TRAFFIC PREDICTION MANAGEMENT

USING MACHINE LEARNING

A DESIGN PROJECT REPORT

Submitted by

NAVEEN KUMAR K

NEELA MUGESH S

SABARI PRIYAN G

SHAI SHARATH VIJEY S

in partial fulfilment for the award of the

degree of

BACHELOR OF TECHNOLOGY

in

ARTIFICIAL INTELLIGENGE AND DATA SCIENCE

K. RAMAKRISHNAN COLLEGE OF TECHNOLOGY

(An Autonomous Institution, affiliated to Anna University Chennai and Approved by AICTE, New Delhi)

SAMAYAPURAM–621112

DECEMBER-2024

K. RAMAKRISHNAN COLLEGE OF TECHNOLOGY
(AUTONOMOUS)
SAMAYAPURAM–621112

BONAFIDE CERTIFICATE

Certified that this design project report titled **"TRAFFIC PREDICTION MANAGEMENT USING MACHINE LEARNING"** is the Bonafide work of **NAVEEN KUMAR K (REG NO: 811721243037) NEELA MUGESH S (REG NO: 811721243039) SABARI PRIYAN G (REG NO: 811721243044) SHAI SHARATH VIJEY S (811721243043)** who carried out the project under my supervision. Certified further, that to the best of my knowledge the work reported here in does not form part of any other project report or dissertation based on which a degree or award was conferred on an earlier occasion on this or any other candidate.

SIGNATURE

Dr. T. AVUDAIAPPAN M.E., Ph.D.,

HEAD OF THE DEPARTMENT

Department of AI

K. Ramakrishnan College of Technology

(Autonomous)

Samayapuram – 621 112

SIGNATURE

Mr. T. PRAVEEN KUMAR M.E.,

ASSISTANT PROFESSOR

Department of AI

K. Ramakrishnan College of Technology

(Autonomous)

Samayapuram – 621 112

Submitted for the viva-voce examination held on ………………

INTERNAL EXAMINER **EXTERNAL EXAMINER**

DECLARATION

We jointly declare that the project report on **"TRAFFIC PREDICTION MANAGEMENT USING MACHINE LEARNING"** is the result of original work done by us and best of our knowledge, similar work has not been submitted to **"ANNA UNIVERSITY CHENNAI"** for the requirement of Degree of **BACHELOR OF TECHNOLOGY**. This design project report is submitted on the partial fulfilment of the requirement of the award of Degree of **BACHELOR OF TECHNOLOGY**.

SIGNATURE

SABARI PRIYAN G
(REG NO: 811721243044)

NAVEEN KUMAR K
(REG NO: 811721243037)

NEELA MUGESH S
(REG NO: 811721243039)

SHAI SHARATH VIJEY S
(REG NO: 811721243050)

PLACE **:** SAMAYAPURAM

DATE **:**

ACKNOWLEDGEMENT

It is with great pride that we express our gratitude and in - debt to our institution "**K. RAMAKRISHNAN COLLEGE OF TECHNOLOGY (AUTONOMOUS)**", for providing us with the opportunity to do this project.

We are glad to credit honorable chairman **Dr. K. RAMAKRISHNAN**, **B.E.,** for having provided for the facilities during the course of our study in college.

We would like to express our sincere thanks to our beloved Executive Director

Dr. S. KUPPUSAMY, MBA., Ph.D., for forwarding to our project and offering adequate duration in completing our project.

We would like to thank our principal **Dr. N. VASUDEVAN, M.E., Ph.D.,** who gave opportunity to frame the project the full satisfaction.

We whole heartily thanks to **Dr. T. AVUDAIAPPAN**, **M.E., Ph.D.,** HEAD OF THE DEPARTMENT, **ARTIFICAL INTELLIGENCE** for providing his encourage pursuing this project.

I express my deep and sincere gratitude to my project guide

Mr. T. PRAVEEN KUMAR M.E., ASSISTANT PROFESSOR, **ARTIFICIAL INTELLIGENCE** for his incalculable suggestions, creativity, assistance and patience which motivated me to carry out the project successfully.

I render my sincere thanks to my project coordinator **Mr. P. B. ARAVIND PRASAD M. Tech.,** other faculties and non-teaching staff members for providing valuable information during the course. I wish to express my special thanks to the officials & Lab Technicians of our departments who rendered their help during the period of the work progress.

ABSTRACT

The paper deals with traffic prediction that can be done in intelligent transportation systems which involve the prediction between previous year's data set and he recent year data which ultimately provides the accuracy and mean square error. This prediction will be helpful for the people who are in need to check the immediate traffic state. The traffic data is predicated on a basis of 1 hour time gap. Live statistics of the traffic is analyzed from this prediction. So, this will be easier to analyze when the user is on driving too. The system compares the data of all roads and determines the most populated roads of the city. I propose the regression model in order to predict the traffic using machine learning by importing Sklearn, Keras and TensorFlow libraries with the increasing urbanization and growth of vehicular traffic, efficient management and prediction of traffic conditions have become paramount for ensuring smooth transportation systems. This paper proposes an Intelligent Traffic Prediction and Management System (ITPMS) that leverages advanced technologies to analyze historical data, monitor real-time traffic patterns, and predict future traffic conditions. The ITPMS integrates machine learning algorithms, data analytics, and sensor technologies to create a comprehensive and dynamic traffic management solution. The system utilizes historical traffic data to train predictive models capable of forecasting traffic congestion, bottleneck locations, and peak hours. Real-time data from various sources, including traffic cameras, GPS devices, and social media, is continuously collected and analyzed to update and refine these predictive models. The predictive models are integrated into a centralized traffic management platform that provides actionable insights for traffic control authorities and commuters.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

GPS	GLOBAL POSITIONING SYSTEM
AI	ARTIFICIAL INTELLIGENCE
CNN	CONVOLUTIONAL NEURAL NETWORKS
V2X	VEHICLE TO EVERYTHING
ARIMA	AUTOREGRESSIVE INTEGRATED MOVING AVERAGE
TSP	TRAFFIC SIGNAL PRIORITY
ETA	ESTIMATED TIME OF ARRIVAL

CHAPTER 1

INTRODUCTION

Machine Learning (ML) is one of the most important and popular emerging branches these days as it is a part of Artificial Intelligence (AI). In recent times, machine learning becomes an essential and upcoming research area for transportation engineering, especially in traffic prediction. Traffic congestion affects the country's economy directly or indirectly by its means. Traffic congestion also takes people's valuable time, cost of fuel every single day. As traffic congestion is a major problem for all classes in society, there has to be a small-scale traffic prediction for the people's sake of living their lives without frustration or tension. For ensuring the country's economic growth, the road user's ease is required in the first place. This is possible only when the traffic flow is smooth. To deal with this, Traffic prediction is needed so that we can estimate or predict the future traffic to some extent.

1.1 BACKGROUND

Many reports of the traffic data are of actual time but it is not favorable and accessible to many users as we need to have prior decision in which route we need to travel. For example, during working days, we need to have daily traffic information or at times we need hourly traffic information but then the traffic congestion occurs; for solving this issue the user needs to have actual time traffic prediction. Many factors are responsible for the traffic congestion. This can be predicted by taking two datasets; one with the past year and one with the recent year's data set. If traffic is so heavy then the traffic can be predicted by referring the same time in the past year's data set and analyzing how congested the traffic would be. With the increasing cost of the fuel, the traffic congestion changes drastically. The goal of this prediction is to provide real-time gridlock and snarl up information. The traffic on the city becomes complex and are out of control these days, so such kind of systems are not sufficient for prediction.

Therefore, research on traffic flow prediction plays a major role in ITS.

1.2 PROBLEM STATEMENT:

To overcome the problem of traffic congestion, the traffic prediction using machine learning which contains regression model and libraries like pandas, os, numpy, matplotlib. pyplot are used to predict the traffic. This has to be implemented so that the traffic congestion is controlled and can be accessed easily. Users can collect the traffic information of the traffic flow and can also check the congestion flow from the start of the day till the end of the day with the time span of one hour data. In this way, Users can know the weather conditions of the roads that they would probably opt to take. This also tells the accuracy of the traffic by comparing their mean square errors of the past year's data and the recent year's data. Users can also know how many vehicles are traveling on average by the traffic prediction.

1.3 AIMS AND OBJECTIVES:

1.3.1 AIM:

- The system aims to enhance the overall efficiency of traffic by predicting congestion points, identifying bottlenecks, and dynamically adjusting traffic control measures.
- This optimization contributes to smoother traffic flow, reducing travel times and enhancing the overall transportation experience.
- By leveraging predictive analytics and real-time monitoring, the aim is to proactively address and mitigate traffic congestion.
- This involves providing timely information to both traffic management authorities and commuters, enabling them to make informed decisions and choose alternative routes when necessary.

1.3.2 OBJECTIVES:

- Develop algorithms and models that can accurately predict traffic conditions, taking into account historical data, real-time inputs, and external factors such as weather and special events.
- Implement a robust real-time monitoring system using sensors, cameras, and other data sources to continuously track and assess current traffic situations on roadways.
- Create mechanisms for dynamic adjustment of traffic signals, lane configurations, and other control measures based on real-time and predicted traffic patterns to optimize the flow of vehicles.
- Develop capabilities to identify potential congestion points and bottlenecks early, allowing for proactive intervention to prevent or alleviate traffic congestion before it significantly impacts the transportation network.
- Design intuitive and user-friendly interfaces, such as mobile applications and online platforms, to disseminate real-time traffic information to commuters. Provide alternative routes and estimated travel times to empower individuals to make informed decisions.
- Establish a comprehensive data analytics framework to process and analyze vast amounts of historical and real-time data. This includes the integration of machine learning algorithms to continuously improve the accuracy of traffic predictions.
- Implement features that enhance road safety, such as identifying accident-prone zones, recommending speed limit adjustments, and facilitating rapid response to emergencies through intelligent traffic management.
- Contribute to environmental sustainability by minimizing traffic congestion and optimizing traffic flow, thereby reducing fuel consumption, emissions, and overall environmental impact associated with urban transportation.

CHAPTER 2

LITERATURE SURVEY

2.1 TITLE: Predicting Traffic for Intelligent Transport System Using Machine Learning Algorithms

AUTHOR: Sai Akshita Kanaparthy, Raveena Reddy Vemula, Sai Niyathi Padakant

YEAR OF PUBLICATION: 2022

ALGORITHM USED: In this paper, different machine algorithms are employed for achieving higher efficiency and accurate results. To identify classification and regression, Decision Tree Algorithm (DT)is used. Based on the parameters supplied, it's a graphical depiction of all possible solutions to a problem/decision.

ABSTRACT: The increasing urbanization and growth in vehicular traffic pose significant challenges to urban transportation systems. This research focuses on leveraging machine learning algorithms to predict traffic patterns for Intelligent Transport Systems (ITS). By analyzing historical and real-time traffic data, our proposed model aims to enhance traffic management, improve safety, and optimize resource allocation. The study utilizes a diverse dataset comprising traffic flow, weather conditions, and historical incident data.

MERITS: Machine learning algorithms can analyze historical and real-time traffic data to predict congestion, enabling better traffic management and optimization of traffic flow.

DEMERITS: Decision trees can be difficult to interpret and explain, particularly as the complexity of the model increases. This can make it challenging to communicate the results of the model to stakeholders, which is essential in traffic prediction for effective decision-making.

2.2 TITLE: Prediction of HDD Failures by Ensemble Learning

AUTHOR: Narendran, Monishraj, Dr. Sathya Srivinas

YEAR OF PUBLICATION: 2022

ALGORITHM USED: In traffic prediction, ensemble learning can be applied in several ways. For example, different models can be trained on different subsets of data, and their predictions can be combined to produce an overall prediction

ABSTRACT: The reliability of Hard Disk Drives (HDDs) is critical for the continuous operation of data storage systems. This research focuses on the application of ensemble learning techniques for the prediction of HDD failures, aiming to provide early detection and minimize the risks associated with data loss and system downtime. The study employs a diverse dataset encompassing various parameters, including but not limited to, SMART (Self-Monitoring, Analysis and Reporting Technology) attributes, environmental factors, and usage patterns.

MERITS: Ensemble learning techniques can effectively combine multiple predictive models, enhancing the ability to detect subtle patterns indicative of impending hard disk drive (HDD) failures. This allows for early identification and intervention, reducing the risk of data loss and downtime.

DEMERITS: Ensemble learning in traffic prediction can have some disadvantages, such as increased computational complexity due to the need to train multiple models. Additionally, ensemble methods may be more challenging to interpret than individual models, making it more difficult to understand how the predictions were made.

2.3 TITLE: Traffic Data prediction in Intelligent Transportation System using mKNN algorithm and Principal Component Analysis.

AUTHOR: P.Pavithra , R.Vadivel

YEAR OF PUBLICATION: 2021

ALGORITHM USED: m-knn and pca

ABSTRACT: As urbanization continues to grow, the efficient management of traffic in Intelligent Transportation Systems (ITS) becomes increasingly crucial. This research proposes a novel approach to traffic data prediction by combining the m-KNN algorithm with Principal Component Analysis (PCA). The objective is to achieve high prediction accuracy, handle spatial correlations, and provide real-time insights for effective traffic management. The study utilizes a comprehensive dataset comprising diverse traffic parameters, including traffic flow, speed, and occupancy.

