Work

In an attempt to enhance prediction accuracy, researchers have proposed a variety of methods to estimate traffic volume, speed, and travel time. Seven machine learning models were employed and evaluated for traffic speed prediction analysis of Thesalonik City of Greece (Fadilah, 2022). According to Fadilah, predicting traffic speed at each time step requires a lot of computational power and time, and then in his study data was processed simultaneously. As a result, tree models, NN, GB, SVR, RF, and LR were employed. The best models were the NN and GB models, but the researcher selected the GB model as it runs quicker than the NN model. Also, Fadilah's result shows redicting the speed per segment is the most effective method for estimating traffic speed according to data used.

Meena et al. (2020) intended to develop the traffic data models in order to work on the traffic flow forecast problem. According to Meena et al., current traffic flow prediction techniques rely on certain traffic prediction models and are still insufficient for handling practical applications. To analyze the large amounts of data for the transportation system, they suggested deep learning, genetic, machine learning, and soft computing algorithms in their work. The best models, with accuracy scores of 88%, 88%, and 91%, were therefore those using machine learning techniques: Decision Tree, SVM, and Random Forest models respectively.

In order to predict traffic flow, Rajabhushanam & Ramchandra (2021) proposed Deep Auto encoder (DAN), Deep Belief Network (DBN), and Random Forest (RF), which they then applied to an online dataset. In order to forecast the traffic flow in a given zone, Rajabhushanam & Ramchandra took into account significant weather, temperature, zone name, and day factors. The accuracy, precision, RMSE, and MSE values were used to assess the proposed system's performance. The random forest (RF) technique outperforms the other two methods.

In order to predict Thessaloniki's traffic status, Salanova et al. (2021) carried out an analysis which compares multiple linear regression, random forests, neural networks, and support vector machines. For this experiment, data were gathered from both fixed and mobile traffic data sources, creating a sizable database that covered the city in both space and time. The researchers summed up the data in 15-minute time windows in order to extract the minimum, maximum, standard deviation, mean, skewness, kurtosis, and frequency of the unique cars that entered each road. After normalizing the data, they used it as training data for machine learning algorithms

that predict mean speeds. It found that the NN and SVR models are more accurate than the LR and RF models when there are significant speed deviations. When there were more significant changes in traffic flow, the NN model adjusted and predicted more accurately, but the SVR model performed better when there were fewer significant changes. Among the three models, the NN model's near-zero prediction errors were the highest. Both linear and nonlinear patterns were handled by the NN and SVR models. When compared to the other models, the RF model produced mostly medium-sized errors, whereas the LR model produced higher errors when traffic fluctuations were a significant factor.

As far as the researcher is aware, there hasn't been any prior local research on traffic prediction systems. On the other hand, local studies have been conducted on intelligent transportation status, traffic management strategies, and other forms of traffic prediction analysis.

The Addis Ababa City Administration's traffic management practices are examined in a study by Hagere Yilma (2014), with special attention to the efforts made to improve traffic problems on the roads and the difficulties encountered during the implementation and execution process that hinder the effective and efficient practice of traffic management. Identified challenges include inadequate resources for operations and implementation, a lack of modern technology to support traffic management practices, and a poor database system. In the Addis Ababa north-south corridor, Kefyalew Tadele (2017) performed an analysis of traffic congestion and its effects on the economy. According to his analysis, the annual cost of congestion, including delays, wasted fuel, and total operating costs for vehicles and passengers, came to about 53.2 million ETB.

The study article by Tezazu Bireda (2018) outlines the primary obstacles to the vehicular transportation system, such as traffic congestion, pollution, and property and life safety. Afterwards, Tezazu suggests implementing ITS to improve the country's mobility, traffic management, and vehicle and network utilization. Tarikwa Tesfa (2020) used machine learning algorithms to predict the severity of accidents and pinpoint the primary causes of accidents in Addis Ababa. The researcher used and tested seven classification algorithms: AdaBoost, K Nearest Neighbor, Random Forest, Decision Tree, Naïve Bayes, and Logistic Regression. The data was gathered from Addis Ababa sub-city police departments between 2017 and 2020. According to Tarikwa's experiment, random forest achieved a higher F1 score of 93.76% using a hyperparameter-tuning 5-fold cross-validation technique. Bus arrival time prediction was the

topic of experimental research conducted by Habtu Reda (2021) using machine learning techniques. Habtu trained and built the model using the same dataset, which contained 140,000 instances, as used in this study. The random forest algorithm performed better according to the experiment's results, with an R-squared score of 0.994, MAE of 0.812, RMSE of 3.780, and MSE of 14.28.

The section on the literature review covers theoretical concepts related to traffic speed, intelligent transportation systems, and their constituent parts, in particular traffic speed forecasts, as well as classical and ensemble machine learning methods and technologies. Finally, global and local traffic speed prediction related researches. Research on traffic speed prediction, both locally and globally, has been conducted. The best performance in the global literature reviewed for this work has been demonstrated by ensemble algorithms based traffic speed prediction, particularly random forest and gradient boosting techniques. To the best of the researcher's knowledge, no previous local study on traffic prediction systems has been done. However, studies on intelligent transportation status, traffic management strategies, and other types of traffic forecast analysis have been carried out locally.

CHAPTER THREE: RESEARCH METHODOLOGY

According to Davidavi꞉cie̩ne (2018), scientific research is conducted using logic, reason, and a methodical examination of the available data. Extensive analysis and carefully designed procedures are characteristics of scientific research (Pandey, 2015). The progress of methodological procedures is making scientific research more demanding, selective, and reliable. Researchers have contributed to these methodological advancements by strongly believing that the best approach to solving many problems is to conduct an unbiased study of every relevant fact (Davidavi꞉cien˗e, 2018). The methods and procedures used in predictive analysis, such as traffic speed prediction, have a significant impact on the accuracy, credibility, and reliability of the results. A number of procedures or steps are involved in this study process in order to complete the study, including defining the problem domain, conducting a thorough literature review, formulating the objective for traffic speed prediction, preparing the collected traffic dataset, and developing efficient traffic speed prediction models.

3.1. Research Design

Overview

In experimental method of conducting research, one or more independent variables are applied to one or more dependent variables in order to quantify the impact of those manipulations on the dependent variables. Researchers are able to deduce a relationship between these two variable types based on the effect of the independent variables on the dependent variables observed (Blakstad O., 2023). One well-known example of an experimental research design is experimental quantitative research design. Some features of an experimental quantitative study include the variables' nature and relationships, experimental treatments that alter the independent variable, and measurements of the dependent variable both before and after the changes to the independent variable.

Using machine learning techniques, this study follows an experimental research approach and then draw a conclusion based on quantitative data analysis. The precision with which one can analyze the relationship between and among variables and make that analysis as objective as possible is the essence of experimental design, and it's perhaps the main reason researchers choose to design and conduct experiments (Mishra & Datta-Gupta, 2018). Based on this idea, this research aims to observe the effect of one independent variables on traffic speed which is the

dependent variable. This process considers several steps based on research questions. The problem is defined and understood in part one, which leads to the formulation of the objective by reviewing various pieces of literature. The second part of the process involves feature selection, data transformation, and preprocessing before the prepared data set is divided into training and testing sets. Create and adjust hyper-parameters for maximum efficiency of the model, and choose an iteration strategy—such as learning rate—to get the best hyper-parameters. The best model is then selected after a model has been developed and compared using evaluation metrics. A report will be written ultimately based on the research's conclusions and findings.

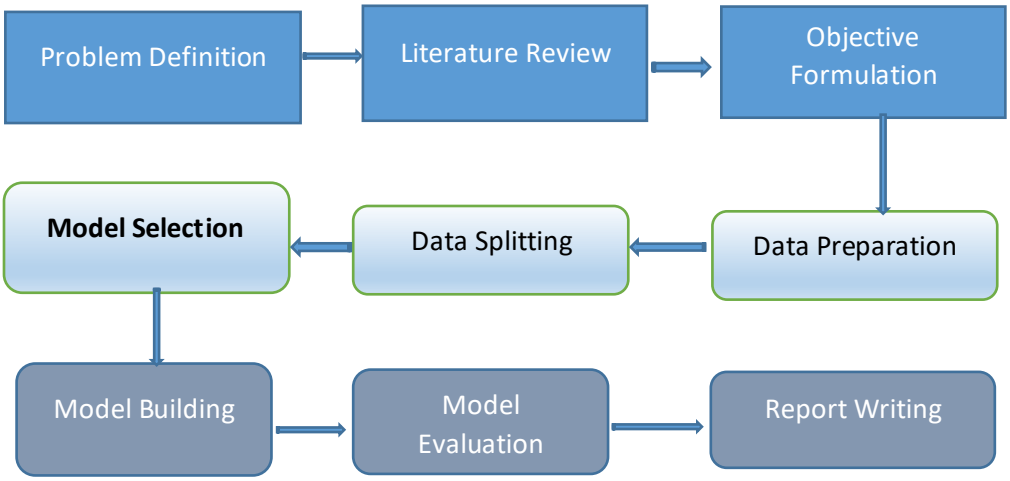


Figure 3.1*: Research Design*

3.2. Data Preparation

Data preparation is fundamental stage of data analysis (Zhang et.al., 2003). The essential phase of the ML lifecycle is data preparation. Data is first gathered from a variety of sources, then cleaned junk data is transformed into real-time machine learning projects to find insights or predict the future. Additionally, machine learning aids in the construction of datasets, proper data transformation, and the discovery of patterns in data that may be used to generate predictions. Data preparation must be done in particular series of steps which includes different tasks.

3.2.1. Data Collection

According to a survey conducted by Zhu Li et.al. regarding data analytics in ITS, they divide it into data gathering, analyzing, and applying layers. Hence, data collection layer is the basis for ITS architecture (Zhu Li et.al.,2018). ITS based data can be collected through video surveillance,

remote sensing, GPS. Among these, data from GPS is suitable for data collection with location tracking. Combined with geographic information system(GIS) or other relevant map displaying technologies, data collected via this technology can be applied to a variety of traffic-related problems, for example, traffic speed prediction. Among issues affecting the traffic flow variation, vehicle speed is the main factor of traffic flow characteristics. These components of traffic flow, serve for traffic system evaluation and optimization (Li et.al., 2022).

The dataset used for this study was collected from 20 buses of Sheger Public Transport within one-year of 2021(January – December) each day, from 6:00 a.m. to 7:30 p.m and then published by Addis Ababa Science and Technology University (AASTU). This GPS data was used for bus station arrival prediction which gathered by Trans-link, a GPS tracking company. The dataset was in comma-separated format (CSV) file format and has 12 attributes and 140,000 total number of records of initial and destination within 123 routes in Addis Ababa. The selected vehicles have been equipped with the AVL system (Automatic Vehicle Location), which keeps track of each bus's exact GPS coordinates (Habtu, 2021).

3.2.2. Understanding Domain Problem

The key to knowing business is to have a thorough awareness of the general process and methods used to solve problems. This can be learned through a variety of activities, including observations, document analysis, and conversations with subject-matter experts. It is valuable to grasp the problem domain in order to better understand the data that will be processed.

There are many problems with the transportation system in the city of Addis Ababa. Lack of transportation facilities is inevitable due to the economy. Vehicle speed and the amount of traffic congestion on the roads are not reliably forecast. This severe issue has not been thoroughly investigated. Additionally, the city of Addis Ababa's transportation administration system relies on outdated manual systems, which led to poor transportation service delivery.

Accurate traffic prediction is essential in this regard since vehicle speed needed to travel between points on a road network is a crucial performance indicator in intelligent transportation and cutting-edge traveler information systems. The vehicle speed in an urban setting depends on the amount of traffic and the condition of the road, which might vary greatly because of congestion.

At another level, one also observes daily patterns (e.g., rush hours), weekly patterns (e.g., weekday versus weekend), and seasonal patterns. Therefore, capturing these features while modeling trip speed can have an immediate impact on commercial transportation companies that distribute goods by allowing them to optimize their routes to be more efficient and minimize environmental impact (Gmira et. al.,2017).

