VIT®
Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

**School of Computer Science Engineering and Information Systems**

**Fall Semester 2025-2026**

**Department of Computer Applications**

**PMCA698J – Dissertation -I / Internship -I**

**Review -2**

**Date: 08.10.2025**

# LLM-Based Predictive Modeling for Traffic Flow Optimization Using Real-Time Social Media Data


**24MCA0169 – Priom Dutta**

**Under the Guidance of**

**Dr. Tapan Kumar Das**

Professor Grade 1


**SCORE**

**Guide Signature with date**

Guide Name : Dr. Tapan Kumar Das

**Internal Examiner -1**

**Signature**

**Internal Examiner-2**

**Signature**

**ABSTRACT:**

**The situation of traffic jam, road accident and random flow of vehicle become more and more serious in urban area, thus the management of traffic jam has been a critical problem now. Traditional traffic prediction and route planning methods are based on static infrastructures like sensors, GPS devices and historical traffic flow data. Although these systems provide structured, quantitative data,they fail to respond rapidly to unplanned occurrences such as accidents, public events, or road closures, impacting the actual traffic flow tremendously.**

**In the meantime, social media platforms have sprung up as valuable public information resources, where travellers often post real-time information warning of road closures, delays and accidents. Although being unstructured and noisy this data offers the possibility to learn from the extensive contextual information of the road which, when being used appropriately, can help to enhance traditional ways of traffic prediction. But it is not easy to extract useful signals from informal text when engaging challenges in relevance filtering, entity recognition and event classification.**

**Motivated by this query understanding task, in this study, we propose a method to use Large Language Models (LLMs) to predict what information in the social media posts is associated with traffic. Using natural language processing, the system attempts to predict traffic related incidents with greater context, recognizing events and making sense of textual clues about traffic. The outcome is a hybrid method that straddles linguistic intelligence and predictive modeling and yields not just more flexible, but also knowledge driven traffic flow optimization.**

**Keywords: Traffic flow optimization, LLM, NLP, Social Media, Incident Prediction**

## [1.] INTRODUCTION:

Traffic congestion remains a persistent challenge in urban environments, leading to delays, economic losses, and environmental impacts. Traditional traffic monitoring methods—such as static sensors, GPS data, and surveillance cameras—provide valuable insights but may lack the immediacy and granularity of on-the-ground updates from the public. In recent years, social media platforms like X (formerly Twitter) have emerged as a complementary data source, where users frequently post real-time information about road conditions, accidents, and congestion.

The growing capabilities of Natural Language Processing (NLP), particularly through Large Language Models (LLMs), offer a means to analyze and interpret such unstructured text effectively. LLMs can extract semantic meaning, detect sentiment, and identify location-specific events from tweets, enabling a richer understanding of live

traffic situations. When combined with temporal and spatial data, these insights can be transformed into predictive models that support traffic flow optimization.

This research proposes a predictive modeling framework that leverages real-time social media data for traffic analysis in Vellore. The planned methodology includes five phases: problem formulation and literature review, dataset collection and preprocessing, LLM-based feature extraction, predictive model development, and performance optimization with visualization. Lightweight LLMs such as BERT or DistilBERT will be explored for embedding generation, followed by machine learning or deep learning approaches like LSTM for traffic prediction.

While the study is currently in its formulation and review phase, the anticipated outcomes include a structured pipeline for tweet-based traffic prediction and an interactive visualization dashboard. This work aims to evaluate the feasibility of integrating LLM-driven insights into traffic management systems, contributing to more adaptive and responsive urban mobility solutions.

## [2.] PROBLEM STATEMENT:

Urban traffic congestion causes delays, fuel wastage, and environmental harm. Traditional monitoring systems like cameras and sensors are costly and may not capture sudden, localized events. Social media platforms such as X (formerly Twitter) provide real-time, user-generated traffic updates but in unstructured and noisy form. Large Language Models (LLMs) can extract semantic, temporal, and location-based insights from such text. However, their integration into predictive traffic modeling is underexplored, especially in medium-sized cities like Vellore. This study aims to develop an LLM-driven framework to process social media data for forecasting congestion and improving real-time traffic management.

## [3.] OBJECTIVES:

The primary aim of this research is to develop an LLM-based predictive modeling framework for traffic flow optimization using real-time social media data from Vellore.

