REAL-TIME TRAFFIC DENSITY ESTIMATION UNDER HETEROGENEOUS MIXED TRAFFIC CONDITION

PROJECT REPORT

Submitted by

ROHINI S

TKM21CETE15

to

the APJ Abdul Kalam Technological University

in partial fulfillment of the requirements for the award of the Degree

of

Master of Technology

in

Transportation Engineering

DEPARTMENT OF CIVIL ENGINEERING

TKM College of Engineering, Kollam

MAY 2023

DECLARATION

I undersigned hereby declare that the project report “Real-Time Traffic Density Estimation under Heterogeneous Mixed Traffic Condition”, submitted for partial fulfillment of the requirements for the award of the degree of Master of Technology of the APJ Abdul Kalam Technological University, Kerala is a bonafide work done by me under the supervision of Dr Kavitha Madhu. This submission represents my idea in my own words and where ideas or words of others have been included, I have adequately and accurately cited and referenced the original sources. I also declare that I have adhered to ethics of academic honesty and integrity and have not misrepresented or fabricated any data or idea or fact or source in my submission. I understand that any violation of the above will be a cause for disciplinary action by the institute and/or the University and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been obtained. This report has not been previously formed as the basis for the award of any degree, diploma or similar title of any other University.

KOLLAM ROHINI S

20-4-2023

DEPARTMENT OF CIVIL ENGINEERING

TKM COLLEGE OF ENGINEERING, KOLLAM

CERTIFICATE

Certified that this report entitled **"REAL-TIME TRAFFIC DENSITY ESTIMATION UNDER HETEROGENEOUS MIXED TRAFFIC CONDITION"** is the report of the project presented by **ROHINI S, TKM21CETE15** during **2022-2023** in partial fulfillment of the requirements for the award of the Degree of Master of Technology in Transportation Engineering of the APJ Abdul Kalam Technological University.

Guide	Project Coordinator	Head of the Department
Dr. Kavitha Madhu	**Dr. Adarsh S**	**Dr. Sajeeb R**
Associate Professor	Associate Professor	Professor
Department of Civil Engineering	Department of Civil Engineering	Department of Civil Engineering
TKMCE, Kollam	TKMCE, Kollam	TKMCE, Kollam

ACKNOWLEDGEMENT

I take this opportunity to express my deep sense of gratitude and sincere thanks to all who helped me to complete the seminar successfully.

I am deeply indebted to my guide, **Dr Kavitha Madhu**, Associate Professor, Department of Civil Engineering and for her excellent guidance, positive criticism and valuable comments.

I am greatly thankful to my project coordinator, **Dr Adarsh S**, Associate Professor, Department of Civil Engineering for his constant supervision as well as for providing necessary information regarding the seminar.

I extend my sincere gratitude to **Dr Sajeeb R**, Professor and Head of the Department of Civil Engineering, for his kind support.

I am grateful to my friend**, Sri Sikhin V C,** Junior Data Scientist, Knowledge Lens, Bangalore, for his immense support during the project work.

Finally, I thank my parents, teachers and friends who directly and indirectly contributed to the successful completion of my project.

Rohini S

TKM21CETE15

ABSTRACT

Traffic density is one of the essential macroscopic parameters and it is of prime importance when a facility is evaluated from the perspective of both users and planners. It is a major congestion indicator. Road traffic congestion is a severe issue with many detrimental implications for the economy and the environment. Traffic conditions are predicted as a result of recent advances in Intelligent Transportation Systems (ITS). Implementing traffic control strategies using ITS requires an accurate real-time estimate of traffic characteristics. One of the key features of ITS applications is predicting real-time traffic density. The present study estimates real-time traffic density using several Machine-Learning (ML) regression models and Kalman Filter (KF). Macroscopic traffic flow models are formulated and using this model, a model-based estimation scheme are designed based on the Kalman filtering technique. The data required for implementing this method in the field requires traffic flow at entry and spot speeds at entry and exit locations. The proposed method is validated using the input-output method of density estimation. The suitable method is identified for predicting traffic density for the given study stretch. It is found that Machine learning methods are more efficient in predicting traffic density when compared to Kalman Filter. Among different ML regression models, Random Forest is found to be more accurate in density prediction.

Keywords: Traffic density, Traffic congestion, Intelligent Transportation System (ITS), Macroscopic traffic flow models, Machine-learning, Kalman Filter

CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 GENERAL

Traffic congestion on urban roadways is an important issue for transportation engineers due to its uncontrolled growth and associated environmental and financial impacts. It's not always practical to solve traffic problems by adding additional infrastructure, as it is the conventional approach. Transportation engineers have recently focused on offering suggestions for the better operational management of existing infrastructure. The use of advanced technologies, such as Intelligent Transportation Systems (ITS), can effectively accomplish this. The real-time traffic operational assessment is a crucial component of ITS implementations. The purpose of real-time condition studies is to immediately gather traffic information, estimate the condition of the traffic, monitor congestion, and gather data on road users. A major ITS application is providing real-time information to drivers about traffic conditions, such as congestion.

Traffic density is a major indicator of road congestion. Density is a measure of the proximity of other vehicles, a factor which influences the freedom to manoeuvre and the psychological comfort of drivers (Al-Sobky and Mousa 2016). Direct measurement of density needs various techniques like aerial photography. Due to its difficulty, density is usually calculated from other traffic flow parameters such as occupancy, speed, and flow. For real-time density estimation, automated methods are used. In this method sensors and detectors are used. Automated devices including detectors, bluetooth devices, video sensors, and probe cars outfitted with Global Positioning System (GPS) are used for the online and offline calculation of traffic-flow variables. The model-based methods provide accurate density for homogenous traffic conditions. The filtering method can provide real-time density for heterogeneous and homogeneous traffic conditions.

In this study, a real-time traffic density estimation of the road stretch of an undivided National Highway (NH) 66 in Kollam was done by using machine learning methods and the Kalman filter. A model-based estimation scheme is used to predict density from techniques like the Kalman filter for the selected study stretch. Several machine learning regression techniques are used for predicting density from average space mean speed and traffic flow. The traffic data are collected from cameras and use the most recent object identification models based on deep learning to find and track moving automobiles. YOLOv5, one of the primary methods for object detection, is utilised in combination with manual counting, also for speed detection, and the results are compared. YOLO performs classification as a single-stage detector and is a lot more efficient than other convolutional neural networks.

1.2 PROBLEM STATEMENT

Traffic congestion is a serious issue on roads due to the increasing number of vehicles on roads and also the increase in the number of trips. In India, traffic flow is heterogeneous in nature. So, it makes more impacts on increases the traffic congestion. Traffic congestion can be monitored and reduced through ITS. Traffic density is a major indicator of road congestion. Density is the proximity of vehicles. Direct measurement of density needs various techniques such as aerial photography. It requires an elevated view of the considered road section. Due to its difficulty density is usually calculated from other traffic flow parameters such as occupancy, speed, and flow.

For real-time density estimation, automated methods are used. In this method sensors and detectors are used. Automated devices including detectors, video sensors, bluetooth devices, and probe cars outfitted with GPS are used for the calculation of traffic-flow variables. The occupancy approaches, model-based methods and filtering methods are three different automated approach methods. The model-based methods provide accurate density for homogenous traffic conditions. The filtering method can provide real-time density for heterogeneous and homogenous traffic conditions. In this study, by integrating model-based and filtering methods real-time traffic density is predicted for a 1km section of National Highway (NH) 66 and also estimates real-time traffic density using several machine-learning methods. The results from different methods were compared and suitable methods were identified. Macroscopic traffic flow models are developed, and a model-based estimation scheme based on the Kalman filtering technique is developed utilising this model. The data required to implement this strategy in the field are the flow of vehicles going through the exit and entry locations, as well as the speeds of vehicles travelling through the exit and entry locations. To reduce the human effort in data extraction, data such as speed and vehicle classification are extracted using object detection algorithms.

1.3 OBJECTIVES

The objectives of this study are to:

1. To detect and classify vehicles using image processing techniques.
2. To compare different traffic density estimation methods using Machine Learning methods and Kalman Filter.
3. To identify a suitable method for density estimation for heterogeneous mixed traffic condition.

1.4 SCOPE

The study focuses on an efficient traffic management system which can monitor and reduce traffic congestion. By estimating real-time traffic density, the information can be passed to travellers, and they can change their mode or routes or time of travel for smooth travel. The location identified for the study is an undivided 1km road section of Kollam Bypass. It is a part of NH 66 that bypasses Kollam City in Kerala, India.

1.5 ORGANIZATION OF THE REPORT

This report is organized into seven chapters. Chapter 1 is the introduction to the project work. It comprises the problem statement, the objectives of the study and the scope of the study. Chapter 2 comprises the details of the literature review as a part of the study. Chapter 3 explains the detailed methodology adopted in the study. It gives a clear idea about the various stages of work. Chapters 4 and 5 deal with the preliminary analysis and the details of methods for traffic density estimation. Chapter 6 discusses the results obtained from the study. Chapter 7 discusses the conclusions drawn from the work.

CHAPTER 2

LITERATURE REVIEW

2.1 GENERAL

The real-time traffic operational assessment is a crucial component of ITS implementations. The purpose of real-time condition studies is to collect the traffic information, estimate the condition of the traffic, monitor congestion, and gather data on road users. Providing real-time information to drivers about traffic conditions, such as congestion, is a major ITS application. Traffic density is a major indicator of road congestion. Density is the proximity of vehicles. There are several methods including the conventional and advanced methods used for density estimation in different conditions which are discussed in the following section. The different object detection algorithms are compared for vehicle detection and classification for use in data extraction from traffic video and they are also discussed in this section.

2.2 TRAFFIC DENSITY ESTIMATION

Traffic density is a gauge of other vehicles' proximity, which affects drivers' freedom of movement and psychological well-being (Al-Sobky and Mousa 2016). Traffic congestion in one of the Indian cities is shown in Figure 2.1. The direct measurement necessitates the application of methods like aerial photography. Usually calculated from other factors like location-based measurements of flow, speed, or occupancy. For defining the level of service (LOS) analysis, ramp metering, and congestion analysis, density is a crucial characteristic. Traffic flow performance is also impacted by traffic density.

A certain section length is used to calculate traffic density. The length is determined by the technology available for data gathering and the maximum distance that automated or conventional equipment can efficiently capture. The speed and flow of traffic at a certain location on the road can be immediately output by automated equipment (sensors or detectors). Videography and time-lapse photography of the traffic are conventional methods used. The traffic-flow variables are then manually extracted using these data. Section lengths of up to 10 m may be used in the detector and sensor sectors. Furthermore, the density can be estimated instantly (in real-time) using specialist algorithms applied to point or space sensor data of numerous variables.

Figure 2.1 Traffic Congestion

(Source: Sen et al., 2013)

One of the important benefits of predicting traffic density in real-time is that the drivers can be updated to avoid congested routes by using an application on smartphones (Ikiriwatte et al., 2019). If there is an accident, construction, etc., the drivers could update the busy routes. This system will improve the transportation system's efficiency. It aids in reducing traffic congestion at the four-way junction where high traffic congestion may occur. By estimating real-time traffic density, the information can be passed to travellers, and they can change their mode or routes or time of travel for smooth travel (Khan et al., 2017). It also saves time and money spent on traffic congestion and enhances transportation efficiency. It improves road safety and lowers pollution. Existing density estimation methods for continuous facilities are divided into two categories: the definition of concept approach and the automation approach (Bharadwaj et al., 2017). Both are discussed in the following sections.

2.2.1 Definition of Concept Approach

The Definition of Concept approach involves the methods which estimate density based on its definition (Kumar et al., 2023). However, it can be a little difficult to use these techniques in real-time. The following four techniques are a part of the Definition of Concept (DOC) approach: Aerial photography approach, Fundamental Traffic Flow (FTF) Equation method, Eddie's method, and cumulative input-output method.

Wagner and May (1963) conducted a freeway operations study in Los Angeles for the California Division of Highways and used aerial photographs for measuring freeway traffic density. The time-lapse photographs were taken by flying aircraft repeatedly above the freeway section under study. Then the complete photographic film coverage was obtained during each pass of the aircraft and aerial cameras over the section of the freeway. The fundamental Traffic Flow Equation Method is based on the variables of traffic flow, namely flow (q), speed (u_s) and density (k). This relationship is known as the fundamental relationship of traffic flow.

Romanowska and Jamroz (2021) suggest that in transportation engineering, their association and the bivariate relations between flow and density, flow and speed, and speed and flow are crucial. They are employed in the planning, construction, and renovation, as well as in the management and operation of transportation systems. It helps to identify a current road's true capacity, evaluate its traffic conditions, or choose the cross-section for a new road based on anticipated traffic levels. This method assumes stationary traffic flow. Due to the non-stationary nature of traffic, this method generates errors in density estimation and it lacks real-time application.

Edie’s Method is the method developed by Edie (1963) is accurate for all traffic-flow states. A time-space plot can be used to determine the definitions of density, space-mean speed and flow based on Edie for a specific facility. Leclercq et al., (2014) suggest that Edie’s method can be only used when the data is from a traffic simulator. It is the only approach to accurately estimate the macroscopic fundamental diagrams experimented by a road network and evaluate the relationship between traffic density and traffic flow without any bias.

Makigami et al., (1971) proposed Cumulative Input-Output Method. Bhaskar et al., (2014) applied cumulative plots for density estimation using loops and Bluetooth data. The "true value of density" should be obtained by this method. Area divided by period duration (λ) and section length yields the mean density over the period (L). In the given, (n, t) curve, n denotes the total number of vehicles that have ever arrived at the section's upstream and downstream ends at any given time, while t denotes the length of time for which a representative density estimate is needed. The cumulative number of vehicle arrivals in the section at the downstream and upstream end is represented by the red and blue curves, respectively are shown in Figure 2.2. The density is then calculated using the area between the downstream and upstream curves. The area divided by the length of the period (λ) and the section length (L) gives the average density during the period. This method is most suited for ITS applications among the four methods mentioned above.

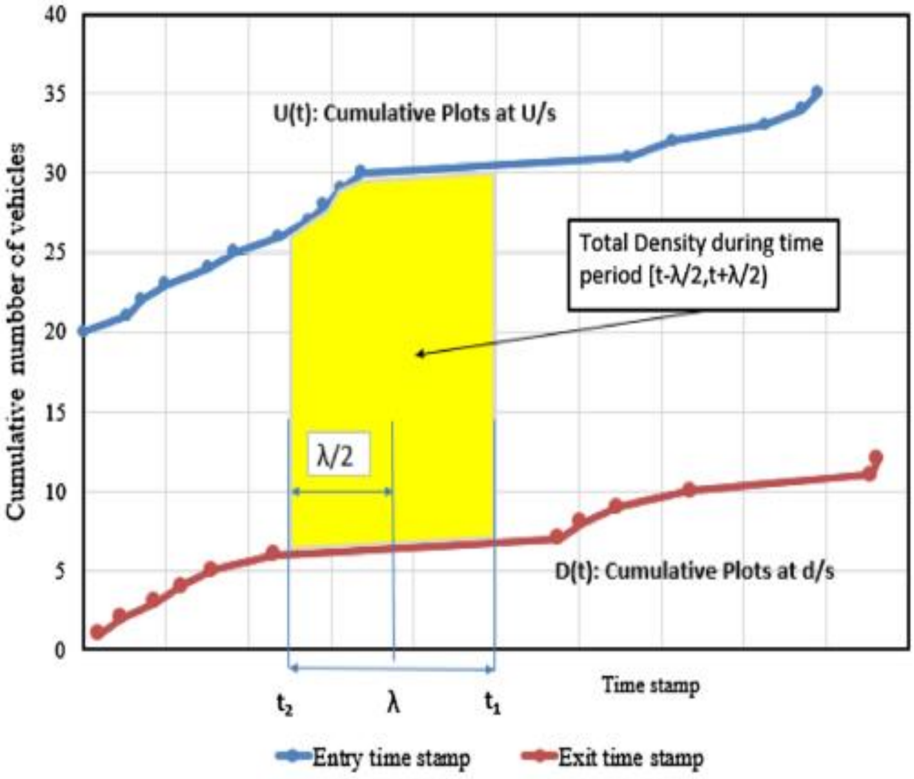


Figure 2.2 Graphical Representation of Cumulative Plot Method

(Source: Bharadwaj et al., 2017)

2.2.2 Automated Methods

Sensors and detectors were used to create real-time automated systems for estimating density. To calculate traffic-flow variables both online and offline, automated devices like detectors, bluetooth devices, video sensors and probe cars with a Global Positioning System (GPS) are used. They are used to cut down on the amount of labour required by conventional data extraction techniques. The automated approach's methodologies are broken down into three categories: the occupancy method, model-based methods, and the filtering method.