Several cities across the world have started to put in place real-time monitoring systems that track the positions of every car in their fleet and make that information available online. This research proposes a trip speed prediction methodology that uses data collected from a GPS device put inside a delivery Sheger Bus to feed machine learning models. It is anticipated that the additional information from data analysis will improve the accuracy of these models' predictions of travel speeds.

3.3. Feature Selection

The success or failure of ML is determined by a variety of factors, including feature selection, careful data collection, and preprocessing. The data collected for this research work only on road traffic and collected on a daily data collection plan. Features with temporal and spatial characteristics are used as features/independent variables are used and based on the objective of the study i.e. to predict traffic speed average speed is used as dependent variable. Therefore, dataset used for this research has 12 variables, in which, 11 independent called labels and 1 is dependent called target variable. Following feature selection, additional preprocessing is performed on the traffic data to filter out all columns (features) that are not very important for analysis.

A	B	C	D	E	F	G	H	I	J	K	L
DayofWeek	TimeRange	Beginning Time	Mileage	Initial_latitude	Initial_longitude	Initial_longitude	Final_longitude	End Time	total_time	Avg_Speed	Month
5	1	12	11.6	8.99131	38.854141	9.066715	38.865112	46	0.56611111	20.4906771	1
5	1	0	23	9.066558	38.86467	9.066586	38.865231	27	1.45333333	15.8256881	1
5	1	53	11.84	9.066731	38.864773	9.021588	38.801731	28	0.58777778	20.1436673	1
5	1	40	3.64	9.021976	38.80088	9.037561	38.774097	48	0.12833333	28.3636364	1
5	1	22	7.21	9.037603	38.774044	9.070862	38.739017	50	0.47611111	15.1435239	1
5	1	52	0.82	9.070875	38.739059	9.06986	38.73901	57	0.08888889	9.225	1
5	1	8	6.15	9.069852	38.738998	9.046659	38.762901	33	0.425	14.4705882	1
5	1	3	1.43	9.028021	38.790092	9.021267	38.801033	9	0.09722222	14.7085714	1
5	1	22	0.16	9.021267	38.801033	9.020801	38.802307	23	0.02444444	6.54545455	1
5	1	31	12.87	9.020803	38.802296	9.06669	38.865128	28	0.95333333	13.5	1
5	1	7	11.41	9.066716	38.865166	9.020481	38.803288	48	0.67277778	16.9595376	1
5	1	12	12	9.020543	38.803135	9.066687	38.865154	9	0.95277778	12.5947522	1
5	1	6	11.46	9.066747	38.864861	9.020774	38.802864	45	0.64333333	17.8134715	1
5	1	54	23	9.020793	38.802872	8.991427	38.855087	59	1.09555556	20.9939148	1
5	1	0	4.06	9.038613	38.877792	9.06888	38.875755	8	0.13722222	29.5870445	1
5	1	37	1.61	9.068903	38.875774	9.066371	38.865421	41	0.08	20.125	1
5	1	5	3.51	9.066387	38.865398	9.066604	38.864716	18	0.21555556	16.2835052	1
5	1	26	11.91	9.066598	38.864708	9.020667	38.802551	4	0.64388889	18.4969802	1
5	1	16	12.79	9.020548	38.802685	9.06664	38.865208	53	0.61555556	20.7779783	1
5	1	27	11.7	9.066709	38.864769	9.021918	38.800968	3	0.59777778	19.5724907	1
5	1	38	0.22	9.021915	38.800953	9.02084	38.802265	39	0.02	11	1
5	1	52	4.86	9.020823	38.80228	9.032675	38.843742	9	0.28166667	17.2544379	1
5	1	15	8.14	9.03268	38.843807	9.066731	38.864925	34	0.33111111	24.5838926	1
5	1	41	15.5	9.066745	38.864922	9.037701	38.774048	42	1.00722222	15.3888583	1
5	1	45	4.19	9.037766	38.77412	9.020841	38.802311	10	0.41833333	10.0159363	1
5	1	22	12.8	9.020852	38.802303	9.066425	38.865265	3	0.68722222	18.6257074	1
5	1	47	11.71	9.066394	38.865314	9.02171	38.801693	23	0.60777778	19.2669104	1
5	1	14	0.21	9.021729	38.801704	9.021159	38.801785	17	0.05055556	4.15384615	1
5	1	33	11.03	9.020986	38.802052	9.068135	38.875046	13	0.66666667	16.545	1

Figure 3.2. *Sample Data Used for the Study.*

The experimental process of this study then selects and appropriately masks the input dataset to ensure that the chosen data satisfy all of the requirements of the research goals. Prior to the assessment, every machine learning (ML) model will be evaluated and its hyper-parameters adjusted to yield the optimal result after training the data input. Ultimately, the prediction outcome will be utilized to evaluate the effectiveness of the ML models and identify the optimal conditions for route based prediction.

Table 3.1: *Feature Selection*

No	Feature name	Data type	Description
1	total_time	Numerical	Total time of the trip
2	Time Range/Peak time	Numerical	Shows the time of crowding/traffic congestion
3	Mileage	Numerical	Total distance in kilometer of the trip
4	Initial Latitude	Numerical	Source latitude of the vehicle
5	Initial Longitude	Numerical	Source longitude of the vehicle
6	Final latitude	Numerical	Destination latitude of the vehicle
7	Final longitude	Numerical	Destination longitude of the vehicle
8	DayofWeek	Numerical	Day of the trip
9	Month	Numerical	Month of the trip
10	Beginning Time	Numerical	Initial time of the trip
11	End Time	Numerical	End Time of the trip
12	Average Speed	Numerical	Deriving average speed of a bus

3.4. Data Preprocessing Using Machine Learning

The method used to process and assess the chosen data is described in this phase. As the outcomes of the algorithms were compared, some evaluation factors were also used to determine which algorithm has the lowest error (Fadilah, 2022).

Initially, the selected data were read using Pandas' library and converted into a data frame after important python libraries are imported.

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import time
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler, MinMaxScaler
from sklearn.preprocessing import LabelEncoder
from sklearn.neighbors import KNeighborsRegressor
from sklearn.neural_network import MLPRegressor
from sklearn.svm import SVR
from sklearn.ensemble import RandomForestRegressor, GradientBoostingRegressor
from sklearn.metrics import r2_score, mean_absolute_error, mean_squared_error
import warnings
warnings.filterwarnings(action='ignore')
```

```
data = ('final_data_from_sheger_transport.csv')

df = pd.read_csv(data)
```

Figure 3.1: *Read Data Using Pandas Library*

3.4.1. Exploratory Analysis

The dataset read and converted to data frame is then visualized using different python command which then used for further analysis.

```
df.shape
```

```
(140707, 12)
```

```
df.head()
```

	DayofWeek	TimeRange	Beginning Time	Mileage	Initial_latitude	Final_latitude	Initial_longitude	Final_longitude	End Time	total_
0	5	1	12	11.60	8.991310	38.854141	9.066715	38.865112	46	0.56
1	5	1	0	23.00	9.066558	38.864670	9.066586	38.865231	27	1.45
2	5	1	53	11.84	9.066731	38.864773	9.021588	38.801731	28	0.58
3	5	1	40	3.64	9.021976	38.800880	9.037561	38.774097	48	0.12
4	5	1	22	7.21	9.037603	38.774044	9.070862	38.739017	50	0.47

```
1 df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 140707 entries, 0 to 140706
Data columns (total 12 columns):
 #   Column             Non-Null Count   Dtype  
---  ------             --------------   -----  
 0   DayofWeek          140707 non-null  int64  
 1   TimeRange          140707 non-null  int64  
 2   Beginning Time     140707 non-null  int64  
 3   Mileage            140707 non-null  float64
 4   Initial_latitude   140707 non-null  float64
 5   Final_latitude     140707 non-null  float64
 6   Initial_longitude  140707 non-null  float64
 7   Final_longitude    140707 non-null  float64
 8   End Time           140707 non-null  int64  
 9   total_time         140707 non-null  float64
 10  Avg_Speed          140590 non-null  float64
 11  Month              140707 non-null  int64  
dtypes: float64(7), int64(5)
memory usage: 12.9 MB
```

```
|: 1 df.describe().T
```

[73]:

	count	mean	std	min	25%	50%	75%	max
DayofWeek	140707.0	2.504943	1.796989	0.000000	1.000000	2.000000	4.000000	6.000000
TimeRange	140707.0	2.907744	1.386030	1.000000	2.000000	3.000000	4.000000	5.000000
Beginning Time	140707.0	27.939292	17.936755	0.000000	12.000000	28.000000	43.000000	59.000000
Mileage	140707.0	10.068103	29.311526	0.000000	3.460000	10.470000	13.160000	10605.000000
Initial_latitude	140707.0	8.999760	0.042827	8.747010	8.961055	9.009662	9.021904	9.073830
Final_latitude	140707.0	38.776208	0.058420	38.668251	38.743702	38.764179	38.803160	38.990940
Initial_longitude	140707.0	9.000592	0.042298	8.746980	8.961244	9.009721	9.021912	9.073930
Final_longitude	140707.0	38.775976	0.058403	38.668259	38.742781	38.763584	38.803246	38.990913
End Time	140707.0	31.339372	18.131988	0.000000	16.000000	31.000000	47.000000	59.000000
total_time	140707.0	0.572561	0.454752	0.000000	0.208333	0.544444	0.768333	14.999444
Avg_Speed	140590.0	16.774217	6.995982	0.000000	12.490355	16.897361	21.101170	707.026186
Month	140707.0	5.939271	3.134082	1.000000	3.000000	6.000000	9.000000	11.000000

```
1 df.columns
```

```
Index(['DayofWeek', 'TimeRange', 'Beginning Time', 'Mileage',
       'Initial_latitude', 'Final_latitude', 'Initial_longitude',
       'Final_longitude', 'End Time', 'total_time', 'Avg_Speed', 'Month'],
      dtype='object')
```

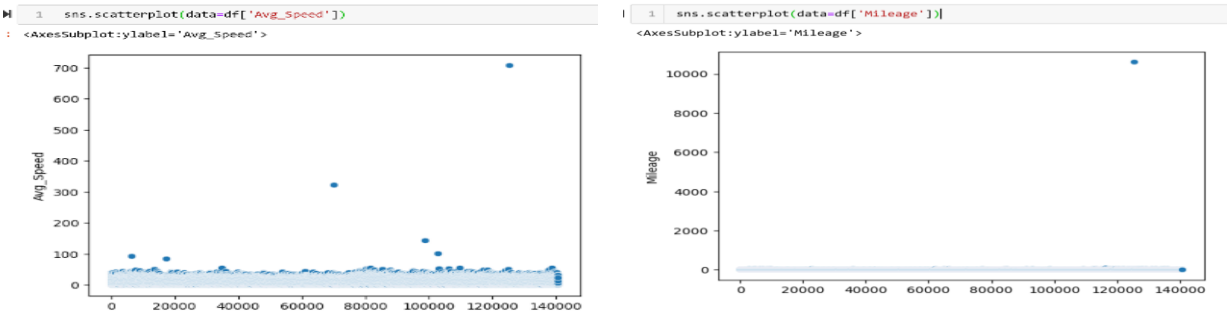
Figure 3.2: *Some Exploratory Analysis*

3.4.2. Data Cleaning

The data used for the study consists of 12 columns or features, as well as 140707 records, as seen in figure 3.2 above. Furthermore, all of the data's features are numerical type, and the data description indicates that the maximum values of the "Mileage" and "Avg_Speed" features, as well as their standard deviation, appear to have outlier values. Techniques for detecting and eliminating outliers should therefore be applied since these values may introduce noise into the models' ability to predict outcomes.

Outliers in a pandas DataFrame can be identified and eliminated using a variety of methods. Among the techniques are the interquartile range (IQR) and the z-score. In this study, IQR is used for analysis. The interquartile range, or IQR, is a statistical measure of the variation between a dataset's first and third quartiles (Priya, 2022). The dataset's first quartile represents the 25th percentile, and the third quartile represents the 75th percentile.

The following Python command finds and filters records that have outliers for the observed features of different maximum and standard deviation values.