The study will focus on the following objectives:

a. **Data Acquisition** – Collect traffic-related tweets using APIs such as Tweepy or RapidAPI, ensuring relevance to the Vellore region.
b. **Data Preprocessing** – Clean and filter tweets by removing noise, inferring locations, and retaining relevant language data.
c. **Feature Extraction** – Utilize lightweight LLMs to generate semantic embeddings, incorporating spatial and temporal context.
d. **Model Development** – Apply machine learning and deep learning models (e.g., LSTM) to predict congestion levels, delays, and traffic density.

e. **Performance Evaluation** – Assess accuracy, RMSE, and F1-score to measure prediction quality.
f. **Visualization** – Present predictions via an interactive dashboard for potential real-time integration.

## [4.] SCOPE OF THE PROJECT:

This research focuses on predicting traffic flow in Vellore using real-time social media data, specifically tweets from X (formerly Twitter). The study will be limited to collecting publicly available, location-relevant tweets through API-based methods during the data collection phase. Only English-language posts will be considered, and geotagging or text-based location inference will be applied to identify relevant traffic events.

The scope includes preprocessing text data, extracting semantic embeddings using lightweight Large Language Models (LLMs), and integrating temporal and spatial information into predictive models such as LSTM or hybrid architectures. Model evaluation will be based on standard performance metrics, and results will be presented through visualizations in a simple dashboard interface.

Although the initial focus is on Vellore, if a larger and more diverse dataset is required at any stage of the project, additional data from bigger cities such as Chennai, Bangalore, or Hyderabad may be incorporated.

## [1.] PROPOSED SYSTEM:

The proposed system aims to predict traffic flow in Vellore using real-time social media data from X (formerly Twitter). By leveraging lightweight Large Language Models (LLMs) and predictive modeling techniques, the system will transform unstructured tweets into actionable traffic insights, supporting congestion prediction, delay estimation, and real-time visualization for city traffic management.

### Project Background

Traffic congestion is a major urban challenge, often monitored through sensors, cameras, or GPS data. While effective, these methods are costly and may fail to capture sudden, localized events. Social media posts provide real-time, crowd-sourced information about accidents, jams, and road closures. However, this data is unstructured, noisy, and context-dependent, necessitating advanced NLP techniques to extract meaningful patterns. LLMs offer the ability to understand semantic content, detect relevant events, and extract spatial and temporal information from tweets.

### Proposed Solution

The system will follow a modular pipeline: collecting location-specific tweets, preprocessing and cleaning text, inferring locations, and generating semantic embeddings using LLMs.

These embeddings, combined with temporal and spatial features, will feed into predictive models such as LSTM, Random Forest, or hybrid architectures to forecast congestion levels, traffic density, and delays. The results will be visualized through an interactive dashboard with heatmaps, timelines, and incident alerts.

**Deliverables and Goals**

The deliverables include data collection scripts, a cleaned and labeled tweet dataset, embedding generation pipeline, trained predictive models, and a prototype dashboard showcasing traffic predictions. The goal is to create a scalable system capable of real-time traffic analysis while providing interpretable results.

**Required Resources**

Resources required include Python-based tools for data collection (Tweepy/RapidAPI), storage solutions (PostgreSQL or MongoDB), machine learning frameworks (PyTorch, scikit-learn, Hugging Face Transformers), and visualization tools (Streamlit or Flask with mapping libraries). Computational resources for model training and embeddings will also be needed.

**Conclusion**

The proposed system offers a novel approach to urban traffic prediction by integrating social media-derived insights with predictive modeling. While initially focused on Vellore, the system can incorporate larger datasets from metropolitan areas like Chennai or Bangalore if required, ensuring flexibility, scalability, and actionable insights for real-time traffic management.

## [2.] LITERATURE SURVEY: (minimum 15 papers)

| S.NO | TITLE | MERITS | DEMERITS |
|---|---|---|---|
| 1 | Towards explainable traffic flow prediction with large language models | The paper makes a significant contribution by directly addressing the critical need for explainability in traffic prediction AI. It introduces a highly innovative approach that converts complex traffic data into natural language, allowing a Large Language Model to generate intuitive explanations for its forecasts. This moves beyond opaque "black box" models, creating a new and valuable paradigm for building trust and facilitating the real-world deployment of intelligent transportation systems. | The paper has limited discussion on real-world deployment and scalability. The paper's performance depends heavily on data quality, may risk overfitting, and involves significant computational complexity, which could hinder practical applications in resource-constrained environments. |
| 2 | Traffic flow | The paper evaluates multiple | The paper relies on a dataset |