MERITS: The m-KNN algorithm, a modified version of the k-Nearest Neighbors algorithm, enhances prediction accuracy by considering multiple neighbors with varying weights. This is particularly advantageous in traffic data prediction, where subtle patterns and variations require a robust and accurate model

DEMERITS: The main difficulty encountered in dealing with problems of text mining is caused by the vagueness of natural language.

2.4 TITLE: Traffic Prediction for Intelligent Transportation System using Machine Learning.

AUTHOR: Gaurav Meena, Deepanjali Sharma, Mehul Mahrishi

YEAR OF PUBLICATION: 2020

ALGORITHM USED: To identify classification and regression we have used a Decision Tree Algorithm (DT). The goal of this method is to predict the value of the target variables. Decision tree learning represents a function that takes as input a vector of attributes value and return a" Decision" a single output value.

ABSTRACT: The escalating challenges posed by urbanization necessitate innovative solutions for efficient traffic management in Intelligent Transportation Systems (ITS). This research focuses on the application of machine learning techniques for traffic prediction, aiming to provide proactive insights for improved traffic control and navigation. The study employs a comprehensive dataset comprising historical traffic patterns, environmental conditions, and other relevant parameters. Machine learning algorithms, including regression models, decision trees, and neural networks, are utilized to develop predictive models capable of forecasting traffic conditions at different times and locations.

MERITS: Machine learning models enable the proactive prediction of traffic conditions, allowing for timely intervention and the implementation of adaptive traffic management strategies to alleviate congestion and improve overall traffic flow.

DEMERIT: Decision trees are prone to overfitting on traffic data. They lack robustness and may not provide accurate predictions on new, unseen traffic patterns.

2.5 TITLE: Machine Learning Methods for Traffic Prediction in Dynamic Optical Networks with Service Chains.

AUTHOR: Daniel Szostak and Krzysztof Walkowiak

YEAR OF PUBLICATION: 2019

ALGORITHM USED: Dynamic Optical Networks (DONs) are being increasingly used in traffic prediction due to their ability to provide high bandwidth and low-latency communication. They enable efficient and reliable data transfer, which is essential for traffic prediction systems to work effectively.

ABSTRACT: The evolution of optical networks towards dynamic configurations and the integration of service chains presents unprecedented challenges in managing and optimizing network resources. This research explores the application of machine learning methods for traffic prediction in dynamic optical networks, with a focus on supporting diverse service chains efficiently. The study utilizes a dataset encompassing dynamic traffic patterns associated with service chains, considering factors such as bandwidth requirements, network topology, and service dependencies.

MERITS: Machine learning methods facilitate the dynamic adaptation of traffic prediction models to changing conditions in optical networks. This adaptability is essential for predicting and managing traffic variations associated with diverse service chains and evolving network configurations.

DEMERITS: DONs may not be scalable enough to handle large- scale traffic prediction problems, as they rely on complex optical components and may require significant computational resources.

CHAPTER 3

SYSTEM SPECIFICATION

3.1 H/W SYSTEM CONFIGURATION:

- Edge Devices/Sensors: IoT devices, traffic cameras, GPS units.
- Microcontrollers: Raspberry Pi, Arduino, or other edge computing devices for local data preprocessing.
- Connectivity: High-speed internet or dedicated communication channels for data transfer.
- Storage: Fast and scalable storage, preferably SSDs for quick data access.

3.2 S/W SYSTEM CONFIGURATION:

- Python programming language - Python 3.x installed on the computer/server
- Operating system - Windows, Linux, or macOS.
- Python libraries such as – opencv , numpy, pandas, tensorflow, Scikit-Learn.
- Django or Flask (Python): For developing the backend of a web-based traffic prediction dashboard.
- HTML/CSS or JavaScript - for UI design

3.3 SOFTWARE DESCRIPTION:

A traffic prediction system is a software solution designed to forecast and analyze traffic conditions to provide valuable insights for efficient traffic management, route planning, and congestion mitigation.

3.3.1 LIBRARY

To develop the Traffic prediction management system, the following libraries are commonly used:

1. Pandas: Data manipulation and analysis library, useful for handling structured data and time-series operations.

2. **NumPy:** NumPy is a fundamental library for numerical computations in Python. It provides efficient numerical operations and arrays, which are essential for processing and manipulating image data in computer vision applications.

3. **TensorFlow or PyTorch:** TensorFlow and PyTorch are deep learning(DL) frameworks commonly used for training and deploying machine learning models. They offer a range of tools and functions for building and training neural networks, including models for face recognition.

4. Prophet: A forecasting tool designed for time-series analysis, developed by Facebook.

5. Stats models: Library for estimating and testing statistical models, which can be useful for analyzing time-series data and building statistical models.

3.3.2 Developing Environment

To develop the Traffic prediction management system, you would typically set up the following environment:

1. **Python:** Python is the primary programming language used for developing the system. Ensure that Python is installed on your system.
2. **Integrated Development Environment (IDE):** Choose an IDE for Python development, such as PyCharm, Visual Studio Code, or Jupyter Notebook. These IDEs provide features like code editing, debugging, and project management, enhancing the development process.

3. **Install Required Libraries:** Use the Python package manager, pip, to install the necessary libraries such as OpenCV, NumPy, TensorFlow or PyTorch, and Flask. You can install them using the command line interface or directly within your IDE.

4. MongoDB, Cassandra, or InfluxDB: NoSQL databases suitable for storing and retrieving traffic-related data.

5. Docker: Containerization allows for consistent deployment across different environments and simplifies dependencies management.

6. Git: Utilize Git for version control to track changes, collaborate with a team, and manage code history.

7. SQL Alchemy: A SQL toolkit and Object-Relational Mapping (ORM) library for Python, useful for interacting with relational databases.

8. **User Interface Design:** Design and develop the user interface using HTML, CSS, and JavaScript. Flask supports template rendering, allowing you to dynamically generate HTML pages based on the data from the backend.
9. **Testing and Deployment:** Test your application thoroughly, checking for any bugs or issues. Once the testing phase is complete, you can deploy the application to a web server or cloud platform for online access.

By following these steps, you can develop the Traffic prediction management system. It's essential to continuously update and maintain the system, ensuring its accuracy, security, and performance over time.

CHAPTER 4

SYSTEM ANALYSIS

4.1 EXISTING SYSTEM

In response to the escalating challenges in urban traffic management, the proposed system aims to revolutionize the way we anticipate and mitigate traffic congestion through an advanced Traffic Prediction Management System. Leveraging cutting-edge technologies and data-driven approaches, the system integrates real-time data streams from diverse sources, including traffic cameras, GPS devices, weather stations, and social media platforms. Through the fusion of machine learning algorithms, statistical models, and time series forecasting techniques, the system analyzes historical and current traffic patterns to make accurate predictions about future conditions.

ALGORITHM USED:

1. ARIMA (AutoRegressive Integrated Moving Average):
A statistical method suitable for analyzing and forecasting time-series data, such as historical traffic patterns.

2. Machine Learning Algorithms:

Regression Models: Linear regression or polynomial regression models can be employed to capture relationships between traffic patterns and various influencing factors.

Decision Trees and Random Forest: Decision trees can model complex relationships, and random forests can improve accuracy through ensemble learning.

Gradient Boosting Models: Algorithms like XGBoost or LightGBM may be used to boost the performance of decision trees for more accurate predictions.

Neural Networks: Deep learning models, particularly recurrent neural networks (RNNs) or long short-term memory networks (LSTMs), can capture intricate temporal dependencies in traffic data

3. **Clustering Algorithms:**

K-Means Clustering: Grouping similar traffic patterns can enhance prediction accuracy by allowing the model to differentiate between different traffic clusters.

4. Feature Engineering:

Feature Selection and Engineering Techniques: Selecting relevant features and engineering new features based on domain knowledge to enhance the predictive power of the models.

5. Spatial-Temporal Models:

Spatio-Temporal Neural Networks: Specialized neural networks designed to capture both spatial and temporal dependencies in traffic data, considering the geographic layout of the road network.

6. Ensemble Learning:

Ensemble Methods (Bagging and Boosting): Combining predictions from multiple models, such as decision trees or regression models, to improve overall prediction accuracy.

7. Probabilistic Forecasting:

Bayesian Methods: Utilizing Bayesian inference to model uncertainties and provide probabilistic traffic forecasts.

4.1.1 DRAWBACKS:

- Data Quality and Reliability
- Model overfitting
- Limited generalizations

- Complexity and Resources intensiveness
- Model Interpretability

4.2 PROPOSED SYSTEM:

In response to the escalating challenges in urban traffic management, we propose an innovative Traffic Prediction Management System designed to revolutionize the way we anticipate and mitigate traffic congestion. Our system leverages state-of-the-art technologies, including advanced machine learning algorithms, real-time data processing, and dynamic visualization tools. The core of our approach involves ingesting diverse data sources, such as traffic cameras, GPS devices, and weather stations, into a unified and scalable platform.

ALGORITHM USED:

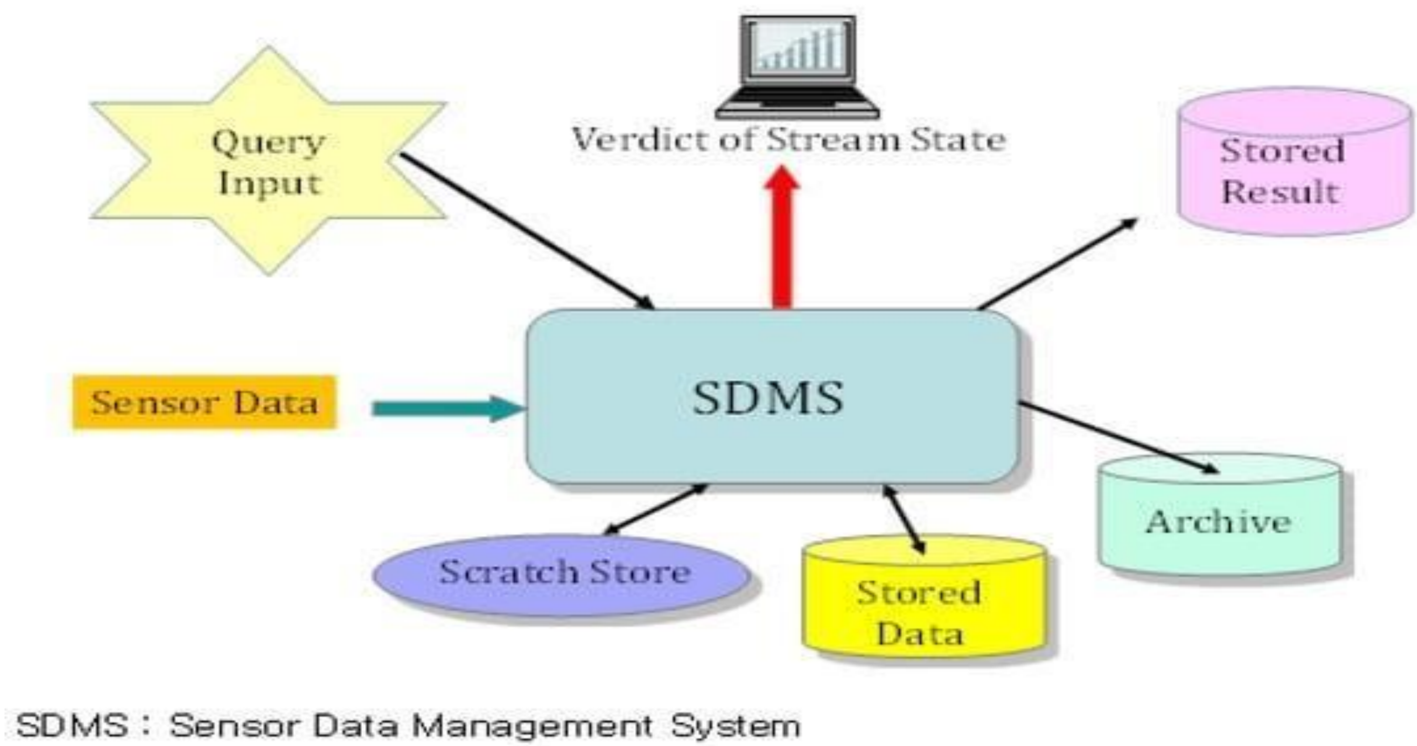


Fig.No.4.1. Design Phases of Algorithm

Data Collection:

Types of Sensors: Deploy a variety of sensors, including traffic cameras, inductive loop detectors, GPS devices, weather stations, and other IoT sensors.

Data Fusion: Integrate data from multiple sensor types to provide a comprehensive view of traffic conditions.

Spatial and Temporal Analysis:

Geospatial Processing: Leverage geospatial analysis to correlate sensor data with the physical layout of the road network.

Temporal Analysis: Analyze data over time to identify trends, patterns, and recurring traffic conditions.

Data Visualization:

Dashboard Integration: Integrate sensor data into the web-based dashboard, providing visual representations and real-time updates for users.

Map Integration: Use mapping libraries (e.g., Leaflet, Mapbox) to display sensor data on interactive maps.

CNN ALGORITHM: Convolutional Neural Networks (CNNs) can be effectively utilized in traffic prediction management systems, especially when dealing with image data obtained from traffic cameras or other visual sensors. CNNs are a type of deep learning architecture designed to automatically and adaptively learn spatial hierarchies of features from input images.

4.2.1 ADVANTAGES:

- Handle large datasets
- Identify vehicle with 97% accuracy.
- Less error rate
- Very accurate and robust • It can prevent from unauthorized user.
- Time saving & avoid complexity.