```
# To detect and remove 'AVg_Speed' outlier values
IQR = 21.101170-12.490355
Q1 = 12.490355 - 1.5*IQR
Q3 = 21.101170 + 1.5*IQR
print(Q1)
print(Q3)
```

```
-0.42586750000000073
34.0173925
```

```
df_filtered = df[(df['Avg_Speed']>Q1) & (df['Avg_Speed']<Q3)]
print(df_filtered)
# Output
```

```
[139493 rows x 12 columns]
```

```
# To detect and remove 'Mileage' outlier values
IQR = 13.160000-3.460000
Q1 = 3.460000 - 1.5*IQR
Q3 = 13.160000 + 1.5*IQR
print(Q1)
print(Q3)
```

```
-11.09
27.71
```

```
df_filtered = df[(df['Mileage']>Q1) & (df['Mileage']<Q3)]
print(df_filtered)
```

```
[137189 rows x 12 columns]
```

The result above shows the defined lower and upper limits and filtered data frame df to only the data points in the interval of Q1 and Q3. Therefore, for 'Avg_Speed' feature 140707 – 139493 = 1214 records those out of the interval have been removed and for 'Mileage' feature (140707 – 137189) = 3518 records those out of the interval have been removed.

Since the dataset being used contains a large number of records, missing values have been handled by filling null values with zero in pandas that contain missing values, as shown below.

```
1 df.isnull().sum()
```

```
DayofWeek              0
TimeRange              0
Beginning Time         0
Mileage                0
Initial_latitude       0
Final_latitude         0
Initial_longitude      0
Final_longitude        0
End Time               0
total_time             0
Avg_Speed            117
Month                  0
dtype: int64
```

```
1 #df = df.dropna(axis = 0)
2 df = df.fillna(0)
```

```
1 df.isnull().sum()
```

```
DayofWeek            0
TimeRange            0
Beginning Time       0
Mileage              0
Initial_latitude     0
Final_latitude       0
Initial_longitude    0
Final_longitude      0
End Time             0
total_time           0
Avg_Speed            0
Month                0
dtype: int64
```

Next, the features that serve as the model's predictor are chosen. Thus, target variable or label which is 'Avg_Speed' is removed from the features and then features are converted to numpy array.

In [89]:
```
1 df.dropna(inplace=True)
2
3 # Split data into features and target
4 X = df.drop('Avg_Speed', axis=1)
5 Y = df['Avg_Speed']
```

Figure 3.3: *Target and Feature Columns*

3.5. Feature Correlation Analysis

For feature correlation analysis, filter method Pearson correlation analysis is employed. A value between -1 and 1 known as a Pearson correlation that quantifies the degree to which two variables are linearly connected (Yemulwar, 2019). When applying correlation analysis to quantitative variables with outlier-free data, the Pearson correlation coefficient is a good option (Turney, 2022). This method was chosen since all of the data used in the study were of the quantitative kind and all records containing values that were outliers had been dropped.

```
#Pearson Correlation Matrix
cor = df.corr()
cor
```

	DayofWeek	TimeRange	Beginning Time	Mileage	Initial_latitude	Final_latitude	Initial_longitude	Final_longitude
DayofWeek	1.000000	-0.353262	-0.005678	0.003699	0.011224	0.004841	0.003158	0.005783
TimeRange	-0.353262	1.000000	-0.001323	0.025538	-0.031743	0.010214	-0.018026	0.004075
eginning Time	-0.005678	-0.001323	1.000000	-0.014740	0.041442	-0.025986	-0.002273	-0.000229
Mileage	0.003699	0.025538	-0.014740	1.000000	0.020022	0.026982	-0.000423	0.024278
Initial_latitude	0.011224	-0.031743	0.041442	0.020022	1.000000	0.338744	0.446387	0.303989
Final_latitude	0.004841	0.010214	-0.025986	0.026982	0.338744	1.000000	0.293593	0.704981
itial_longitude	0.003158	-0.018026	-0.002273	-0.000423	0.446387	0.293593	1.000000	0.348645
inal_longitude	0.005783	0.004075	-0.000229	0.024278	0.303989	0.704981	0.348645	1.000000
End Time	-0.019993	-0.012605	0.103227	0.008977	0.018456	0.006592	0.006274	0.010198
total_time	-0.039603	0.107621	-0.014592	0.323876	0.062598	-0.031958	0.007302	-0.025149
Avg_Speed	0.082507	0.013924	-0.084721	0.360534	0.014940	0.192439	0.039230	0.190512
Month	0.010478	0.238339	-0.004732	0.009940	-0.026417	0.013154	-0.025590	0.010849

```
# Visualizing Features Correlation Using heatmap
sns.set(rc={'figure.figsize':(30,10)}, font_scale=2)
sns.heatmap(df.corr(), annot=True, vmin= -1.0, cmap='mako')
plt.title('Data Correlation')
plt.show()
```

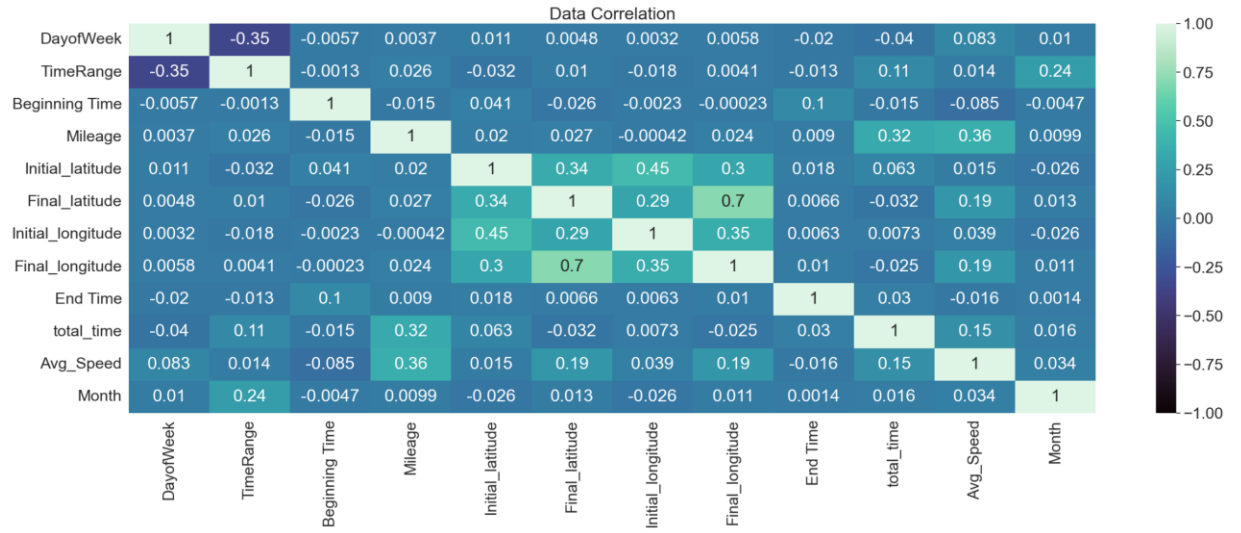


Figure 3.4: *Features Correlation Analysis*

A significant dependency or association between the independent variables or predictor variables must be taken into account after applying data cleaning and receiving the cleaned dataset in order to show the presence or absence of multi-collinearity. Results from multi-collinearity may be unreliable since the data you're comparing are typically identical. The above-mentioned correlation matrix and heat map demonstrate that the features do not show multi-collinearity. Additionally, some predictor variables have a moderate correlation and a higher number have a weak correlation when the degree to which each predictor variable is correlated with the target variable and with itself is taken into consideration. Consequently, all of the features are used to build the model.

3.6. Model Selection

Regression machine learning techniques were employed because the speed value, the attribute for which this study requires prediction, is continuous. Moreover, machine learning methods, comprising NN, GB and RF were used as they shown promising results according to evaluation of relevant works on traffic speed prediction (TSP) topic. The chosen regression predictive analysis techniques were applied during model building. To determine which ML techniques

would be best for the data, these selected algorithms were tested and compared with SVM and KNN classical ML algorithms.

3.7. Model Training and Testing

Machine learning requires both training and test sets of data. Having accurate data is essential for creating successful models since machine learning algorithms learn from data. We can assess the effectiveness of our models and gain understanding of how they operate with the use of training and test data sets (Zabolotnyy, 2023).

3.7.1. Train-Test Split

A number of variables, such as data volume, diversity, and velocity, as well as the nature of the task at hand, can have an impact on how well machine-learning algorithms perform (Habtu, 2021). It separates the datasets into a testing set and a training set. The method can learn from the data that will later be mapped to other new or undiscovered data because it is trained using the training set. How well the model predicts on this subset is assessed using the test dataset (Brownlee, 2020).

The traffic dataset, which consists of 140707 preprocessed records, which is large; therefore, it is divided into two sets: a training set with 70% of the records and a testing set with 30%. As a result, 42213 instances were used as the testing set and 98494 instances from the dataset's total instances were used as the training set, as shown in the following figure.

```
1 print(X_train.shape)
2 print(X_test.shape)
3 print(Y_train.shape)
4 print(Y_test.shape)
5
```