Occupancy is the point estimate of concentration or density. Lane, time, and area occupancy are the three different types of occupancy. The ratio of the sum of the lengths of the vehicles in the section to the number of vehicles is referred to as lane occupancy. The density is calculated by dividing lane occupancy by the average vehicle length.

Temporal occupancy does not provide actual density results in the presence of heterogeneous traffic. To properly analyse heterogeneous traffic, the entire road width must be considered. Therefore, the area of the detection zone and the vehicles must be considered when the occupancy concept is applied. Mallikarjun and Rao (2006) proposed the area-occupancy concept for heterogeneous traffic. It is modified by Arasan and Dhivya (2010). It is defined as the sum of the total vehicle area projected on the ground per unit area of the roadway.

The demand for real-time occupancy or density estimates has primarily led to the implementation of filtering techniques in research. These techniques use correlated traffic-flow parameters that can be managed online to estimate traffic density. The Kalman filter, Extended Kalman filter and other filtering techniques have all been applied so far. When the system's governing equations are linear, estimation and prediction use the Kalman filter. It can be used in both stationary and non-stationary traffic conditions (Emami et al., 2019). Non-linear estimators are used for non-linear equations. The Extended Kalman Filter (EKF), Particle Filter (PF), and Unscented Kalman Filter (UKF) are three important non-linear estimators. UKF is a method for calculating the statistics of a random variable that undergoes a non-linear transformation and uses a deterministic sampling approach with a minimum set of samples (Julier and Uhlmann 2004). The EKF is used for nonlinear estimates. It linearizes the non-linear model around the previous state estimates and uses a first-order Taylor series approximation (Wang and Papageorgiou 2005). The PF is a powerful approximate solution to a general non-linear, non-gaussian filtering problem (Mihaylova et al., 2007).

Dhivyabharathi et al., (2015) compared the effectiveness of several filtering methods. It was found that the errors are less than 15% in different situations. The majority of the time, EKF and UKF errors were higher than PF errors under shorter periods. For longer periods, EKF and UKF dominate over PF. Guo et al., (2014) used an adaptive Kalman filter approach for estimating short-term traffic flow rate and uncertainty quantification and suggested that it is suitable for highly volatile traffic. Ma et al., (2017) proposed a 2-dimensional forecasting which is more accurate than standard Kalman filtering approach and uses historic traffic data for density prediction.

A few online and offline model-based methods developed for density estimation include regression analysis, Artificial Neural Networks (ANN), Vehicle Ad Hoc Networks

(VANET) and data fusion. Anand et al., (2011) found that information derived from the same sources could occasionally be more erroneous than information derived using the data fusion technique, where the flow was estimated from video and space mean speed was determined from a probe vehicle's GPS. Ozkurt et al., (2009) proposed a method to estimate traffic density using an Artificial Neural Network (ANN). Aljamal et al., (2020) proposed a data-driven method for determining density for connected vehicle systems. Ajitha et al., (2015) developed a model-based estimation scheme for real-time traffic density computation using the Kalman filter. Kumar et al., (2015) used ANN for traffic flow forecasting and used traffic volume, speed, density as input parameters.

2.3 TRAFFIC DENSITY PREDICTION USING MACHINE LEARNING AND DEEP LEARNING

Due to its capacity to learn from and recognize patterns in huge and complicated data sets, machine learning is becoming more and more significant in the prediction of traffic density. To effectively estimate traffic density in real time, these algorithms can be trained on a range of data sources, such as traffic count data, GPS data, and real-time traffic cameras. To precisely estimate traffic density, Support Vector Machines (SVMs) and Artificial Neural Networks (ANNs) can efficiently assess and interpret massive volumes of data. Convolutional Neural Networks (CNNs), a type of deep learning technique, have produced positive traffic density prediction outcomes. By precisely modelling the non-linear interactions between the different elements that affect traffic density, these cutting-edge tools enable more precise predictions.

Aydin et al., (2021) suggested an approach that aims to forecast the traffic density at certain crossings during peak periods, such as commuting and rush hours and proposed a Long short-term memory (LSTM) based model. The new LSTM-based model's performance is evaluated against machine learning techniques like decision trees, linear regression, random forests, and the traditional deep learning approach (DL). Istanbul Metropolitan Municipality (IMM) traffic density data from August 2020 were used in this analysis. IMM Open Dataset consists of time, intersection locations, the number of vehicles based at those intersections, and the average speed, top speed, and bottom speed of those vehicles are all provided. Using actual IMM data, experimental assessments of the linear regression, decision trees, random forest, traditional deep learning, and LSTM methods were conducted. The effectiveness of the approaches was evaluated using the

Root Mean Square Error (RMSE) and Mean Absolute Error (MAE) values. The MAE error values revealed that the LSTM method produced predictions with lower error rates in comparison to other methods. The RMSE values are shown in Figure 2.3. The proposed LSTM algorithm outperforms the compared strategies in terms of estimating traffic density.

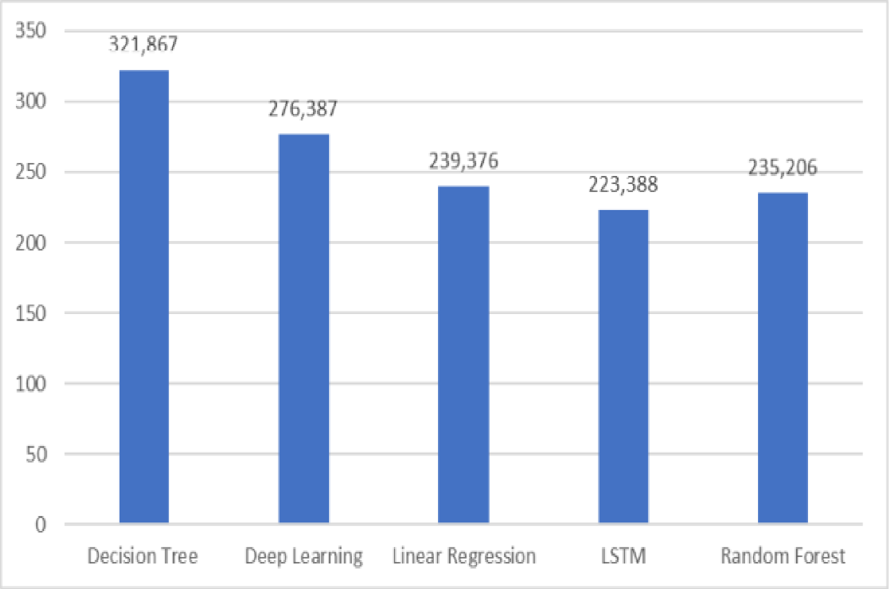


Figure 2.3 Total RMSE values of the Method.

(Source: Aydin et al., 2021)

Raj et al., (2016) investigated how automated sensor data and data-driven algorithms can be used to anticipate traffic density under Indian traffic circumstances. This study uses location-based sensors that can measure factors like density and Time Mean Speed (TMS) to address the issue of estimating traffic density. Artificial Neural Networks (ANN) and k-Nearest Neighbour (k-NN or KNN) are chosen as the estimation and prediction tools in machine learning. Most sensors provide data which consists of volume and speed. These data are used in this study to estimate and forecast density. A model which uses the output from k-NN as a training set for ANN was also tested. Using data collected at 5-minute intervals, density estimation using speed and volume as inputs to estimate the target variable density generated Mean Absolute Percentage Error (MAPE) in the range of roughly 2-5%. The MAPE ranged from 10 to 12 per cent in the density prediction, which used the densities of earlier time steps to forecast the current density. Therefore, for ITS applications under Indian traffic circumstances, the employment of data-driven approaches like ANN and k-NN together with automated sensor data seems promising. Combining these two strategies, though, did not result in noticeably better performance and is therefore not advised.

Asmaa et al., (2013) conducted a comparative study using the microscopic and macroscopic parameters for road traffic density estimation. The extracted parameters are applied to three classifiers, the K Nearest Neighbour (KNN) classifier, the Learning Vector Quantization (LVQ) classifier and the SVM classifier. The highest result was obtained by combining SVM with the mean speed of the vehicle and the density macroscopic characteristics obtained by the block matching technique. Ramchandra and Rajabhushanam (2022) used DAN (Deep Autoencoder), RF (Random Forest), DBN (Deep Belief Network), and LSTM (Long Short Term Memory) for density prediction. The accuracy, precision, recall, and error value metrics of machine learning algorithms can be used to assess performance of the proposed model. LSTM achieves 95.2% accuracy among the four techniques.

Deep learning's key benefit is that it can extract the features on its own and doesn't require any outside data. Mane et al., (2022) proposed a method for classifying traffic images according to traffic density using a customized convolution neural network (CCNN) to aid in driving. The proposed system classifies the present traffic situation into low, medium, and high categories utilizing footage taken by installed cameras to monitor traffic. To increase accuracy, elaborate deep neural networks are implemented and the training process is parallelized using NVIDIA GPUs. The efficiency of the proposed system is assessed using a real-time traffic video from Pune, India, and recorded highway CCTV footage from Seattle, Washington (WA). The suggested system's CCNN architecture is shown in Figure 2.4.

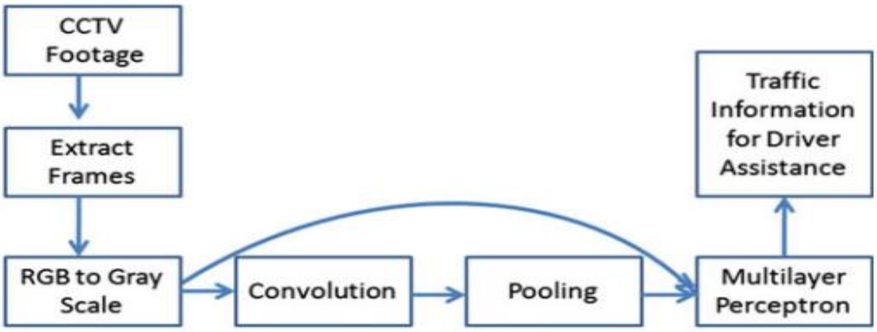


Figure 2.4 General Framework Architecture for Traffic Density Classification

(Source: Mane et al., 2022)

One application of deep learning is in traffic monitoring, which serves a variety of functions. Numerous activities, including vehicle identification, traffic infraction monitoring, vehicle speed monitoring, etc., can be accomplished by utilising cameras placed in strategic locations along the highways. Typically, deep learning-based techniques or machine learning techniques are used to detect objects. Vehicle identification with AI in transportation is one of the most significant applications of object detection, which is demonstrated in Figure 2.5.

Figure 2.5 Deep Learning-Based Object Detection for Vehicles

(Source: https://pythonawesome.com)

CNN architecture used in Deep Learning (DL) is well known for its excellent performance in Computer Vision (CV), particularly in object recognition and classification. Convolutional neural networks (CNN) constitute the foundation for the majority of object detectors currently in use. These detectors can be broadly categorised into two classes: single-stage and two-stage detectors (Khazukov et al., 2020). The single-stage detectors are often quick and anticipate classes and objects bounding boxes during a single network run. You Look Only Once (YOLO) and Single Stage Detector (SSD) are well-known examples of single-stage detectors. Chakraborty et al., (2018) studied the application of the YOLO deep learning model on traffic congestion detection using camera images. These architectures perform especially well when the target items take up a huge part of the image (Fedorov et al., 2019). The two-stage models classify in the second stage after generating areas. This method compromises picture processing speed for greater accuracy. Srivastava et al., 2021 compared three widely used object detection algorithms. YOLO was found to be the fastest, followed closely by SSD and Faster RCNN, who came in last.

2.4 GAPS IDENTIFIED

Most of the studies use manual data extraction techniques after collecting traffic video when machine learning and other statistical methods are used for density prediction. In this study computer vision methods are used for data extraction. Computer Vision is a field of artificial intelligence that involves the use of algorithms to analyse and interpret visual data from images or video. In the context of traffic engineering, computer vision plays a critical role in extracting traffic data from camera feeds and other visual sources. Using computer vision for data extraction was found to be cost-effective compared to other methods such as manual traffic counts, which are labour-intensive and time-consuming. Quite a few research were conducted on comparing the accuracy of different machine learning models. But limited studies were conducted on comparing the performance of machine learning regression models and the Kalman filter for density prediction.

2.5 SUMMARY

The Kalman filter and the Extended Kalman filter are the traditional statistical method used for traffic flow prediction. The prediction results could be the input for Intelligent Transportation System applications. Machine learning methods are also used to predict traffic density. By comparing the results, the best method for density prediction can be identified.

CHAPTER 3

METHODOLOGY

3.1 GENERAL

The methodology includes the study area identification, site selection, data collection, data extraction, data analysis and modelling. The study focuses on an efficient traffic management system which can monitor and reduce traffic congestion. By estimating real-time traffic density, the information can be passed to travellers, and they can change their mode or routes or time of travel for smooth travel. The location identified for the study is an undivided 1km road section of Kollam Bypass. It is a part of NH 66 that bypasses Kollam City in Kerala, India. This chapter explains the methodology proposed for the study.

3.2 DETAILED METHODOLOGY

The methodology planned to achieve the mentioned objectives is discussed below: The first step is the study of literature. A background study was done on various kinds of literature to understand the different methods used to estimate traffic density in real time and by definition. Next is the data collection. A videography survey is used to collect the traffic volume data. Cameras are to be set up on the exit and entry sides of a selected road stretch. Videography covers the number of vehicles passing through road stretch and records the traffic condition. The location identified for the study is an undivided 1km road section of Kollam Bypass. After data collection, the object detection technique YOLOv5 is used to detect vehicles and speed calculation. Vehicles are classified and speed is estimated using YOLOv5 and Deepsort. It is compared with manual counting and manual speed calculations. Using speed and classified count, macroscopic traffic flow models are developed, and a model-based estimation scheme based on the Kalman filtering technique is developed. The only data required to implement this strategy in the field are the flow of vehicles going through the entry and exit locations, as well as the speeds of vehicles travelling through the entry and exit locations. Object detection algorithms are used to extract data such as speed and vehicle classification to reduce human effort in data extraction. The proposed method will be validated using the input-output method of density estimation. Several machine learning regression methods are also used to predict the traffic density using traffic flow and average space mean speed from the field. The methodology adopted for the study is shown in Figure 3.1.

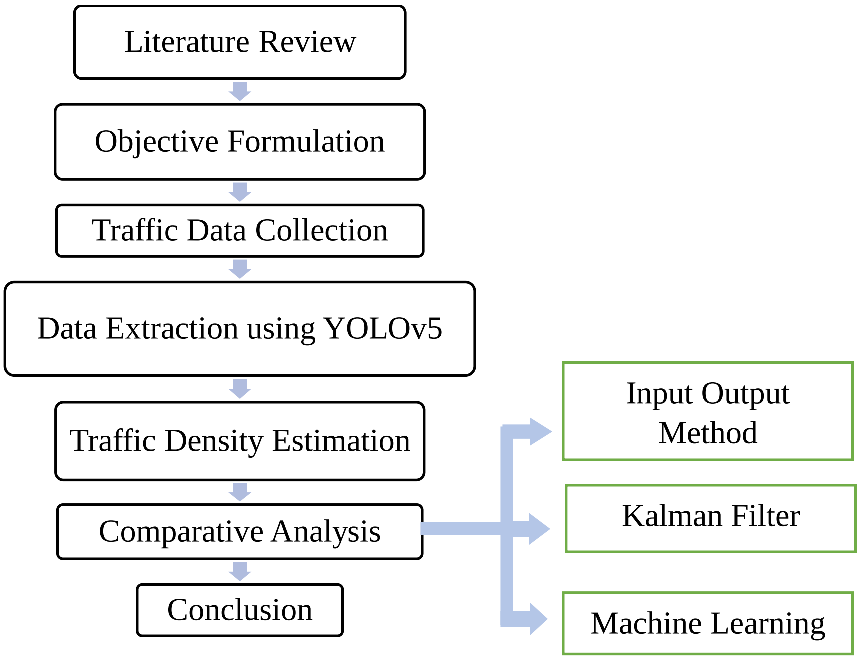


Figure 3.1 Flowchart of Methodology

3.3 STUDY AREA IDENTIFICATION AND DATA COLLECTION

The location identified for the study was an undivided 1km road section of Kollam Bypass which is shown in Figure 3.2 and Figure 3.3 respectively. It is a part of NH 66 that bypasses Kollam City in Kerala, India.