CHAPTER 5

ARCHITECTURAL DESIGN

5.1 SYSTEM DESIGN:

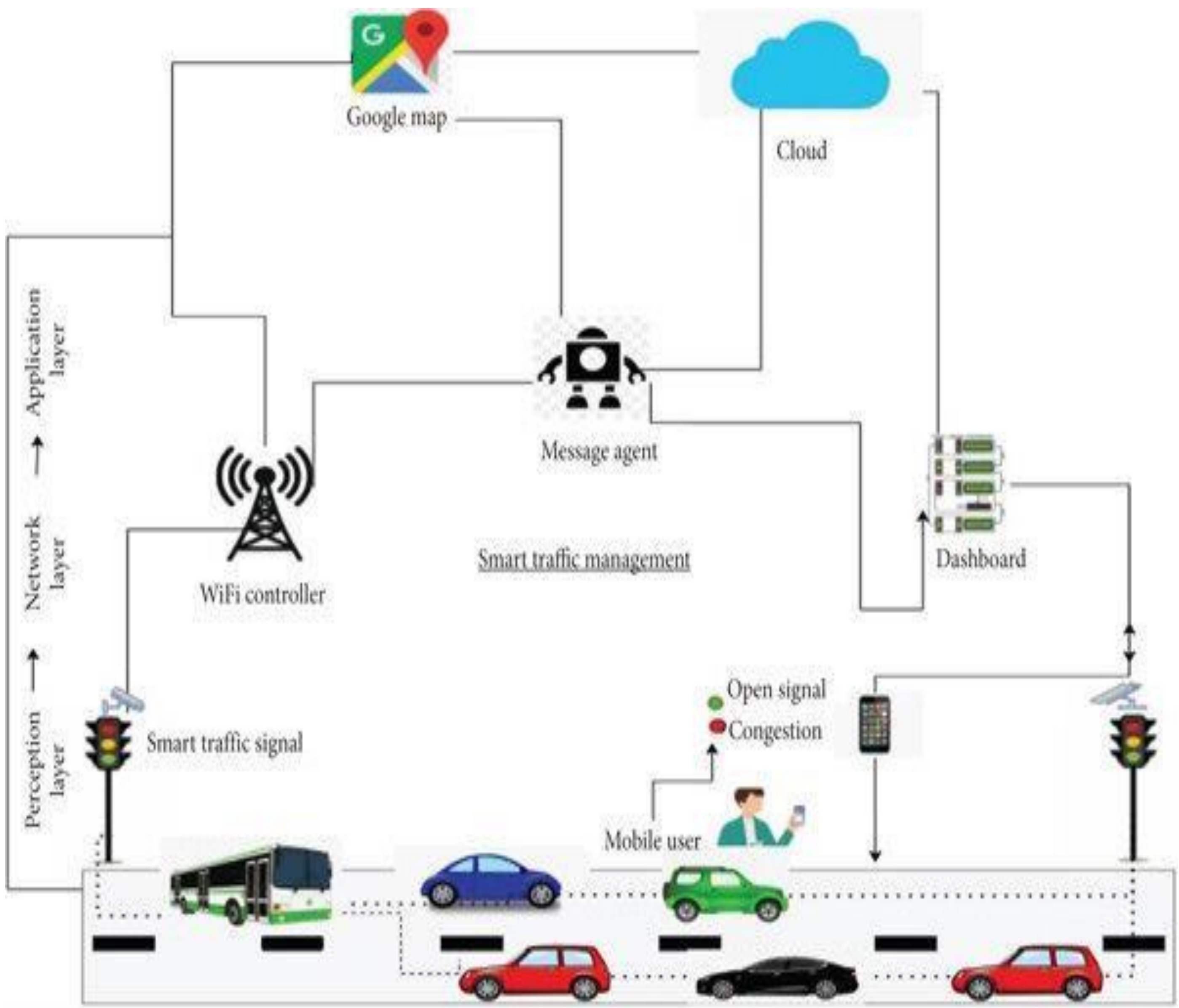


Fig.No.5.1. Architectural Design

5.2 DATA FLOW DIAGRAM:

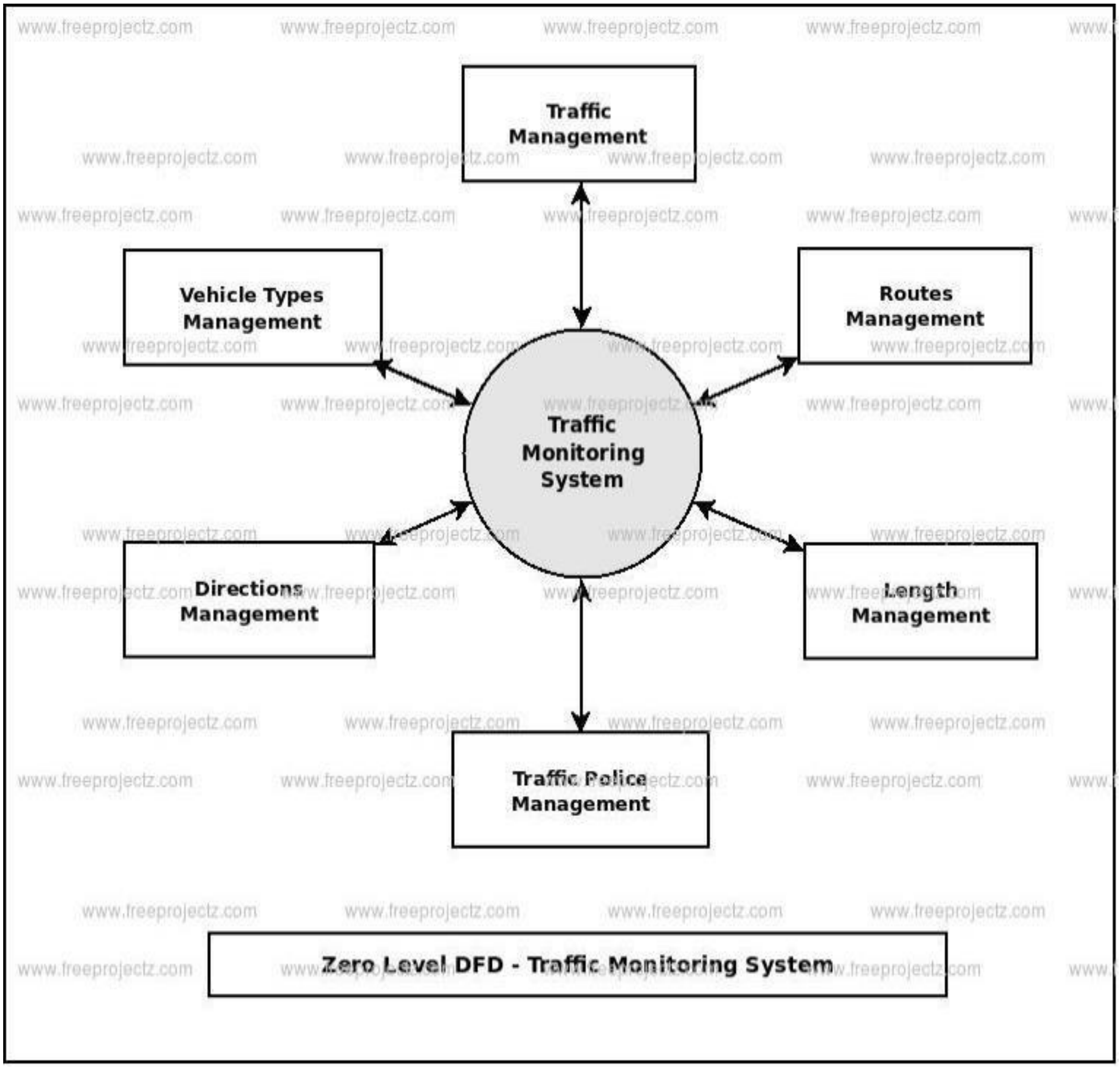


Fig.No.5.2. Data Flow diagram

5.3 USE CASE DIAGRAM:

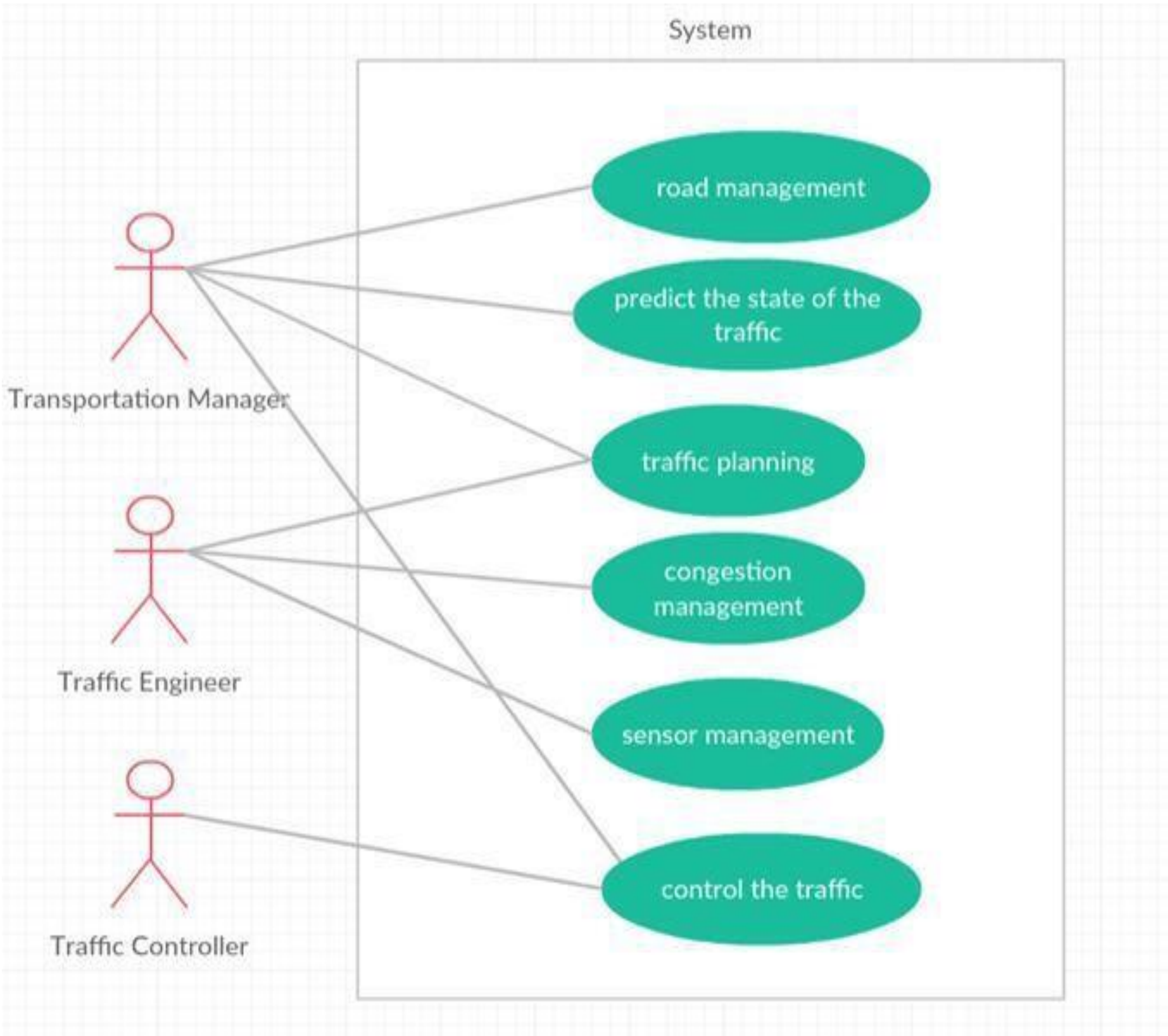


Fig.No.5.3. Use Case Diagram

5.4 ACTIVITY DIAGRAM:

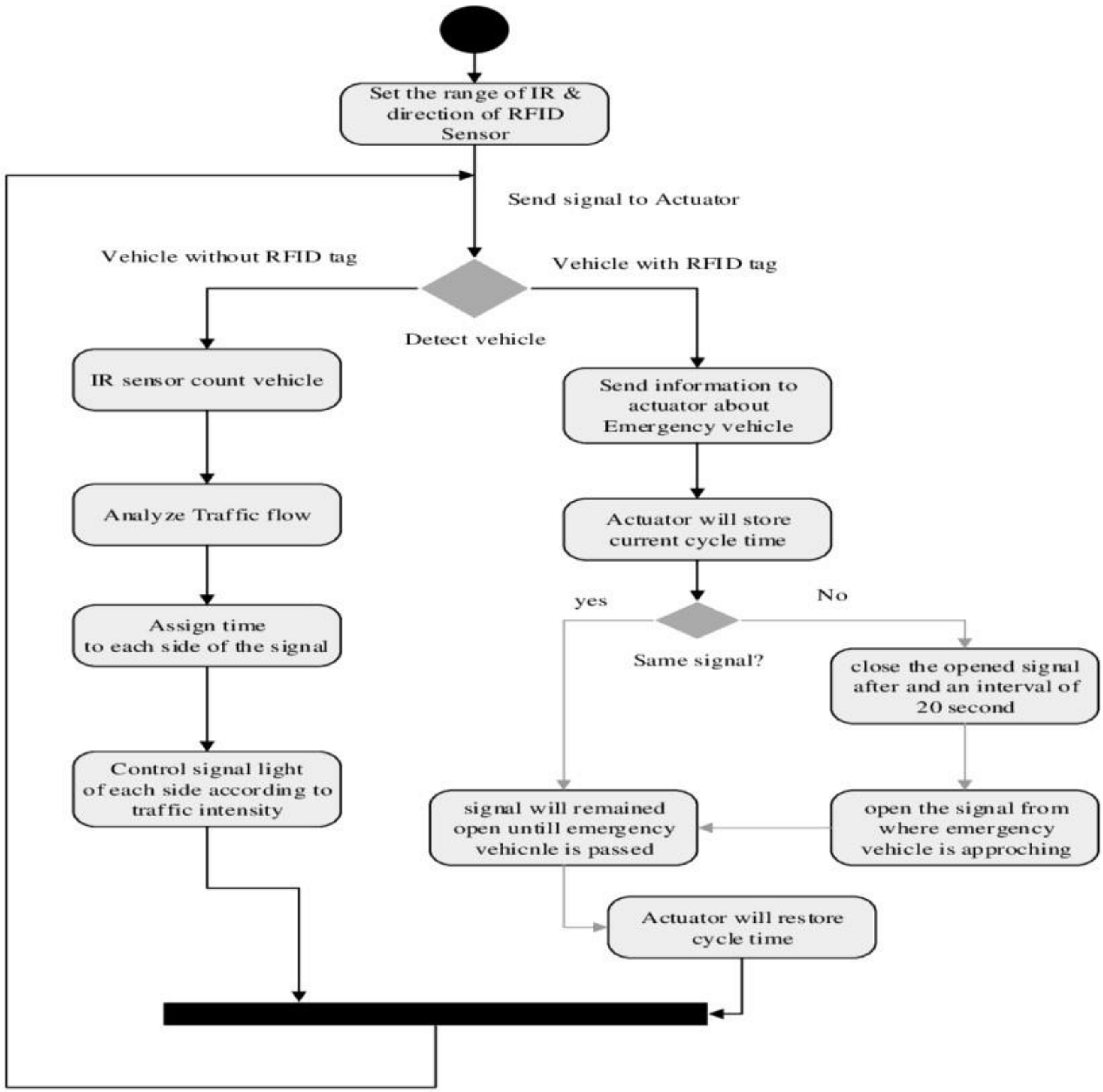


Fig.No.5.4. Activity Diagram

5.5 SEQUENCE DIAGRAM:

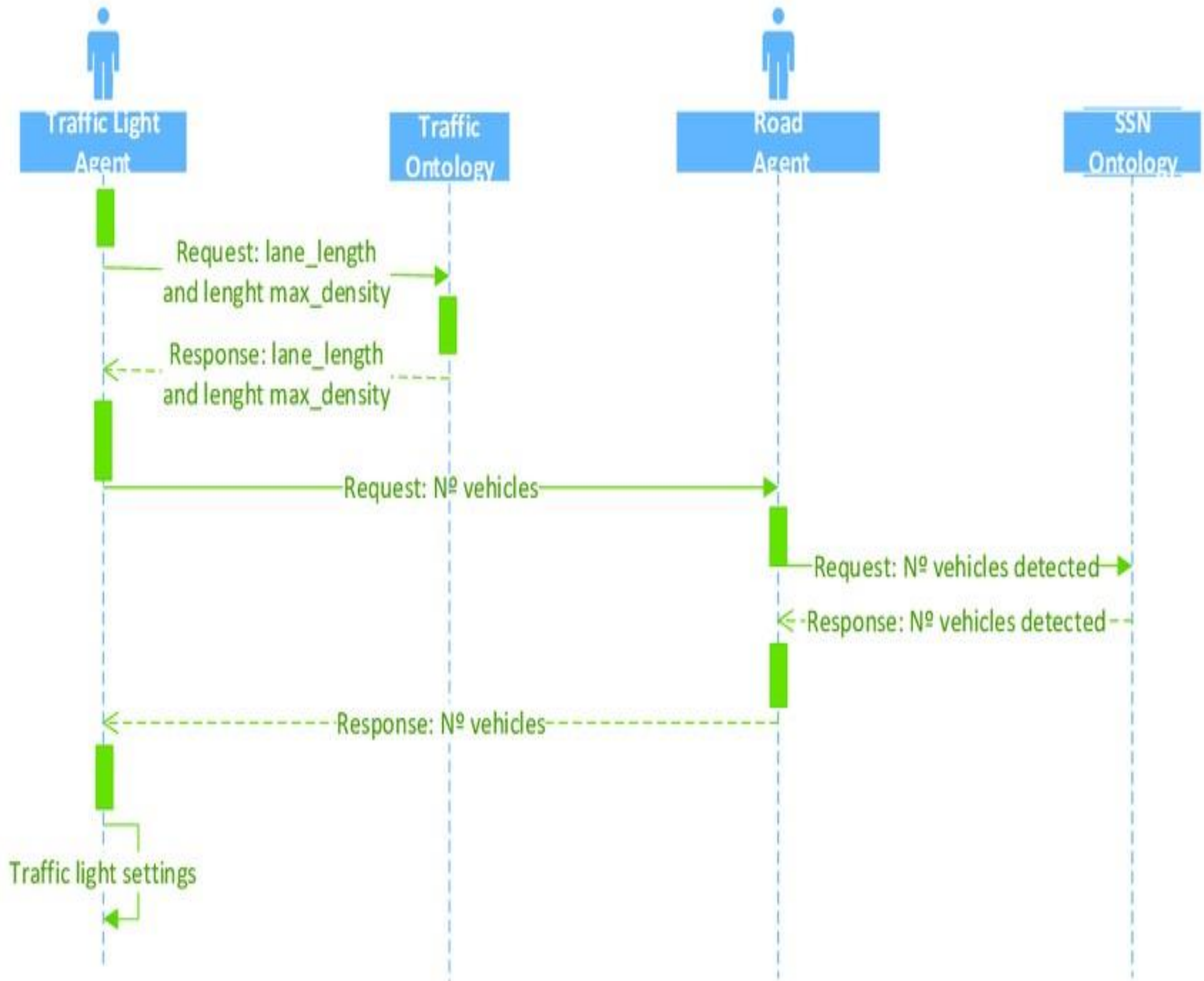


Fig.No.5.5. Sequence Diagram

CHAPTER 6
MODULE DESCRIPTION

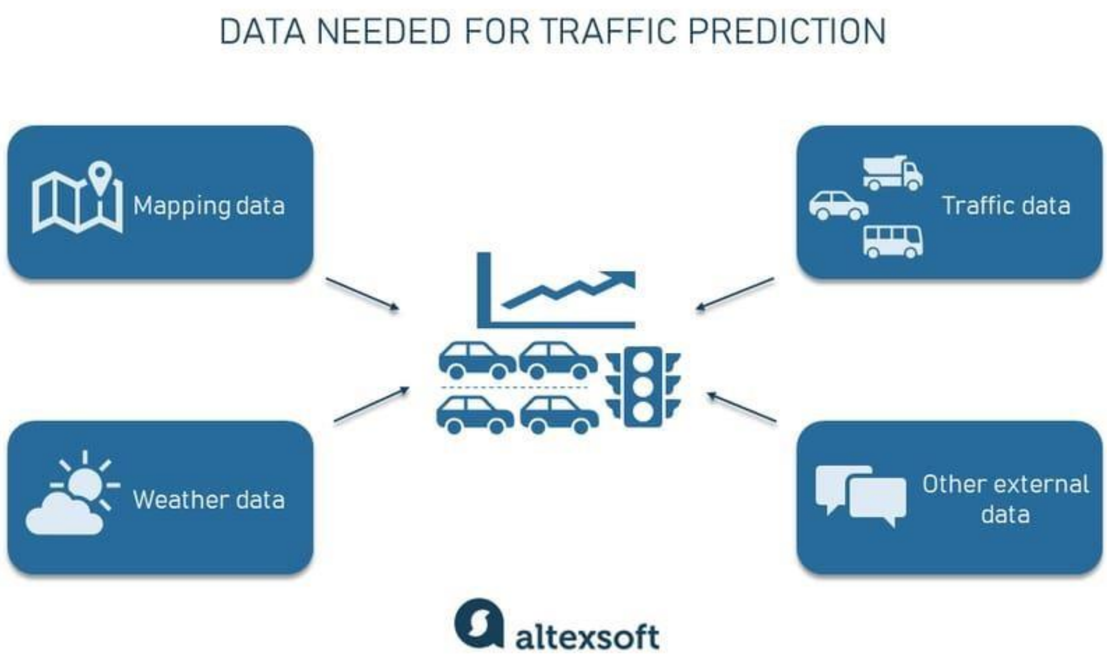


Fig.No.6.1. Phases Of Proposed System

6. 1 MODULES:

6.1.1 MAPPING DATA

First of all, you need to have a detailed map with road networks and related attributes. connecting to such global mapping data providers as Google Maps, TomTom, HERE, or OSM is a great way to obtain complete and up-to-date information.

Here are the key tasks involved in the Mapping data module:

1.Route optimization: Integrate mapping features to suggest optimized routes based on predicted traffic conditions. This helps users navigate efficiently and avoid congested areas

2.Integration with traffic cameras: Link traffic camera feeds to the map interface, allowing users to view live images from specific locations.

6.1.2 TRAFFIC DATA MODULE:

Traffic information. Then, you'll have to collect both historical and current traffic-related information such as the number of vehicles passing at a certain point, their speed, and type (trucks, light vehicles, etc.). Devices used to collect this data are

- loop detectors,
- cameras,
- weigh in motion sensors, and
- radars, or other sensor technologies.

Fortunately, you don't have to install these devices all over the place on your own. It's easier to get this information from the aforementioned providers that gather data from a system of sensors, diverse third-party sources, or make use of GPS probe data. (Just in case you're unfamiliar with how infrastructures for collecting, processing, and storing data are designed and work, you can visit our related post on data engineering to get an idea.)Other platforms such as Otonomo use an innovative Vehicle to Everything (V2X) technology to collect so-called connected car data from embedded modems.

6.1.3 WEATHER INFORMATION MODULE:

Weather data (historical, current, and forecasted) is also necessary as meteorological conditions impact the road situation and driving speed. There are lots of weather data providers you can connect to — such as Open Weather or Tomorrow.io.

Here are the key tasks involved in the Weather information module:

- Weather Stations: Utilize data from local weather stations to obtain real-time information on factors such as temperature, precipitation, wind speed, and visibility.
- Time Synchronization: Align weather data with the corresponding traffic data to ensure temporal synchronization.

- Weather-Triggered Alerts: Implement alerts and notifications based on predicted adverse weather conditions that may affect traffic. This allows for proactive measures to be taken.

6.1.4 WEB INTERFACE MODULE:

The web interface module of a Traffic Prediction Management System serves as the user interface, providing a platform for visualizing, analyzing, and interacting with traffic-related data. Here are key components and features that can be included in the web interface module:

- Present a dashboard that offers a high-level overview of current traffic conditions, key metrics, and alerts.
- Integrate interactive maps using mapping libraries like Leaflet, Mapbox, or Google Maps.
- Represent traffic flow using color-coded markers, lines, or heatmaps on the map.