```
(98494, 11)
(42213, 11)
(98494,)
(42213,)
```

Figure 3.7: *Read Data Using Pandas Library*

3.7.2. Performance Evaluation Metrics

An evaluation of a model's performance and effectiveness is known as a model evaluation. There are numerous metrics for assessing the performance of prediction models. After training, the

model should be tested with never-before-seen (unseen) data to determine its effectiveness. This makes it possible to determine if the model is effective at forecasting new data or only data that has already been fed into it but hasn't been seen before. To make sure the designed model forecasts correctly, the model should be rigorously evaluated. Mean squared error, mean absolute error, determination coefficients, and root mean squared error are the most often utilized performance evaluation metrics for regression prediction issues (Goyal, 2021).

Root Mean Square Error (RMSE), Mean Absolute Error (MAE), and Mean Square Error evaluation metrics are used to evaluate all learning models, which are defined in the following equations

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{i}^{n}\left(X_i - \hat{X}_i\right)^2} \qquad 3.2$$

$$\mathbf{MAE} = \frac{1}{n}\sum_{i=1}^{n}\left|X_i - \hat{X}_i\right| \qquad 3.3$$

$$\mathrm{MAPE} = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{X_i - \hat{X}_i}{X_i} \times 100\%\right| \qquad 3.4$$

where X_i denotes the observed traffic speed at time i, and X_i prime denotes the forecast at next i.

$$R^2 = \frac{SS_{reg}}{SS_{tot}} = \frac{\sum_j (y_j - \bar{y})^2}{\sum_i (y_i - \bar{y})^2} \qquad 3.5$$

Predicted labels are indicated by the letter y_j, whereas real labels are indicated by the letter y_i. SS_{reg} is the regression sum of squares, and SS_{tot} is the total square value, which is proportional to the variance of the data.

3.8. Analysis Method

3.8.1. Analysis Tool

Data analysis is used to extract relevant information from data so that sensible judgments can be made. Sklearn 1.3.0 and Python 3.9.13 in anaconda environment were utilized for the analysis in this study.

Python Libraries

1. **Pandas**

Pandas is a Python package designed primarily for working quickly and logically with relational or labeled data. It is an open-source Python package that offers high-performance, user-friendly data structures and data analysis tools for the labeled data in the Python programming language. Pandas creates a Python object with rows and columns called a data frame from the data in a CSV, TSV, or SQL database. The Python NumPy library is the foundation upon which this library is based (nikhilaggarwal3, 2023).

2. **NumPy**

Another popular library for numerical analysis is NumPy, which is a helpful way to store common multi-dimensional data. It's a package for general-purpose array processing. This package contains high-performance multidimensional array objects and tools for working with the arrays.

3. **Matplotlib**

Object-oriented API The Python plotting library called Matplotlib offers an API that enables you to incorporate plots into programs. The Python programming language's integration of MATLAB is strikingly similar.

4. **Seaborn**

Seaborn, a Matplotlib modification with more sophisticated capabilities, is a Python library. The Matplotlib-based data visualization library is what offers a high-level interface for creating eye-catching and educational statistical visualizations (Desai, 2020).

Sklearn

Python's Scikit-learn (Sklearn) library for machine learning is the most effective and reliable. Through a consistent Python interface, it offers a variety of effective techniques for statistical modeling and machine learning, such as dimensionality reduction, clustering, and classification.

5. **Haversine**

It is a python package that used to calculate the distance between two geolocations using latitude and longitude.

3.8.2. Package Management and Running Environments

Package Management

Anaconda for scientific computing (data science, machine learning applications, large-scale data processing, predictive analytics, etc.), Anaconda is a free and open-source distribution. Data-science packages are included in the distribution that are compatible with Windows, Linux, and macOS. To make package management and deployment easier, Anaconda uses the Python and R programming languages.

Jupyter Notebook

Jupyter is open source tool that it supports—Jupyter, Python, and R—are referenced in the name of the tool. Python programs can be made using the IPython kernel, which is included with Jupyter.. This notebook allows you to create and share documents with live code, equations, visualizations, and text. Jupyter Notebooks were created by the IPython project, which previously had an IPython Notebook project of its own. But you can also use one of the approximately 100 different kernels that are currently available.

Environments

The tools used for implementation were installed on a personal computer Hp Spectre x360 Laptop with Intel(R) Core(TM) i7-7500U CPU @ 2.70GHz 2.90 GHz, 16.0 GB (15.9 GB usable) memory, Intel® HD Graphics and 500 Gigabyte(SSD) hard disk storage capacities. The operating system is Window 10 Pro, x64-based processor.

CHAPTER FOUR: EXPERIMENT IMPLEMENTATION AND RESULTS

The outcomes of the processing phase described in the previous section are presented in this chapter. This section outlines and explains the research's observations, as well as its implementations and experimental findings. Finally, the chapter is concluded by assessing the effectiveness of the regression models and presenting the findings.

4.1. System Architecture

The following is a system architecture that explains the interactions and relationships between several components that collectively provide a certain output related to the study objective.

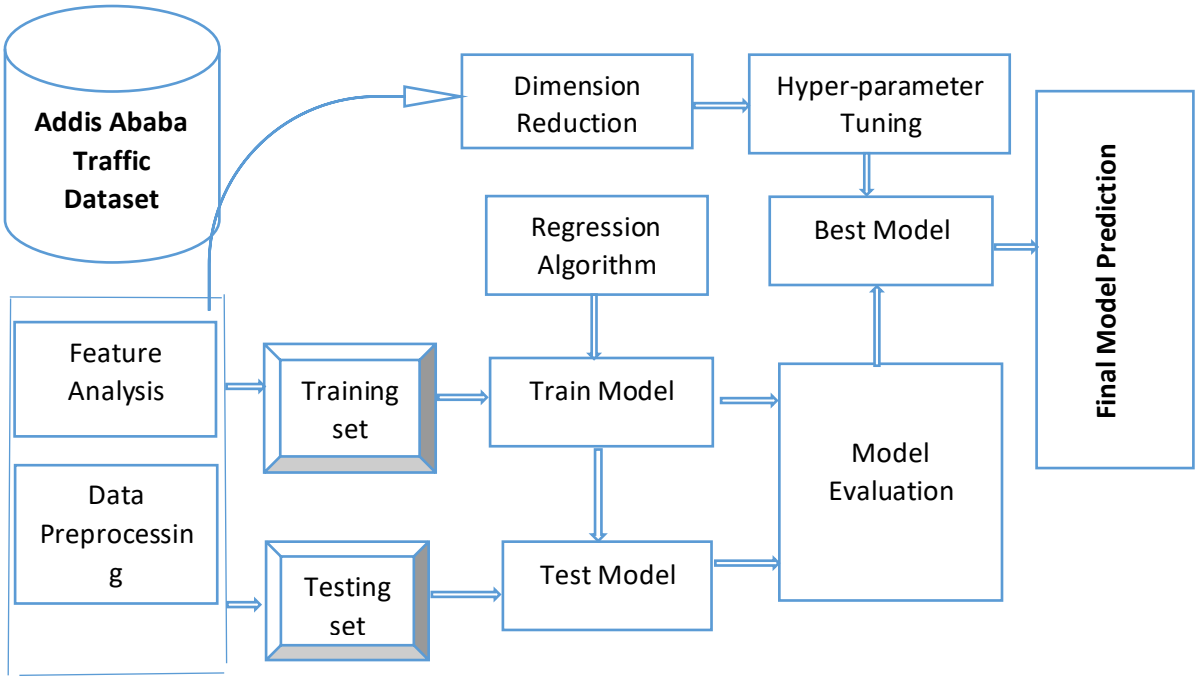


Figure 4.1: *Proposed System Architecture*

The Addis Ababa Traffic Dataset Analysis is the first step in the proposed system architecture, as it is shown in the figure 4.1. Preprocessing and choosing the best features are the next steps in the data preparation phase. Following preparation, the dataset is split into training and testing sets. The testing set is used to evaluate or test the model's performance, while the training set is used to train the model. The phase of model evaluation involves evaluating and comparing the model that was developed using the chosen algorithms. The best-performing model is then chosen to go through additional hyper-parameter tuning. Finally, the selected best model is used for the traffic speed prediction.

4.2. Experiment Result of all Features and Model Parameters

The experiments are conducted using the 12 features. The twelve selected features are 'DayofWeek', 'TimeRange', 'Beginning Time', 'Mileage', 'Initial_latitude', 'Final_latitude', 'Initial_longitude', 'Final_longitude', 'End Time', 'total_time', 'Month' as independent features and 'Avg_Speed' as dependent feature.

At this point, the model has been trained using the dataset's normal distribution using all of the conventional algorithms chosen: KNN regression, SV regression, MLP regression, as well as ensemble algorithms selected; gradient boosting algorithm and random forest regression.

The results of the trained algorithm using the model parameter indicate that the SVM model performed the worst and learned more from the training set than the testing set. While the KNN learnt training set data in the same way as it learned testing set data and produced good results, the MLP model learned testing set data but not as much training set data. In both the training and testing sets, RF and GB learned the data efficiently. The RF model performed the best, followed by the GB model.

As shown in Figure 4.2, the performance of all models is evaluated and compared. The R^2 Score, RMSE (Root Mean Square Error), MAE (Mean Absolute Error) and Mean Squared Error(MSE), achieved by each model is summarized in the following Table 4.1.

Table 4.1. *R^2 Score, RMSE, MAE and MSE Score Values of Baseline Models.*

	KNN	**MLP**	**SVM**	**RF**	**GB**
R^2 Score	81%	91%	45%	99%	95%
RMSE	2.88	2.04	4.98	0.24	1.56
MAE	1.63	1.24	3.44	0.06	0.99
MSE	8.32	4.5	24.81	0.06	2.39

As seen in Table 4.1, the performance of both ensemble algorithm, Random forest achieved the highest score with 99%, followed by GB algorithm. Furthermore, the RF model is the best model based on RMSE, MAE, and MSE score values.

4.3. Features Importance and Experimental Results

By removing unnecessary and redundant data, feature selection offers an efficient method for resolving high-dimensional data analysis problems. This method can also speed up computation, increase learning accuracy, and enable a deeper understanding of the learning model or data

(Sheng et.al., 2018). Thus, seven features with highest importance value are selected for further training and then evaluated as shown in the following figure 4.3.

```
#Feature Importance Selection
```

```
model.fit(X,Y)
print(model.feature_importances_)
```

```
[3.33537602e-03 3.24396628e-03 5.04984997e-03 6.02560969e-01
 3.49891355e-03 2.06453994e-02 1.73969754e-03 3.57522868e-02
 1.32104130e-06 3.24076054e-01 9.61667264e-05]
```

```
df.dropna(inplace=True)

# Split data into features and target
X = df.drop(['Beginning Time', 'Initial_latitude', 'Final_longitude','Avg_Speed'], axis=1)
Y = df['Avg_Speed']

```

```
from sklearn.model_selection import train_test_split

X_train, X_test, Y_train, Y_test = train_test_split(X, Y, train_size = 0.7, random_state = 1)
```

```
print(X_train.shape)
print(X_test.shape)
print(Y_train.shape)
print(Y_test.shape)
```