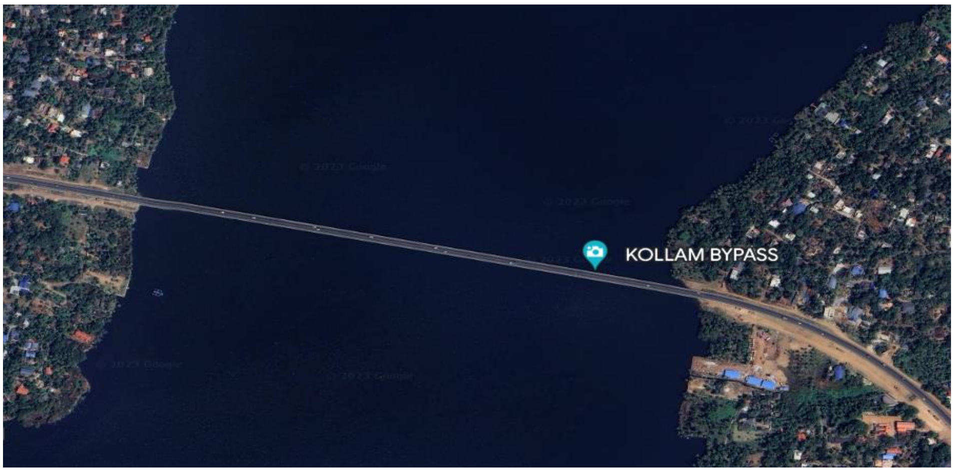


Figure 3.2 Map of 1 km Stretch of Kollam Bypass at Kadavoor Bridge

(Source: Google Earth Pro)

Videographic Survey was used to collect the traffic volume data. Cameras were placed on the exit and entry sides of a selected road stretch. Videography covers the number of vehicles passing through road stretch and records the traffic condition. Video data of 2 hours including peak and off-peak hours for 3 days were collected.

Figure 3.3 Images of Study Location

3.4 DATA EXTRACTION

In this work, traffic density is predicted for a particular section of the road. The number of vehicles entering the section and average space mean speed is the data needs. The traffic video data are collected using video cameras. For the recognition and tracking of vehicles on the road, this study makes use of the most recent deep learning-based object detection models. YOLOv5 is one of the most common object detection algorithms. YOLO object detection employs a single neural network as a real-time object detection system. It is capable of detecting any type and number of items. YOLO, which stands for "You Only Look Once," is an object detection technique that divides photos into grids. Each grid cell is in charge of detecting items within itself. Because of its speed and precision, YOLO is one of the most well-known object detection techniques. A two-stage tracker is included with YOLOv5 + Deep Sort with PyTorch. The detections produced by YOLOv5, a series of object detection architectures and models trained on the COCO dataset, are fed into a Deep Sort algorithm, which tracks the items. It can detect and track any object that the YOLOv5 model has been trained to detect. Figure 3.4 shows the steps used to detect a vehicle and its speed.

The video data needed for vehicle detection is collected using cameras. The video from an elevated position is preferred for use in YOLOv5 because by taking the video from an elevated position overlapping can be avoided, so the accuracy of the result can be increased. The angle of fixing the camera is also important it is suitable to fix the camera straight parallel to the road like cameras or perpendicular to the direction of the road. Cameras fixed at an in-between angle increase the chance of overlapping vehicles.

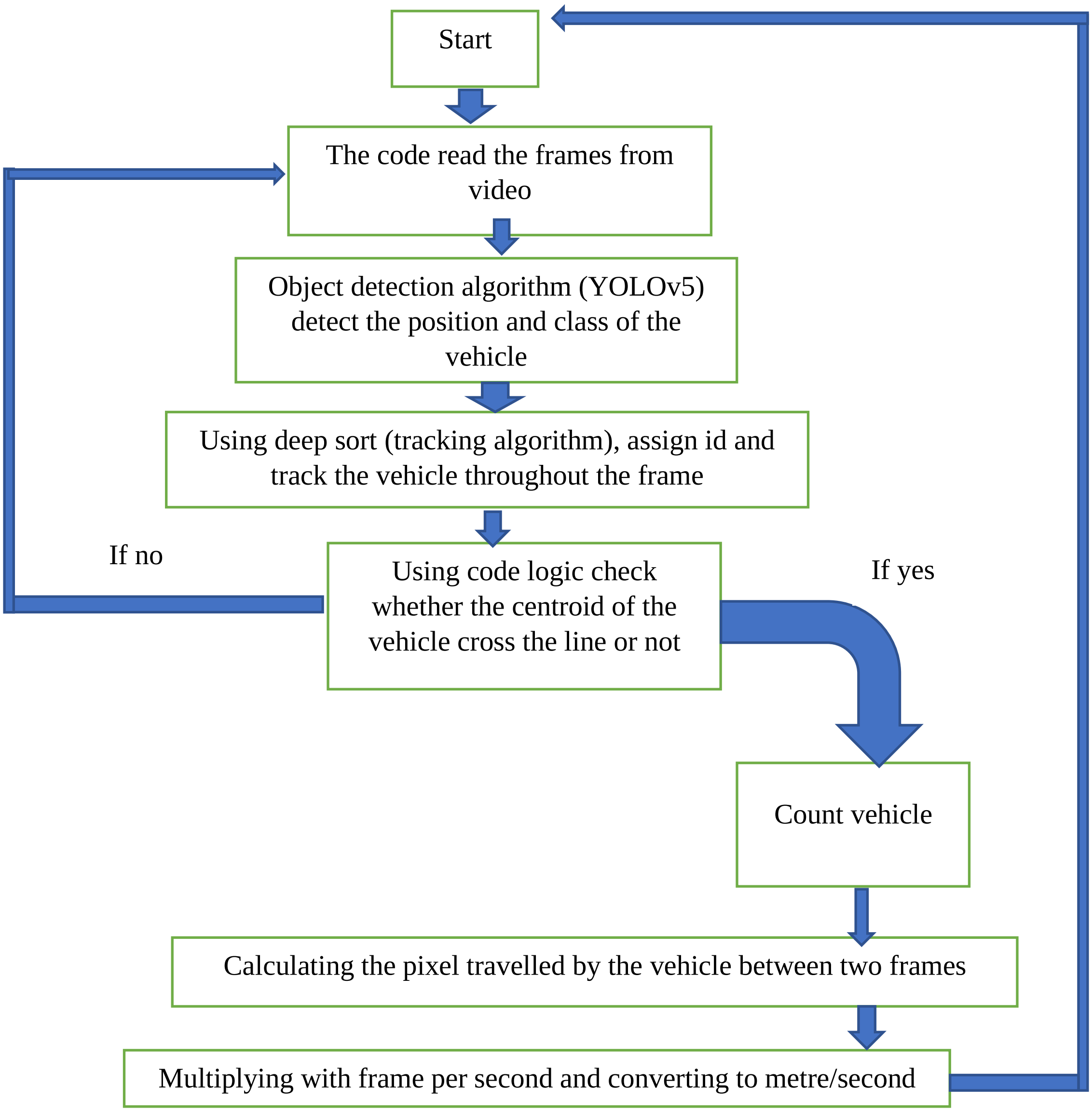


Figure 3.4 Steps for Vehicle Classification and Speed Detection

For only vehicle detection, YOLO is used. After vehicle classifications, the results are compared with manual counting. The results are almost the same. Then a code is used to find the pixels travelled by each vehicle. When the pixels travelled by each vehicle are obtained, it is multiplied by frame per second and converted to metres/second. To avoid misclassification annotation is done using labellmg tool. Annotation is the process of labelling each vehicle in a frame with its class. Before that, the videos were converted into frames. Figure 3.5 shows the process of converting video into frames and Figure 3.6 shows the process of annotation. Figure 3.7 shows vehicle classification and speed detection in YOLOv5. Figure 3.8 shows the results of volume and speed data (pixel/s) of tracked vehicles obtained in Excel.

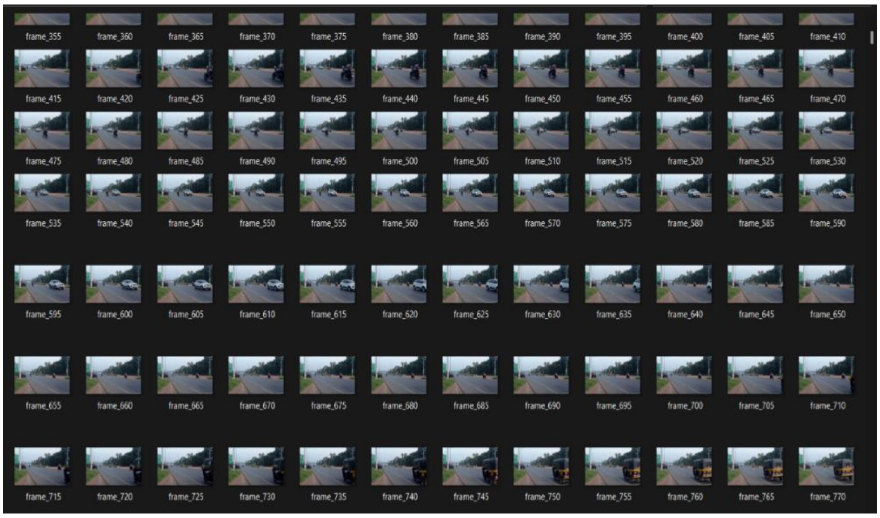
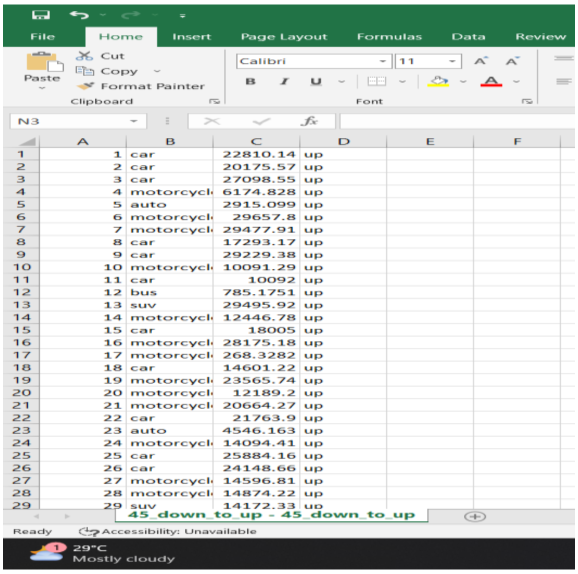


Figure 3.5 Video-to-Frame Conversion

Figure 3.6 Annotation Process

Figure 3.7 Vehicle Classification and Detection



	A	B	C	D	E	F
1	1	car	22810.14	up		
2	2	car	20175.57	up		
3	3	car	27098.55	up		
4	4	motorcycl	6174.828	up		
5	5	auto	2915.099	up		
6	6	motorcycl	29657.8	up		
7	7	motorcycl	29477.91	up		
8	8	car	17293.17	up		
9	9	car	29229.38	up		
10	10	motorcycl	10091.29	up		
11	11	car	10092	up		
12	12	bus	785.1751	up		
13	13	suv	29495.92	up		
14	14	motorcycl	12446.78	up		
15	15	car	18005	up		
16	16	motorcycl	28175.18	up		
17	17	motorcycl	268.3282	up		
18	18	car	14601.22	up		
19	19	motorcycl	23565.74	up		
20	20	motorcycl	12189.2	up		
21	21	motorcycl	20664.27	up		
22	22	car	21763.9	up		
23	23	auto	4546.163	up		
24	24	motorcycl	14094.41	up		
25	25	car	25884.16	up		
26	26	car	24148.66	up		
27	27	motorcycl	14596.81	up		
28	28	motorcycl	14874.22	up		
29	29	suv	14172.33	up		

Figure 3.8 Volume and Speed Data (pixel/s) of Tracked Vehicles obtained in Excel

Figure 3.9 shows the. confusion matrix from model training through annotation. Annotation is the process of labelling each vehicle in a frame with its particular class. By increasing the annotation, we can increase the accuracy of vehicle detection. If large misclassification was observed after annotation, we can improve the accuracy by augmentation. Augmentation is a process of creating large amount of training dataset by creating new transformed versions of images for a dataset to increase its diversity.

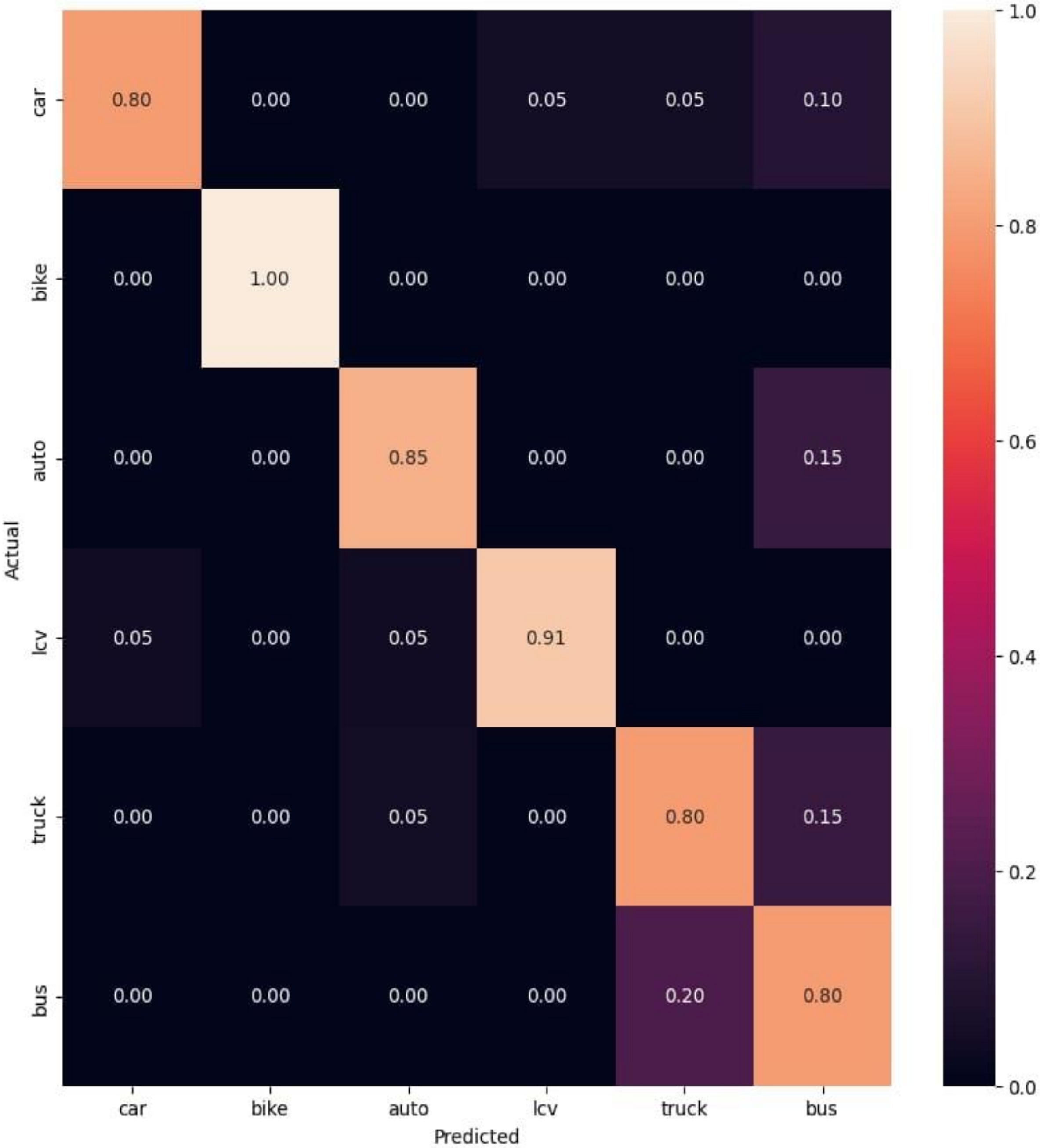


Figure 3.9 Confusion Matrix from Model Training

To measure the performance of vehicle detection, the confusion matrix is used. The different types of vehicles considered are bikes or two-wheelers (TW), three-wheelers or autorickshaws (3W), cars, heavy goods vehicles (trucks), light commercial vehicles (LCV) and buses. The confusion matrix displays properly and inaccurately classified cases for all classes, providing a clearer understanding of the model's performance.

3.5 TRAFFIC DENSITY PREDICTION

In this study, traffic density is predicted using machine learning methods and Kalman filter**.** Machine learning (ML) regression is a popular technique used for predicting numerical values based on a set of input features. It is a type of supervised learning, where the algorithm learns to map input features to output values by analysing a dataset of labelled examples. In ML regression, the goal is to find a mathematical function that best fits the data and can be used to predict the output values for new input features. Kalman Filtering is a technique used to estimate some unknown variables given the measurements over time containing statistical noise and other inaccuracies and produces estimates of unknown variables. The Kalman filter works like a predictor-corrector algorithm. Kalman filtering is used when the governing equations are linear. Another technique, the Extended Kalman Filter (EKF) is used when the equations of the system are non-linear. Density values needed for ML regression models and for of the estimation scheme are obtained through the input output method.