6.1.5 INFO PREDICTION MODULE:

In a Traffic Prediction Management System, an information prediction module is crucial for forecasting and disseminating relevant information regarding traffic conditions. This module leverages predictive analytics to anticipate future traffic patterns, incidents, and congestion. Here are key components and features of an information prediction module:

1. **Predictive model:** Develop machine learning models, such as regression models, time series forecasting models, or neural networks, to predict future traffic conditions based on historical data.

2. **Real time data integration:** Integrate real-time data streams from sensors, traffic cameras, and other sources into the predictive models for up-to-the-minute accuracy.

3. **Event prediction:** Predict potential events that could impact traffic, such as accidents, road closures, or planned construction activities.

6.1.6 RELATION MODULE:

The relation module within a Traffic Prediction Management System plays a pivotal role in establishing connections and dependencies among various data sources, components, and entities involved in the traffic management ecosystem. This module serves as the backbone for integrating diverse data streams, facilitating effective communication between predictive models, visualization interfaces, and external systems. It establishes the relationships needed to interpret the complex interplay of factors influencing traffic conditions. By orchestrating these relationships, the module enables a holistic understanding of the urban mobility landscape. It manages the integration of traffic data from sensors, weather information, and historical patterns, fostering a comprehensive view for predictive modeling. Additionally, the relation module facilitates the seamless interaction between the predictive models and the visualization interfaces, ensuring that the predicted traffic conditions are accurately conveyed to users in real-time. Integration with external systems, such as traffic signal control systems or emergency services, is also streamlined through this module, fostering collaborative efforts in traffic management. The relation module acts as a dynamic hub, adapting to evolving data sources, technological advancements, and user requirements, thereby playing a vital role in the efficiency and adaptability of the overall Traffic Prediction Management System.

CHAPTER 7

CONCLUSION & FUTURE SCOPE

7.1 CONCLUSION:

In the system, it has been concluded that we develop the traffic flow prediction system by using a machine learning algorithm. By using regression model, the prediction is done. The public gets the benefits such as the current situation of the traffic flow, they can also check what will be the flow of traffic on the right after one hour of the situation and they can also know how the roads are as they can know mean of the vehicles passing though a particular junction that is 4 here. The weather conditions have been changing from years to years. The cost of fuel is also playing a major role in the transportation system. Many people are not able to afford the vehicle because of the fuel cost. So, there can be many variations in the traffic data. There is one more scenario where people prefer going on their own vehicle without carpooling, this also matters in the traffic congestion. So, this prediction can help judging the traffic flow by comparing them with these 2 years data sets. The forecasting or the prediction can help people or the users in judging the road traffic easier beforehand and even they can decide which way to go using their navigator and also this will prediction will be also helpful.