```
(98494, 8)
(42213, 8)
(98494,)
(42213,)
```

Figure 4.3: *Features Selected for Train_Test_Split*

Following feature importance selection, the performance of each model is assessed and contrasted, as shown in figure 4.4 below. The following Table 4.2 provides a summary of the R2 Score, RMSE (Root Mean Square Error), MAE (Mean Absolute Error), and Mean Squared Error (MSE) obtained by each model.

Table 4.2. *R^2 Score, RMSE, MAE and MSE Score Values of Each Selected Models after Feature Importance.*

	KNN	**MLP**	**SVM**	**RF**	**GB**
R^2 Score	50%	91%	48%	99%	95%
RMSE	4.72	2.00	4.82	0.22	1.56
MAE	3.59	1.20	3.52	0.57	0.94
MSE	22.24	4.00	23.28	0.05	2.36

As seen in table 4.2; Random Forest performance based on; RMSE is improved by 0.02 and MSE improved by 0.07 score values. However, MAE and R^2 is still the same to baseline models result after feature importance selection. Again the best model is RF followed by GB.

4.4. Hyper-Parameter Tuning and Experimental Results

In response to data, machine learning algorithms automatically detect new information and adjust their internal parameters. This class of parameters is called model parameters, or simply parameters. Conversely, some variables need to be pre-established and cannot be changed while the learning process is underway. These parameters are also known as hyper-parameters. The model parameters demonstrate how the input data are converted into the intended output, whereas the hyper-parameters display the structure of the model. Depending on the selection and values of its hyper-parameters, the performance of a machine learning model can significantly change (Elgeldawi et.al., 2021).

Machine learning professionals use the cross-validation (CV) technique to evaluate a model's performance and generalization potential. Iteratively performing model training and evaluation on various training and testing datasets each time entails creating numerous folds, or subsets of datasets, for the datasets. Stronger ML models can be trained by combining cross-validation methods with hyper-parameter tuning (Scott, 2023).

Some of the common methods for tuning hyper-parameters include grid search cross-validation and random search cross-validation. As a result, Grid Search Cross-Validation is a popular tuning method that selects the best set of hyper-parameters for a model by iterating and evaluating through all potential combinations of provided parameters. The input must be a hyper-parameter grid in the form of a Python dictionary with the names and values of the parameter names. A modified version of Grid Search CV is Randomized Search CV. Randomized Search samples random combinations from the hyper-parameter space parameter grid for a predetermined number of iterations, in contrast to Grid Search, which exhaustively trains the model for all combinations from the parameter grid. The Random Search CV trains and evaluates the model for the predetermined number of iterations in order to identify the best model parameters (Scott, 2023). For this study experiment, a random search CV was used. The chosen algorithm is trained with the optimal hyper-parameters following RandomizedSearchCV,

which determines the optimal hyper-parameters based on the sklean library's 'best_params_' attributes.

Table 4.3. R^2 Score, RMSE, MAE and MSE Score Values of Each Selected Models after Hyper-Parameter Tuning.

	KNN	**MLP**	**SVM**	**RF**	**GB**
R^2 Score	67%	81%	67%	99%	99%
RMSE	4.43	2.91	3.87	0.23	0.23
MAE	3.33	1.92	3.29	0.06	0.13
MSE	19.63	3.74	14.99	0.05	0.05

Table 4.3 illustrates that, at this stage, GB performance has significantly improved, outperforming the same as RF model in terms of R2, RMSE, MSE, and MAE score values. Likewise, all models' performance improved except for the MLP.

4.5. Results and Discussion

To address the overall objective of the study, this section will discuss the findings from the previous section. The main objective of this study is to forecast traffic speed using a number of significant characteristics or features aimed at decreasing the negative effects of traffic congestion in Addis Ababa City. As a result, the findings presented in the previous parts and the overall results of the study contribute to addressing the initial research questions.

Question1: What are the most relevant features for predicting traffic speed?

The Addis Ababa Sheger Bus Traffic Dataset includes spatial traffic, daily and monthly time series, as well as the length and distance of each trip. These features were initially used as the training input for a five regression algorithms, such as KNN, MLP, SVR, GB, and RF models. Then the most relevant features were found using sklearn library feature_importances_ attributes and used to build models. Using presented dataset, DayofWeek, TimeRange, End Time, Mileage, Initial_longitude, Final_latitude, total_time, and Moth have been identified as most key factors in predicting traffic speed based on the results of the best model. Accordingly, from this point of view, it can be said that temporal elements, including the duration of the vehicle's trip, peak hours, the trip's end time, and the month or season of the trip have a significant influence on traffic speed. Furthermore, the traffic speed is greatly influenced by geographical or region-specific characteristics, such as the overall travel distance, latitude, and longitude.

Question2: Which machine learning algorithm performs best in predicting traffic speed?

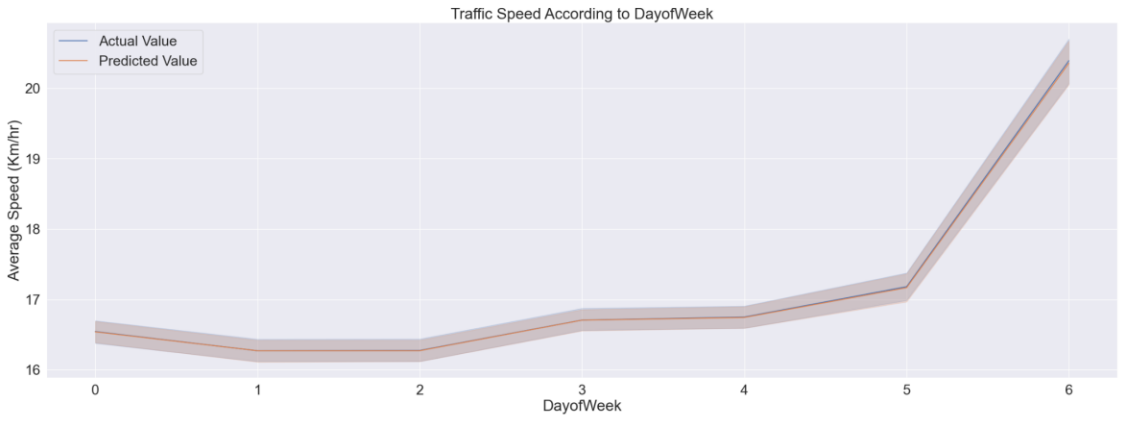
This research question expects to analyze regression algorithms forecasting abilities. Using model parameters, SVM scored worst performance, RF outperforms and GB perform best whereas other models have moderate performance. At feature importance selection phase, it can be concluded that KNN have the worst performance followed by SVM. However, based on the precision testing (using R^2 metric) and recall testing (using RMSE, MAE, MSE metrics) the performance of other models was improved. Based hyper-parameter tuning, GB performance has significantly improved, outperforming the same as RF model in terms of R2, RMSE, MSE, and MAE score values. Likewise, all models' performance improved except for the MLP, and this indicates that NN may automatically learn data without the need for parameter modification

Question3: How is city-wide traffic speed during days of the week and months of the year scenarios?

From research question1 perspective, researcher identified that temporal factors have high influence on traffic speed. Therefore, based on presented features for this study, further analysis of features with temporal characteristics, such as days in a week and months in a year is needed. As a result, predictive analysis of this factors are shown in the following plot chart.

```
In [39]:
customPalette = sns.set_palette(sns.color_palette("hls", 8))
ax = sns.set(rc={'figure.figsize':(30,10)}, font_scale=2)
ax = sns.lineplot(x=X_test['DayofWeek'], y=Y_test, label='Actual Value') # ,
#hue=X_train['DayofWeek']
ax = sns.lineplot(x=X_test['DayofWeek'], y=Y_predictions, label='Predicted Value')
ax.set_title("Traffic Speed According to DayofWeek")
ax.set_xlabel("DayofWeek")
ax.set_ylabel("Average Speed (Km/hr)")

Out[39]: Text(0, 0.5, 'Average Speed (Km/hr)')
```



```
customPalette = sns.set_palette(sns.color_palette("hls", 8))
ax = sns.set(rc={'figure.figsize':(30,10)}, font_scale=2)
ax = sns.lineplot(x=X_test['Month'], y=Y_test, label='Actual Value') # ,
#hue=X_train['DayofWeek']
ax = sns.lineplot(x=X_test['Month'], y=Y_predictions, label='Predicted Value')
ax.set_title("Traffic Speed According to Months of a Year")
ax.set_xlabel("Month")
ax.set_ylabel("Average Speed (Km/hr)")
```