3.5.1 Input-Output Method

This calculation was carried out at every one-minute interval using Equation 3.1. This method provides actual density values, and it is used for comparing the best model by calculating the Mean Absolute Percentage Error (MAPE).

$$k_{n+1} = k_n + [N_{en} - N_{ex}] \tag{3.1}$$

where,

k_{n+1} = density at (n+ 1)th instant

k_n= density at kth instant

N_{en}= Number of vehicles entering the section through entry point

N_{ex}= Number of vehicles leaving the section through exit point

3.5.2 Traffic Density Estimation using Kalman Filter

Kalman filtering is a technique used to estimate some unknown variables given the measurements over time containing statistical noise and other inaccuracies and produces estimates of unknown variables. It is often used for the prediction and filtering of time-series data. Kalman filtering is used when the governing equations are linear. Another technique, the Extended Kalman Filter (EKF) is used when the equations of the system are non-linear. The steps for implementing the Kalman filter for density prediction are provided in Figure 3.10.

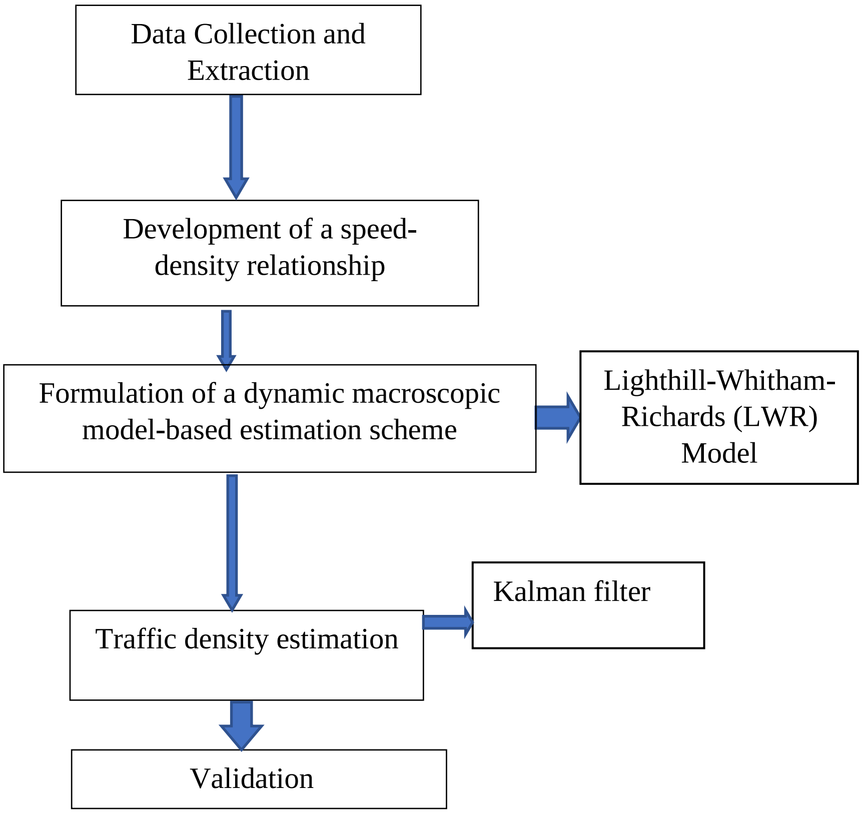


Figure 3.10 Steps for Implementing Kalman Filter

Kalman Filtering algorithm consists of two steps and they are prediction and update. The Kalman filter functions similarly to a predictor-corrector algorithm. It initially predicts an a priori estimate of the state variables using the system model, system inputs, and the preceding time interval's state estimate, and then corrects it using measurements to give an a posteriori estimate.

Let X_k represent the state variables that are used to depict the flow of vehicles approaching a segment over time. The Kalman algorithm is used to forecast the following

X_{k+1} flow. The φ_k denotes the state transition matrix which converts the previous state X_k to the following state X_{k+1}. Consider a noise vector W_k with normal distribution zero and a variance of Q_k.

The prediction model can be represented in Equation 3.2.

$$x_{k+1} = \psi_k \cdot x_k + w_k \tag{3.2}$$

Z_k denotes the observations made at the moment when the flow of connected vehicles approaches the reference line. H denotes the relationship between the measured variables and the state variables. V_k represents the measurement error, which is a Gaussian noise with a mean of zero and a variance of R_k. W_k and V_k have no association and are statistically independent and it is indicated using Equation 3.3 for all i and j and E is the expected value.

$$E\left[w_i v_j j^\wedge T\right] = 0 \tag{3.3}$$

The flow observation is expressed in Equation 3.4.

$$z_k = H_{xk} + y_k \tag{3.4}$$

The steps of the prediction are shown below,

1. Initialization

   If k = 0, the Equation 3.5 and 3.6 are used.

$$E[x_0] = \tilde{x}_0 \tag{3.5}$$

$$E[(x_0 - \tilde{x}_0)^2] = P_0 \tag{3.6}$$

   Where, X_k and P_k are the estimates of the state and error covariance at time k.

2. Extrapolation

   The extrapolation of state and error covariance are superscript dash stands for prior estimation of the state or error covariance.

3. Calculation of Kalman gain
4. State and error covariance update
5. The value of k = k+1 return to the step 2 and continue the process until the end of the present time period.

3.6 SUMMARY

This chapter deals with the study area identification, data collection and extraction and different methods used in the study for traffic density prediction. There is a variation in vehicle count obtained using the object detection algorithm and the vehicle count obtained from manual counting. For accurate detection and classification, the object detection algorithm needs data training. Details of Kalman Filter are also provided in this Chapter. More details about traffic density prediction are provided in Chapter 4, 5 and 6.

CHAPTER 4

MACHINE LEARNING (ML) REGRESSION MODELS

4.1 GENERAL

Machine learning (ML) regression is a popular technique used for predicting numerical values based on a set of input features. It is a type of supervised learning, where the algorithm learns to map input features to output values by analysing a dataset of labelled examples. In ML regression, the goal is to find a mathematical function that best fits the data and can be used to predict the output values for new input features. The accuracy of the predictions is measured by calculating the difference between the predicted values and the actual values, known as the error or loss.

4.2 MACHINE LEARNING (ML) REGRESSION FOR DENSITY PREDICTION

Machine learning regression can be applied to traffic engineering to predict traffic flow, congestion, and travel time on roads and highways. By using ML regression algorithms, engineers can better understand the patterns and trends in traffic data, and develop models that accurately predict traffic conditions. Machine learning (ML) regression can be applied to predict traffic density on roads and highways, which can help traffic engineers and transportation planners to improve traffic management and reduce congestion. ML regression algorithms can be used to analyse historical traffic data, as well as other factors such as weather, time of day, and special events, to develop models that accurately predict traffic density. In this study average space mean speed and traffic flow were used to predict the traffic density. Random forest regression, Support vector regression, Linear regression, XGBoost regression, and KNN regression are types of machine learning regression algorithms. Each of these regression algorithms has its strengths and weaknesses, and their suitability for a particular problem depends on factors such as the size and complexity of the data, the nature of the input variables, and the desired level of prediction accuracy.

4.2.1 Random Forest Regression

Random forest regression is a sophisticated machine learning approach that builds a forest out of a collection of decision trees. Each decision tree is built using a random subset of the data as well as a random subset of the characteristics. The programme then aggregates all of the decision trees' predictions to generate a final forecast. The random forest algorithm is well-known for its excellent accuracy, handling of missing data, and resistance to overfitting. Overfitting happens when the model is too tightly fitted to the training data, resulting in poor performance on new data. By employing numerous decision trees that are less likely to fit the noise in the data, random forest regression helps to avoid overfitting. The random forest algorithm follows a two-step process as shown in Figure 4.1.

1. Generates an array of n decision tree regressors (estimators). The number of estimators n in Scikit Learn (the machine learning Python library) is set to 100 by default and is referred to as n-estimators. The trees are built using the specified hyperparameters (for example, the minimum number of samples at leaf nodes, the maximum depth to which a tree can grow, and so on).

2. Determine the average forecast among all estimators. Each decision tree regression predicts a number as an output for a given input. The average of the forecasts is the 'final' output of random forest regression.

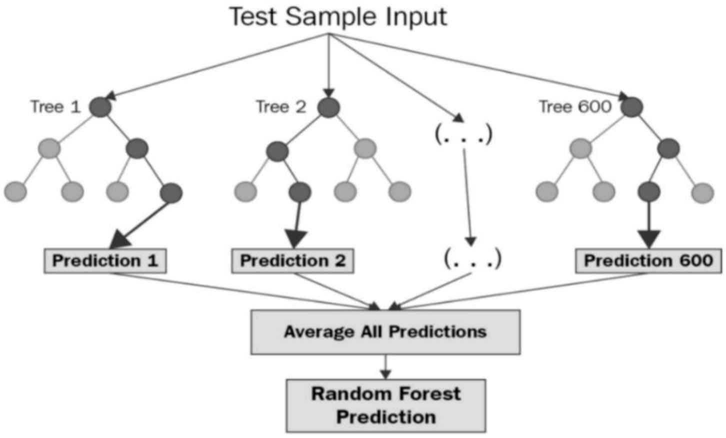


Figure 4.1 Steps for Random Forest Regression

(Source: https://www.keboola.com)

4.2.2 Support Vector Regression

Support vector regression is a machine learning algorithm that uses a support vector machine (SVM) to find the best possible line that fits the data. The algorithm creates a hyperplane that maximizes the distance between the data points and the hyperplane. The hyperplane is then used to predict the output variable. Support vector regression is particularly effective in handling nonlinear data and outliers. Nonlinear data refers to data that does not follow a linear relationship between the input and output variables, whereas outliers are data points that are significantly different from the rest of the data. Support vector regression can handle these types of data by transforming the input variables into a higher-dimensional space, where the data can be separated by a hyperplane. Support vector regression has been used in a wide range of applications, including in predicting stock prices, weather forecasting, and the analysis of medical data.

4.2.3 Linear Regression

Linear regression is a statistical method that uses a linear equation to predict the output variable based on one or more input variables. The equation is represented by a straight line that minimizes the distance between the data points and the line. Linear regression is a simple and easy-to-interpret algorithm, making it a popular choice for many applications. Linear regression assumes that there is a linear relationship between the input and output variables, which may not always be the case. It also assumes that the errors in the data are normally distributed, which may not be true for all datasets. Linear regression has many applications, including in predicting housing prices, estimating the impact of advertising on sales, and in the analysis of social and economic data.

4.2.4 XGBoost Regression

XGBoost regression is an advanced machine-learning algorithm that uses a gradient-boosting framework to create a model. The algorithm starts with a simple model and then iteratively improves it by adding more models. XGBoost regression is known for its high accuracy, speed, and ability to handle large datasets. It is particularly effective in handling unstructured data, such as text and images. XGBoost regression has been used in a wide range of applications, including in predicting customer churn, detecting fraud, and in the analysis of genomics data.

4.2.5 KNN Regression

KNN regression is a machine learning algorithm that uses the k-nearest neighbour approach to predict the output variable. The algorithm finds the k-nearest data points to the input variables and then takes the average of the output variables of those data points to make a prediction. KNN regression has been used in a wide range of applications, including in predicting crop yields, predicting the cost of living, and in the analysis of medical data.

4.3 EVALUATION CRITERIA

Evaluation criteria used for comparing different regression models are R-Squared (R^2), Mean Absolute Percentage Error (MAPE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and Mean Absolute Error (MAE). The model having high R^2 and minimum error values is considered as the best model.

1. R-Square (R^2): R^2 measures the proportion of the variance in the dependent variable that is explained by the independent variables. RSS is the sum of squares of residuals and TSS is the total sum of squares.

$$R^2 = 1 - \frac{RSS}{TSS} \tag{4.1}$$

2. Mean Absolute Percentage Error (MAPE): It is a metric used to measure the accuracy of a forecasting model by calculating the percentage difference between the actual and predicted values.

$$MAPE = \frac{1}{n} \Sigma \left| \frac{(Actual - Predicted)}{Actual} \right| * 100 \tag{4.2}$$

3. Mean Squared Error (MSE): MSE measures the average squared difference between the predicted and actual values.

$$MSE = \frac{1}{n} \Sigma (Actual - Predicted)^2 \tag{4.3}$$

4. Root Mean Squared Error (RMSE): it takes the square root of the MSE value

$$RMSE = \sqrt{\frac{1}{n}\Sigma(Actual - Predicted)^2} \tag{4.4}$$

5. Mean Absolute Error (MAE): MAE measures the average absolute difference between the predicted and actual values.

$$MAE = \frac{1}{n}\Sigma|Actual - Predicted| \tag{4.5}$$

4.4 TRAFFIC DENSITY PREDICTION

For prediction, the traffic density values estimated from the input-output method are taken as the independent variable and the average space mean speed (km/hr) and traffic flow (PCU/hr) is taken as the independent variable. These were extracted from the traffic video data.

4.5 SUMMARY

For the regression, 5 different algorithms were used to find the best model which gives the best prediction. The selected algorithms include Linear Regression, Random Forest, Support Vector Machine, XGBoost regression and KNN regression. The algorithms were selected based on the literature survey. The data set was initially set up to it standard form to make use of model. Results are provided in Chapter 6.

CHAPTER 5

PRELIMINARY ANALYSIS

5.1 GENERAL

This chapter is dedicated to the preliminary analysis of the data collected. The data collected from the field is used for the data analysis. The chapter consists of the vehicle composition of the study stretch and discusses the method of finding actual values of density using input output method.

5.2 VEHICLE COMPOSITION OF STUDY STRETCH

In this study, real-time traffic density will be estimated for a 1km road stretch of an undivided National Highway. The camera was placed at the entry and exit of the 1km study stretch. YOLOv5 + Deep Sort is used in the data extraction process to detect and classify vehicles from video. The classified volume count was extracted from the video in both directions of traffic namely Mevaram to Kavanad and Kavanad to Mevaram. The spot speed of vehicles was also obtained from the video. The different types of vehicles considered for traffic volume analysis are two wheelers (TW), three-wheelers or autorickshaws (3W), cars, Light Commercial Vehicles (LCV), heavy goods vehicles (trucks) and buses.

The flow entering was extracted from the video at every 1-min interval and was also expressed in PCU/hr. The spot speeds of vehicles were also obtained from the video. The harmonic mean (HM) of spot speeds was then used to compute the space mean speeds. Vehicle average space meaning speeds were calculated by averaging the HM values at the entry and exit sections.

The average traffic composition of the different vehicle classes observed in the study stretch of Mevaram to Kavanad is shown in Figure 5.1. Vehicle composition of the Mevaram to Kavanad stretch consists of 41%, 32%, 9%, 7%, 9% and 2% of TWs, Car, 3W, LCVs, trucks, and buses respectively. The average traffic composition of the different vehicle classes observed in the study stretch of Kavanad to Mevaram is shown in Figure 5.2. The traffic consists of 39%, 37%, 9%, 6%, 8% and 1% of TWs, Car, 3W, LCVs, trucks, and buses respectively for the Kavanad to Mevaram stretch.

