# ADVANCED TRAFFIC VOLUME ESTIMATION

## AN INTERNSHIP PROJECT REPORT

SMARTBRIDGE

**Department of Computer Science and Engineering**

**GAYATRI VIDYA PARISHAD COLLEGE OF ENGINEERING FOR WOMEN**


**INTERNSHIP DURATION** : January 2025 - April 2025

**Submitted By**

Doddi Likhitha

**Submitted To**

Smart Bridge

Hyderabad

# TABLE OF CONTENTS

# ABSTRACT

Efficient traffic management is essential for keeping cities moving smoothly, but traditional methods often struggle with delays and inaccuracies. This can lead to frustrating congestion, inefficient road networks, and unreliable navigation for commuters.

*TrafficTelligence* is designed to change that by providing real-time and predictive insights into traffic patterns. With smarter data analytics and machine learning, it helps authorities optimize traffic signals, adjust lane configurations, and manage congestion dynamically. City planners can also use these insights to design better roads, transit systems, and urban spaces that accommodate future traffic growth more effectively.

For daily commuters, TrafficTelligence enhances navigation by offering up-to-the-minute traffic volume estimates, ensuring more efficient route planning and travel decisions. Unlike conventional systems that rely on outdated data, this technology delivers live updates, helping users avoid traffic bottlenecks and save time.

By integrating cutting-edge technology with real-world applications, TrafficTelligence transforms urban mobility, making cities smarter, roads less congested, and commutes more seamless.



## Key words

Real-time traffic monitoring , Traffic volume estimation , Smart urban planning , Machine learning in transportation , Predictive traffic analytics , Adaptive traffic control , Intelligent transportation systems , Commuter navigation optimization , Data-driven mobility solutions , Traffic flow management.

# 1. INTRODUCTION

Imagine rushing to work on a Monday morning, only to find yourself stuck in an unexpected traffic jam. The frustration builds as minutes pass, and you wonder why no navigation app warned you in advance. Cities worldwide face this daily challenge—unpredictable traffic congestion that slows down commuters, disrupts businesses, and strains urban infrastructure. Traditional traffic monitoring methods, which rely on fixed sensors, manual counting, or outdated data, often fail to provide real-time insights, leading to inefficiencies in traffic management and urban planning.

Now, imagine a city where traffic lights adjust dynamically based on real-time congestion levels, where city planners can anticipate and prepare for future traffic demands, and where commuters receive instant, accurate route recommendations to avoid unnecessary delays. This is the vision behind *TrafficTelligence*—a next-generation traffic volume estimation solution that combines cutting-edge data analytics, machine learning, and real-time monitoring. By providing precise, up-to-the-minute insights, TrafficTelligence empowers transportation authorities, city planners, and daily commuters with the information they need to navigate urban landscapes efficiently. With this smart approach to traffic management, we can create smoother commutes, less congested roads, and more sustainable cities—transforming the way we move every day.

## 1.1 Problem Statement

Efficient traffic management, urban planning, and commuter navigation depend on accurate traffic volume estimations. However, traditional methods often fall short due to outdated data collection techniques, lack of real-time monitoring, and limited predictive capabilities. This results in congestion, inefficient infrastructure development, and suboptimal route guidance for commuters.

**1. Dynamic Traffic Management:** Traffic authorities need real-time, high-precision data to optimize signal timings, adjust lane configurations, and reduce congestion. Without accurate and timely insights, road networks become inefficient, leading to delays and increased fuel consumption.

**2. Urban Development Planning:** City planners must design infrastructure that accommodates future traffic patterns, but traditional forecasting methods struggle to capture evolving trends. Poorly planned road networks and public transit systems lead to overcrowding and inefficient land use.

**3. Commuter Navigation:** Navigation systems often rely on historical data rather than real-time insights, resulting in suboptimal route recommendations. Commuters lack accurate, up-to-the-minute information that could help them avoid traffic congestion and reduce travel time.

TrafficTelligence addresses these challenges by leveraging advanced analytics and machine learning to provide real-time, predictive traffic volume insights. By enabling smarter decision-making across traffic management, urban planning, and navigation, TrafficTelligence paves the way for a more efficient and sustainable urban future.

# 2. Related Work

Several studies have explored real-time traffic estimation and predictive modeling using various techniques, including sensor-based monitoring, GPS data analysis, and machine learning algorithms. Traditional methods such as inductive loop detectors and camera-based monitoring systems provide valuable insights but often face limitations related to cost, maintenance, and data accuracy. Recent advancements in artificial intelligence and big data analytics have improved predictive capabilities, allowing for more dynamic and efficient traffic management. Research has demonstrated that integrating real-time data from multiple sources, including IoT sensors, GPS, and crowdsourced mobile data, enhances accuracy in traffic volume predictions. Solutions like Google Maps and Waze utilize user-generated data for congestion tracking, but they primarily depend on historical trends rather than real-time predictive analytics, which TrafficTelligence aims to address.

# 3. System Analysis

## 3.1 Software Requirement Specification

TrafficTelligence is designed to deliver real-time, predictive traffic volume estimations through a combination of data analytics and machine learning. The system will collect and process live traffic data from IoT sensors, GPS devices, and public data sources to generate insights for transportation authorities, urban planners, and commuters. The software must support data ingestion from multiple sources, implement predictive modeling using machine learning techniques, and provide an intuitive interface for real-time visualization. Additionally, the system should feature cloud-based storage for scalability, API integration for third-party applications, and a robust backend to ensure high-performance data processing.

## 3.2 System requirements

### 3.2.1 Minimum Software requirements

-Operating System: Windows 10 or later, macOS, or Linux

- Programming Languages: Python, Java, or C++

- Frameworks and Libraries: TensorFlow/PyTorch (for machine learning), Apache Kafka (for real-time data streaming), OpenCV (for image processing, if applicable)

- Database: PostgreSQL/MySQL for structured data, MongoDB for unstructured data

- Web Technologies: HTML, CSS, JavaScript, React or Angular (for front-end development)

- API Support: RESTful API for third-party integration

### 3.2.2 Minimum Hardware requirements

- Processor: Intel Core i5 or AMD Ryzen 5 (or higher)

- RAM: 8GB (minimum), 16GB (recommended for machine learning tasks)

- Storage: SSD with at least 256GB of free space (recommended for handling large datasets)

- GPU: NVIDIA GTX 1050 or equivalent (for machine learning and real-time image processing)

- Network: Stable internet connection with high bandwidth for real-time data transfer

TrafficTelligence is designed to work efficiently with these specifications, ensuring seamless traffic data processing, real-time analytics, and predictive insights to enhance traffic management, urban planning, and commuter navigation.