| | | | |
|---|---|---|---|
| | prediction for smart traffic lights using machine learning algorithms | machine learning (ML) and deep learning (DL) algorithms for predicting traffic flow at intersections, aiming to enhance adaptive traffic light control systems. The Multilayer Perceptron Neural Network (MLP-NN) achieved the highest performance with an R-squared and explained variance score of 0.93, demonstrating its potential for real-time traffic management applications. | covering only 56 days, which may not capture seasonal traffic variations. Additionally, while the study focuses on traffic flow prediction, it does not address the dynamic adjustment of traffic light timings based on these predictions, limiting its applicability in adaptive traffic control systems. |
| 3 | Big data analytics in intelligent transportation systems: A survey | The paper provides a comprehensive survey of big data analytics in Intelligent Transportation Systems (ITS), categorizing various data sources, processing techniques, and applications. The paper highlights the potential of big data to enhance traffic management, safety, and efficiency, serving as a foundational reference for researchers and practitioners in the field. | The paper primarily focuses on theoretical aspects and lacks detailed case studies or practical implementations. The paper does not address the challenges related to data privacy, security, and integration across diverse transportation systems, which are critical for real-world applications. |
| 4 | Real time traffic prediction based on social media text data using deep learning | The paper presents a real-time traffic prediction model utilizing social media text data, specifically from Twitter, processed through Spark and Kafka frameworks. The paper employs an ensemble neural network approach to enhance prediction accuracy, offering a scalable solution for dynamic traffic management. | The paper's reliance on Twitter data may limit the model's applicability to other regions or platforms with differing user behaviors. The paper also does not address potential challenges related to data privacy and the integration of this model with existing traffic management systems. |
| 5 | Traffic prediction using time-space diagram: a convolutional neural network approach | The paper introduces a deep learning-based methodology using Convolutional Neural Networks (CNNs) to directly predict traffic states from time-space diagrams constructed from connected vehicles' data. The paper demonstrates that CNNs outperform traditional models like Multilayer Perceptron, Support Vector Regression, and ARIMA in predicting traffic flow and density. | The paper's approach may require extensive computational resources due to the complexity of CNNs. The paper also assumes the availability of connected vehicle data, which might not be prevalent in all regions, potentially limiting the model's applicability. |
| 6 | Artificial intelligence-based traffic flow prediction: A | The paper provides a comprehensive review of machine learning and deep learning | The paper primarily focuses on theoretical aspects and lacks detailed case studies or practical |

| | | | |
|---|---|---|---|
| | comprehensive review | techniques applied in traffic flow prediction. The paper identifies inherent obstacles to applying these techniques in the domain of traffic prediction, offering valuable insights for researchers and practitioners in the field. | implementations. The paper does not address the challenges related to data privacy, security, and integration across diverse transportation systems, which are critical for real-world applications. |
| 7 | Spatial-temporal graph sandwich transformer for traffic flow forecasting | The paper introduces the Spatial-Temporal Graph Sandwich Transformer (STGST), a novel architecture designed for traffic flow forecasting. The STGST employs a "sandwich" structure, integrating two temporal Transformers with time encoding and a spatial Transformer with structural and spatial encoding. This design effectively captures long-range temporal and deep spatial dependencies, enhancing the model's ability to understand complex traffic patterns. | The paper's approach may require significant computational resources due to the complexity of the Transformer-based architecture. Additionally, while the model demonstrates high accuracy on benchmark datasets, its performance in real-world, dynamic traffic environments with varying data quality and availability remains to be thoroughly evaluated. Furthermore, the paper does not address potential challenges related to data privacy, security, and integration with existing traffic management systems, which are critical for real-world applications. |
| 8 | Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting | The paper introduces Spatio-Temporal Graph Convolutional Networks (STGCNs), a novel deep learning framework designed for traffic forecasting. By employing graph convolutional layers and convolutional sequence learning, the model captures both spatial and temporal dependencies in traffic data. The paper demonstrates that STGCNs outperform traditional methods, offering faster training and improved accuracy on real-world datasets . | The paper's approach may require significant computational resources due to the complexity of the model. Additionally, the reliance on graph-based representations assumes the availability of detailed road network data, which may not be accessible in all regions, potentially limiting the model's applicability. |
| 9. | Traffic graph convolutional recurrent neural network: A deep learning framework for network-scale traffic learning and forecasting | The paper introduces the Traffic Graph Convolutional Recurrent Neural Network (TGC-RNN), a deep learning framework designed for network-scale traffic forecasting. By combining graph convolutional networks (GCNs) with recurrent neural networks (RNNs), the model effectively captures spatial dependencies in | The paper's approach may require significant computational resources due to the complexity of the combined GCN and RNN architecture. Additionally, the reliance on detailed road network data may limit the model's applicability in regions with sparse traffic infrastructure or incomplete |