7.2 FUTURE SCOPE:

In the future, the system is often further improved using more factors that affect traffic management using other methods like deep learning, artificial neural network, and even big data. The users can then use this technique to seek out which route would be easiest to achieve on destination. The system can help in suggesting the users with their choice of search and also it can help to find the simplest choice where traffic isn't in any crowded environment. Many forecasting methods have already been applied in road traffic jam forecasting. While there's more scope to create the congestion prediction more precise, there are more methods that give precise and accurate results from the prediction. Also, during this period, the employment of the increased available traffic data by applying the newly developed forecasting models can improve the prediction accuracy. These days, traffic prediction is extremely necessary for pretty much a part of the state and also worldwide. So, this method of prediction would be helpful in predicting the traffic before and beforehand. For better congestion purpose grade and accuracy are prominent in traffic prediction. within the future, the expectation is going to be the estimation of established order accuracy prediction with much easier and user-friendly methods so people would find the prediction model useful and that they won't be wasting their time and energy to predict the information. There will be some more accessibility like weather outlook, GPS that's the road and accident-prone areas will be highlighted in order that people wouldn't prefer using the paths which aren't safe and simultaneously they'll predict the traffic. This will be done by deep learning, big data, and artificial neural network.

APPENDIX (SAMPLE CODE)

APPENDIX 1:

FRONTEND:

LOGIN PAGE:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Traffic Prediction Management - Login</title>
  <style>
    body {
      font-family: Arial, sans-serif;
      margin: 0;
      padding: 0;
      background-color: #f4f4f4;
    }
    .header {
      background-color: #333;
      color: #fff;
      padding: 15px 20px;
      text-align: center;
    }
    .container {
      max-width: 400px;
      margin: 100px auto;
      background: #fff;
      padding: 20px;
      box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);
    }
    .form-group {
      margin-bottom: 15px;
```

```
}
label {
   display: block;
   font-weight: bold;
   margin-bottom: 5px;
}
input[type="text"],
input[type="password"] {
   width: 100%;
   padding: 10px;
   border: 1px solid #ccc;
   border-radius: 5px;
}
button {
   background-color: #333;
   color: #fff;
   padding: 10px 15px;
   border: none;
   border-radius: 5px;
   cursor: pointer;
   width: 100%;
}
button:hover {
   background-color: #555;
}
.link-buttons {
   display: flex;
   justify-content: space-between;
   margin-top: 20px;
}
.link-buttons a {
   text-decoration: none;
   color: white;
   padding: 10px 15px;
```

```
        background-color: #333;
        border-radius: 5px;
        text-align: center;
    }
    .link-buttons a:hover {
        background-color: #555;
    }

    /* Popup styles */
    .popup {
        display: none;
        position: fixed;
        left: 50%;
        top: 50%;
        transform: translate(-50%, -50%);
        background-color: rgba(0, 0, 0, 0.7);
        color: #fff;
        padding: 20px;
        border-radius: 10px;
        text-align: center;
    }
    .popup button {
        background-color: #4CAF50;
        border: none;
        padding: 10px 20px;
        color: white;
        border-radius: 5px;
        cursor: pointer;
        margin-top: 10px;
    }
    .popup button:hover {
        background-color: #45a049;
    }
</style>
```

```html
    <script>
        function login() {
            // Display the pop-up message
            document.getElementById('login-popup').style.display = 'block';

            // After a short delay, redirect to the User page
            setTimeout(function() {
                window.location.href = "User.html"; // Redirect to User.html after 2
seconds
            }, 2000);  // 2000ms = 2 seconds
        }
    </script>
</head>
<body>
    <div class="header">
        <h1>Traffic Prediction Management</h1>
    </div>
    <div class="container">
        <div class="form-group">
            <label for="username">Username:</label>
            <input type="text" id="username" placeholder="Enter username">
        </div>
        <div class="form-group">
            <label for="password">Password:</label>
            <input type="password" id="password" placeholder="Enter password">
        </div>
        <button onclick="login()">Login</button>
        <div class="link-buttons">
            <a href="settings.html">Settings</a>
        </div>
    </div>

    <!-- Pop-up message -->
    <div id="login-popup" class="popup">
```

```
        <p>Login Successful!</p>
        <button>Redirecting...</button>
    </div>
</body>
</html>
```

USER:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Traffic Prediction Management</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            margin: 0;
            padding: 0;
            background-color: #f4f4f4;
            display: flex;
            flex-direction: column;
            align-items: center;
            justify-content: center;
            height: 100vh;
        }
        .header {
            position: absolute;
            top: 0;
            width: 100%;
            background-color: #333;
            color: #fff;
            padding: 15px 20px;
            text-align: center;
        }
```

```
        .container {
          text-align: center;
        }
        button {
          background-color: #333;
          color: #fff;
          padding: 15px 20px;
          border: none;
          border-radius: 5px;
          cursor: pointer;
          width: 200px;
          margin: 10px;
          font-size: 16px;
        }
        button:hover {
          background-color: #555;
        }
    </style>
    <script>
        function goToLiveTraffic() {
          window.location.href = "live_traffic.html"; // Redirect to the Live Traffic
page
        }

        function goToLiveViolation() {
          window.location.href = "live_violation.html"; // Redirect to the Live
Violation page
        }
    </script>
</head>
<body>
    <div class="header">
        <h1>Traffic Prediction Management</h1>
    </div>
```

```
    <div class="container">
        <button onclick="goToLiveTraffic()">Live Traffic</button>
        <button onclick="goToLiveViolation()">Live Violation</button>
    </div>
</body>
</html>
```

LIVE TRAFFIC:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Live Traffic</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            margin: 0;
            padding: 0;
            background-color: #f4f4f4;
            display: flex;
            flex-direction: column;
            align-items: center;
            justify-content: center;
            height: 100vh;
        }
        button {
            background-color: #333;
            color: #fff;
            padding: 15px 20px;
            border: none;
            border-radius: 5px;
            cursor: pointer;
            font-size: 16px;
```

```
        }

        button:hover
 {
            background-color: #555;
        }
    </style>
    <script>
        function watchLiveTraffic() {
            alert("Redirecting to Live Traffic Stream...");
            // Replace with actual live traffic stream URL
            window.location.href = "live_traffic_stream.html";
        }
    </script>
</head>
<body>
    <button onclick="watchLiveTraffic()">Click to Watch Live Traffic</button>
</body>
</html>
```

LIVE VIOLATION:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Live Violation</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            margin: 0;
            padding: 0;
            background-color: #f4f4f4;
```

```
            display: flex;
            flex-direction: column;
            align-items: center;
            justify-content: center;
            height: 100vh;
        }
        button {
            background-color: #333;
            color: #fff;
            padding: 15px 20px;
            border: none;
            border-radius: 5px;
            cursor: pointer;
            font-size: 16px;
        }
        button:hover {
            background-color: #555;
        }
    </style>
    <script>
        function watchLiveViolation() {
            alert("Redirecting to Live Violation Stream...");
            // Replace with actual live violation stream URL
            window.location.href = "live_violation_stream.html";
        }
    </script>
</head>
<body>
    <button onclick="watchLiveViolation()">Click to Watch Live Violation</button>
</body>
</html>
```

SETTINGS:

```
<!DOCTYPE html>
```

```
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Settings</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            margin: 0;
            padding: 0;
            background-color: #f4f4f4;
        }
        .header {
            background-color: #333;
            color: #fff;
            padding: 15px 20px;
            text-align: center;
        }
        .container {
            max-width: 600px;
            margin: 50px auto;
            background: #fff;
            padding: 20px;
            box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);
        }
        .form-group {
            margin-bottom: 15px;
        }
        label {
            display: block;
            font-weight: bold;
            margin-bottom: 5px;
        }
        input[type="text"] {
```

```
            width: 100%;
            padding: 10px;
            border: 1px solid #ccc;
            border-radius: 5px;
        }
        button {
            background-color: #333;
            color: #fff;
            padding: 10px 15px;
            border: none;
            border-radius: 5px;
            cursor: pointer;
            width: 100%;
        }
        button:hover {
            background-color: #555;
        }
    </style>
    <script>
        function saveSettings() {
            alert("Settings Saved!"); // Display pop-up message
        }
    </script>
</head>
<body>
    <div class="header">
        <h1>Settings</h1>
    </div>
    <div class="container">
        <h3>Update Your Settings</h3>
        <div class="form-group">
            <label for="username">Update Username:</label>
            <input type="text" id="username" placeholder="New username">
        </div>
```

```
        <div class="form-group">
            <label for="email">Update Email:</label>
            <input type="text" id="email" placeholder="New email">
        </div>
        <button onclick="saveSettings()">Save Changes</button> <!-- Save button
triggers alert -->
    </div>
</body>
</html>
```

Figure.No.A.1.5 Settings

BACKEND:

```
# Import necessary libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.ensemble import RandomForestRegressor
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error

# Load the data
file_path = 'Traffic.csv'  # Update with the correct path if needed
traffic_data = pd.read_csv(file_path)

# Convert Time to a usable feature (hour of the day)
traffic_data['Hour'] = pd.to_datetime(traffic_data['Time'], format='%I:%M:%S %p').dt.hour

# Part 1: Traffic Prediction
X = traffic_data[['Hour', 'CarCount', 'BikeCount', 'BusCount', 'TruckCount']]
y = traffic_data['Total']

# Train-test split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Train a Random Forest Regressor
rf_model = RandomForestRegressor(random_state=42)
rf_model.fit(X_train, y_train)
```

```
# Predict on test data
traffic_data['Predicted_Total'] = rf_model.predict(X)

# Visualize Predictions
plt.figure(figsize=(14, 7))
plt.plot(traffic_data['Time'][:500], traffic_data['Total'][:500], label='Actual Traffic',
color='blue', marker='o', linewidth=1)
plt.plot(traffic_data['Time'][:500], traffic_data['Predicted_Total'][:500], label='Predicted
Traffic', color='orange', linestyle='--', linewidth=1)
plt.xticks(rotation=90, fontsize=8)
plt.title('Traffic Prediction vs Actual', fontsize=16)
plt.xlabel('Time (First 500 Observations)', fontsize=14)
plt.ylabel('Number of Vehicles', fontsize=14)
plt.legend(fontsize=12)
plt.grid(axis='y', linestyle='--', linewidth=0.7)
plt.tight_layout()
plt.show()

# Print model evaluation
mse = mean_squared_error(y_test, rf_model.predict(X_test))
print(f"Model Performance: Mean Squared Error = {mse:.2f} vehicles.")

# Part 2: Traffic Violation Analysis
traffic_situation_counts = traffic_data['Traffic Situation'].value_counts()

# Pie chart for Traffic Situations
plt.figure(figsize=(8, 8))
traffic_situation_counts.plot.pie(
    autopct='%1.1f%%',
    startangle=90,
    colors=sns.color_palette('pastel'),
    wedgeprops={'linewidth': 1, 'edgecolor': 'black'}
)
plt.title('Distribution of Traffic Situations', fontsize=16)
plt.ylabel('')
plt.show()

# Bar chart for High Traffic Days
traffic_data['High Traffic'] = traffic_data['Total'] > traffic_data['Total'].mean()
high_traffic_days = traffic_data.groupby('Day of the week')['High Traffic'].sum()
```

```
plt.figure(figsize=(10, 6))
high_traffic_days.plot(kind='bar', color='teal', edgecolor='black')
plt.title('High Traffic Days (Total Above Average)', fontsize=16)
plt.xlabel('Day of the Week', fontsize=14)
plt.ylabel('Number of High Traffic Periods', fontsize=14)
plt.xticks(rotation=45)
plt.grid(axis='y', linestyle='--', linewidth=0.7)
plt.tight_layout()
plt.show()

# Correlation Heatmap for Vehicle Counts
correlation_matrix = traffic_data[['CarCount', 'BikeCount', 'BusCount', 'TruckCount',
'Total']].corr()

plt.figure(figsize=(10, 7))
sns.heatmap(
    correlation_matrix,
    annot=True,
    cmap='Blues',
    fmt='.2f',
    linewidths=1,
    linecolor='black'
)
plt.title('How Different Vehicle Counts Relate to Total Traffic', fontsize=16)
plt.show()

# Part 3: 14 Plots for All Seven Days

# Chart 1 and Chart 2 for each day
unique_days = traffic_data['Day of the week'].unique()

for day in unique_days:
    # Filter data for the current day
    day_data = traffic_data[traffic_data['Day of the week'] == day]

    # Chart 1: Vehicle counts by hour
    hourly_vehicle_counts = day_data.groupby('Hour')[['CarCount', 'BikeCount', 'BusCount',
'TruckCount']].sum()
    plt.figure(figsize=(12, 6))
```

```
    hourly_vehicle_counts.plot(kind='bar', stacked=True, figsize=(12, 6), color=['blue', 
'green', 'orange', 'red'], edgecolor='black')
    plt.title(f"Vehicle Counts by Hour: {day}", fontsize=16)
    plt.xlabel("Hour of the Day", fontsize=14)
    plt.ylabel("Number of Vehicles", fontsize=14)
    plt.xticks(rotation=0)
    plt.legend(title="Vehicle Type", fontsize=12)
    plt.grid(axis='y', linestyle='--', linewidth=0.7)
    plt.tight_layout()
    plt.show()

    # Chart 2: Average traffic per hour
    hourly_avg_traffic = day_data.groupby('Hour')['Total'].mean()
    plt.figure(figsize=(12, 6))
    hourly_avg_traffic.plot(kind='line', marker='o', linestyle='-', color='purple', linewidth=2)
    plt.title(f"Average Traffic per Hour: {day}", fontsize=16)
    plt.xlabel("Hour of the Day", fontsize=14)
    plt.ylabel("Average Total Traffic", fontsize=14)
    plt.grid(axis='both', linestyle='--', linewidth=0.7)
    plt.tight_layout()
    plt.show()
# Chart 1: Vehicle counts by hour for each day
unique_days = traffic_data['Day of the week'].unique()

for day in unique_days:
    day_data = traffic_data[traffic_data['Day of the week'] == day]
    hourly_vehicle_counts = day_data.groupby('Hour')[['CarCount', 'BikeCount', 'BusCount', 
'TruckCount']].sum()

    plt.figure(figsize=(12, 6))
    hourly_vehicle_counts.plot(kind='bar', stacked=True, figsize=(12, 6), color=['blue', 
'green', 'orange', 'red'], edgecolor='black')
    plt.title(f"Vehicle Counts by Hour: {day}", fontsize=16)
    plt.xlabel("Hour of the Day", fontsize=14)
    plt.ylabel("Number of Vehicles", fontsize=14)
    plt.xticks(rotation=0)
    plt.legend(title="Vehicle Type", fontsize=12)
    plt.grid(axis='y', linestyle='--', linewidth=0.7)
    plt.tight_layout()
    plt.show()
```

```
# Chart 2: Average traffic per hour for each day
for day in unique_days:
    day_data = traffic_data[traffic_data['Day of the week'] == day]
    hourly_avg_traffic = day_data.groupby('Hour')['Total'].mean()

    plt.figure(figsize=(12, 6))
    hourly_avg_traffic.plot(kind='line', marker='o', linestyle='-', color='purple', linewidth=2)
    plt.title(f"Average Traffic per Hour: {day}", fontsize=16)
    plt.xlabel("Hour of the Day", fontsize=14)
    plt.ylabel("Average Total Traffic", fontsize=14)
    plt.grid(axis='both', linestyle='--', linewidth=0.7)
    plt.tight_layout()
    plt.show()
```