# 4. System Design

## 4.1 Introduction

TrafficTelligence is a cutting-edge system designed to enhance traffic volume estimation, addressing the inefficiencies of traditional monitoring methods. It integrates real-time data analytics, machine learning, and predictive modeling to optimize traffic management, urban planning, and commuter navigation. By continuously analyzing live traffic feeds from sensors, cameras, and GPS data, the system provides dynamic traffic insights to transportation authorities, enabling adaptive traffic control, optimized signal timings, and efficient lane management. This reduces congestion and enhances road efficiency. For urban planners, TrafficTelligence forecasts future traffic patterns, aiding in infrastructure development, public transit planning, and sustainable city design. Unlike traditional methods that rely on outdated or static data, this system dynamically adapts to evolving trends, ensuring roads and transit networks remain efficient and future-proof. Commuters benefit from accurate, real-time navigation assistance, helping them make informed travel decisions and avoid congested routes. By leveraging AI-driven predictions, the system enhances route optimization for navigation apps, improving travel efficiency. TrafficTelligence thus transforms urban mobility by providing precise, actionable insights, ensuring seamless traffic flow, sustainable urban growth, and improved commuter experiences. This intelligent, data-driven approach enables smarter cities, reducing delays and enhancing overall transportation efficiency in rapidly evolving urban environments.