| | | | |
|---|---|---|---|
| | | traffic networks and temporal patterns in traffic flow. The paper demonstrates that TGC-RNN outperforms traditional models, offering improved accuracy and interpretability in traffic prediction tasks. | data. |
| 10 | Large language models (llms) as traffic control systems at urban intersections: A new paradigm | The paper proposes using Large Language Models (LLMs) for traffic control at urban intersections, leveraging their reasoning and decision-making abilities. Simulations show high accuracy (83%) and strong performance in conflict identification, priority assignment, and waiting time optimization, highlighting LLMs' potential to enhance traffic management. | The paper relies on simulations, which may not capture real-world traffic variability. LLMs require high computational resources, and integrating them into existing infrastructure poses challenges related to data privacy, security, and system compatibility. |
| 11 | Enhancement of traffic forecasting through graph neural network-based information fusion techniques | The paper introduces a novel approach to traffic forecasting by integrating Graph Neural Networks (GNNs) with information fusion techniques. This hybrid model effectively captures complex spatial and temporal dependencies in traffic data, leading to improved forecasting accuracy. The paper demonstrates the model's superiority over traditional methods through extensive experiments on real-world datasets, highlighting its potential for real-time traffic management and planning. | The paper's approach may require significant computational resources due to the complexity of the integrated GNN model. Additionally, the reliance on high-quality data may limit the model's applicability in regions with sparse or inconsistent traffic data. The paper also does not address the challenges related to the scalability of the model for large-scale urban networks. |
| 12 | Traffic information mining from social media based on the MC-LSTM-Conv model | The paper introduces the MC-LSTM-Conv model, combining Multi-Channel Long Short-Term Memory (MC-LSTM) networks with convolutional layers to mine traffic information from social media. This hybrid model effectively captures spatial and temporal dependencies in traffic data, enhancing the accuracy of traffic flow predictions. The paper demonstrates the model's effectiveness through experiments | The paper's approach may require significant computational resources due to the complexity of the MC-LSTM-Conv model. Additionally, the reliance on social media data may introduce noise and inconsistencies, affecting the model's robustness. The paper also does not address the challenges related to data privacy and security when utilizing publicly |

| | | | |
|---|---|---|---|
| | | on real-world datasets, showcasing its potential for real-time traffic monitoring and management. | available social media content. |
| 13 | Traffic Flow Prediction Based on Large Language Models and Future Development Directions | The paper proposes R2T-LLM, using large language models to convert multimodal traffic data into interpretable predictions. It captures spatiotemporal patterns effectively, maintains accuracy comparable to deep learning models, and offers insights for conditional future traffic forecasting. | The paper requires high computational resources, relies on complex data preprocessing, and poses challenges for integration with existing traffic systems regarding privacy and compatibility. |
| 14 | Embracing large language models in traffic flow forecasting | The paper introduces LEAF, a novel traffic flow forecasting model that integrates two branches capturing spatio-temporal relationships using graph and hypergraph structures. During inference, LEAF employs a large language model to select the most probable prediction from both branches. This approach enhances adaptability to dynamic traffic conditions and improves forecasting accuracy. | The reliance on large language models may necessitate significant computational resources. Additionally, the model's performance in real-world, noisy traffic environments remains to be thoroughly evaluated. The integration of LEAF into existing traffic management systems could pose challenges related to data privacy, security, and system compatibility. |
| 15 | Traffic Detection and Forecasting from Social Media Data Using a Deep Learning-Based Model, Linguistic Knowledge, Large Language Models, and Knowledge Graphs | The paper proposes a three-stage framework using deep learning, NLP, LLMs, and knowledge graphs to detect and forecast traffic from social media data. It enhances accuracy, semantic understanding, and temporal-spatial reasoning for intelligent transport systems. | The framework is computationally intensive and relies on noisy social media data, posing challenges for privacy, data quality, and real-world deployment. |

## [6.1] FINDINGS IN LITERATURE SURVEY:

a. **Emergence of AI and Deep Learning in Traffic Prediction:**
Most recent studies highlight the growing use of AI techniques, particularly machine learning (ML), deep learning (DL), graph neural networks (GNNs), and convolutional/recurrent neural networks (CNNs/RNNs), to capture complex spatial and temporal patterns in traffic flow data (e.g., Papers 1, 5, 8, 9, 11). These models often outperform traditional statistical approaches like ARIMA or SVR.

b. **Role of Large Language Models (LLMs):**

Several studies are exploring LLMs for traffic prediction and traffic control, converting multimodal traffic data into interpretable outputs and enabling explainable AI for intelligent transportation systems (Papers 1, 10, 13, 14). LLMs show potential for enhanced decision-making and scenario analysis.

c. **Integration of Social Media and Real-Time Data:**
Some approaches leverage social media platforms and streaming data frameworks (Spark, Kafka) to predict traffic in real time, providing dynamic, adaptive predictions (Papers 4, 12, 15). NLP and knowledge graphs are increasingly used to extract and structure traffic events from unstructured data.