```
[40]: Text(0, 0.5, 'Average Speed (Km/hr)')
```

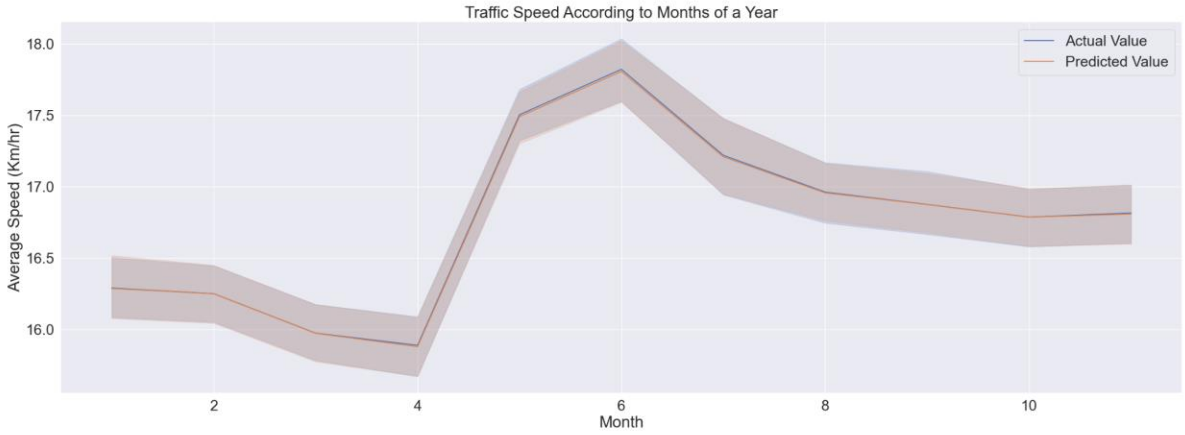
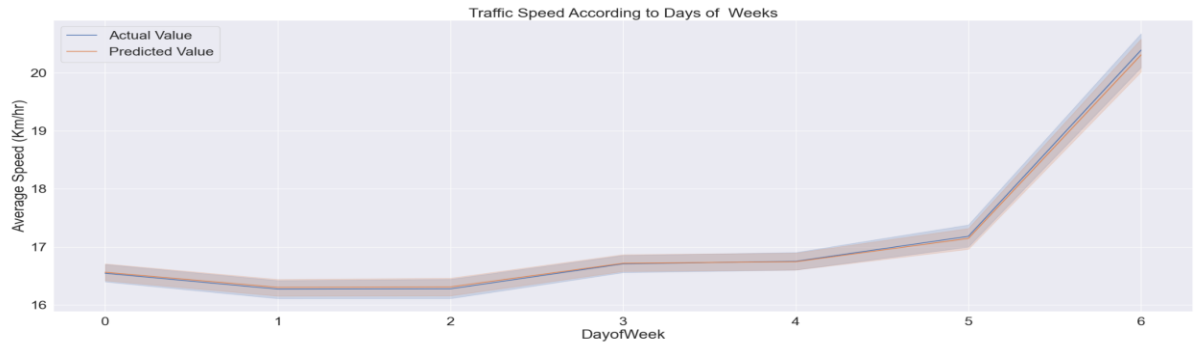


Figure 4.7: *Traffic Speed Analysis During Days of week and Months in a Year at Model Training Phase*



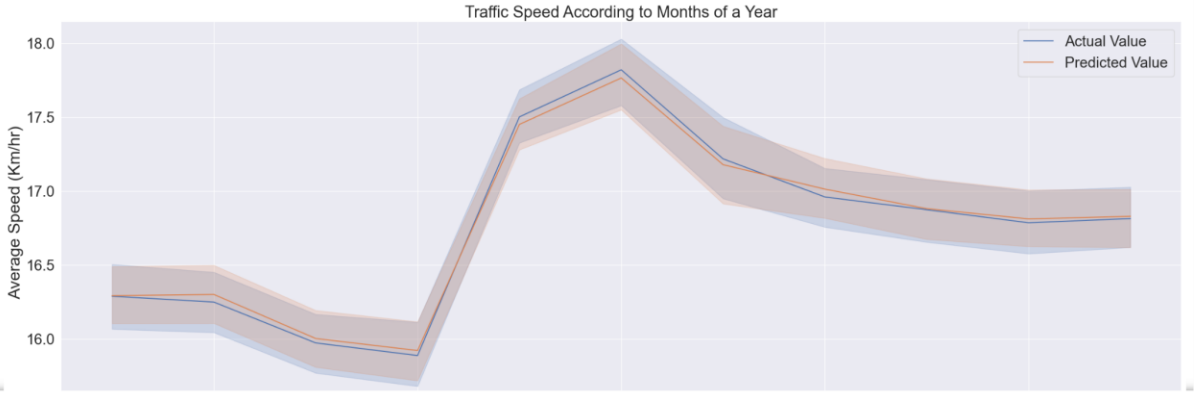


Figure 4.8: *Traffic Speed Analysis During Days of week and Months in a Year at after feature importance identification and hyper-parameter tuning phase*

According to the data analysis result shown in figure 4.7 and 4.8, it is noticeable that traffic speed is comparatively high on weekends and in the months of May, June, and July. Baseline model predicts traffic speed the same level to actual value whereas as after features and parameters change predicted value is slightly different from actual value. As a result, traffic speed is more high on weekends and more less in the month of June and July.

CHAPTER FIVE: CONCLUSION AND FUTURE WORKS

5.1. Conclusion

The study's researcher is aware that the city's expanding urban and industrial populations, which demand more vehicles, are making the problem of traffic congestion worse. And also, given the state of the economy and organizational challenges, constructing road facilities along with ITS infrastructures that complies with international standards is a difficult and long-term solution. As it was described in theoretical framework traffic congestion in the city arisen due to mismatch of road facilities and number vehicles. According to secondary data gathered from the Addis Ababa Traffic management Agency, which was given responsibility for traffic management by the city government, a traffic count has been being done along 50 junctions in the city. But due to lack of permanent electronic traffic recording devices, traffic count was undertaken manually by observers who visually count and record traffic on a hand-held electronic device. Then, in addition to junction count, route analysis should be carried out, which in turn, helps to provide alternative path in case of excessive amount of traffic. One way to address these problems is with traffic speed prediction, which is part of an intelligent transportation system. Therefore, to predict traffic speed in this study, machine learning techniques were used.

Machine learning (ML) regression algorithms were used to train presented dataset and predict traffic speed value. Using the default parameter, feature importance selection, and hyper-parameter tuning, the scoring values of the random forest and gradient boosting algorithms tested and evaluated were the best. As a result, determinant factors with temporal and spatial characteristics have significant impact on traffic speed of Addis Ababa. And also, further analysis of temporal characteristics with this study scope identifies the traffic status in the days of week and moths of a year. In the future, different cases or scenarios as well as strategies can be developed in order to find the most effective method for performing city-wide traffic speed prediction using ML algorithms and other methods to experience intelligent transportation system in Addis Ababa. Machine learning in general and the ensemble models in particular shown positive performance traffic prediction result in this study.

5.2. Recommendations and Future Works

The following recommendations and future works are made to address Addis Ababa's traffic congestion issues and to establish an intelligent transportation system, taking into account the city's general traffic management practices and their shortcomings as well as the results and findings of this research.

5.2.1. Recommendations

- It is crucial to consider the application of all kinds of traffic management techniques when building the road and its infrastructure. Thus, the researcher suggests that the development and implementation of comprehensive intelligent transportation systems are necessary.
- Using installed sensors, the city government should think of network-wide traffic data collecting system, which in turn, suitable for different varies machine learning techniques and also deep learning frameworks.

5.2.2. Future Works

- In order to predict a specific time traffic speed, traffic data must be collected over a period of years using small time steps on various days. This will enable the traffic management agency to dynamically control traffic congestion at different time steps.
- One of the study's future projects might be to include external factors, such as accidents, weather condition information in the input features.
- In light of the findings of this study, the Addis Ababa city government should plan for designing database management system which can be integrated with machine learning, particularly ensemble models, in order to frequently perform route-based traffic speed analyses.
- Finally, based on identified determinant factors those can highly influence traffic speed as per this study result, experimental research which dynamically consider spatial and temporal characteristics of a road traffic should be one of a future work.

References

Nazirkar, R. & Rajabhushanam, C. (2021). Traffic Prediction System Using Machine Learning Algorithms. *CAC, Chennai, India.*

Seada, M. & Emer, T. (2019). Model-Based Urban Road Network Performance Measurement Using Travel Time Reliability: A Case Study of Addis Ababa City, Ethiopia. *American Journal of Civil Engineering and Architecture,* Vol. 7, No. 5, 202-207

Tola, A.M., Demissie, T.A., & Saathoff, F., & Gebissa, A. (2021). Severity, Spatial Pattern and Statistical Analysis of Road Traffic Crash Hot Spots in Ethiopia. *Appl. Sci*. 11, 8828.

Li, Zhu., & Fei, R.Y. (2019). Big Data Analytics in Intelligent Transportation Systems: A Survey.

IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS.

Bedada, B., & Kekeba, T. (2020). Developing Traffic Congestion Detection Model Using Deep Learning Approach: A Case Study of Addis Ababa City Road. *Department of Software Engineering, Big Data and HPC CoE,Addis Ababa Science and Technology University*, Addis Ababa, Ethiopia.

Kashif, N.Q., & Hanan, A. (2013). A Survey on Intelligent Transportation Systems. *Middle-East Journal of Scientific Research*, 15 (5): 629-642, ISSN 1990-923.

Zhang, J. (2011). Data-Driven Intelligent Transportation Systems: A Survey. *IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS*, VOL. 12, NO. 4.

Li, Y. (2017). Intelligent Transportation System(ITS): Concept, Challenge and Opportunity. *IEEE*

3rd International Conference on Big Data Security on Cloud.

Ethiopia Urbanization Review: Urban Institutions for Middle- Income Ethiopia. (2015). *World Bank,* Tech. Report 100238

Muhammad, A., & Joaquim, F. (2016). Intelligent Transportation Systems. *Springer International*

Publishing Switzerland.

Mahmuda A., & Sara M. (2021). A Review of Traffic Congestion Prediction Using Artificial

Intelligence. *Journal of Advanced Transportation.* Article ID 8878011.

Ray, S.(2019). A Quick Review of Machine Learning Algorithms. *International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (Com-IT-Con)*, India, 14th -16th Feb.

Gaurav, M., Deepanjali, Sh., & Mehul, M. (2020). Traffic Prediction for Intelligent Transportation System using Machine Learning. *3rd International Conference on Emerging Technologies in Computer Engineering: Machine Learning and Internet of Things*, IEEE Conference Record 48199.

Trupti, A., & Santosh V.C. (2014). An Overview of Association Rule Mining Algorithms. *(IJCSIT) International Journal of Computer Science and Information Technologies*, Vol. 5 (1), 927-930

Peter, D.C., & Neofytos D. (2021). Precision Medicine in Digital Pathology Via Image Analysis and Machine Learning, *Elsevier Inc.*

Noor, A., Mat, R., Nuraini, S., Khairul, K.I., Suzaimah, R., Mohd, F.A., & Sazali, S. (2021). Gap, Techniques and Evaluation: Traffic Fow Prediction Using Machine Learning and Deep Learning, *Journal of Big Data*, 8:152.

Zeroual, A., Harrou, F., & Sun, Y. (2021). Predicting Road Traffic Density Using a Machine Learning-Driven Approach. *International Conference on Electrical, Computer and Energy Technologies*.

Chungi, L., Yeonjun, K., Seungmin, J., Dongmin, K., & Ross, M. (2015). A Visual Analytics System for Exploring, Monitoring, and Forecasting Road Traffic Congestion. *Journal of Latex Class Files*, VOL. 14, NO. 8.

Wang, Y., Cao, J., Li, W., Gu, T., & Shi, W. (2017). Exploring Traffic Congestion Correlation from Multiple Data Sources, *Pervasive and Mobile Computing*.

Wang, J., Mao, Y., Li, J., Xiong, Z., & Wang W.X. (2015). Predictability of Road Traffic and

Congestion in Urban Areas. *PLoS ONE,* 10(4): e0121825. doi: 10.1371/journal.pone.0121825.

Zeroual, A., Harrou, F., & Sun, Y. Road Traffic Density Estimation and Congestion Detection with a Hybrid Observer-Based Strategy, (2018)

Harrou, F., Zeroual, A., & Sun Y. (2020). Traffic Congestion Monitoring Using an Improved KNN

Strategy, Measurement.

Tezazu, B. (2018). Intelligent Transport System in Ethiopia: Status and the Way Forward. *ICST Institute for Computer Sciences, Social Informatics and Telecommunications Engineering.*

Jain, V., Sharma, A. (2012). Road Traffic Congestion in the Developing World. *ACM*, 978-1-4503-1262-2/12/03.

Gmira, M., Gendreau, M., Lodi, A., & Potvin, J.Y. (2017). Travel Speed Prediction Using Machine

Learning Techniques. *ITS World Congress Montreal*.

Mahesh, B. (2018). Machine Learning Algorithms - A Review. *International Journal of Science and Research IJSR,* ISSN: 2319-7064.

Issam, E., & Martin, J.M. (2015). Machine Learning in Radiation Oncology: Theory and Applications. *Springer International Publishing Switzerland*, DOI: 10.1007/978-3-319-18305-3_1

Bratsas, Ch., Koupidis, K., Salanova, J.M., Giannakopoulos, K., Kaloudis, A., & Aifadopoulou, G. (2019). A Comparison of Machine Learning Methods for the Prediction of Traffic Speed in Urban Places. *Centre for Research and Technology Hellas—Hellenic Institute of Transport*, P.C. 57001 Thessaloniki, Greece.

Liu, Y., & Wu, H. (2017). Prediction of Road Traffic Congestion Based on Random Forest. *10th International Symposium on Computational Intelligence and Design.*

Kustrin, S.A. (2000). Basic Concepts of Artificial Neural network (ANN) Modeling and Its Application in Pharmaceutical Research. *Journal of Pharmaceutical and Biomedical Analysis*, 717–727

Rzeszotko, J., & Nguyen, S.H. (2017). Machine Learning for Traffic Prediction. *Polish-Japanese Institute of Information Technology,* Koszykowa 86, 02-008 Warsaw, Polska.

Boukerche, A., & Wang, J. (2020). Machine Learning-based traffic prediction models for Intelligent Transportation Systems. *Paradise Research Laboratory, School of Electrical Engineering and Computer Science, University of Ottawa*, 800 King Edward Ave., Ottawa, K1N 6N5 ON, Canada.

George, S & Kumar, S. (2020). Traffic Prediction Using Multifaceted Techniques: A Survey. *Springer Science + Business Media* LLC, part of Springer Nature.

Nagy, A.M. & Simon, V. (2018). Survey On Traffic Prediction in Smart Cities. *Pervasive and Mobile* Computing, https://doi.org/10.1016/j.pmcj.2018.07.004

Ayana, M. (2021). Computational Modeling and Analysis of Traffic Crash and Traffic Volume in Addis – Adama Expressway. *Master's Thesis, Department of Computer Science*, Addis Ababa University.

Habtu, R. (2021). Public Bus Arrival Time Prediction Using Machine Learning: In Case of Addis Ababa. *Master's Thesis, Department of Software Engineering*, Addis Ababa Science and Technology University.

Endale, M. (2021). Mobile Data Traffic Prediction Using Multivariate Time Series Data: The Case of LTE Network in Addis Ababa. *Master's Thesis, Department of Telecommunication Engineering,* Addis Ababa University.

Tamene, B. (2020). Video Streaming Data Traffic Prediction by Using Long Short Term Memory

(LSTM) Model: In the case of UMTS Network in Addis Ababa. *Master's Thesis, Department of Telecommunication Engineering*, Addis Ababa University.

Tewodros, K. (2019). Recurrent Neural Network-based Base Transceiver Station Power System Failure Prediction. *Master's Thesis, Department of Telecommunication Engineering,* Addis Ababa University.

Kefyalew, T.A. (2017). Analysis of Traffic Congestion and its Economic Cost in Addis Ababa A Case Study of Meskel Square to Kality Interchange. *Master's Thesis, Department Civil Engineering.* Addis Ababa Science and Technology University.

Andualem, D. (2018). Assessing the Level of Traffic Congestion at major intersections in Addis Ababa: A case for East-West corridor (Lem-Hotel & Urael Junction). *Master's Thesis, Department of Road and Transport Engineering,* Addis Ababa science and Technology University.

Addis Ababa Non-Motorised Transport Strategy 2019-2028. (2018). *Addis Ababa Road and Construction Bureau*.

Kumar, S., Tiwari, P., Zymbler, M. (2019). Internet of Things is a Revolutionary Approach for Future Technology Enhancement: A Review. *J Big Data*, https://doi.org/10.1186/s40537-019-0268-2.

United Nations ESCAP (2021). Intelligent Transportation Systems for Sustainable Development in Asia and the Pacific. pp. 1–38, http:// www. unescap.org/sites/ default/ fles/ ITS. pdf.

Hassn, H., Ismail, A., Borhan, M.N., & Syamsunur, D. (2016). The Impact of Intelligent Transport System Quality: Drivers' Acceptance Perspective. *Int. J Technol*, 7(4):553–61, https://doi.org/10.14716/ijtech.v7i4.2578

Chang, X., Li, H., Rong, J., Huang, Z., Chen, X., & Zhang, Y. (2019). Effects of On-board Unit On Driving Behavior in Connected Vehicle Traffic Flow. *J Adv Transp,* https://doi.org/10.1155/

Miglani, A., & Kumar N. (2019). Deep Learning Models for Traffic Flow Prediction in

Autonomous Vehicles: A Review, Solutions, And Challenges. *Veh Commun.20:100184*, https://doi.org/10.1016/j.

Sun, Y., Li, Z., Li, X., & Zhang, J. (2021). Classifier Selection and Ensemble Model for Multi-Class Imbalance Learning in Education Grants Prediction. *Appl. Artif. Intell*., vol. 35, no. 4, pp. 290-303.

Nti, K., Adekoya, A. F., & Weyori, B. A. (2020). A Comprehensive Evaluation of Ensemble Learning for Stock-Market Prediction. *J. Big Data,* vol. 7, no. 1, pp. 20.

Fawagreh, K., Gaber, M. M., & Elyan, E. (2014) Random Forests: From Early Developments to Recent Advancements. *Syst. Sci. Control Eng.*, vol. 2, no. 1, pp. 602-609.

Mienye, D., & Sun, Y. (2022). A Survey of Ensemble Learning: Concepts, Algorithms, Applications and Prospects. *IEEE Access*, vol.10, pp.99129-99149, doi: 10.1109/ACCESS.2022.3207287.

Li, Q.F., & Song, Z.M. (2022). High-Performance Concrete Strength Prediction Based On Ensemble Learning. *Construct. Building Mater*., vol. 324

Oza, N. C. & Tumer, K. (2020). Classifier Ensembles: Select Real-World Applications. *Inf. Fusion*, vol. 9, pp. 4-20.

Alelyani, S. (2021). Stable Bagging Feature Selection on Medical Data. *J. Big Data,* vol. 8, no. 1, pp. 11.

Mrabah, N. M., Bouguessa & Ksantini, R. (2022). Adversarial Deep Embedded Clustering: On A Better Trade-Off Between Feature Randomness and Feature Drift. *IEEE Trans. Knowl. Data Eng*., vol. 34, no. 4, pp. 1603-1617.

Han, S., & Kim, H. (2019). On The Optimal Size of Candidate Feature Set in Random Forest. *Appl. Sci.*, vol. 9, no. 5, pp. 898.

Brownlee, J. (2020). Train-Test Split for Evaluating Machine Learning Algorithms. *Machine Learning Mastery,* 23 7 2020: https://machinelearningmastery.com/train-test-split-for-evaluating-machinelearning-algorithms/

Mishra, S., & Datta-Gupta, A. (2018). Experimental Design and Response Surface Analysis. *Applied Statistical Modeling and Data Analytics*, 169–193. doi:10.1016/b978-0-12-803279-4.00007-9

Davidavičienė, V. (2018). Research Methodology: An Introduction. Modernizing the Academic Teaching and Research Environment, 1–23. doi:10.1007/978-3-319-74173-4_1

Bratsas, Charalampos, Kleanthis Koupidis, Josep-Maria Salanova, Konstantinos Giannakopoulos,

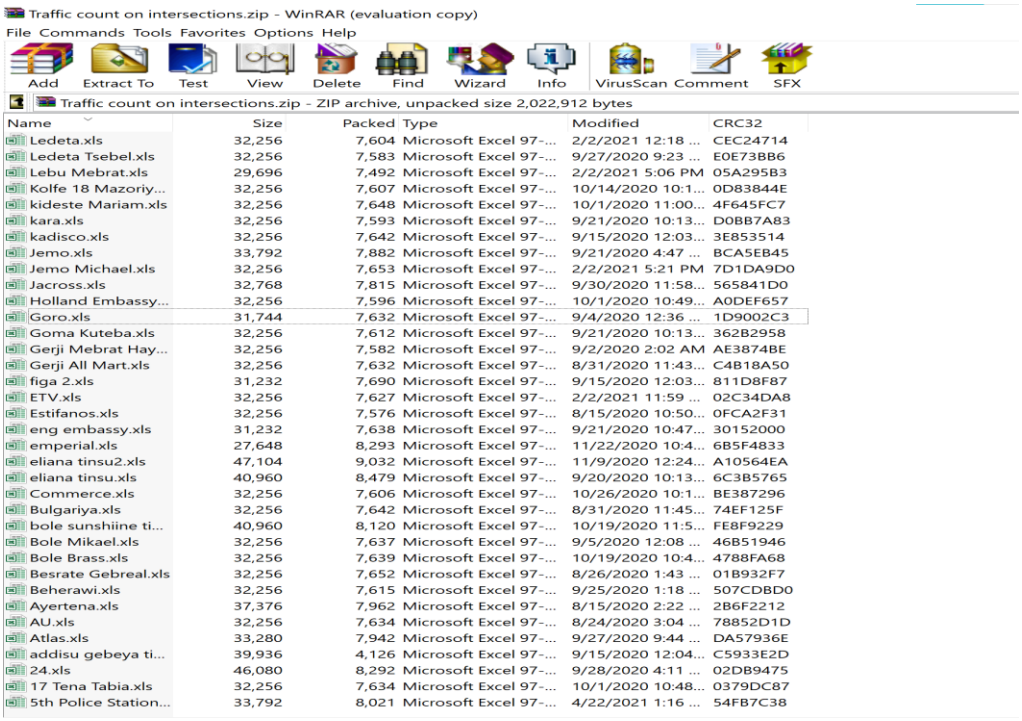
Aristeidis Kaloudis, and Georgia Aifadopoulou. 2020. "A Comparison of Machine Learning Methods for the Prediction of Traffic Speed in Urban Places" *Sustainability* 12, no. 1: 142. https://doi.org/10.3390/su12010142

Boukerche, A., & Wang, J. (2020). Machine Learning-based traffic prediction models for

Intelligent Transportation Systems. *Computer Networks*, 181, 107530. doi: 10.1016/j.comnet.2020.107530

Appendixes

Appendix A: List of Addis Ababa Junction-Point on Which Traffic Count Carried Out



Traffic count on intersections.zip - WinRAR (evaluation copy)

File Commands Tools Favorites Options Help

Add Extract To Test View Delete Find Wizard Info VirusScan Comment SFX

Traffic count on intersections.zip - ZIP archive, unpacked size 2,022,912 bytes

Name	Size	Packed	Type	Modified	CRC32
Ledeta.xls	32,256	7,604	Microsoft Excel 97-...	2/2/2021 12:18 ...	CEC24714
Ledeta Tsebel.xls	32,256	7,583	Microsoft Excel 97-...	9/27/2020 9:23 ...	E0E73BB6
Lebu Mebrat.xls	29,696	7,492	Microsoft Excel 97-...	2/2/2021 5:06 PM	05A295B3
Kolfe 18 Mazoriy...	32,256	7,607	Microsoft Excel 97-...	10/14/2020 10:1...	0D83844E
kideste Mariam.xls	32,256	7,648	Microsoft Excel 97-...	10/1/2020 11:00...	4F645FC7
kara.xls	32,256	7,593	Microsoft Excel 97-...	9/21/2020 10:13...	D0BB7A83
kadisco.xls	32,256	7,642	Microsoft Excel 97-...	9/15/2020 12:03...	3E853514
Jemo.xls	33,792	7,882	Microsoft Excel 97-...	9/21/2020 4:47 ...	BCA5EB45
Jemo Michael.xls	32,256	7,653	Microsoft Excel 97-...	2/2/2021 5:21 PM	7D1DA9D0
Jacross.xls	32,768	7,815	Microsoft Excel 97-...	9/30/2020 11:58...	565841D0
Holland Embassy...	32,256	7,596	Microsoft Excel 97-...	10/1/2020 10:49...	A0DEF657
Goro.xls	31,744	7,632	Microsoft Excel 97-...	9/4/2020 12:36 ...	1D9002C3
Goma Kuteba.xls	32,256	7,612	Microsoft Excel 97-...	9/21/2020 10:13...	362B2958
Gerji Mebrat Hay...	32,256	7,582	Microsoft Excel 97-...	9/2/2020 2:02 AM	AE3874BE
Gerji All Mart.xls	32,256	7,632	Microsoft Excel 97-...	8/31/2020 11:43...	C4B18A50
figa 2.xls	31,232	7,690	Microsoft Excel 97-...	9/15/2020 12:03...	811D8F87
ETV.xls	32,256	7,627	Microsoft Excel 97-...	2/2/2021 11:59 ...	02C34DA8
Estifanos.xls	32,256	7,576	Microsoft Excel 97-...	8/15/2020 10:50...	0FCA2F31
eng embassy.xls	31,232	7,638	Microsoft Excel 97-...	9/21/2020 10:47...	30152000
emperial.xls	27,648	8,293	Microsoft Excel 97-...	11/22/2020 10:4...	6B5F4833
eliana tinsu2.xls	47,104	9,032	Microsoft Excel 97-...	11/9/2020 12:24...	A10564EA
eliana tinsu.xls	40,960	8,479	Microsoft Excel 97-...	9/20/2020 10:13...	6C3B5765
Commerce.xls	32,256	7,606	Microsoft Excel 97-...	10/26/2020 10:1...	BE387296
Bulgariya.xls	32,256	7,642	Microsoft Excel 97-...	8/31/2020 11:45...	74EF125F
bole sunshiine ti...	40,960	8,120	Microsoft Excel 97-...	10/19/2020 11:5...	FE8F9229
Bole Mikael.xls	32,256	7,637	Microsoft Excel 97-...	9/5/2020 12:08 ...	46B51946
Bole Brass.xls	32,256	7,639	Microsoft Excel 97-...	10/19/2020 10:4...	4788FA68
Besrate Gebreal.xls	32,256	7,652	Microsoft Excel 97-...	8/26/2020 1:43 ...	01B932F7
Beherawi.xls	32,256	7,615	Microsoft Excel 97-...	9/25/2020 1:18 ...	507CDBD0
Ayertena.xls	37,376	7,962	Microsoft Excel 97-...	8/15/2020 2:22 ...	2B6F2212
AU.xls	32,256	7,634	Microsoft Excel 97-...	8/24/2020 3:04 ...	78852D1D
Atlas.xls	33,280	7,942	Microsoft Excel 97-...	9/27/2020 9:44 ...	DA57936E
addisu gebeya ti...	39,936	4,126	Microsoft Excel 97-...	9/15/2020 12:04...	C5933E2D
24.xls	46,080	8,292	Microsoft Excel 97-...	9/28/2020 4:11 ...	02DB9475
17 Tena Tabia.xls	32,256	7,634	Microsoft Excel 97-...	10/1/2020 10:48...	0379DC87
5th Police Station...	33,792	8,021	Microsoft Excel 97-...	4/22/2021 1:16 ...	54FB7C38

Appendix B: - Sample of Addis Ababa Intersections Traffic Count Data

Start Date: 9/25/2020
Start Time: 2:30:00 AM
Site Code: 00000000
Comment 1: Default Comments
Comment 2: Change These in The Preferences Window
Comment 3: Select File/Preference in the Main Scree
Comment 4: Then Click the Comments Tab

	MEXICO From North				From East				TORHAYLOCH From South				LEDETA CONDOMINIUM From West			
Start Time	Right	Thru	Left	Peds	Right	Thru	Left	Peds	Right	Thru	Left	Peds	Right	Thru	Left	Peds
2:30 AM	0	204	0	0	0	0	0	0	0	181	84	0	0	0	68	0
4:30 AM	0	299	0	0	0	0	0	0	0	256	64	0	0	0	52	0
6:30 AM	0	211	0	0	0	0	0	0	0	381	46	0	0	0	45	0
10:30 AM	0	244	0	0	0	0	0	0	0	331	49	0	0	0	50	0

Appendix C: - Commands when all features and model parameter used