## 4.2 UMLdiagrams

### 4.2.1 Use case diagram

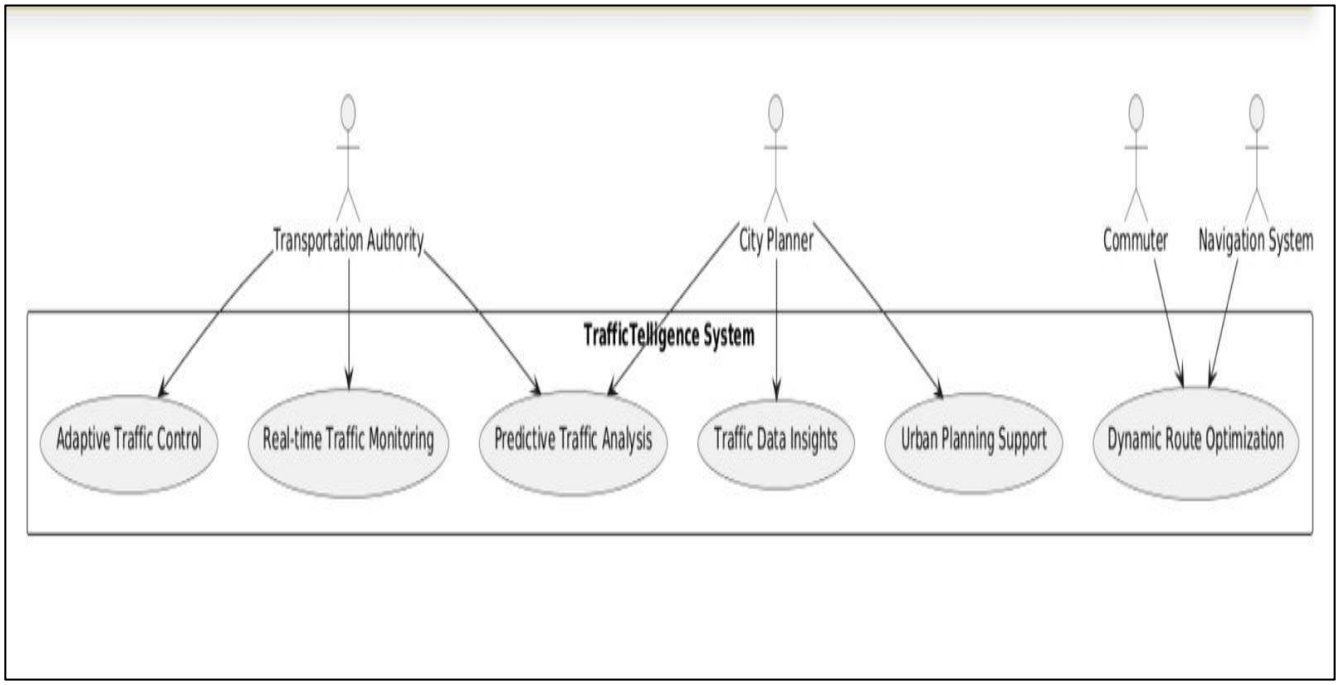


**Fig : Use case Diagram**

### 4.2.2 Class diagram

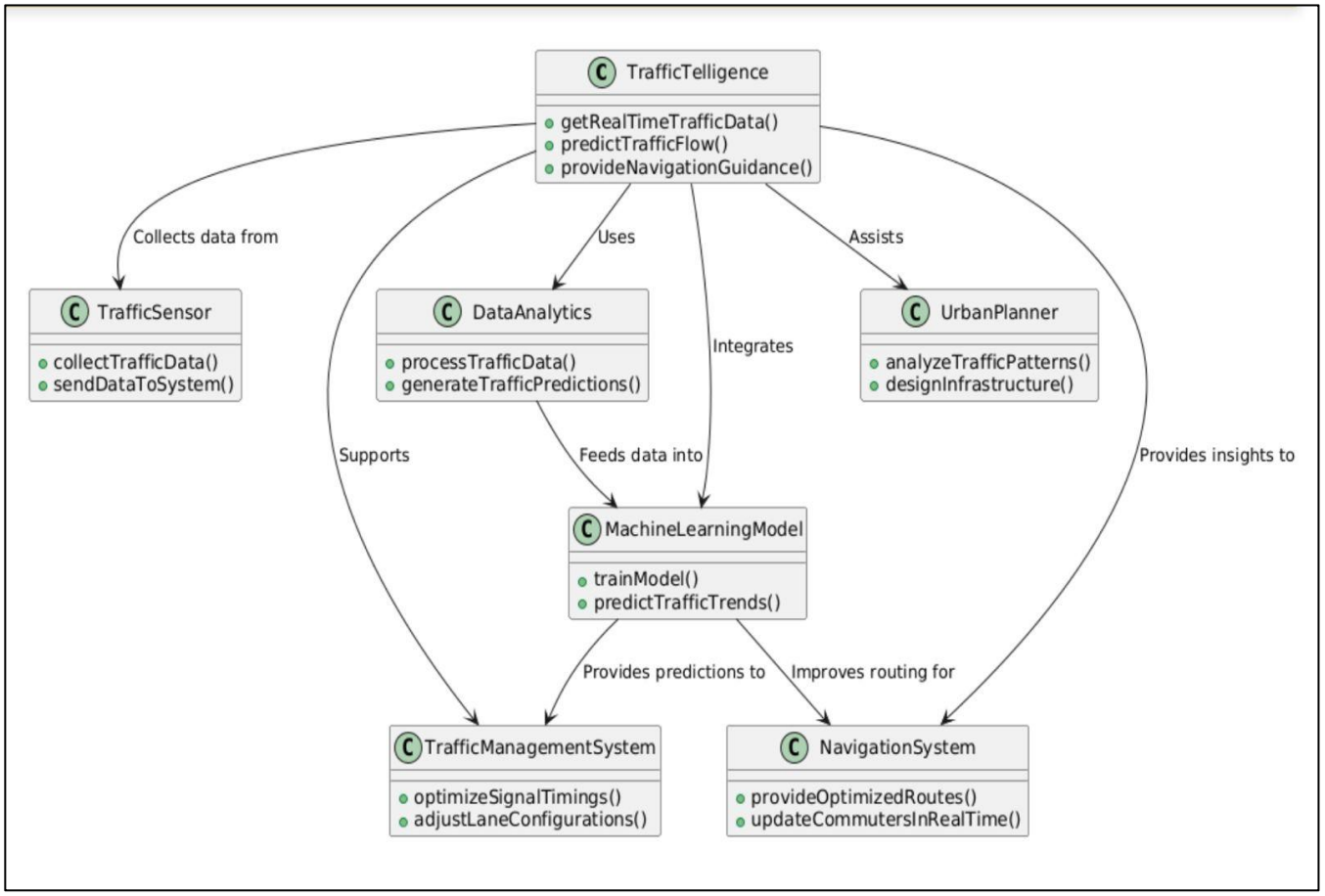


**Fig : Class Diagram**

### 4.2.3 Sequence diagram

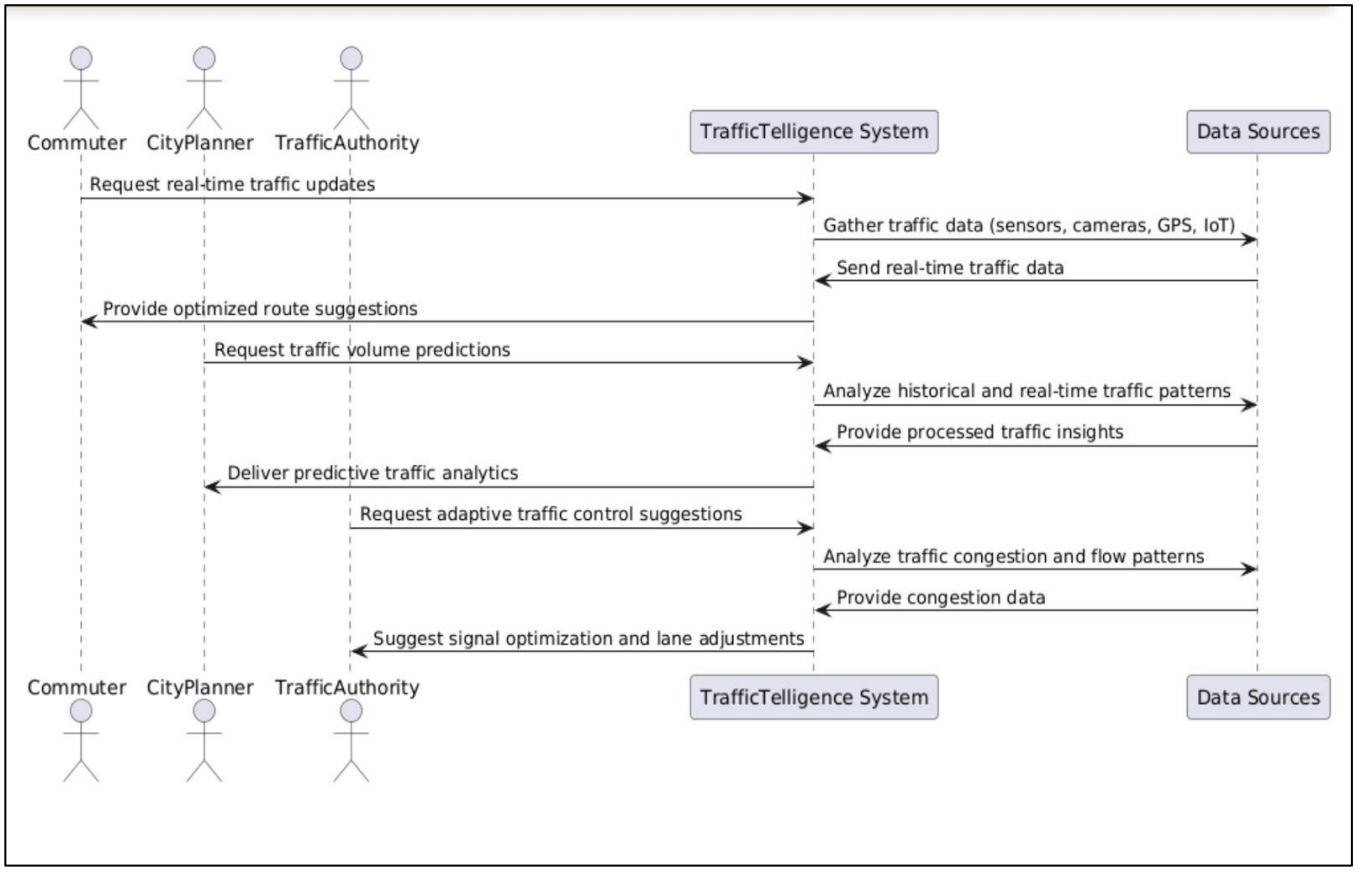


**Fig : Sequence Diagram**

### 4.2.4 Activity diagram

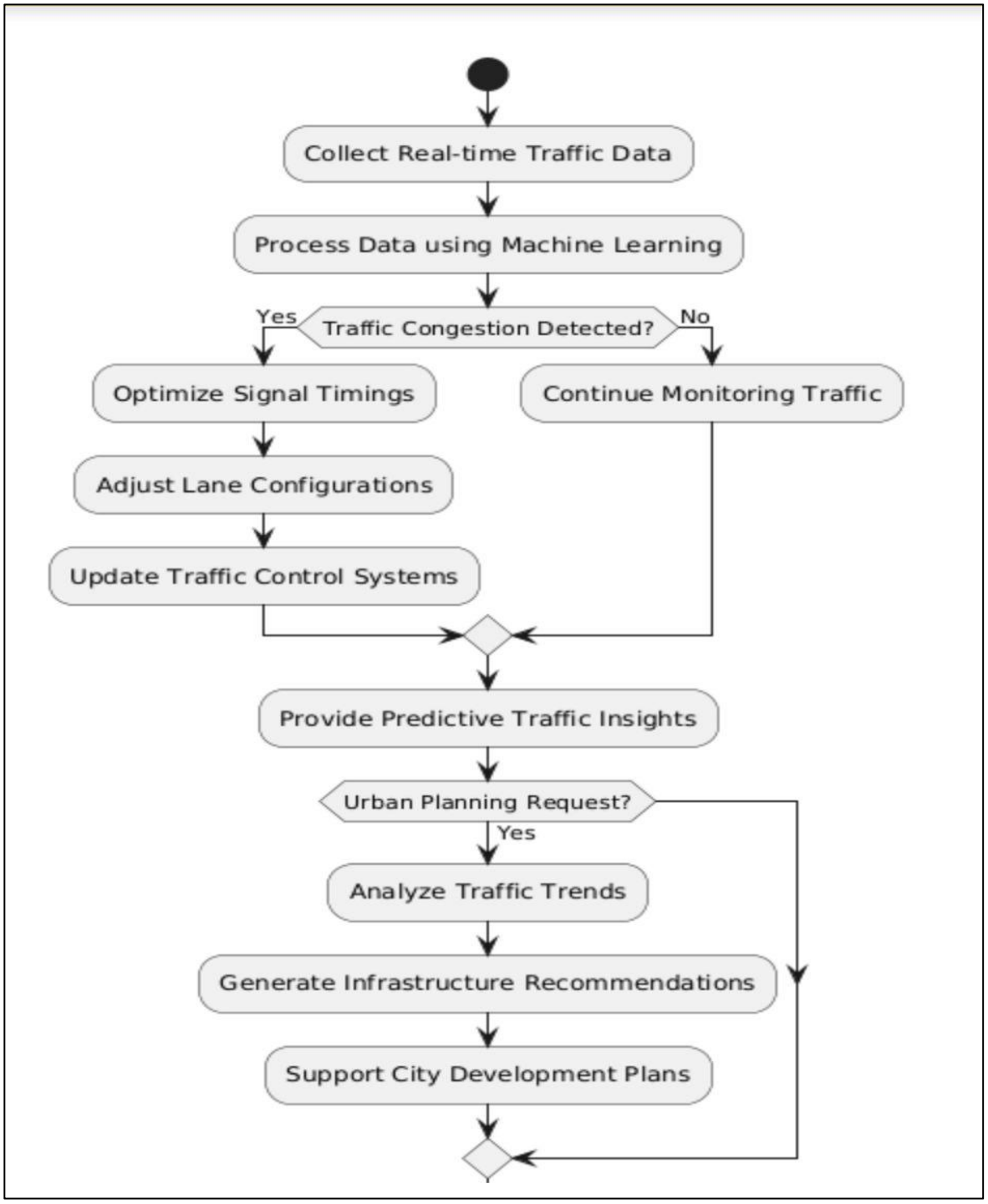


**Fig : Activity Diagram**

# 5. Methodology

## 5.1 ARCHITECTURE DIAGRAM / WORK FLOW DIAGRAM

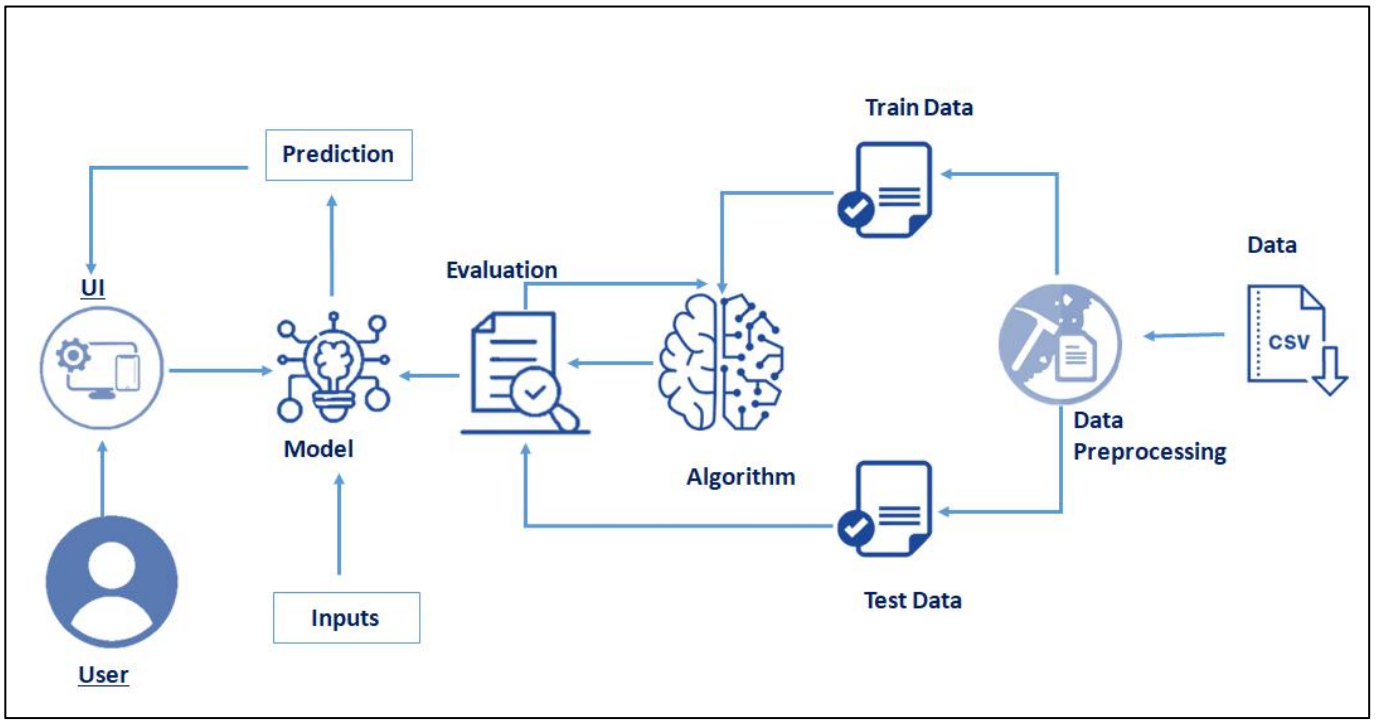


**Fig : Architecture Digram**

To develop an advanced traffic volume estimation system, a structured approach was followed, starting from data collection to model deployment. The system utilizes machine learning algorithms to analyze traffic patterns based on historical data. The methodology involves data preprocessing, model training, evaluation, and integration with a user-friendly interface for real-time predictions.

- **Data Collection**
  - The dataset used for traffic volume estimation is sourced in CSV format.
  - The data includes various features such as timestamp, vehicle count, weather conditions, and other relevant parameters.
- **Data Preprocessing**
  - The collected raw data undergoes preprocessing to remove inconsistencies and missing values.
  - Data normalization and transformation techniques are applied to ensure optimal performance.
  - The dataset is split into **train data** and **test data** for model development.
- **Algorithm Selection & Model Training**
  - A machine learning model is selected for traffic volume prediction, incorporating techniques like regression or deep learning.
  - The training dataset is used to train the model using selected algorithms.

- Hyperparameter tuning is performed to optimize the model performance.

- **Model Evaluation**

  - The trained model is tested using the **test dataset** to measure its accuracy and efficiency.
  - Various evaluation metrics such as Mean Absolute Error (MAE), Mean Squared Error (MSE), and R-squared score are calculated to assess model performance.

- **User Interface (UI) Development**

  - A user-friendly web-based UI is developed to allow users to input data and get real-time predictions.