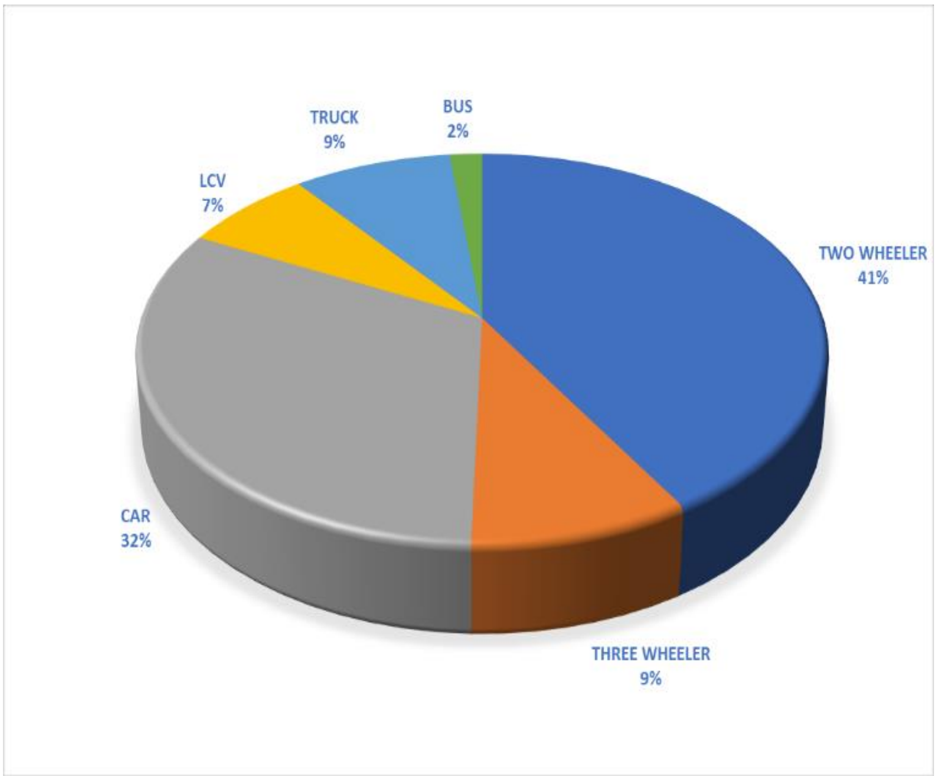


Figure 5.1 Vehicle Composition of Mevaram to Kavanad Stretch

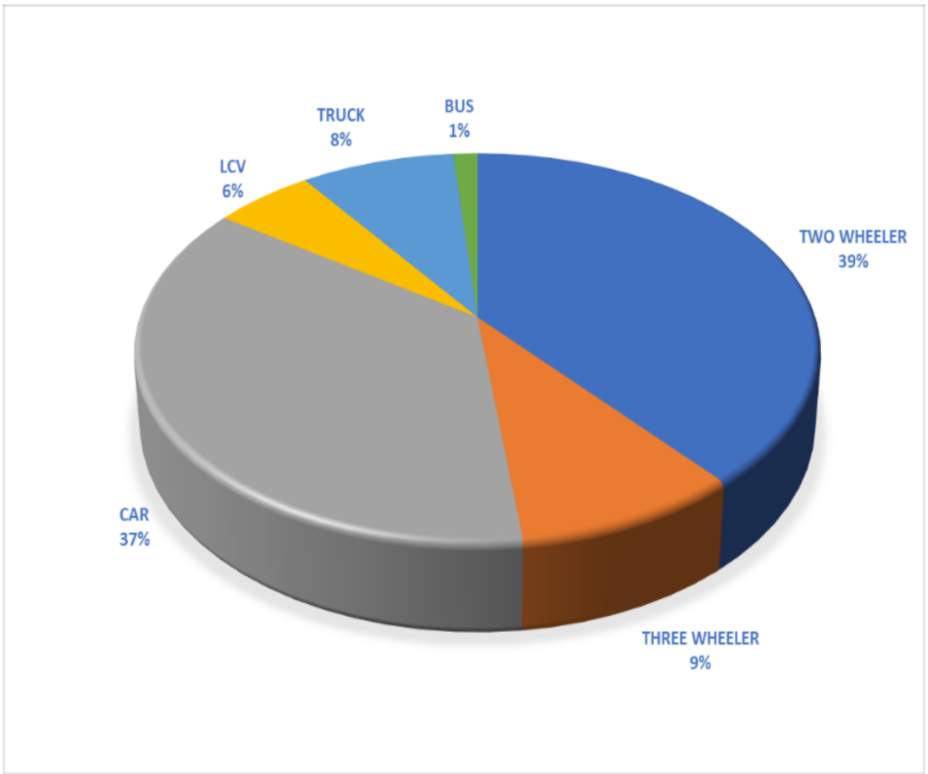


Figure 5.2 Vehicle Composition of Kavanad to Mevaram Stretch

The traffic volumes in terms of vehicle counts were transformed into per-hour flow numbers for each minute. This was done at the classified level for the six major vehicle categories, namely two-wheelers (TWs), autos, cars, buses, light commercial vehicles (LCVs), and trucks, and then converted to PCU values recommended by the Indian Road Congress (IRC) 64 1990. The PCU values utilised in this study are shown in Table 5.1.

Table 5.1 Recommended PCU values as per IRC 64 1990

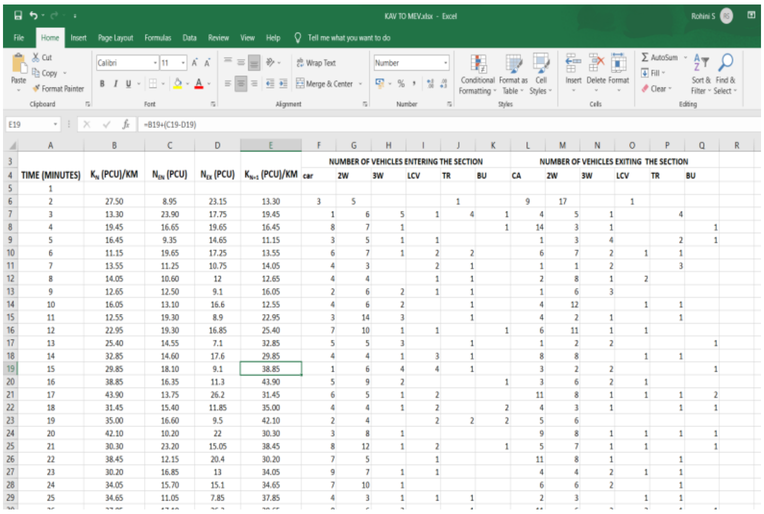
Vehicle Type	**Equivalency Factor**
Fast Vehicles	
Motorcycle or Scooter	0.50
Passenger Car, Pick-up Van or Auto-rickshaw	1.00
Agricultural Tractor, Light Commercial Vehicle	1.50
Truck or Bus	3.00
Truck Trailer, Agricultural Tractor Trailer	4.50
Slow Vehicles	
Cycle	0.50
Cycle Rickshaw	2.00
Horse Drawn Vehicle	3.00
Hand Cart	4.00
Bullock Cart	8.00

5.3 TRAFFIC DENSITY PREDICTION USING INPUT-OUTPUT METHOD

The actual densities are calculated using input-output analysis and expressed in PCU/km at 1-min intervals. To calculate the density inside the section, an initial count of the number of vehicles exiting the roadway between the entry and exit points is made, to which the number of vehicles entering the section is continuously added and the number of vehicles leaving the section is continuously subtracted. The starting number of distinct classes of vehicles present inside the section necessary for the input-output analysis was determined by photographing the section before the commencement of data collection, as illustrated in Figure 5.3.

Figure 5.3 Initial Density from Images

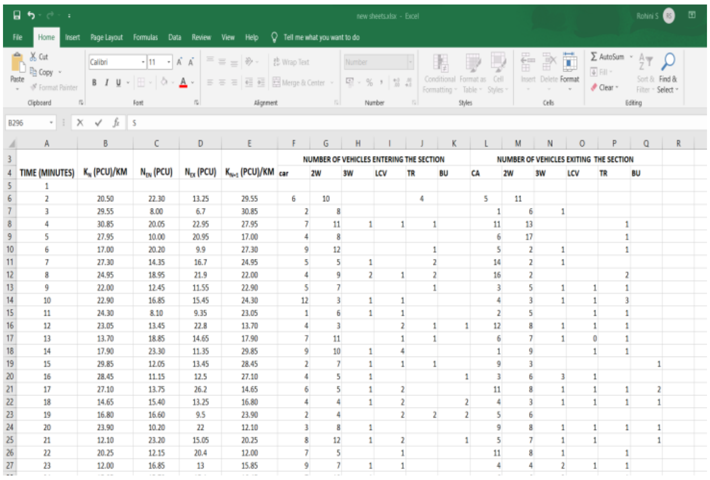
The density calculation was done using MS Excel for both directions using the Equation 3.1 are shown in Figure 5.4 and 5.5.



E19 =B19+(C19-D19)

TIME (MINUTES)	K_N (PCU)/KM	N_{EN} (PCU)	N_{EX} (PCU)	K_{N+1} (PCU)/KM	NUMBER OF VEHICLES ENTERING THE SECTION						NUMBER OF VEHICLES EXITING THE SECTION					
					car	2W	3W	LCV	TR	BU	CA	2W	3W	LCV	TR	BU
1																
2	27.50	8.95	23.15	13.30	3	5			1		9	17		1		
3	13.30	23.90	17.75	19.45	1	6	5	1	4	1	4	5	1		4	
4	19.45	16.65	19.65	16.45	8	7	1			1	14	3	1			1
5	16.45	9.35	14.65	11.15	3	5	1	1			1	3	4		2	1
6	11.15	19.65	17.25	13.55	6	7	1	2	2		6	7	2	1	1	
7	13.55	11.25	10.75	14.05	4	3		2	1		1	1	2		3	
8	14.05	10.60	12	12.65	4	4		1	1		2	8	1	2		
9	12.65	12.50	9.1	16.05	2	6	2	1	1		1	6	3			
10	16.05	13.10	16.6	12.55	4	6	2		1		4	12		1	1	
11	12.55	19.30	8.9	22.95	3	14	3		1		4	2	1		1	
12	22.95	19.30	16.85	25.40	7	10	1	1		1	6	11	1	1		
13	25.40	14.55	7.1	32.85	5	5	3		1		1	2	2			1
14	32.85	14.60	17.6	29.85	4	4	1	3	1		8	8		1	1	
15	29.85	18.10	9.1	38.85	1	6	4	4	1		3	2	2			1
16	38.85	16.35	11.3	43.90	5	9	2			1	3	6	2	1		
17	43.90	13.75	26.2	31.45	6	5	1	2			11	8	1	1	1	2
18	31.45	15.40	11.85	35.00	4	4	1	2		2	4	3	1		1	1
19	35.00	16.60	9.5	42.10	2	4		2	2	2	5	6				
20	42.10	10.20	22	30.30	3	8	1				9	8	1	1	1	1
21	30.30	23.20	15.05	38.45	8	12	1	2		1	5	7	1	1		1
22	38.45	12.15	20.4	30.20	7	5		1			11	8	1		1	
23	30.20	16.85	13	34.05	9	7	1	1			4	4	2	1	1	
24	34.05	15.70	15.1	34.65	7	10	1				6	6	2		1	
25	34.65	11.05	7.85	37.85	4	3	1	1	1		2	3		1	1	

Figure 5.4 Density Estimation done in MS Excel for the direction Kavanad to Mevaram



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
3						NUMBER OF VEHICLES ENTERING THE SECTION						NUMBER OF VEHICLES EXITING THE SECTION						
4	TIME (MINUTES)	K_N (PCU)/KM	N_{EN} (PCU)	N_{EX} (PCU)	K_{N+1} (PCU)/KM	car	2W	3W	LCV	TR	BU	CA	2W	3W	LCV	TR	BU	
5	1																	
6	2	20.50	22.30	13.25	29.55	6	10			4		5	11					
7	3	29.55	8.00	6.7	30.85	2	8					1	6	1				
8	4	30.85	20.05	22.95	27.95	7	11	1	1	1		11	13			1		
9	5	27.95	10.00	20.95	17.00	4	8					6	17			1		
10	6	17.00	20.20	9.9	27.30	9	12			1		5	2	1		1		
11	7	27.30	14.35	16.7	24.95	5	5	1		2		14	2	1				
12	8	24.95	18.95	21.9	22.00	4	9	2	1	2		16	2			2		
13	9	22.00	12.45	11.55	22.90	5	7			1		3	5	1	1	1		
14	10	22.90	16.85	15.45	24.30	12	3	1	1			4	3	1	1	3		
15	11	24.30	8.10	9.35	23.05	1	6	1	1			2	5		1	1		
16	12	23.05	13.45	22.8	13.70	4	3		2	1	1	12	8	1	1	1		
17	13	13.70	18.85	14.65	17.90	7	11		1	1		6	7	1	0	1		
18	14	17.90	23.30	11.35	29.85	9	10	1	4			1	9		1	1		
19	15	29.85	12.05	13.45	28.45	2	7	1	1	1		9	3				1	
20	16	28.45	11.15	12.5	27.10	4	5	1			1	3	6	3	1			
21	17	27.10	13.75	26.2	14.65	6	5	1	2			11	8	1	1	1	2	
22	18	14.65	15.40	13.25	16.80	4	4	1	2		2	4	3	1	1	1	1	
23	19	16.80	16.60	9.5	23.90	2	4		2	2	2	5	6					
24	20	23.90	10.20	22	12.10	3	8	1				9	8	1	1	1	1	
25	21	12.10	23.20	15.05	20.25	8	12	1	2		1	5	7	1	1		1	
26	22	20.25	12.15	20.4	12.00	7	5		1			11	8	1		1		
27	23	12.00	16.85	13	15.85	9	7	1	1			4	4	2	1	1		

Figure 5.5 Density Estimation done in MS Excel for the direction Mevaram to Kavanad

Initial density was converted PCU/km in this method and the vehicles entering and exiting the section were also converted into PCU. This method is very simple and gives the actual or true values of density. The density obtained from this method is used for further analysis.

5.4 SUMMARY

The density in 1-minute interval was obtained in this method. It is used for validation and for training the machine learning models. Other methods used for density prediction are detailed in the next chapters.

CHAPTER 6

RESULTS AND DISCUSSIONS

6.1 GENERAL

In this study, a real-time traffic density estimation of the road stretch of an undivided National Highway, NH 66 in Kollam was done by using the Kalman filter and machine learning regression models. A model-based estimation scheme using different prediction techniques like the Kalman filter was employed for traffic density estimation of the selected study stretch for both directions.

6.2 TRAFFIC DENSITY PREDICTION USING KALMAN FILTER

6.2.1 Speed density relationship for both directions

The best fit of speed density relationship is found out based on the highest R square value. The speed density relationship for this lane is developed by using average space mean speed data and density data used by machine learning models for testing. The speed density relationship was obtained from MS Excel. The details of the curve fit obtained are shown in Table 6.1 and shown in Figures 6.1 and 6.2 for both directions. The green shield's linear model is used to obtain the values of free flow speed (u_f) and Jam density (k_j).

Table 6.1 Speed Density Relationship

Direction	**R-Square**	**Equation**	**Parameters**	
			u_f	k_j
Kavanad to Mevaram	0.969	Y= 77.55-0.998k	77.75	77.90
Mevaram to Kavanad	0.931	y = 66.818-1.1028k	66.818	60.621

The developed linear speed density models are implemented in the dynamic model-based estimation scheme using Kalman filter for estimating the traffic density along the study stretch. The static speed-density relationship will not give the complete information on temporal variation of macroscopic variables. For studying the temporal variation of macroscopic variables, dynamic models are necessary for real time traffic state estimation of variables including density for ITS applications.

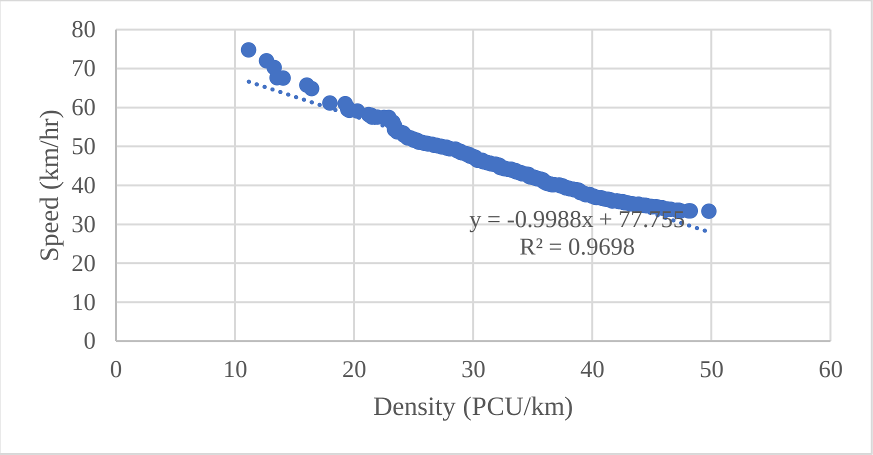


Figure 6.1 Linear Speed Density Model for Kavanad to Mevaram Direction

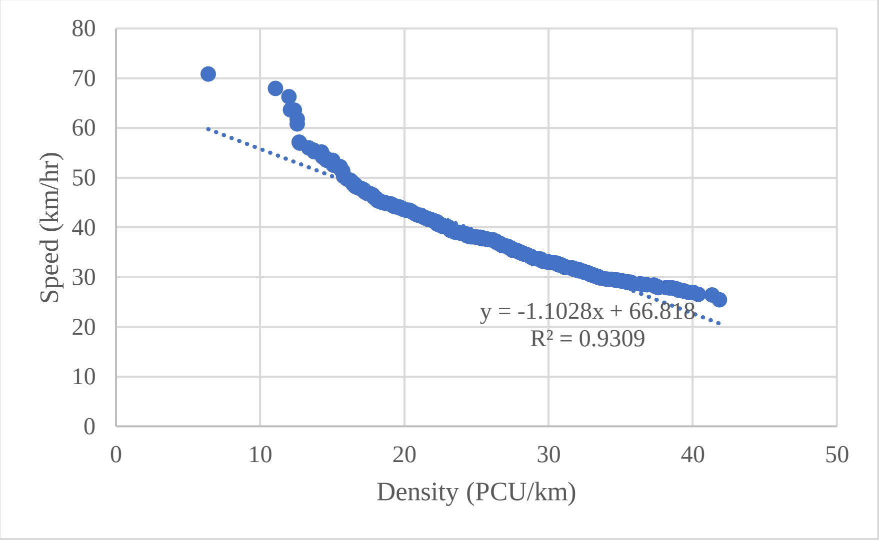


Figure 6.2 Linear Speed Density Model for Mevaram to Kavanad Direction

6.2.2 Development of Model-Based Estimation Scheme

KF was used as the model the linear in nature. Three flow variables (q_{en} and q_{ex}) and one density variable. Based on the flow data, the Kalman filter is utilised to estimate the density variable. To update the density and velocity, the algorithm uses a linearized version of the LWR (Lighthill-Whitham-Richards) traffic flow model. The continuity equation and an assumed speed density relationship are used in LWR models to represent the behaviour of traffic streams. These models are based on the assumption that traffic flows are always in equilibrium. It is assumed that the speed and density values are in equilibrium at every place in the stream at any moment. The LWR model is made up of a continuity equation, a traffic flow fundamental equation, and a speed-density relationship. The equation of the LWR model is given in the Equation 6.1.