APPENDIX 2

SCREEN SHOTS

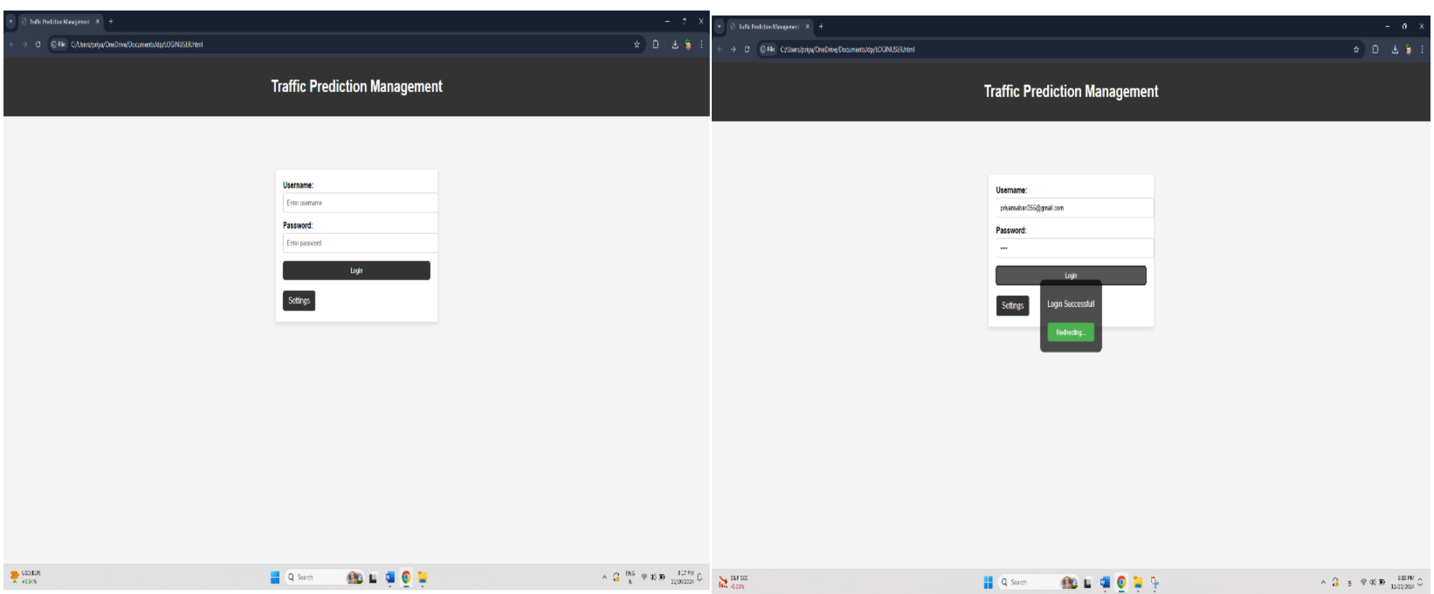


Fig.No.A.1.1 Login

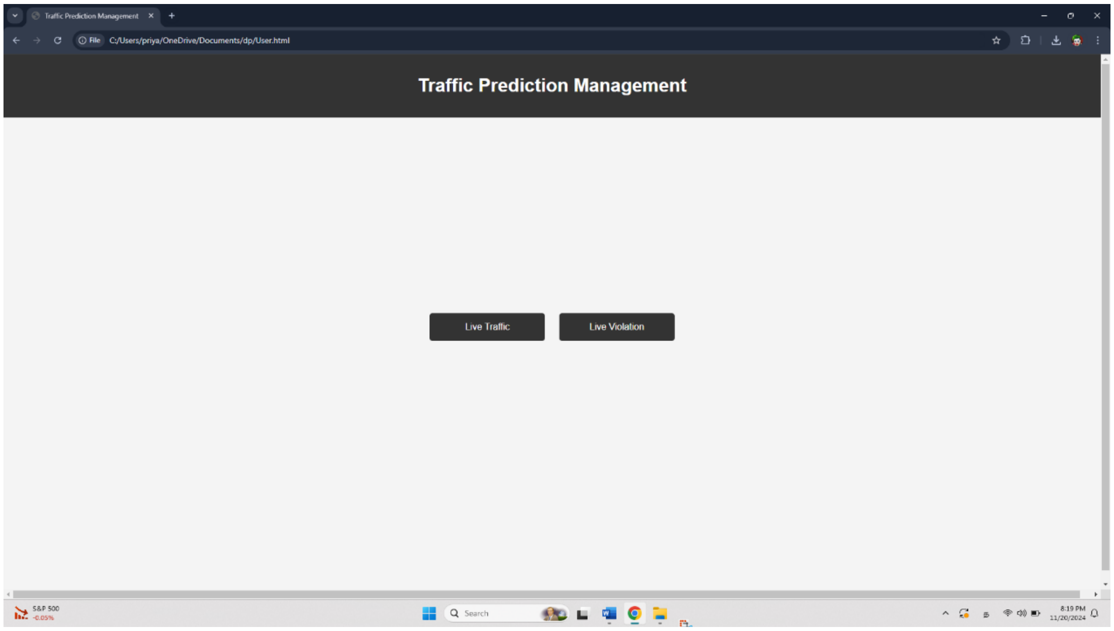


Fig.No.A.1.2 User

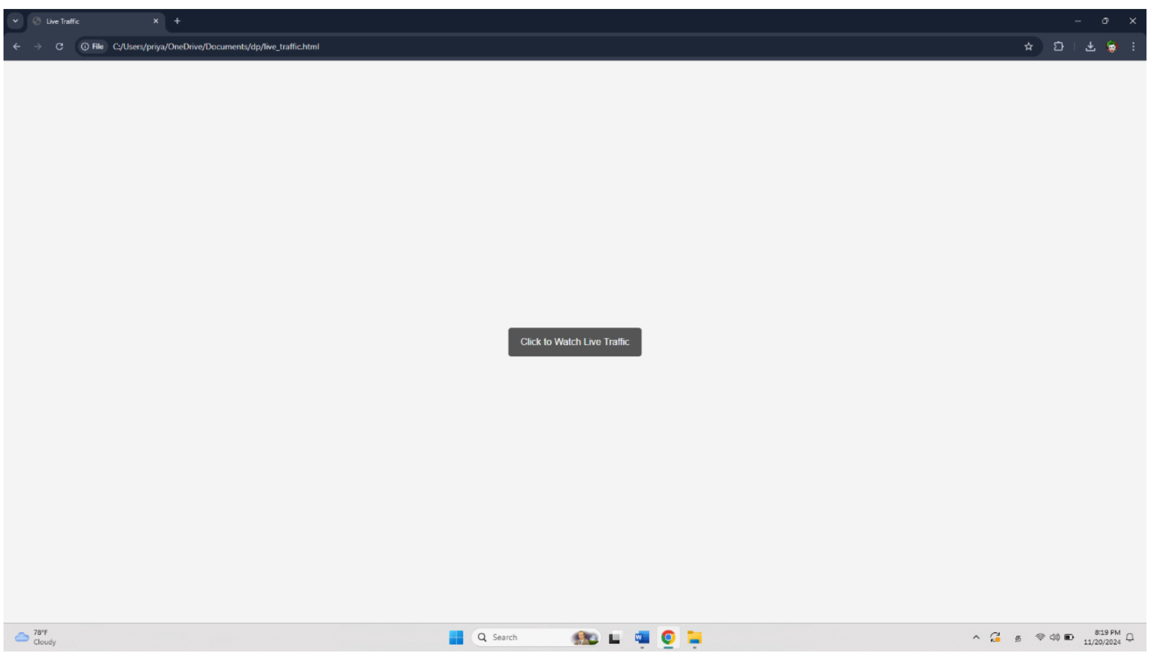


Fig.No.A.1.3 Live Traffic

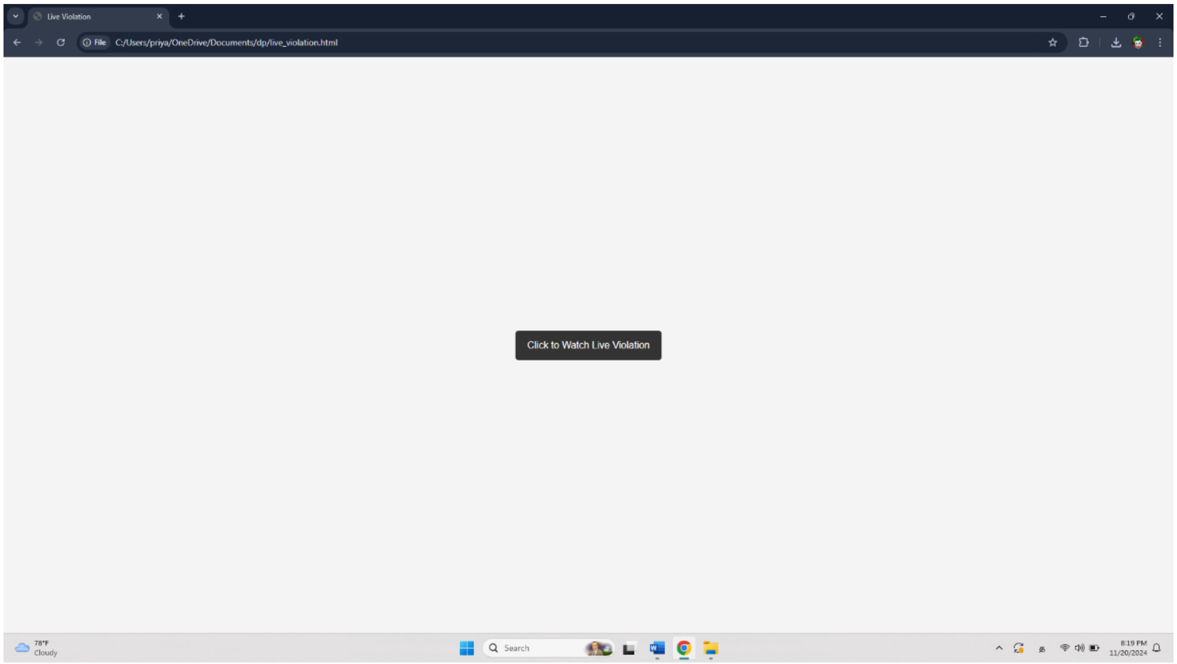


Fig.No.A.1.4 Live Violation

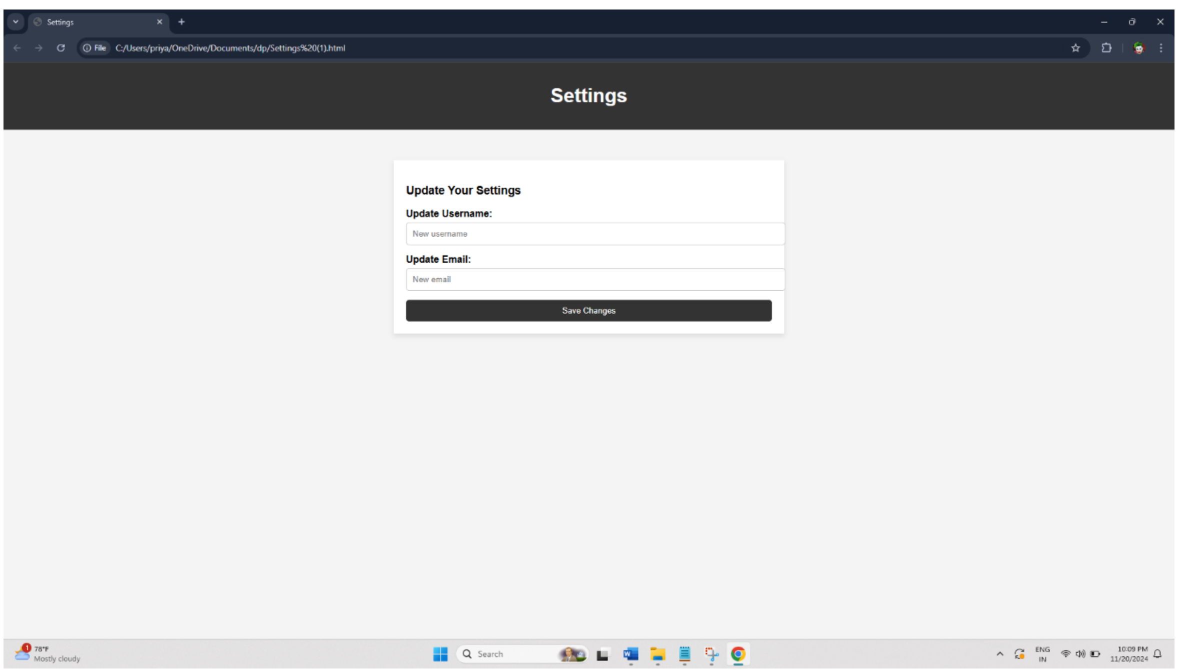


Fig.No.A.1.5 Settings

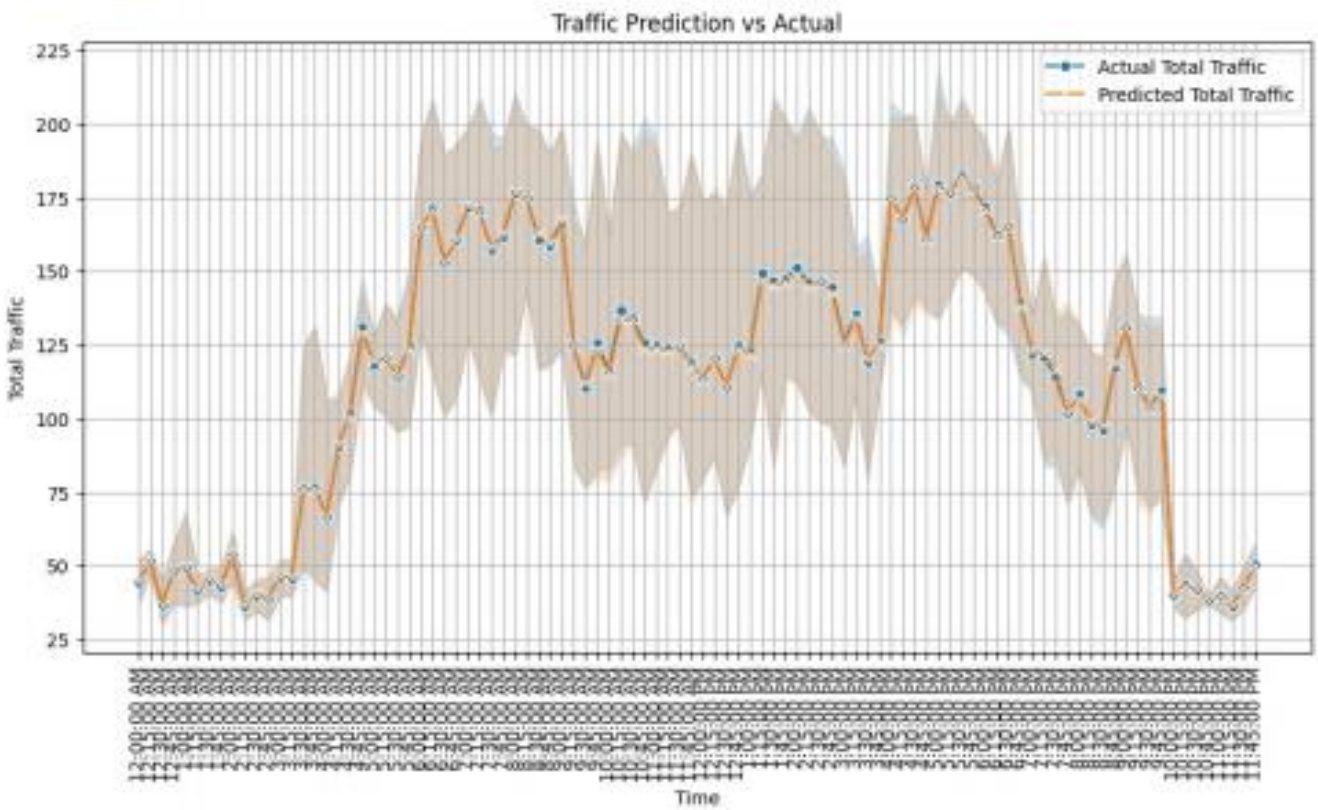
```python
sns.countplot(data=traffic_data, x='Traffic Situation', order=traffic_data['Traffic Situation'].value_counts().i
plt.title('Traffic Situation Distribution')
plt.xlabel('Traffic Situation')
plt.ylabel('Count')
plt.grid(True, axis='y')
plt.show()

# Analyze High-Traffic Hours
traffic_data['High Traffic'] = traffic_data['Total'] > traffic_data['Total'].mean()

plt.figure(figsize=(12, 6))