  - The UI interacts with the trained model to display predictions dynamically.

- **Prediction Generation**

  - The trained model takes input from the UI, processes the data, and provides predictions on traffic volume.
  - The predictions are displayed on the UI for user interpretation.

- **Final Integration & Testing**

  - The frontend and backend components are integrated to ensure seamless interaction between users and the machine learning model.
  - The complete system is tested for real-world performance and optimized based on user feedback.

## 5.2 MODULES ,DESCRIPTIONS

### 1. Data Collection

- **Description:** This module is responsible for gathering traffic data from various sources, including government traffic databases, IoT sensors, and surveillance cameras. The collected data is stored in a structured format (CSV) containing details such as vehicle count, timestamps, weather conditions, and road type. This serves as the foundation for building an accurate traffic volume prediction model.

### 2. Data Preprocessing

- **Description:** The raw collected data often contains inconsistencies, missing values, and noise. This module cleans and prepares the data by handling missing values, removing outliers, normalizing numerical data, and converting categorical variables into a machine-readable format. The dataset is then split into training and testing subsets for further processing.

### 3. Algorithm Development

- **Description:** This module focuses on selecting the best machine learning techniques for traffic volume estimation. It involves implementing algorithms such as XGBoost, Decision Trees, and LSTMs, fine-tuning hyperparameters, and optimizing feature selection to improve the model's accuracy. The selected model learns from past traffic patterns to make reliable predictions.

### 4. Model Training and Testing

- **Description:** The preprocessed data is fed into the machine learning model for training. The model learns patterns from historical traffic data, and its performance is evaluated on unseen test data. Evaluation metrics such as Mean Absolute Error (MAE), Mean Squared Error (MSE), and R-squared scores are used to measure the accuracy of predictions.