$$k^{n+1}= k^n\ [1-\frac{\triangle t}{\triangle x}u_f(1-\frac{k^n}{k_j})]+k^n-1\ [1-\frac{\triangle t}{\triangle x}u_f(1-\frac{k^n-1}{k_j})] \tag{6.1}$$

Speed-based model is developed as,

$$\mathrm{u}(n+1)=u(n)+bh(u(\rho)-u(n))+\frac{h}{L}\frac{d(u(k))}{dk}(q_{en}(n)-\rho(n).u(n)) \tag{6.2}$$

Implementing the speed density relationship in speed-based model equation of Kavanad to Mevaram direction, it is obtained as

$$\mathrm{u}(n+1)=u(n)+bh\left\{77.75(1-\frac{k}{77.90})-u(n)\right\}-\ \frac{77.75h.k(n)}{77.90.L}.77.75(1-\frac{k}{77.90})(q_{en}(n)-k(n).u(n)) \tag{6.3}$$

The corresponding value of f and A_1 which are the inputs to Kalman filter were obtained for study stretch for linear is shown in Equation 6.4 and 6.5.

$$\boldsymbol{f}=\left|\begin{array}{c} k^{n+1}= k^n\ [1-\frac{\triangle t}{\triangle x}\ u_f(1-\frac{k^n}{k_j})] \\ u(n+1)=u(n)+bh\left\{77.75\left(1-\frac{k}{77.90}\right)-u(n)\right\}-\ \frac{77.75h.k(n)}{77.90.L}.77.75\left(1-\frac{k}{77.90}\right)(q_{en}(n)-k(n).u(n) \end{array}\right| \tag{6.4}$$

$$A_1=\begin{bmatrix} 1-\frac{u(n)}{L} & -\frac{h.k(n)}{L} \\ \frac{77.55h}{77.90}.77.55\left(1-\frac{k(n)}{77.90}\right)\begin{Bmatrix} b.k(n)+\frac{(q_{en}(n))}{L}* \\ \left[1-\frac{k(n)}{77.90}\ \right]+\frac{k(n).u(n)}{L}\left[2-\frac{k(n)}{77.90}\right] \end{Bmatrix} & \begin{array}{c} 1-bh+\frac{77.75h}{L*77.90}* \\ \left(k(n)\right)^2*77.75(1-\frac{k(n)}{77.90}) \end{array} \end{bmatrix} \tag{6.5}$$

Implementing the speed density relationship in speed-based model equation of Mevaram to Kavanad direction, it is obtained as

$$\mathrm{u}(n+1)=u(n)+bh\left\{66.818(1-\frac{k}{60.621})-u(n)\right\}-\ \frac{66.818h.k(n)}{60.621L}.66.8181-\frac{k}{60.621})(q_{en}(n)-k(n).u(n)) \tag{6.6}$$

The corresponding value of f and A_1 were obtained for study stretch for linear is shown in Equation 6.7 and 6.8

$$\boldsymbol{f}=\left|\begin{array}{c} k^{n+1}= k^n\ [1-\frac{\triangle t}{\triangle x}\ u_f(1-\frac{k^n}{k_j})] \\ u(n+1)=u(n)+bh\left\{66.818(1-\frac{k}{60.621})-u(n)\right\}-\ \frac{66.818h.k(n)}{60.621.L}.66.818(1-\frac{k}{60.621})(q_{en}(n)-k(n).u(n) \end{array}\right| \tag{6.7}$$

$$A_1=\begin{bmatrix} 1-\frac{u(n)}{L} & -\frac{h.k(n)}{L} \\ \frac{66.818h}{60.621}.66.818\left(1-\frac{k(n)}{60.621}\right)\begin{Bmatrix} b.k(n)+\frac{(q_{en}(n))}{L}* \\ \left[1-\frac{k(n)}{60.621}\ \right]+\frac{k(n).u(n)}{L}\left[2-\frac{k(n)}{60.621}\right] \end{Bmatrix} & \begin{array}{c} 1-bh+\frac{66.818h}{L*60.621}* \\ \left(k(n)\right)^2*66.818(1-\frac{k(n)}{60.621}) \end{array} \end{bmatrix} \tag{6.8}$$

6.2.3 Results

Actual density values (expressed in PCU/km) generated from input output analysis were used to confirm the results, and performance was assessed using Mean Absolute Percentage Error (MAPE). The plots of the estimated values of densities against the actual values for both direction is shown in Figure 6.3 and Figure 6.4. The MAPE values for traffic density estimated for both direction of study stretch for linear traffic flow is given in Table 6.2.

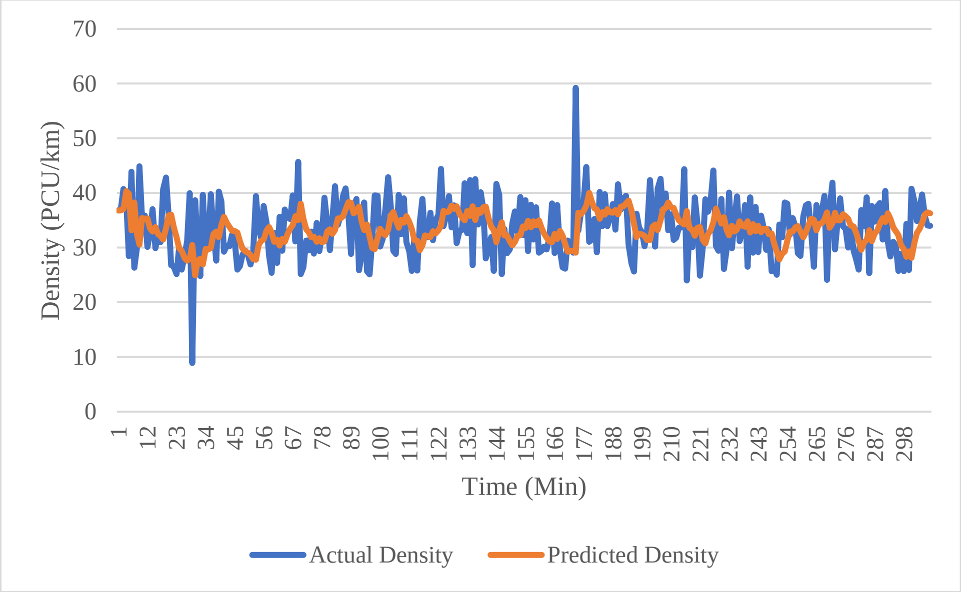


Figure 6.3 Actual and Estimated Density along Kavanad to Mevaram

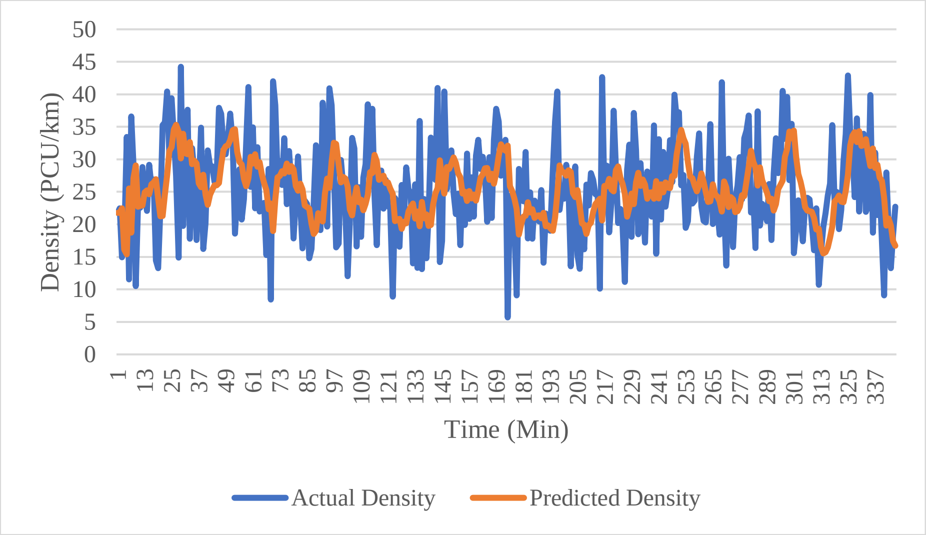


Figure 6.4 Actual and Estimated Density along Mevaram to Kavanad

Table 6.2 MAPE value for Model - Based Estimation Scheme

Direction	MAPE
Kavanad to Mevaram	14.55%
Mevaram to Kavanad	28.89%

6.3 TRAFFIC DENSITY PREDICTION USING MACHINE LEARNING REGRESSION MODELS

The flow entering was extracted from the video at every 1-min interval and was also expressed in PCU/hr. The spot speeds of vehicles were also obtained from the video. The harmonic mean (HM) of spot speeds was then used to compute the space mean speeds. Vehicle average space meaning speeds were calculated by averaging the HM values at the entry and exit sections. Density for every 1-min interval was obtained from the input output method. The density is the dependent variable and average space mean speed and traffic flow per hour were taken as the independent variable for all models.

Random forest regression, Support vector regression, Linear regression, XGBoost regression, and KNN regression are all types of machine learning regression algorithms. These algorithms are used to predict the value of a continuous variable based on the values of one or more input variables. Each of these regression algorithms has its strengths and weaknesses, and their suitability for a particular problem depends on factors such as the size and complexity of the data, the nature of the input variables, and the desired level of prediction accuracy. 80% data is used for testing and 20% for training the models. Models were developed in two directions.

6.3.1. Kavanad to Mevaram

Table 6.2 shows the error and R-squared values obtained from different Machine-Learning models. The comparison of MAE, R-squared values, MSE, RMSE and MAPE are shown in figures 6.5 to 6.9. R-squared is a measure of regression model goodness-of-fit. R-squared measures the strength of the relationship between the model and the dependent variable on a convenient 0 – 100% scale. 100% denotes a model that explains all of the variation in the response variable's mean. The scatter of data points around the fitted regression line is calculated using R-squared. It is also known as the coefficient of determination or the coefficient of multiple determination in the case of multiple regression. Higher R-squared values indicate less disparities between observed and fitted values for the same data set. It was observed that the R square values obtained for XGBoost and Random Forest regression are very close to each other. R square is 0.986 and 0.981 for Random Forest regression and XGBoost regression respectively. But it is high for Random Forest regression.

Table 6.3 Evaluation Metrics for Kavanad to Mevaram Direction

Method	MAE	MSE	RMSE	MAPE	R-square
Linear Regression	2.037492	7.681319	2.771519	0.072453	0.864691
Random Forest Regression	0.3201	0.75834	0.870827	0.017644	0.986642
Support Vector Regression	0.838987	3.322257	1.822706	0.034754	0.941477
XGBoost Regression	0.436541	0.761842	0.97959	0.019953	0.981864
KNN Regression	3.931458	27.09307	5.205101	0.141213	0.522746

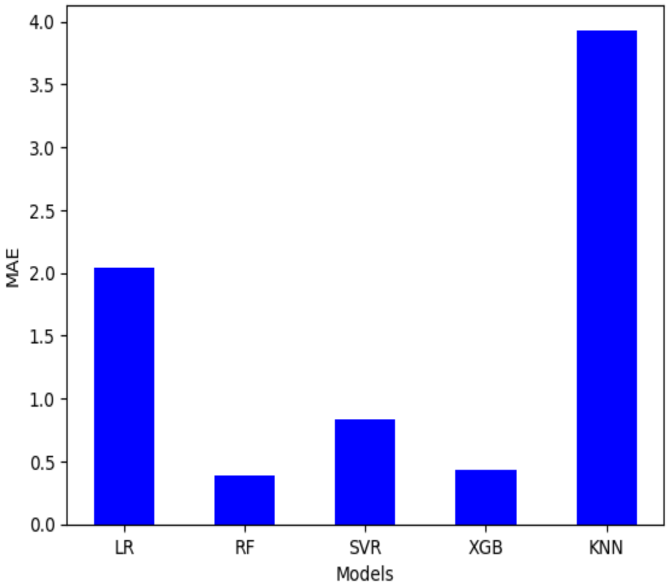


Figure 6.5 Comparison of MAE values for Regression model

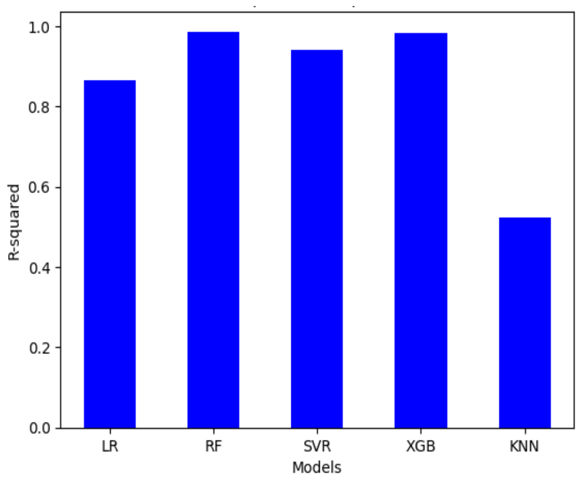


Figure 6.6 Comparison of R-squared values for Regression models

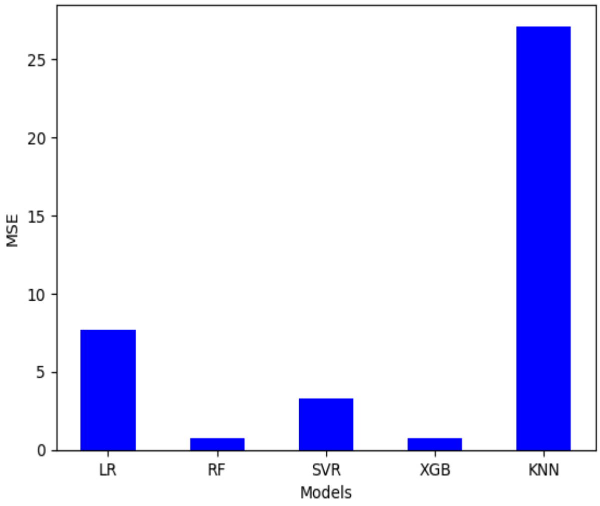


Figure 6.7 Comparison of MSE for Regression models

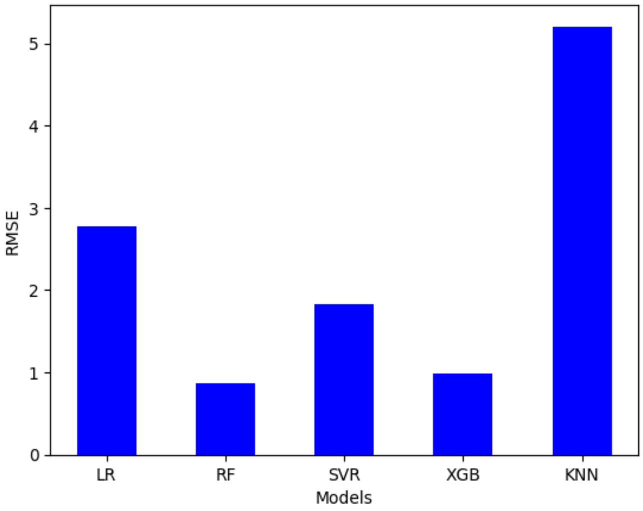


Figure 6.8 Comparison of RMSE for Regression models

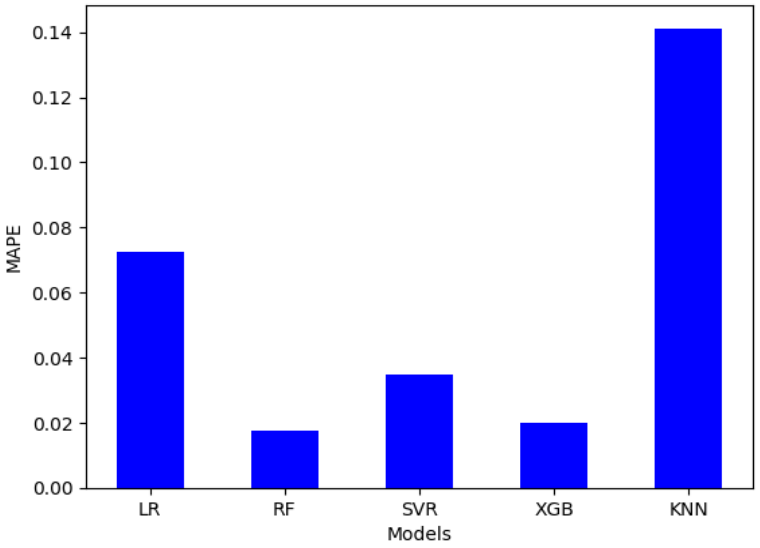


Figure 6.9 Comparison of MAPE for Regression models

For evaluating the performance of each model, RMSE, MSE, MAE and MAPE values were used. The model having the least error and high R square value is considered as best. It was observed that the Random Forest Regression have the largest R square value, that is 0.986. The RMSE, MSE, MAE and MAPE values were found to be very small for Random Forest Regression compared to other methods. So, it can be considered the best method for predicting density in this direction. KNN regression results in high error values and low R square values when compared with others. So, it cannot be used for accurate prediction. The actual density and predicted density obtained from the best and less accurate models are also shown in Figure 6.10 and 6.11.