```
#Model Building and Evaluation Using Default/Model Parameters
```

```
models = {

"                   K-Nearest Neighbors": KNeighborsRegressor(),
"                        Neural Network": MLPRegressor(),
"   Support Vector Machine (RBF Kernel)": SVR(),
"                         Random Forest": RandomForestRegressor(),
"                     Gradient Boosting": GradientBoostingRegressor() # n_estimators=100, learning_rate
}
for name, model in models.items():
    model.fit(X_train, Y_train)
    print(name + " trained.")
for name, model in models.items():
    print(name + " R^2 Score: {:.5f}".format(model.score(X_test, Y_test)))
    print(name + " R^2 Score: {:.5f}".format(model.score(X_train, Y_train)))
```

```
                  K-Nearest Neighbors trained.
                       Neural Network trained.
Support Vector Machine (RBF Kernel) trained.
                        Random Forest trained.
                    Gradient Boosting trained.
                  K-Nearest Neighbors R^2 Score: 0.81435
                  K-Nearest Neighbors R^2 Score: 0.81268
                       Neural Network R^2 Score: 0.90733
                       Neural Network R^2 Score: 0.40120
Support Vector Machine (RBF Kernel) R^2 Score: 0.44616
Support Vector Machine (RBF Kernel) R^2 Score: 0.38683
                        Random Forest R^2 Score: 0.99872
                        Random Forest R^2 Score: 0.98298
                    Gradient Boosting R^2 Score: 0.94653
                    Gradient Boosting R^2 Score: 0.94857
```