### 5. Prediction System

- **Description:** After the model is trained, this module handles real-time prediction tasks. Users can input relevant traffic conditions, and the system provides an estimated traffic volume based on the trained model. This module plays a crucial role in traffic planning and management.

### 6. Evaluation and Optimization

- **Description:** This module continuously evaluates the model's performance and fine-tunes it for better accuracy. It applies various optimization techniques like feature engineering, hyperparameter tuning, and cross-validation to enhance the efficiency and reliability of traffic predictions.

### 7. User Interface (UI) Development

- **Description:** A web-based user interface is developed to allow users to interact with the system. Users can input data and receive predictions in an easy-to-understand graphical format. The UI is developed using HTML, CSS, JavaScript, and frameworks like React to ensure a seamless experience.

### 8. Integration and Deployment

- **Description:** All modules are integrated into a unified system, ensuring smooth data flow and model execution. The final model is deployed on a cloud platform or web server, making it accessible for real-time use. APIs may be used for data integration with external traffic management systems.

## 9. Security and Performance Optimization

- **Description:** This module ensures the security of user data and system performance. It implements encryption techniques, secure authentication, and data access control to prevent unauthorized usage. Performance optimization techniques such as caching, model compression, and load balancing are applied to reduce response time and enhance efficiency.

## 10. Final Testing and Validation

- **Description:** Before deployment, the system undergoes rigorous testing, including unit testing, integration testing, and user acceptance testing. Various test cases are executed to ensure that the system performs well under different traffic conditions. Based on the test results, final refinements are made to enhance accuracy and usability.