sns.boxplot(data=traffic_data, x='Day of the week', y='Total', hue='High Traffic')
plt.title('Traffic Levels by Day of the Week')
plt.xlabel('Day of the Week')
plt.ylabel('Total Traffic')
plt.legend(title='High Traffic')
plt.grid(True, axis='y')
plt.show()

# Correlation Heatmap for Vehicle Counts
correlation_matrix = traffic_data[['CarCount', 'BikeCount', 'BusCount', 'TruckCount', 'Total']].corr()

plt.figure(figsize=(8, 6))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt='.2f')
plt.title('Correlation Matrix for Vehicle Counts')
plt.show()
```




```
Mean Squared Error on Test Data: 12.63
```

Fig.No.A.2.1 Prediction vs Actual

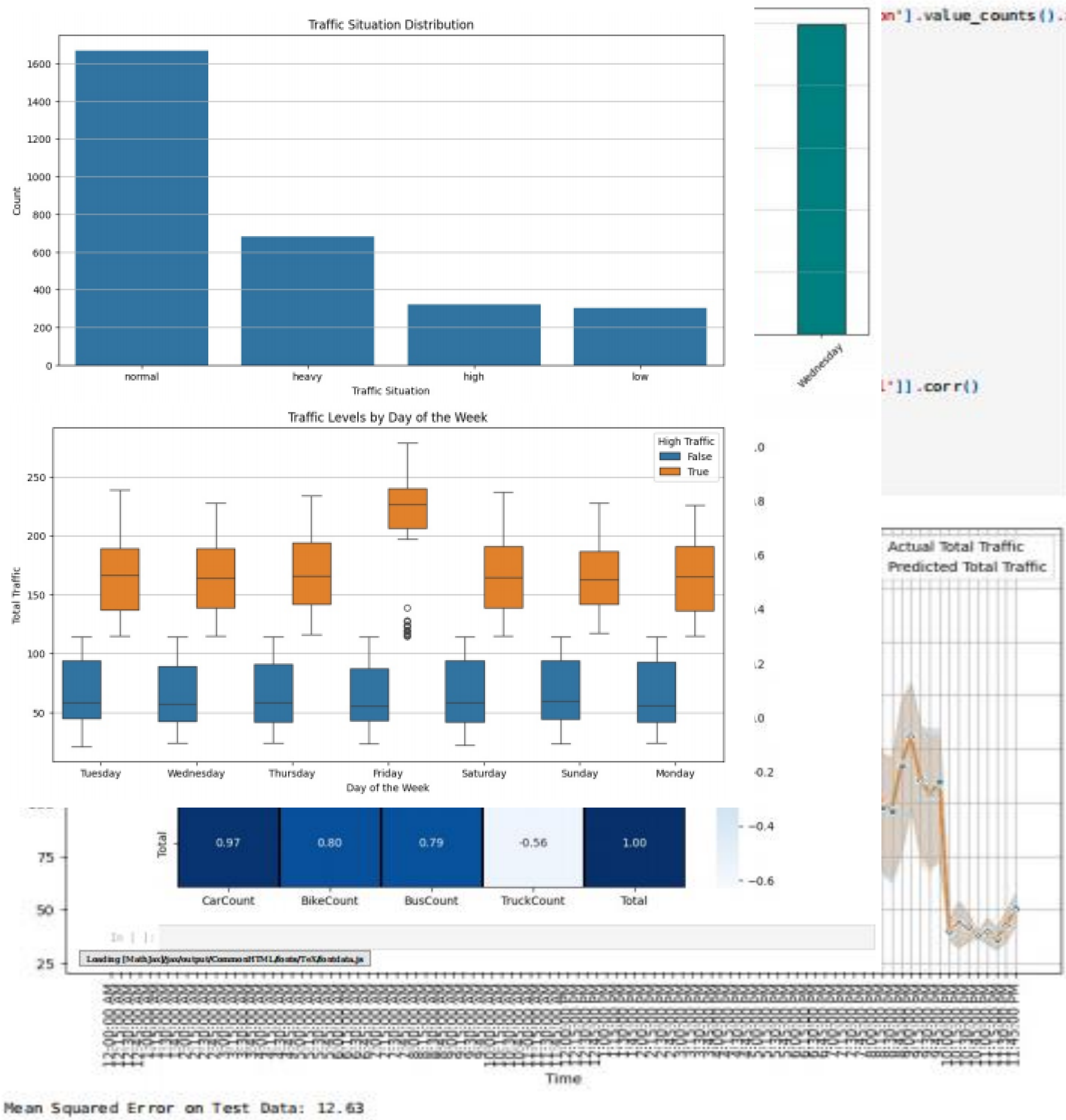


Fig.No.A.2.2 Distribution of Data

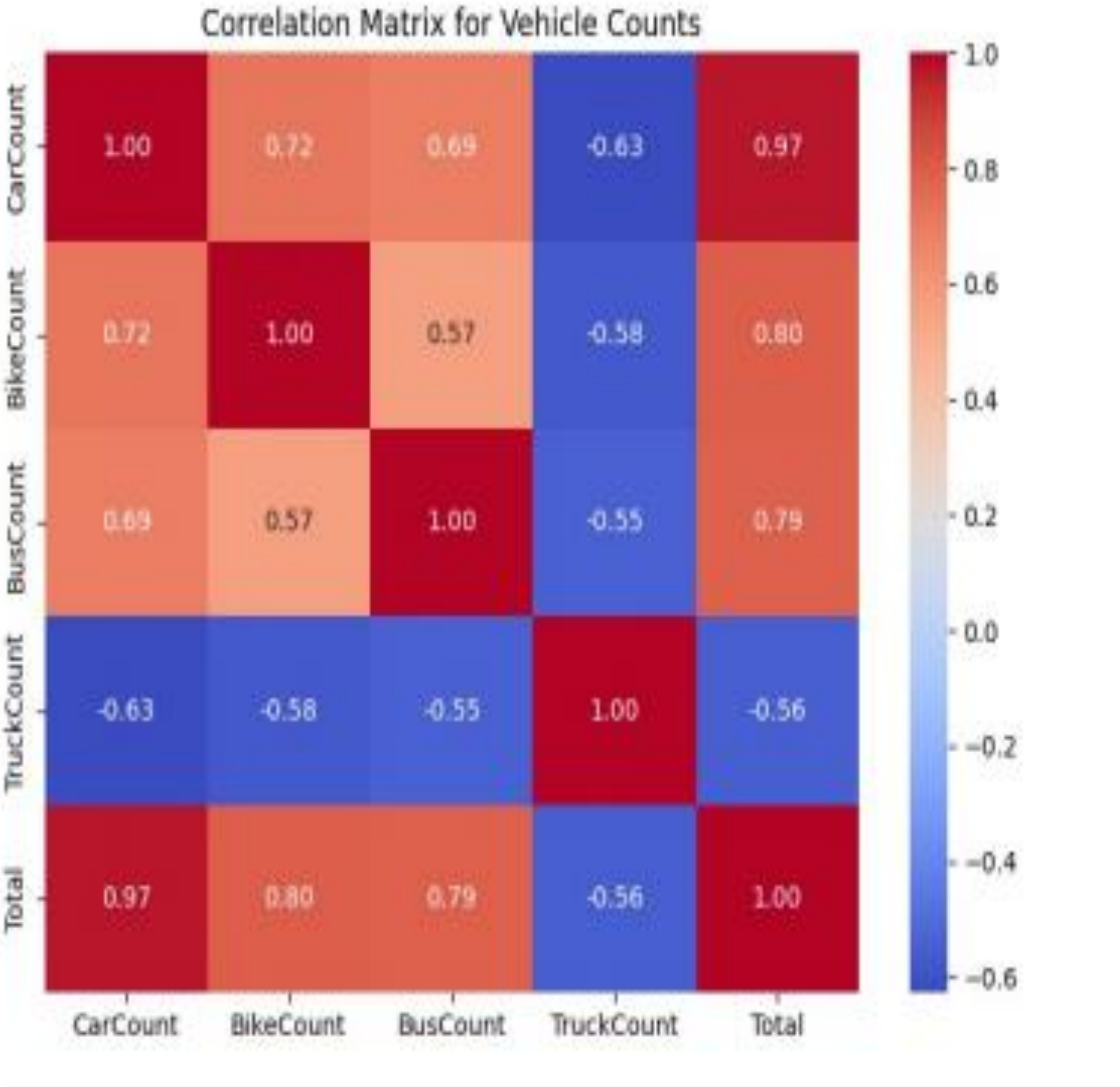
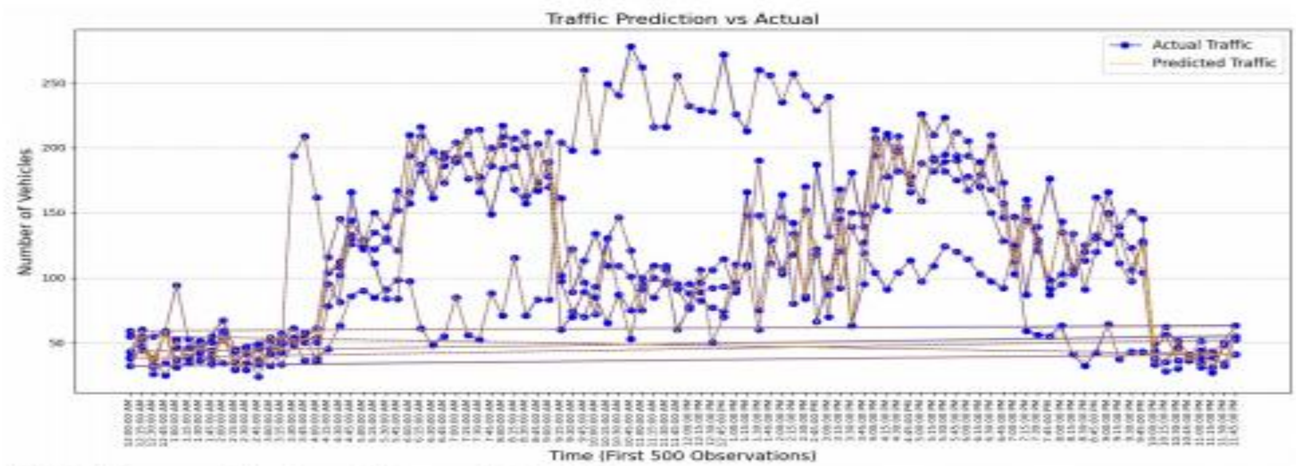