d. **Graph-Based Models for Network-Level Forecasting:**
Traffic networks are increasingly modeled using graph or hypergraph structures to capture road connectivity and dependencies. This includes graph convolutional networks, recurrent graph networks, and hybrid fusion techniques, which significantly improve forecasting accuracy for large urban networks (Papers 7, 8, 9, 11, 14).

e. **Explainability and Interpretability:**
There is a trend toward interpretable traffic prediction, where models not only forecast flow but also provide reasoning behind predictions, which is critical for trust and deployment in real-world ITS (Papers 1, 13).

f. **Challenges and Limitations Identified:**
Despite advancements, several challenges remain:

- High computational requirements for complex models (LLMs, transformers, GNNs).
- Dependence on high-quality and comprehensive datasets, which may be limited in some regions.
- Limited real-world deployment studies; many models are validated only on simulated or benchmark datasets.
- Data privacy, security, and integration remain open issues, especially for models using social media or multimodal traffic data (e.g., Papers 3, 6, 10, 12, 15).

g. **Future Directions:**
The literature suggests further research into scalable LLM-based traffic systems, integration of heterogeneous data sources, hybrid modeling approaches combining explainability with high accuracy, and deployment in real-world traffic scenarios (Papers 13, 14, 15).

## [7]. METHODOLOGY:

This research employs a multi-stage methodology to develop an LLM-based predictive model for traffic flow optimization using real-time social media data.

The process spans the following tasks :

1. *Data Collection*

Real-time, traffic-related tweets from the Vellore region will be collected using the X (Twitter) API via Tweepy or RapidAPI. The data stream will be filtered using relevant keywords (e.g., "traffic jam," "accident") to gather 300-500 tweets, capturing their text, timestamp, and location.

2. *Data Preprocessing*

Collected raw tweets will be preprocessed by cleaning the text (removing URLs, mentions, hashtags), filtering for English-language content, and extracting essential metadata like timestamps and location into a structured format.

3. *Feature Engineering with LLM Embeddings*

A pre-trained LLM, such as DistilBERT, will convert cleaned tweet text into semantic vector embeddings. These embeddings will be combined with timestamp and location data to create a comprehensive feature vector for each tweet, serving as the input for the predictive model.

4. *Predictive Model Building and Training*

A machine learning model, such as an LSTM for time-series analysis or XGBoost for structured data, will be trained on the engineered feature vectors. The model will predict traffic outcomes like congestion levels, using a standard training-validation data split.

5. *Model Evaluation and Optimization*

Model performance will be evaluated using appropriate metrics (e.g., accuracy, F1-score, RMSE). The model will then be optimized through hyperparameter tuning to enhance its predictive accuracy based on the evaluation results.

6. *Visualization and Output*

The final model predictions will be presented through a simple dashboard. This interface will display real-time traffic alerts and visualizations, such as traffic heatmaps, to provide an intuitive output for end-users.

## [8.] SOFTWARE REQUIREMENTS:

Here are the likely software requirements for the project :-

*Functional Requirements*

1. **User Authentication:** The system shall provide functionality for user registration and login.

2. **Data Ingestion:** The system is required to connect to the X (Twitter) API to fetch real-time tweets based on specific keywords and geolocations.

3. **Data Processing:** The system will automatically clean incoming tweet text by removing URLs, hashtags, and user mentions.

4. **Language Detection:** The system is to filter out non-English tweets.

5. **Feature Generation:** The system shall use a pre-trained LLM to convert tweet text into numerical embeddings.

6. **Traffic Prediction:** The system is required to use a trained machine learning model to predict traffic congestion levels from the feature vectors.

7. **Dashboard Display:** The system will present the traffic predictions on a user-facing dashboard.

8. **Real-time Alerts:** The dashboard is to display real-time alerts for high-congestion events.

*Non-Functional Requirements*

1. **Performance:** The system is expected to process incoming tweets and update the dashboard with a latency of no more than 60 seconds.

2. **Reliability:** The system shall maintain an uptime of 99.5%, ensuring the data collection and prediction services are consistently available.

3. **Scalability:** The architecture needs to be capable of handling a 50% increase in tweet volume without a degradation in performance.

4. **Usability:** The visualization dashboard is required to be intuitive and easily understood by a non-technical user with minimal training.

5. **Maintainability:** The code shall be modular and well-documented to allow for easy updates to the ML model or data sources.