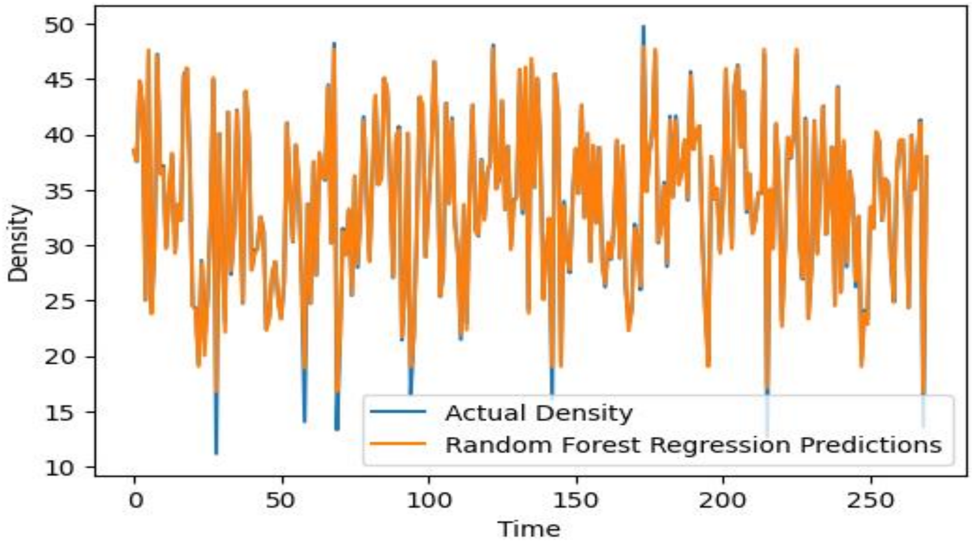


Figure 6.10 Actual and Predicted Density using Random Forest Regression

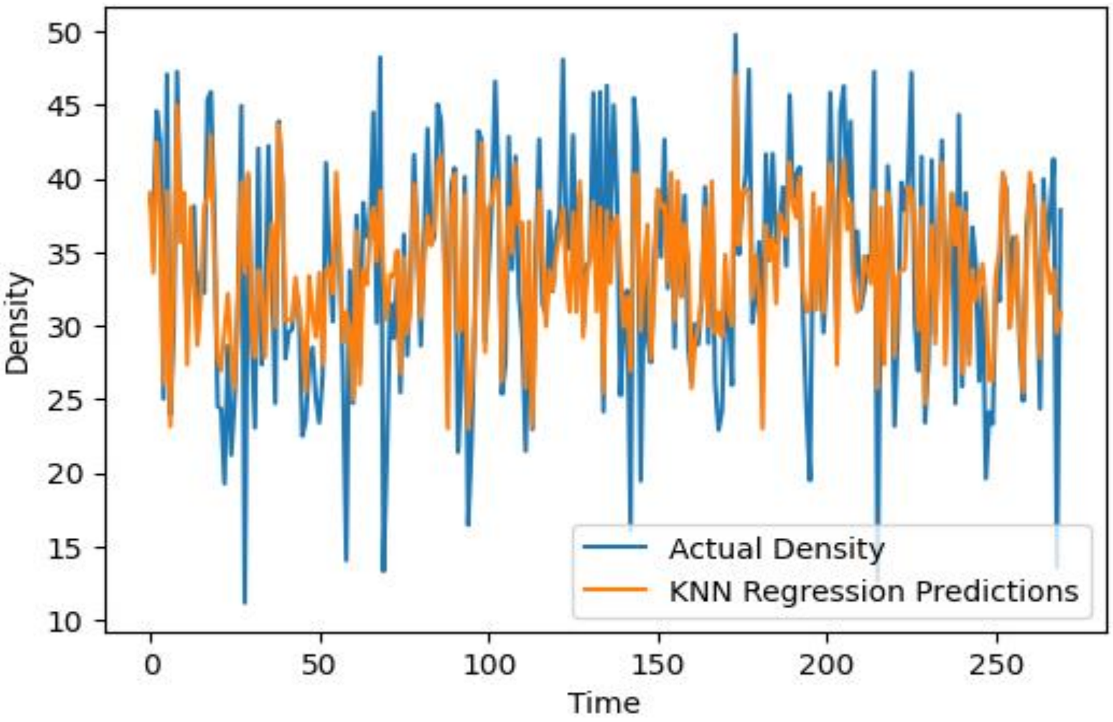


Figure 6.11 Actual and Predicted Density using KNN Regression

The graph indicates the actual values and the predicted values. The blue line indicates the actual values and the orange line indicates the predicted value. The X axis represents the time and Y axis represents the value predicted. So, in figure 6.10 curve corresponding to the random forest, the actual curve and predicted curve are more over similar. So, among the selected machine learning techniques, Random Forest is performing the best.

6.3.2 Mevaram to Kavanad

Table 6.4 shows the error and R-square values obtained from different Machine-Learning models. R-squared is a measure of regression model goodness-of-fit. The scatter of data points around the fitted regression line is calculated using R-squared. It is also known as the coefficient of determination or the coefficient of multiple determination in the case of multiple regression. Higher R-squared values indicate less disparities between observed and fitted values for the same data set. It was observed that the R square values obtained for all regression are very close to each other. It is greater than 97% for all machine learning regression models. The error values obtained from all models for this direction are small when compared to other direction. R square is 0.987 and 0.986 for Random Forest regression and XGBoost regression respectively. But it is high for Random Forest regression. For evaluating the performance of each model, RMSE, MSE, MAE AND

MAPE values were also used. The model having the least error and high R square value is considered as best.

Table 6.4 Evaluation Metrics for Mevaram to Kavanad Direction

Method	**MAE**	**MSE**	**RMSE**	**MAPE**	**R-square**
Linear Regression	0.714741	0.802043	0.895569	0.032138	0.98414
Random Forest Regression	0.407808	0.655716	0.799763	0.018269	0.987033
Support Vector Regression	0.748354	0.9493	0.97432	0.035143	0.981228
XGBoost Regression	0.461754	0.666648	0.807902	0.020682	0.98641
KNN Regression	0.703495	1.510984	1.229221	0.029436	0.970121

The comparison of MAE, R-squared values, MSE, RMSE and MAPE are shown in figures 6.12 to 6.16. The error values are also small for Random Forest regression when compared with other models. KNN regression have small R square value. But it is not very small when compared to other models. KNN regression have high error values when compared to others.

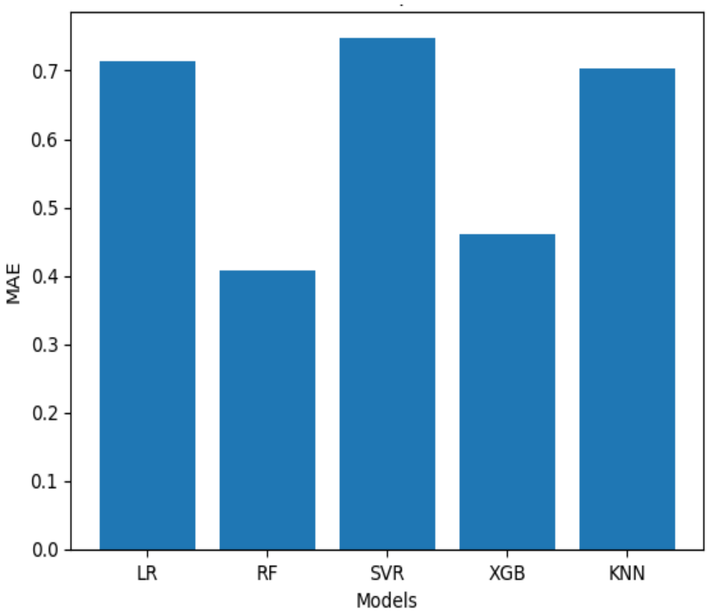


Figure 6.12 Comparison of MAE for Regression Models

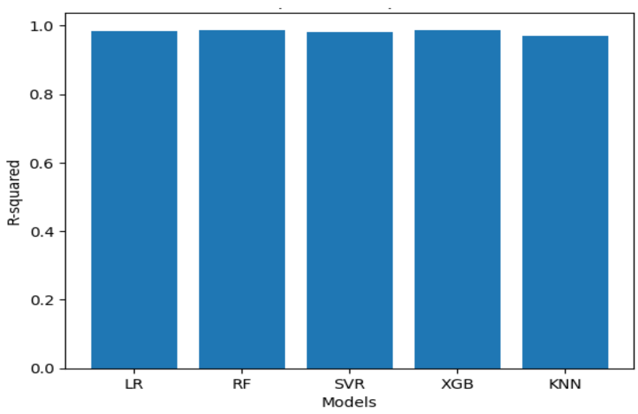


Figure 6.13 Comparison of R-squared values for Regression Models

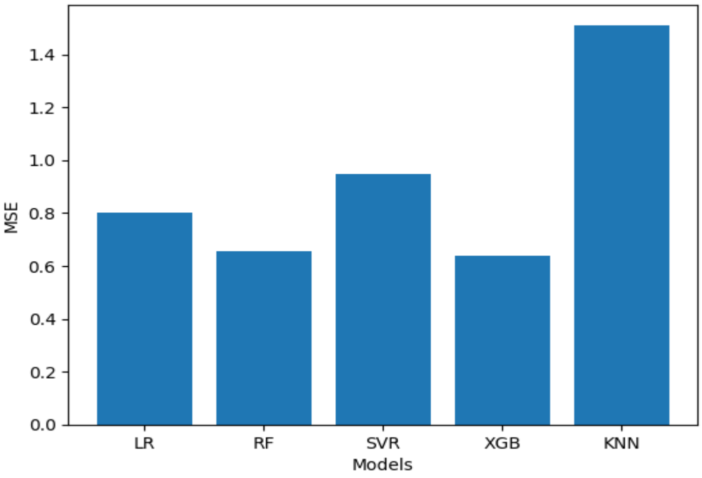


Figure 6.14 Comparison of MSE for Regression Models

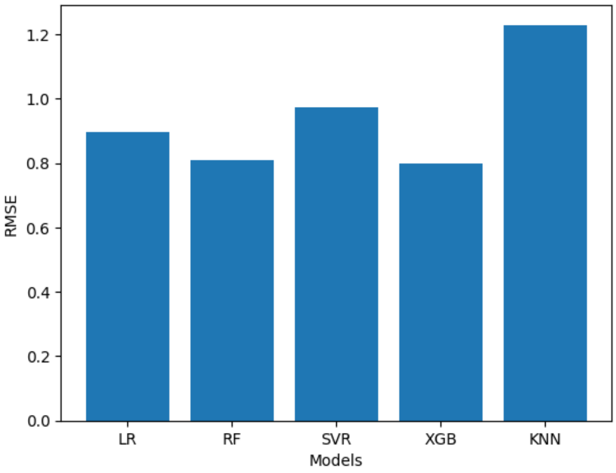


Figure 6.15 Comparison of RMSE for Regression Models

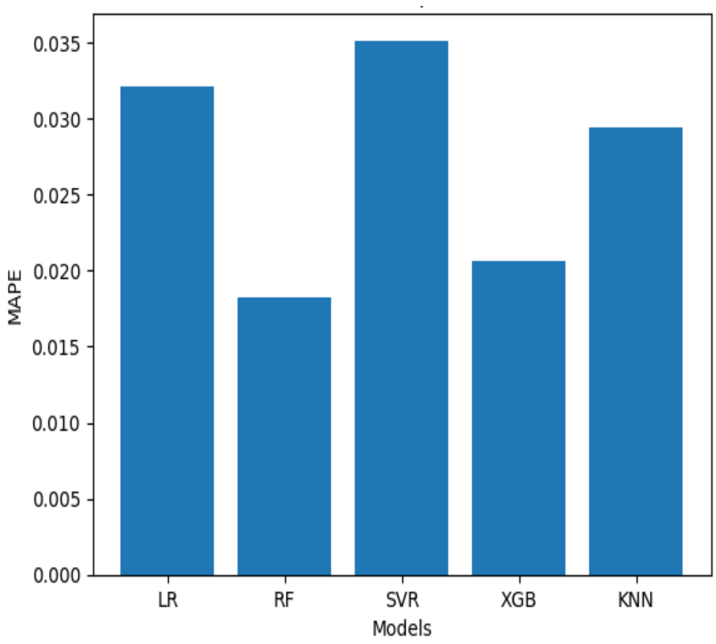


Figure 6.16 Comparison of MAPE for Regression Models

The RMSE, MSE, MAE and MAPE values were found to be very small for Random Forest regression compared to other methods. So, it can be considered the best method for predicting density in this direction. KNN regression results in high error values and low R square value when compared with others. The actual density and predicted density obtained from the best and less accurate models are also shown in Figure 6.17 and 6.18.

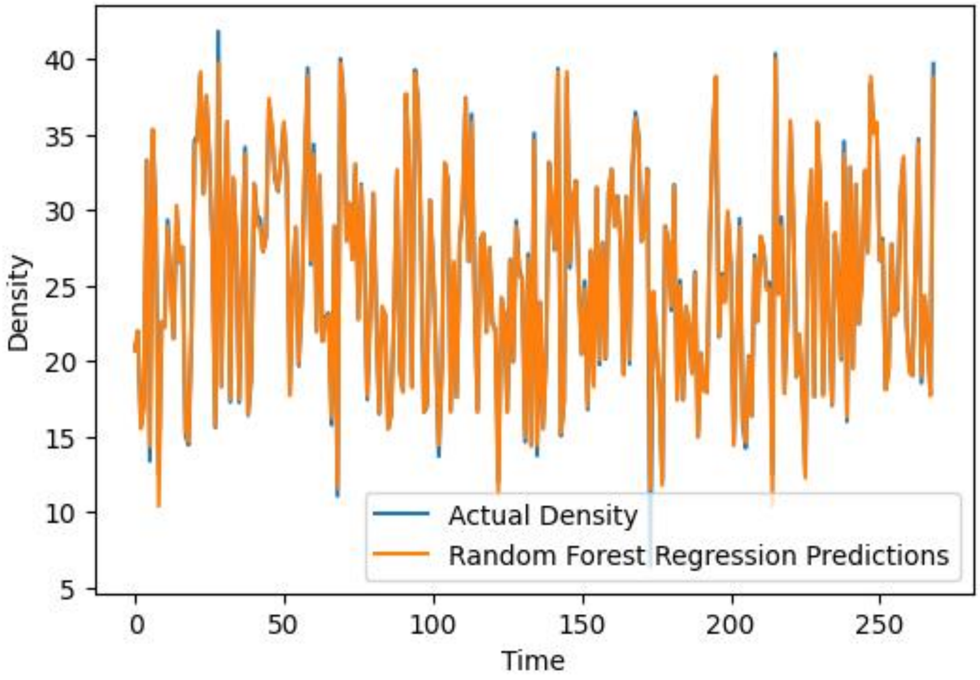


Figure 6.17 Actual and Predicted Density using Random Forest Regression

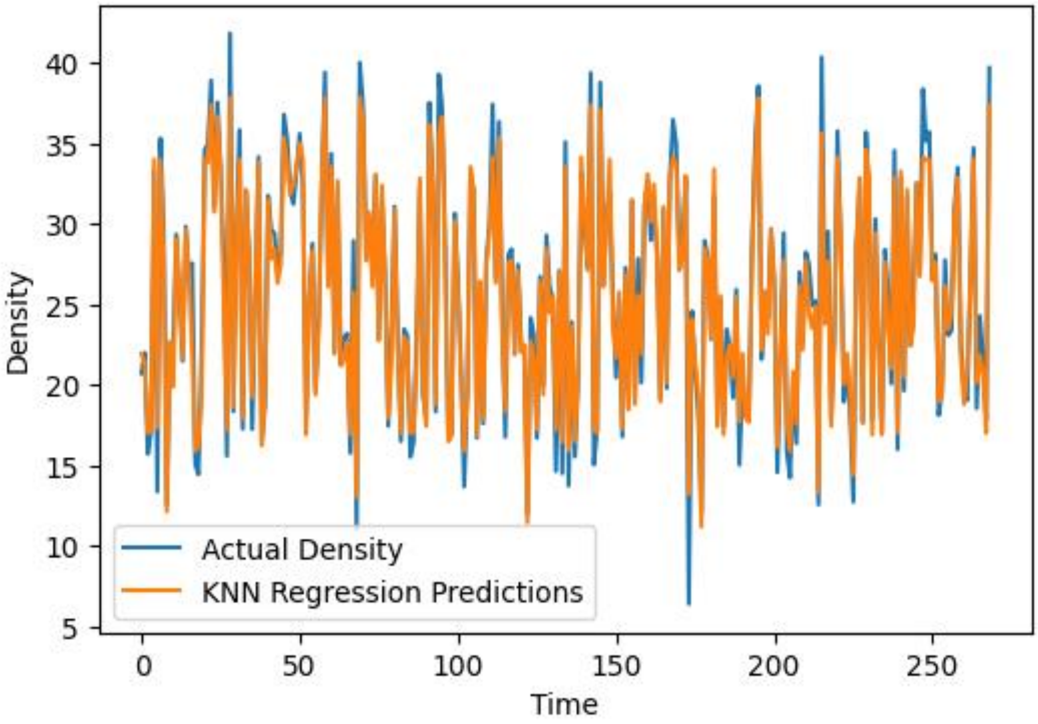


Figure 6.18 Actual and Predicted Density using KNN Regression

The graph indicates the actual values and the predicted values. The blue line indicates the actual values, and the orange line indicates the predicted value. The X axis represents the time and Y axis represents the value predicted. So, in figure 6.10 curve corresponding to the random forest, the actual and predicted curves were almost similar. So, it can be considered as the best method for predicting density in this direction.