# 6.Implementation

## Database:

| | A | B | C | D | E | F | G | H | I |
|---|---|---|---|---|---|---|---|---|---|
| 1 | holiday | temp | rain | snow | weather | date | Time | traffic_volume | |
| 2 | None | 288.28 | 0 | 0 | Clouds | 02-10-2012 | 9:00:00 | 5545 | |
| 3 | None | 289.36 | 0 | 0 | Clouds | 02-10-2012 | 10:00:00 | 4516 | |
| 4 | None | 289.58 | 0 | 0 | Clouds | 02-10-2012 | 11:00:00 | 4767 | |
| 5 | None | 290.13 | 0 | 0 | Clouds | 02-10-2012 | 12:00:00 | 5026 | |
| 6 | None | 291.14 | 0 | 0 | Clouds | 02-10-2012 | 13:00:00 | 4918 | |
| 7 | None | 291.72 | 0 | 0 | Clear | 02-10-2012 | 14:00:00 | 5181 | |
| 8 | None | 293.17 | 0 | 0 | Clear | 02-10-2012 | 15:00:00 | 5584 | |
| 9 | None | 293.86 | 0 | 0 | Clear | 02-10-2012 | 16:00:00 | 6015 | |
| 10 | None | 294.14 | 0 | 0 | Clouds | 02-10-2012 | 17:00:00 | 5791 | |
| 11 | None | 293.1 | 0 | 0 | Clouds | 02-10-2012 | 18:00:00 | 4770 | |
| 12 | None | 290.97 | 0 | 0 | Clouds | 02-10-2012 | 19:00:00 | 3539 | |
| 13 | None | 289.38 | 0 | 0 | Clear | 02-10-2012 | 20:00:00 | 2784 | |
| 14 | None | 288.61 | 0 | 0 | Clear | 02-10-2012 | 21:00:00 | 2361 | |
| 15 | None | 287.16 | 0 | 0 | Clear | 02-10-2012 | 22:00:00 | 1529 | |
| 16 | None | 285.45 | 0 | 0 | Clear | 02-10-2012 | 23:00:00 | 963 | |
| 17 | None | 284.63 | 0 | 0 | Clear | 03-10-2012 | 0:00:00 | 506 | |
| 18 | None | 283.47 | 0 | 0 | Clear | 03-10-2012 | 1:00:00 | 321 | |
| 19 | None | 281.18 | 0 | 0 | Clear | 03-10-2012 | 2:00:00 | 273 | |
| 20 | None | 281.09 | 0 | 0 | Clear | 03-10-2012 | 3:00:00 | 367 | |
| 21 | None | 279.53 | 0 | 0 | Clear | 03-10-2012 | 4:00:00 | 814 | |
| 22 | None | 278.62 | 0 | 0 | Clear | 03-10-2012 | 5:00:00 | 2718 | |
| 23 | None | 278.23 | 0 | 0 | Clear | 03-10-2012 | 6:00:00 | 5673 | |
| 24 | None | 278.12 | 0 | 0 | Clear | 03-10-2012 | 8:00:00 | 6511 | |
| 25 | None | | 0 | 0 | Clear | 03-10-2012 | 9:00:00 | 5471 | |
| 26 | None | 291.97 | 0 | 0 | Clear | 03-10-2012 | 12:00:00 | 5097 | |
| 27 | None | 293.23 | 0 | 0 | Clear | 03-10-2012 | 13:00:00 | 4887 | |
| 28 | None | 294.31 | 0 | 0 | Clear | 03-10-2012 | 14:00:00 | 5337 | |
| 29 | None | 295.17 | 0 | 0 | Clear | 03-10-2012 | 15:00:00 | 5692 | |
| 30 | None | 295.13 | 0 | 0 | Clear | 03-10-2012 | 16:00:00 | 6137 | |
| 31 | None | | 0 | 0 | Clouds | 03-10-2012 | 18:00:00 | 4623 | |
| 32 | None | 290.65 | 0 | 0 | Clouds | 03-10-2012 | 19:00:00 | 3591 | |

# Code of Model Building:

```
lin_reg=linear_model.LinearRegression()

Dtree=tree.DecisionTreeRegressor()

Rand=ensemble.RandomForestRegressor()

svr=svm.SVR()

XGB = xgboost.XGBRegressor()


lin_reg.fit(x_train,y_train)

Dtree.fit(x_train,y_train)

Rand.fit(x_train,y_train)

svr.fit(x_train,y_train)

XGB.fit(x_train,y_train)


p1=lin_reg.predict(x_train)

p2=Dtree.predict(x_train)

p3=Rand.predict(x_train)

p4=svr.predict(x_train)

p5=XGB.predict(x_train)


print(p1)

from sklearn import metrics


print(metrics.r2_score(p1,y_train))

print(metrics.r2_score(p2,y_train))

print(metrics.r2_score(p3,y_train))

print(metrics.r2_score(p4,y_train))

print(metrics.r2_score(p5,y_train))


p1=lin_reg.predict(x_test)

p2=Dtree.predict(x_test)

p3=Rand.predict(x_test)

p4=svr.predict(x_test)

p5=XGB.predict(x_test)


print(metrics.r2_score(p1,y_test))

print(metrics.r2_score(p2,y_test))

print(metrics.r2_score(p3,y_test))

print(metrics.r2_score(p4,y_test))

print(metrics.r2_score(p5,y_test))
```

```
MSE = metrics.mean_squared_error(p3,y_test)


np.sqrt(MSE)


import pickle

pickle.dump(Rand,open("model.pkl",'wb'))

pickle.dump(le,open("encoder.pkl",'wb'))

 from sklearn.preprocessing import StandardScaler

import pickle


# Assume 'X_train' is your training data (features)

scaler = StandardScaler()

X_train_scaled = scaler.fit_transform(x_train)  # Fit and transform on training data


# Save the fitted scaler

pickle.dump(scaler, open("scale.pkl", 'wb'))
```