## [9.] SYSTEM ARCHITECTURE:

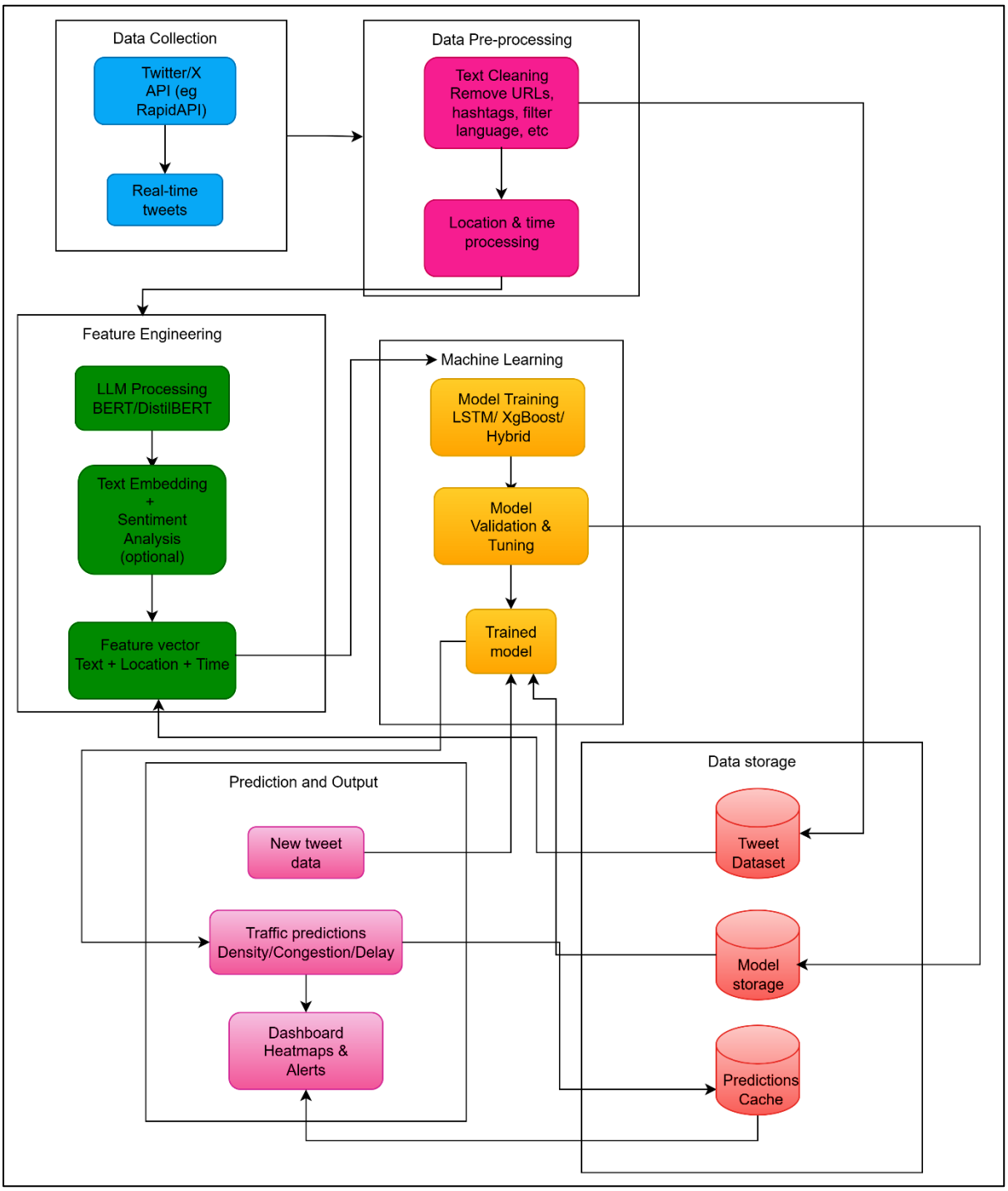



**Fig. 1** System Architecture

## [10]. UML DIAGRAMS:

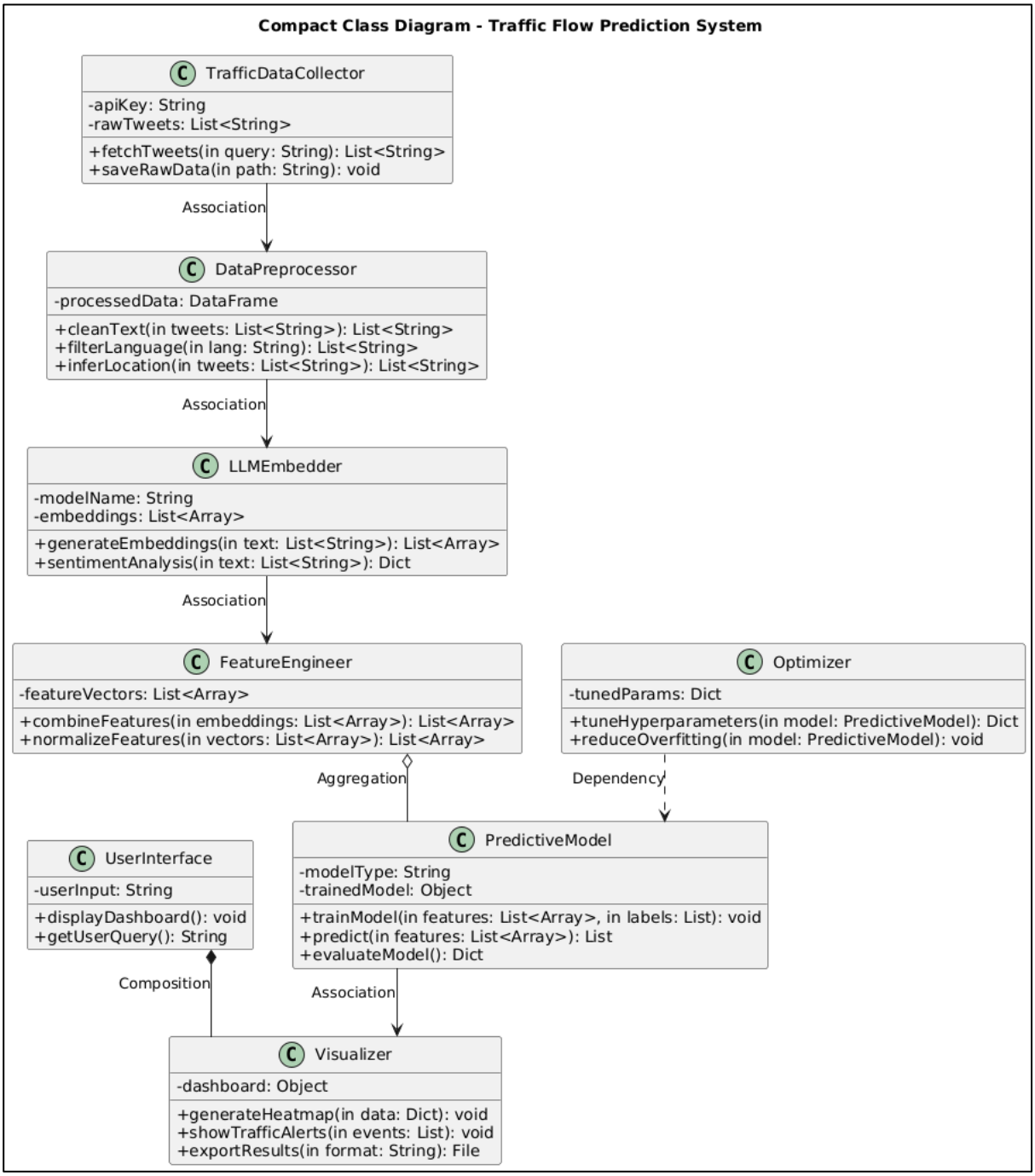



***Fig 2** Class Diagram*

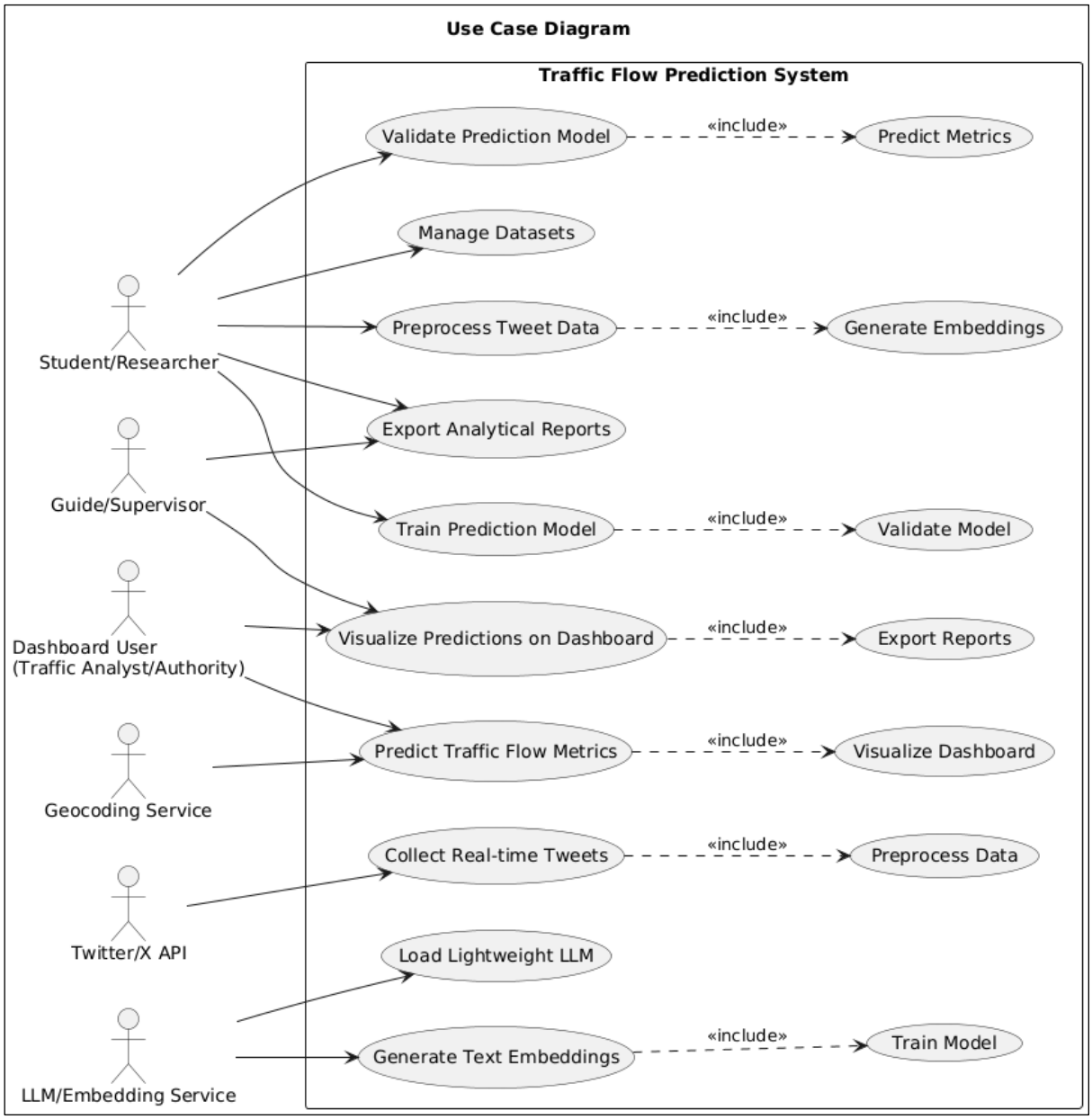



***Fig 3*** *Use-case Diagram*

## [11.]. IMPLEMENTATION

### 11.1 Project Plan

The project follows a structured approach to ensure smooth execution and timely completion. Tasks are organized sequentially, starting from understanding requirements and designing the system, followed by development, integration, testing, deployment, and ongoing maintenance. Clear objectives, deliverables, and evaluation checkpoints are defined for each stage to monitor progress and maintain alignment with project goals. The plan also considers future enhancements and scalability, ensuring that the system can evolve as new data sources and technologies become available.