6.4 COMPARISON OF KALMAN FILTER AND MACHINE LEARNING REGRESSION MODELS

Machine learning is a branch of Artificial Intelligence (AI) and can be used to predict traffic density by training a model on historical traffic data, which can be useful for predicting traffic patterns and optimizing traffic flow. Kalman filter, on the other hand, is a mathematical algorithm used to predict the state of a system based on its previous state and a series of measurements. When the MAPE value obtained for KF is high when compared with MAPE from Random Forest Regression, and XGBoost Regression. Implementing a Kalman filter requires a good understanding of the underlying mathematical concepts, and it can be challenging to tune the parameters of the filter for optimal performance. When Kalman filter was compared with ML regression methods, ML is found to be easily adaptable and gives better performances.

6.5 SUMMARY

Both ML and Kalman filter are used to predict traffic density. ML Regression models such as Random Forest Regression, Support Vector Regression, Linear Regression, XGBoost Regression and K-NN Regression were used. For evaluating the performance of each model, RMSE, R square, MSE, MAE and MAPE values were used. Among different ML regreesion models, the best method to predict traffic density in both direction is Random Forest regression. MAPE values are used to compare the performance of Kalman filter and MlL regression models. It is found that to optimize traffic flow and make long-term predictions, machine learning is more appropriate when compared with Kalman filter.

CHAPTER 7

CONCLUSIONS

7.1 GENERAL

Traffic density estimation of the undivided National Highway was conducted in this study. Traffic data was extracted using object detection algorithm YOLOv5. Density is estimated using the input-output method, Kalman filter and machine learning method. ML Regression models such as Random Forest Regression, Linear Regression, Support Vector Regression, XGBoost Regression and K-NN Regression were used. For evaluating the performance of each model, RMSE, R square, MSE, MAE and MAPE values were used. It was observed that the Random Forest Regression gives better performance measure and goodness of fit compared to other ML regression models. When comparing the MAPE values obtained from the Kalman filter and Random Forest Regression, it is very small for Random Forest Regression.

7.2 CONCLUSIONS

Present study compared different methods of traffic density estimation. Traffic video data collected from study stretch and data such as classified volume count disaggregate speed data are extracted using the YOLOv5 algorithm. Lane detection is used to obtain the count and speed of vehicles from each lane. Using computer vision for data extraction was found to be cost-effective compared to other methods such as manual traffic counts, which are labour-intensive and time-consuming. Real-time density estimation techniques used in this study are Machine learning regression methods and Kalman filter.

Kalman filter is a powerful tool for the prediction and filtering of time-series data, and it has a well-established theoretical foundation. Actual values of density are obtained from the input-output method. To predict density using the Kalman filter, speed density relationship developed from the speed and density data. Best fit for speed density relationship is found out by curve fitting. The estimation techniques used are the Kalman filter for the linear model. The MAPE value is calculated to find out the accuracy of the model-based estimation scheme. MAPE value for the speed-density linear model was found good for both directions. Implementing a Kalman filter requires a good understanding of the underlying mathematical concepts. It can be challenging to tune the parameters of the filter for optimal performance. When Kalman filter was compared with regression methods ML was found to implement easily.

Different machine learning regression models were also used to predict the density. Density (PCU/km) was taken as the dependent variable and average space mean speed (km/hr) and traffic flow (PCU/hr) were the independent variables. ML regression has gained popularity in recent years due to the availability of large datasets and advances in computational power. ML regression is flexible so that it can handle a wide range of data types and can learn complex relationships between predictors and outputs. ML Regression models such as Random Forest Regression, Support Vector Regression, Linear Regression, XGBoost Regression and K-NN Regression were used. For evaluating the performance of each model, RMSE, R square, MSE, MAE and MAPE values were used.

For Kavanad to Mevaram stretch, it was observed that the R square values obtained for XGBoost and Random Forest regression are very close to each other. R square is 0.986 and 0.981 for Random Forest regression and XGBoost regression respectively. But, it is high for Random Forest regression. The error values are also small for Random Forest regression when compared with other models. KNN regression have small R square and high error values when compared to others. So, it cannot be used for accurate prediction. The MAPE value obtained for Kalman filter and Random Forest regression are 0.145 and 0.0176 respectively. So, Random Forest is the best model to predict density accurately.

For Mevaram to Kavanad stretch, it was observed that the R square values obtained from all regression are very close to each other. The error values obtained from all models for this direction are small when compared to other direction. R square is 0.987 and 0.986 for Random Forest regression and XGBoost regression respectively. But it is high for Random Forest regression. So, Random Forest is the best model to predict density accurately. The error values are also small for Random Forest regression when compared with other models. KNN regression have small R square value and it is 0.970. But it is not very small when compared to other models. KNN regression have high error values when compared to others. The MAPE value obtained for Kalman filter and Random Forest regression are 0.288 and 0.0182 respectively. So, Random Forest is the best model to predict density accurately.

Among different ML regression models Random Forest Regression have high R square value and low values for RMSE, MSE, MAE and MAPE. When compared with the Kalman filter, MAPE was very small for Random Forest Regression. Random Forest Regression was found to be more accurate in predicting traffic density for both directions.

7.3 APPLICATIONS

Research is being done on predicting traffic density using machine learning regression tools and Kalman filter because these techniques can improve the accuracy and reliability of real-time traffic data. Real-time traffic data can be noisy and unreliable, and can be affected by factors such as incomplete or incorrect data, sensor failures, and changes in traffic patterns due to incidents or road work. Machine learning techniques can help to filter out noise and improve the accuracy of traffic density estimates.

Apart from providing real-time congestion information, predicting traffic density using machine learning techniques can have several other applications. Traffic density prediction can help traffic management authorities regulate traffic flow more effectively and reduce congestion. Authorities can optimise traffic flow by estimating traffic density in advance and adjusting traffic lights, rerouting traffic, and taking other actions. Traffic density forecasting can be used to build Intelligent Transportation Systems that optimise travel routes and types of transportation. These systems can recommend the optimal travel route and mode of transportation for users by predicting traffic density in real time. Authorities can estimate emissions and conduct emission-cutting measures by predicting traffic density. Estimating traffic density can aid in the design of new infrastructure such as highways, bridges, and tunnels. Authorities can ensure that new infrastructure projects are developed to meet future traffic demands by forecasting traffic density. Authorities can evaluate the economic impact of traffic congestion and take efforts to mitigate the negative impact on the economy by predicting traffic density.

7.4 SCOPE FOR FUTURE RESEARCH

Due to data collecting limits, only one section is considered for real-time traffic density estimation. The calculation of traffic density in this study was done at an aggregate level and did not take into account the classified density of vehicles. Further study is possible to estimate density at the classified level. In this study only limited amount of data were used. More studies can be conducted to test the proposed strategy using a greater amount and a more diversified sort of data. The research can be expanded to include roadways with varying geometric and traffic characteristics.

REFERENCES

1. Ajitha, T., Vanajakshi, L., and Subramanian, S. C. (2015). Real-time traffic density estimation without reliable side road data. *Journal of Computing in Civil Engineering*, *29* (2), 04014033.
2. Aljamal, M. A., Farag, M., and Rakha, H. A. (2020). Developing data-driven approaches for traffic density estimation using connected vehicle data. *IEEE Access*, *8*, 219622-219631.
3. Al-Sobky, A. S. A., and Mousa, R. M. (2016). The traffic density determination and its applications using smartphones. *Alexandria Engineering Journal*, *55* (1), 513-523.
4. Anand, R. A., Vanajakshi, L., and Subramanian, S. C. (2011). Traffic density estimation under heterogeneous traffic conditions using data fusion. In *2011 IEEE Intelligent Vehicles Symposium (IV)* 31-36.
5. Arasan, V. T., and Dhivya, G. (2010). Simulation of highly heterogeneous traffic flow characteristics. In *ECMS*, 81-87.
6. Asmaa, O., Mokhtar, K., and Abdelaziz, O. (2013). Road traffic density estimation using microscopic and macroscopic parameters. *Image and Vision Computing*, *31* (11), 887-894.
7. Aydin, S., Taşyürek, M. U. R. A. T., and Öztürk, C. Traffic Density Estimation using Machine Learning Methods. *Journal of Artificial Intelligence and Data Science*, *1*(2), 136-143.
8. Bharadwaj, N., Kumar, P., Mane, A. S., Arkatkar, S. S., Bhaskar, A., and Joshi, G. J. (2017). Comparative evaluation of density estimation methods on different uninterrupted roadway facilities: few case studies in India. *Transportation in developing economies*, *3* (1), 1-14.
9. Bhaskar, A., Tsubota, T., & Chung, E. (2014). Urban traffic state estimation: Fusing point and zone based data. *Transportation Research Part C: Emerging Technologies*, *48*, 120-142.
10. Chakraborty, P., Adu-Gyamfi, Y. O., Poddar, S., Ahsani, V., Sharma, A., and Sarkar, S. (2018). Traffic congestion detection from camera images using deep convolution neural networks. *Transportation Research Record*, *2672* (45), 222-231.
11. Dhivyabharathi, B., Fulari, S., Amrutsamanvar, R., Vanajakshi, L., Subramanian, S. C., and Panda, M. (2015). Performance comparison of filtering techniques for real-

time traffic density estimation under Indian urban traffic scenario. In *2015 IEEE 18th International Conference on Intelligent Transportation Systems* 1442-1447.

12. Edie, L. C. (1963). *Discussion of traffic stream measurements and definitions*. New York: Port of New York Authority,139-154.
13. Emami, A., Sarvi, M., and Asadi Bagloee, S. (2019). Using Kalman filter algorithm for short-term traffic flow prediction in a connected vehicle environment. *Journal of Modern Transportation*, *27*, 222-232.
14. Fedorov, A., Nikolskaia, K., Ivanov, S., Shepelev, V., and Minbaleev, A. (2019). Traffic flow estimation with data from a video surveillance camera. *Journal of Big Data*, *6* (1), 1-15.
15. Guo, J., Huang, W., and Williams, B. M. (2014). Adaptive Kalman filter approach for stochastic short-term traffic flow rate prediction and uncertainty quantification. *Transportation Research Part C: Emerging Technologies*, *43*, 50-64.
16. Ikiriwatte, A. K., Perera, D. D. R., Samarakoon, S. M. M. C., Dissanayake, D. M. W. C. B., and Rupasignhe, P. L. (2019). Traffic density estimation and traffic control using convolutional neural network. *In 2019 IEEE International Conference on Advancements in Computing (ICAC)*, 323-328.
17. Indian Roads Congress (1990). Guidelines for capacity of roads in rural area. IRC code of Practice, IRC: 64, 1990, New Delhi.
18. Julier, S. J., and Uhlmann, J. K. (2004). Unscented filtering and nonlinear estimation. *Proceedings of the IEEE*, *92* (3), 401-422.
19. Khan, S. M., Dey, K. C., and Chowdhury, M. (2017). Real-time traffic state estimation with connected vehicles. *IEEE Transactions on Intelligent Transportation Systems*, *18* (7), 1687-1699.
20. Khazukov, K., Shepelev, V., Karpeta, T., Shabiev, S., Slobodin, I., Charbadze, I., and Alferova, I. (2020). Real-time monitoring of traffic parameters. *Journal of big data*, *7* (1), 1-20.
21. Kumar, K., Parida, M., and Katiyar, V. K. (2015). Short term traffic flow prediction in heterogeneous condition using artificial neural network. *Transport*, *30* (4), 397-405.
22. Kumar, P., Arkatkar, S., Joshi, G., and Easa, S. (2023). Traffic Density Estimation Methods for Uninterrupted Roadway Facilities: Review and Guidelines. *Journal of Transportation Engineering, Part A: Systems*, *149* (2), 03122004.

23. Leclercq, L., Chiabaut, N., and Trinquier, B. (2014). Macroscopic fundamental diagrams: A cross-comparison of estimation methods. Transportation Research Part B: Methodological, 62, 1-12.
24. Ma, M., Liang, S., Guo, H., and Yang, J. (2017). Short-term traffic flow prediction using a self-adaptive two-dimensional forecasting method. *Advances in Mechanical Engineering*, *9* (8), 1687814017719002.
25. Makigami, Y., G. Newell, and R. Rothery. (1971). “Three-dimensional representation of traffic flow.” *Transportation Science,* 5, 302-313.
26. Mallikarjuna, C., and Rao, K. R. (2006). Area occupancy characteristics of heterogeneous traffic. *Transportmetrica*, *2* (3), 223-236.
27. Mane, D., Bidwe, R., Zope, B., and Ranjan, N. (2022). Traffic Density Classification for Multiclass Vehicles Using Customized Convolutional Neural Network for Smart City. In *Communication and Intelligent Systems: Proceedings of ICCIS 2021* (pp. 1015-1030). Singapore: Springer Nature Singapore.
28. Mihaylova, L., Boel, R., and Hegyi, A. (2007). Freeway traffic estimation within particle filtering framework. *Automatica*, *43* (2), 290-300.
29. Ozkurt, C., and Camci, F. (2009). Automatic traffic density estimation and vehicle classification for traffic surveillance systems using neural networks. *Mathematical and Computational Applications*, *14* (3), 187-196.
30. Raj, J., Bahuleyan, H., and Vanajakshi, L. D. (2016). Application of data mining techniques for traffic density estimation and prediction. *Transportation Research Procedia*, *17*, 321-330.
31. Ramchandra, N. R., and Rajabhushanam, C. (2022). Machine learning algorithms performance evaluation in traffic flow prediction. *Materials Today: Proceedings*, *51*, 1046-1050.
32. Romanowska, A., and Jamroz, K. (2021). Comparison of Traffic Flow Models with Real Traffic Data Based on a Quantitative Assessment. Applied Sciences, 11 (21), 9914.
33. Sen, R., Cross, A., Vashistha, A., Padmanabhan, V. N., Cutrell, E., and Thies, W. (2013). Accurate speed and density measurement for road traffic in India. In *Proceedings of the 3rd ACM Symposium on Computing for Development*,1-10.
34. Srivastava, S., Divekar, A. V., Anilkumar, C., Naik, I., Kulkarni, V., and Pattabiraman, V. (2021). Comparative analysis of deep learning image detection algorithms. *Journal of Big Data*, *8* (1), 1-27.

35. Wagner, F. A., and May Jr, A. D. (1963). Use of aerial photography in freeway traffic operations studies. *Highway Research Record*, *19*, 24-34.
36. Wang, Y., and Papageorgiou, M. (2005). Real-time freeway traffic state estimation based on extended Kalman filter: a general approach. *Transportation Research Part B: Methodological*, *39* (2), 141-167.

VISIBLE OUTPUT

Rohini, S and Madhu, k (2023). Advances in Traffic Density Estimation: A Review of Traditional and Modern Methods. International Conference on "Innovative Methods and Practical Applications for Cognizant Transportation Systems"(IMPACTS2023), March.