```python
from sklearn.metrics import r2_score, mean_absolute_error, mean_squared_error
for name, model in models.items():
    Y_predict = model.predict(X_test)
    print(name + " R^2 Score: {:.5f}".format(r2_score(Y_test, Y_predict)))
    print(name + " RMSE: {:.5f}".format(np.sqrt(mean_squared_error(Y_test, Y_predict))))
    print(name + " MAE: {:.5f}".format(mean_absolute_error(Y_test, Y_predict)))
    print(name + " MSE: {:.5f}". format(mean_squared_error(Y_test, Y_predict)))
# Specifying and creating the MODEL
# Change the model accordingly!
#RF_model = RandomForestRegressor()
#RF_model.fit(X_train, Y_train)
# PREDICTING and VALIDATING the values
#Y_predictions = RF_model.predict(X_test)
#print('R^2 Score =', r2_score(Y_test, Y_predictions))
#print('Mean Absolute Error =', mean_absolute_error(Y_test, Y_predictions))
#print('Root Mean Squared Error =', np.sqrt(mean_squared_error(Y_test, Y_predictions)))
#print('Mean Squared Error =', mean_squared_error(Y_test, Y_predictions))
```

```
                K-Nearest Neighbors R^2 Score: 0.81435
                K-Nearest Neighbors RMSE: 2.88405
                K-Nearest Neighbors MAE: 1.62743
                K-Nearest Neighbors MSE: 8.31775
                     Neural Network R^2 Score: 0.90733
                     Neural Network RMSE: 2.03761
                     Neural Network MAE: 1.23962
                     Neural Network MSE: 4.15184
Support Vector Machine (RBF Kernel) R^2 Score: 0.44616
Support Vector Machine (RBF Kernel) RMSE: 4.98134
Support Vector Machine (RBF Kernel) MAE: 3.43734
Support Vector Machine (RBF Kernel) MSE: 24.81374
                      Random Forest R^2 Score: 0.99872
                      Random Forest RMSE: 0.23971
                      Random Forest MAE: 0.05886
                      Random Forest MSE: 0.05746
                  Gradient Boosting R^2 Score: 0.94653
                  Gradient Boosting RMSE: 1.54782
                  Gradient Boosting MAE: 0.97651
                  Gradient Boosting MSE: 2.39574
```

Appendix D: - Commands and experimental results after Feature Importance Selection

```
7]:
models = {

"                   K-Nearest Neighbors": KNeighborsRegressor(),
"                        Neural Network": MLPRegressor(),
"   Support Vector Machine (RBF Kernel)": SVR(),
"                         Random Forest": RandomForestRegressor(),
"                     Gradient Boosting": GradientBoostingRegressor() # n_estimators=100, learning_rate=0
}
for name, model in models.items():
    model.fit(X_train, Y_train)
    print(name + " trained.")
for name, model in models.items():
    print(name + " R^2 Score: {:.5f}".format(model.score(X_test, Y_test)))
    print(name + " R^2 Score: {:.5f}".format(model.score(X_train, Y_train)))
```

```
                   K-Nearest Neighbors trained.
                        Neural Network trained.
Support Vector Machine (RBF Kernel) trained.
                         Random Forest trained.
                     Gradient Boosting trained.
                   K-Nearest Neighbors R^2 Score: 0.50351
                   K-Nearest Neighbors R^2 Score: 0.63248
                        Neural Network R^2 Score: 0.91052
                        Neural Network R^2 Score: 0.58811
Support Vector Machine (RBF Kernel) R^2 Score: 0.48035
Support Vector Machine (RBF Kernel) R^2 Score: 0.41599
                         Random Forest R^2 Score: 0.99890
                         Random Forest R^2 Score: 0.98174
                     Gradient Boosting R^2 Score: 0.94729
                     Gradient Boosting R^2 Score: 0.94977
```

Appendix E: - Commands and experimental results after hyper-parameter tuning

```
from sklearn.model_selection import RandomizedSearchCV
from pprint import pprint

# Number of trees in random forest
n_estimators = [int(x) for x in np.linspace(start = 30, stop = 300, num = 6)]
# Number of features to consider at every split
max_features = ['auto', 'sqrt']
# Maximum number of levels in tree
max_depth = [int(x) for x in np.linspace(10, 110, num = 6)]
max_depth.append(None)
# Minimum number of samples required to split a node
min_samples_split = [2, 5, 10]
# Minimum number of samples required at each leaf node
min_samples_leaf = [1, 2, 4]
# Method of selecting samples for training each tree
bootstrap = [True, False]

# Create the random grid
param_grid = {'n_estimators': n_estimators,
              'max_features': max_features,
              'max_depth': max_depth,
              'min_samples_split': min_samples_split,
              'min_samples_leaf': min_samples_leaf,
              'bootstrap': bootstrap}

pprint(param_grid)
```

```
{'bootstrap': [True, False],
 'max_depth': [10, 30, 50, 70, 90, 110, None],
 'max_features': ['auto', 'sqrt'],
 'min_samples_leaf': [1, 2, 4],
 'min_samples_split': [2, 5, 10],
 'n_estimators': [30, 84, 138, 192, 246, 300]}
```

```
random_search = RandomizedSearchCV(RandomForestRegressor(),
                                   param_grid)
random_search.fit(X_train, Y_train)
print(random_search.best_params_)
```

```
{'n_estimators': 300, 'min_samples_split': 5, 'min_samples_leaf': 2, 'max_features': 'sqrt', 'max_depth': 70, 'bootstrap': F
alse}
```

```
models = {

"                  K-Nearest Neighbors": KNeighborsRegressor(),
"                       Neural Network": MLPRegressor(),
"   Support Vector Machine (RBF Kernel)": SVR(),
"                        Random Forest": RandomForestRegressor(n_estimators = 138, min_samples_split =10,
                                                               min_samples_leaf = 1,
                                                               max_features = 'sqrt', max_depth = 30,
                                                               bootstrap = False),
"                    Gradient Boosting": GradientBoostingRegressor() # n_estimators=100, learning_rate=0.1, max_depth=3
}
for name, model in models.items():
    model.fit(X_train, Y_train)
    print(name + " trained.")
for name, model in models.items():
    print(name + " R^2 Score: {:.5f}".format(model.score(X_test, Y_test)))
    print(name + " R^2 Score: {:.5f}".format(model.score(X_train, Y_train)))
```

```
                 K-Nearest Neighbors trained.
                      Neural Network trained.
Support Vector Machine (RBF Kernel) trained.
                       Random Forest trained.
                   Gradient Boosting trained.
                 K-Nearest Neighbors R^2 Score: 0.53933
                 K-Nearest Neighbors R^2 Score: 0.65470
                      Neural Network R^2 Score: 0.91645
                      Neural Network R^2 Score: 0.71719
Support Vector Machine (RBF Kernel) R^2 Score: 0.49856
Support Vector Machine (RBF Kernel) R^2 Score: 0.43103
                       Random Forest R^2 Score: 0.97545
                       Random Forest R^2 Score: 0.99175
                   Gradient Boosting R^2 Score: 0.94901
                   Gradient Boosting R^2 Score: 0.95018
```

```python
from sklearn.metrics import r2_score, mean_absolute_error, mean_squared_error
for name, model in models.items():
    Y_predict = model.predict(X_test)
    print(name + " R^2 Score: {:.5f}".format(r2_score(Y_test, Y_predict)))
    print(name + " RMSE: {:.5f}".format(np.sqrt(mean_squared_error(Y_test, Y_predict))))
    print(name + " MAE: {:.5f}".format(mean_absolute_error(Y_test, Y_predict)))
    print(name + " MSE: {:.5f}". format(mean_squared_error(Y_test, Y_predict)))
# Specifying and creating the MODEL
# Change the model accordingly!
PT_RF_model = RandomForestRegressor()
PT_RF_model.fit(X_train, Y_train)
# PREDICTING and VALIDATING the values
Y_predictions = PT_RF_model.predict(X_test)
print('R^2 Score =', r2_score(Y_test, Y_predictions))
print('Mean Absolute Error =', mean_absolute_error(Y_test, Y_predictions))
print('Root Mean Squared Error =', np.sqrt(mean_squared_error(Y_test, Y_predictions)))
print('Mean Squared Error =', mean_squared_error(Y_test, Y_predictions))
```

```
                K-Nearest Neighbors R^2 Score: 0.53933
                K-Nearest Neighbors RMSE: 4.54306
                K-Nearest Neighbors MAE: 3.44258
                K-Nearest Neighbors MSE: 20.63937
                     Neural Network R^2 Score: 0.91645
                     Neural Network RMSE: 1.93480
                     Neural Network MAE: 1.18495
                     Neural Network MSE: 3.74344
   Support Vector Machine (RBF Kernel) R^2 Score: 0.49856
   Support Vector Machine (RBF Kernel) RMSE: 4.73982
   Support Vector Machine (RBF Kernel) MAE: 3.46427
   Support Vector Machine (RBF Kernel) MSE: 22.46593
                       Random Forest R^2 Score: 0.97545
                       Random Forest RMSE: 1.04875
                       Random Forest MAE: 0.53931
                       Random Forest MSE: 1.09987
                  Gradient Boosting R^2 Score: 0.94901
                  Gradient Boosting RMSE: 1.51140
                  Gradient Boosting MAE: 0.93441
                  Gradient Boosting MSE: 2.28432
R^2 Score = 0.9988576654589824
Mean Absolute Error = 0.056804102353283946
Root Mean Squared Error = 0.22622963322083367
Mean Squared Error = 0.05117984694723293
```

```
3]:
```

```python
from sklearn.metrics import r2_score, mean_absolute_error, mean_squared_error
for name, model in models.items():
    Y_predict = model.predict(X_test)
    print(name + " R^2 Score: {:.5f}".format(r2_score(Y_test, Y_predict)))
    print(name + " RMSE: {:.5f}".format(np.sqrt(mean_squared_error(Y_test, Y_predict))))
    print(name + " MAE: {:.5f}".format(mean_absolute_error(Y_test, Y_predict)))
    print(name + " MSE: {:.5f}". format(mean_squared_error(Y_test, Y_predict)))
# Specifying and creating the MODEL
# Change the model accordingly!
#RF_model = RandomForestRegressor()
#RF_model.fit(X_train, Y_train)
# PREDICTING and VALIDATING the values
#Y_predictions = RF_model.predict(X_test)
#print('R^2 Score =', r2_score(Y_test, Y_predictions))
#print('Mean Absolute Error =', mean_absolute_error(Y_test, Y_predictions))
#print('Root Mean Squared Error =', np.sqrt(mean_squared_error(Y_test, Y_predictions)))
#print('Mean Squared Error =', mean_squared_error(Y_test, Y_predictions))
```

```
                K-Nearest Neighbors R^2 Score: 0.50351
                K-Nearest Neighbors RMSE: 4.71640
                K-Nearest Neighbors MAE: 3.58945
                K-Nearest Neighbors MSE: 22.24438
                     Neural Network R^2 Score: 0.91052
                     Neural Network RMSE: 2.00223
                     Neural Network MAE: 1.20182
                     Neural Network MSE: 4.00894
Support Vector Machine (RBF Kernel) R^2 Score: 0.48035
Support Vector Machine (RBF Kernel) RMSE: 4.82512
Support Vector Machine (RBF Kernel) MAE: 3.52447
Support Vector Machine (RBF Kernel) MSE: 23.28176
                      Random Forest R^2 Score: 0.99890
                      Random Forest RMSE: 0.22173
                      Random Forest MAE: 0.05697
                      Random Forest MSE: 0.04917
                  Gradient Boosting R^2 Score: 0.94729
                  Gradient Boosting RMSE: 1.53667
                  Gradient Boosting MAE: 0.93671
                  Gradient Boosting MSE: 2.36136
```