Fig.No.A.2.3 Correlations for Vehicle

```python
# 3. Correlation Heatmap for Vehicle Counts with Plain Language
correlation_matrix = traffic_data[['CarCount', 'BikeCount', 'BusCount', 'TruckCount', 'Total']].corr()

plt.figure(figsize=(10, 7))
sns.heatmap(
    correlation_matrix,
    annot=True,
    cmap="Blues",
    fmt=".2f",
    linewidths=1,
    linecolor='black'
)
plt.title('How Different Vehicle Counts Relate to Total Traffic', fontsize=16)
plt.show()
```




```
Model Performance: Mean Squared Error = 12.63 vehicles.
```

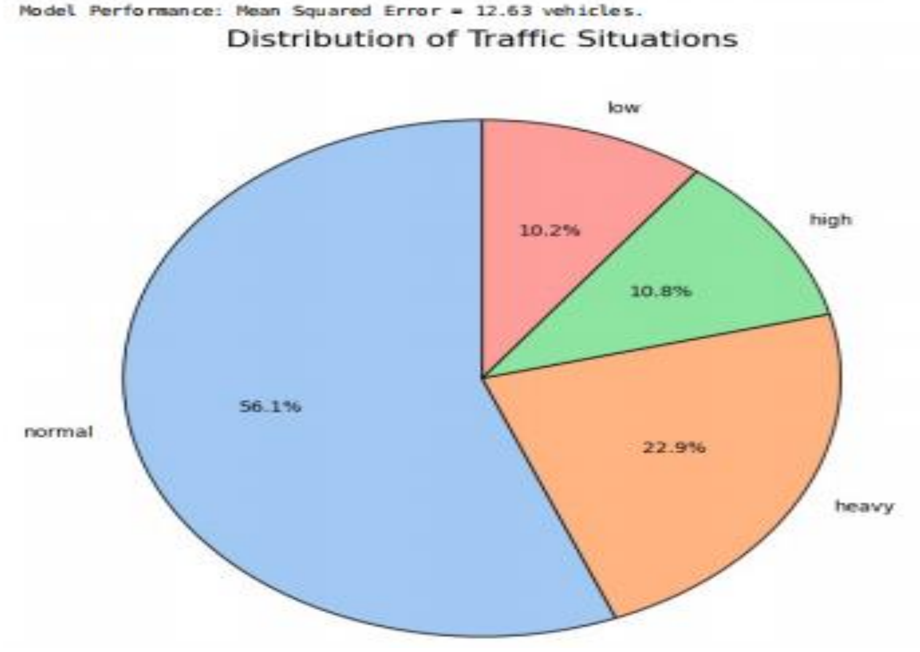


Fig.No.A.2.4 Distribution of Traffic

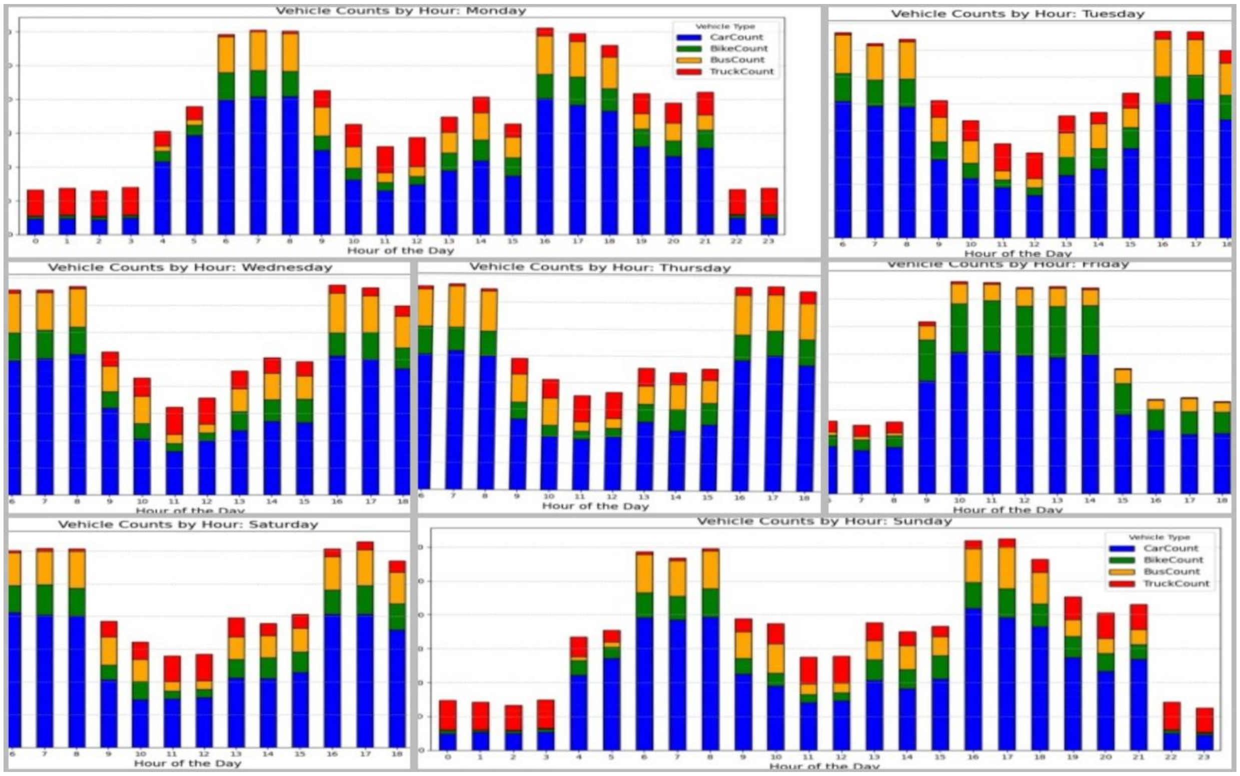


Figure.No.A.2.5 Vehicle Count per Hour

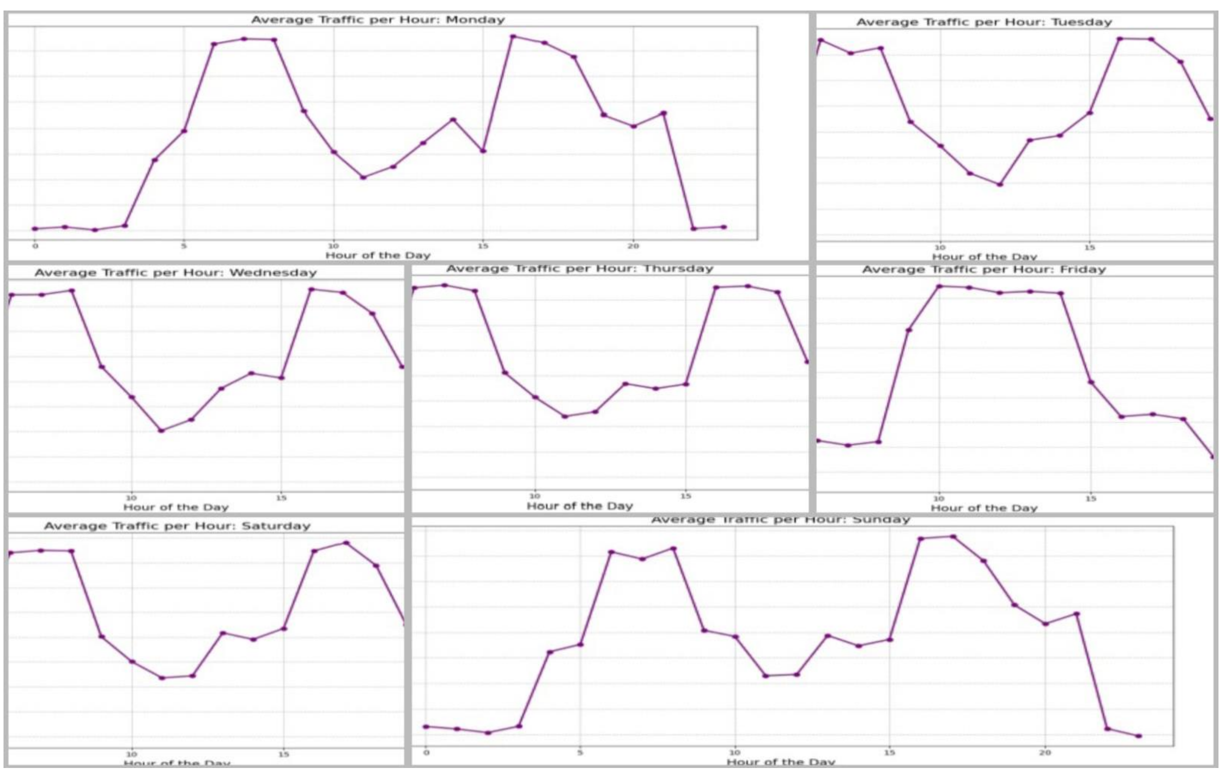


Fig.No.A.2.6 Average Traffic per Hour

REFERENCES

1. Vlahogianni, E. I., M. G. Karlaftis, and J. C. Golias. Optimized and MetaOptimized Neural Networks for Short Term Traffic Flow Prediction: A Genetic Approach. Transportation Research Part C: Emerging Technologies, Vol. 13, No. 3, 2005, pp. 211–234.
2. Machine Learning Approach to Short-Term Traffic Congestion Prediction in a Connected Environment AmrElfar1, Alireza Talebpour2, and Hani S. Mahmassani1, National Academy of Sciences: Transportation Research Board 2018.
3. Big data-driven machine learning-enabled traffic flow prediction Fanhui Kong1 Jian Li1 Bin Jiang2 Tianyuan Zhang3 Houbing Song3, 2018.
4. Bao G, Zeng Z, Shen Y. Region stability analysis and tracking control of memristive recurrent neural network. Neural Netw. 2017;5(1):74-89.
5. Jiang, X., and H. Adeli.Dynamic Wavelet Neural Network Model for Traffic Flow Forecasting. Journal of Transportation Engineering, Vol. 131, No. 10, 2005, pp. 771–779
6. Identification Traffic Flow Prediction Parameters AnuchitRatanaparadorn Department ofIndustrialEngineering,KasetsartUniversity,Thailand,AnuchitRatanaparadorn,S asivim olMeeampol,ThaneeratSiripachana,Pornthep,Anussornnitisarn, 19-21 2013, zadar, Croatia, international conference
7. YuhanJia, Jianping Wu, and Ming Xu, Traffic Flow Prediction with Rainfall Impact Using a Deep Learning Method,Journal of Advanced Transportation, 2017.
8. Felix Kunde Alexander Hartenstein Stephan Pieper Petra Sauer, Traffic prediction using a Deep Learning paradigm, CEUR-WS.org, 2017