## Build HTML Code:

Index.html:

```
<!DOCTYPE html>

<html>

<head>

    <meta charset="UTF-8">

    <title>Traffic Volume Estimation</title>

</head>

<body background="https://cdn.vox-cdn.com/thumbor/voARJfEKvTp6iMSzW3ExPn06TDM=/0x78:3000x1766/1600x900/cdn.vox-cdn.com/uploads/chorus_image/image/44219366/72499026.0.0.jpg" text="black">

    <div class="Login">

        <center><h1>Traffic Volume Estimation</h1></center>


        <form action="{{ url_for('predict') }}" method="post" target="_blank">

            <h1>Please enter the following details</h1>


            <label for="holiday">Holiday:</label>

            <select id="holiday" name="holiday">

                <option value=7>None</option>

                <option value=1>Columbus Day</option>

                <option value=10>Veterans Day</option>

                <option value=9>Thanksgiving Day</option>

                <option value=0>Christmas Day</option>

                <option value=6>New Years Day</option>

                <option value=11>Washington's Birthday</option>

                <option value=5>Memorial Day</option>

                <option value=2>Independence Day</option>

                <option value=8>State Fair</option>

                <option value=3>Labor Day</option>

                <option value=4>Martin Luther King Jr Day</option>

            </select>

            <br><br>


            <label>Temperature:</label>

            <input type="number" name="temp" placeholder="temp" required="required" /><br><br>


            <label>Rain:</label>

            <input type="number" min="0" max="1" name="rain" placeholder="rain" required="required" /><br><br>
```

```
            <label>Snow:</label>
            <input type="number" min="0" max="1" name="snow" placeholder="snow" required="required" /><br><br>

            <label for="weather">Weather:</label>
            <select id="weather" name="weather">
                <option value=1>Clouds</option>
                <option value=0>Clear</option>
                <option value=6>Rain</option>
                <option value=2>Drizzle</option>
                <option value=5>Mist</option>
                <option value=4>Haze</option>
                <option value=3>Fog</option>
                <option value=10>Thunderstorm</option>
                <option value=8>Snow</option>
                <option value=9>Squall</option>
                <option value=7>Smoke</option>
            </select>
            <br><br>

            <label>Year:</label>
            <input type="number" min="2012" max="2022" name="year" placeholder="year" required="required" /><br><br>

            <label>Month:</label>
            <input type="number" min="1" max="12" name="month" placeholder="month" required="required" /><br><br>

            <label>Day:</label>
            <input type="number" min="1" max="31" name="day" placeholder="day" required="required" /><br><br>

            <label>Hours:</label>
            <input type="number" min="0" max="24" name="hours" placeholder="hours" required="required" /><br><br>

            <label>Minutes:</label>
            <input type="number" min="0" max="60" name="minutes" placeholder="minutes" required="required"
/><br><br>

            <label>Seconds:</label>
            <input type="number" min="0" max="59" name="seconds" placeholder="seconds" required="required"
/><br><br>
```

```
            <button type="submit" class="btn btn-primary btn-block btn-Large"
style="height:30px;width:200px">Predict</button>

        </form>

    </div>

</body>

</html>
```

## Main Python code:

```
import numpy as np
import pickle
import pandas as pd
import os
from flask import Flask, request, render_template

app = Flask(__name__)

# Load the model and scaler
try:
    model = pickle.load(open('model.pkl', 'rb'))
    scale = pickle.load(open('scale.pkl', 'rb'))
except FileNotFoundError as e:
    print(f"Error: {e}. Ensure model.pkl and scale.pkl exist in the correct path.")
    exit(1)

@app.route('/')  # Home Page
def home():
    return render_template('index.html')

@app.route('/predict', methods=["POST"])  # Prediction Route
def predict():
    try:
        # Reading inputs given by the user
        input_feature = [float(x) for x in request.form.values()]

        # Ensure inputs match the expected feature count
        if len(input_feature) != len(scale.feature_names_in_):
            return render_template('result.html', prediction_text="Error: Incorrect number of inputs.")

        # Convert input to DataFrame with correct feature names
        data = pd.DataFrame([input_feature], columns=scale.feature_names_in_)

        # Transform data using the preloaded scaler
        data = scale.transform(data)

        # Make predictions using the loaded model
```

```
        prediction = model.predict(data)[0]


        # Render the result page with the prediction
        return render_template('result.html', prediction_text=f"Predicted Traffic Volume: {prediction}")


    except Exception as e:
        return render_template('result.html', prediction_text=f"Error: {str(e)}")


if __name__ == "__main__":
    port = int(os.environ.get('PORT', 5000))
    app.run(port=port, debug=True, use_reloader=False)
```

# 7.Results

## 7.1 Output Screens:

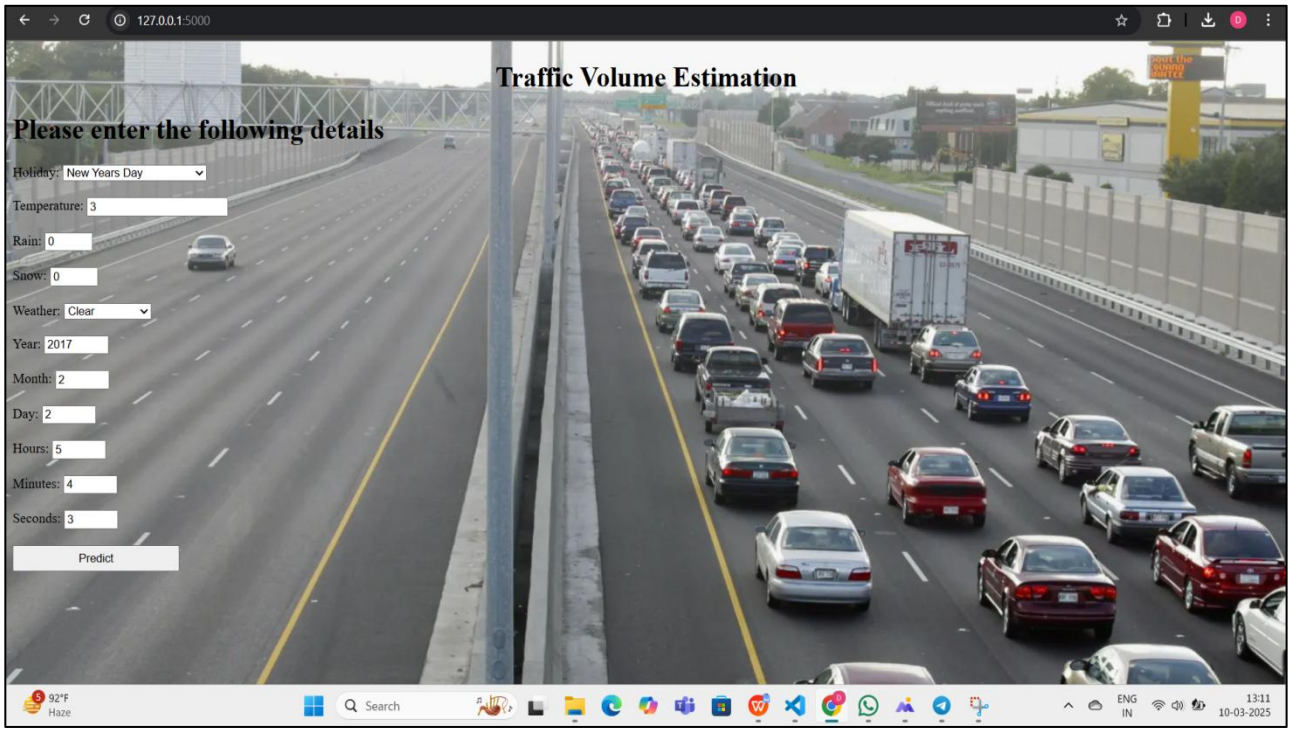


**Fig : Output Screen 1**

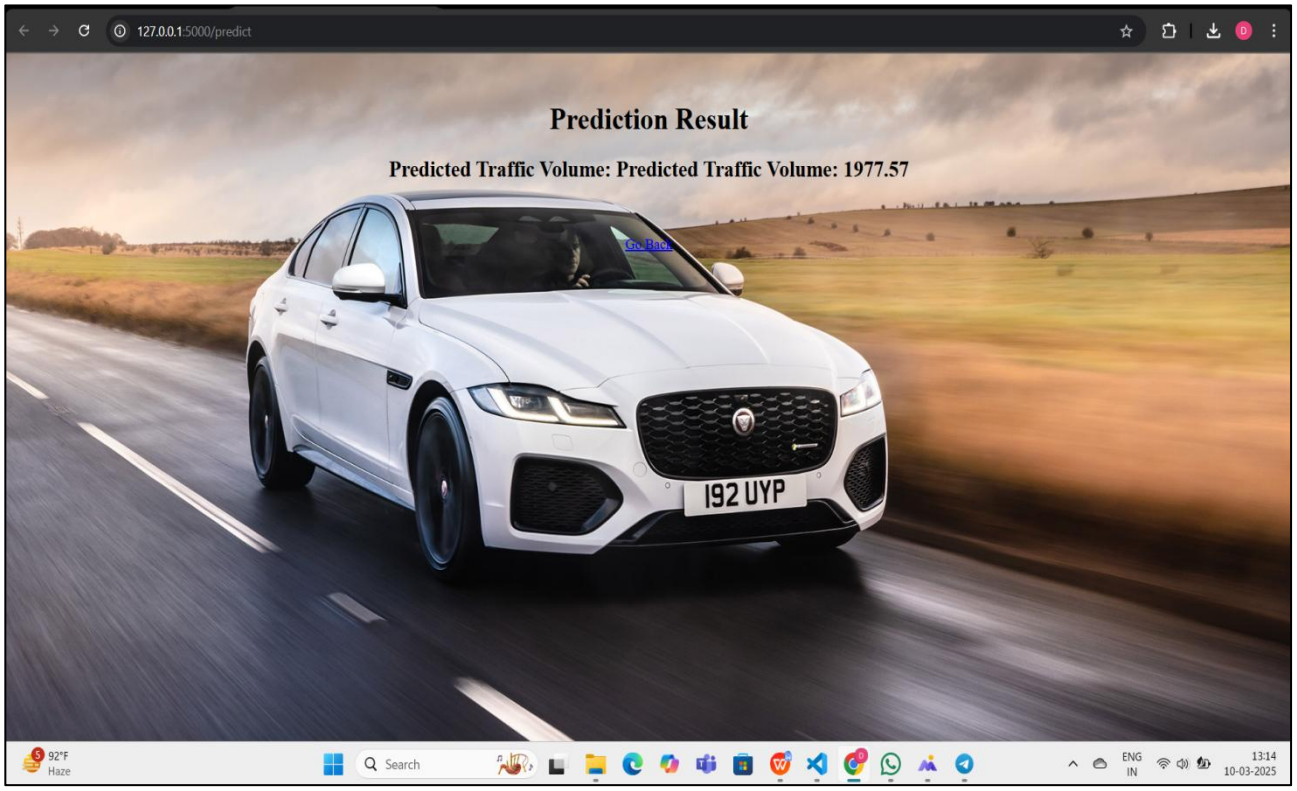


**Fig : Output Screen 2**

# 8.Conclusion

In conclusion, accurate traffic volume estimation is essential for efficient traffic management, urban planning, and commuter navigation. Traditional methods often fail to provide real-time, precise insights, leading to congestion, delays, and inefficient infrastructure planning. TrafficTelligence addresses these challenges by utilizing advanced data analytics, machine learning, and real-time monitoring to offer dynamic traffic flow predictions. This innovative solution empowers transportation authorities to implement adaptive traffic control strategies, enables urban planners to design infrastructure that meets future demands, and enhances commuter navigation by providing real-time route optimizations. By integrating cutting-edge technology, TrafficTelligence transforms traffic management, making cities more efficient, sustainable, and commuter-friendly.

# 9.FUTURE SCOPE

The TrafficTelligence: Advanced Traffic Volume Estimation with Machine Learning project has vast potential for future improvements. It can be integrated with real-time data sources like IoT sensors and GPS to provide accurate traffic predictions. Further enhancements include smart city integration, where traffic lights and urban planning can adapt based on AI-driven insights. The model can also incorporate anomaly detection for accident prediction and multi-modal analysis for public transport optimization. Additionally, deploying a mobile or web-based application can make traffic insights accessible to commuters and authorities in real-time.

# 10.References


1. **Li, Y., & Shahabi, C. (2018).** "A Brief Overview of Machine Learning Methods for Short-Term Traffic Forecasting." *Transportation Research Part C: Emerging Technologies, 95*, 323-345.
2. **Zhang, J., Zhang, X., & Qi, H. (2020).** "Deep Learning Models for Traffic Flow Prediction: A Review." *IEEE Transactions on Intelligent Transportation Systems, 21(6),* 2309-2322.
3. **Wu, Y., & Tan, H. (2016).** "Short-Term Traffic Flow Prediction with LSTM Neural Network." *IEEE 19th International Conference on Intelligent Transportation Systems (ITSC),* 537-542.
4. **Google Developers. (2023).** "Machine Learning for Traffic Estimation: Google Maps Case Study." Retrieved from Google AI Blog.
5. **Scikit-Learn Documentation.** "Preprocessing Techniques for Machine Learning Models." Available at https://scikit-learn.org.
6. **TensorFlow Team.** "Building Deep Learning Models for Traffic Prediction." Available at https://www.tensorflow.org.