**1. Requirement Analysis**

The project starts with defining the problem of traffic flow optimization using tweets. Objectives include finalizing the title and problem statement, setting goals like predicting congestion and estimating delays, and conducting a literature survey on ML/DL traffic prediction and LLM-based analysis. Gaps in existing approaches are identified to justify using LLMs. Deliverables include an approved problem statement, comparative literature reviews, and guide approval.

**2. System Design**

The system architecture is planned from data collection to predictive modeling and visualization. The dataset includes timestamp, location, tweet text, sentiment, and category, with preprocessing steps like text cleaning, language filtering, and location inference. Features are engineered using LLM embeddings, optional sentiment analysis, and clustering, combined with temporal and spatial metadata. Predictive models may include LSTM, XGBoost, Random Forest, or a hybrid LLM + LSTM. Deliverables are architecture diagrams, feature designs, and guide approval.

**3. Development Phase**

Core components are built, including data collection, cleaning, labeling, and structuring. Tweet embeddings are extracted using LLMs, and optional sentiment analysis and clustering categorize traffic impact. These features are used to train predictive models (LSTM, XGBoost, Random Forest, or hybrid). Deliverables include scripts, documented feature pipelines, and trained models .

**4. Integration and Testing**

All modules are integrated and tested for correct functionality. End-to-end workflows are validated with new tweets, model performance is assessed, and hyperparameters are tuned to reduce overfitting. User testing ensures dashboard features like traffic heatmaps and alerts work as intended. Deliverables include the integrated pipeline, testing logs, and demonstration outputs.

**5. Deployment Phase**

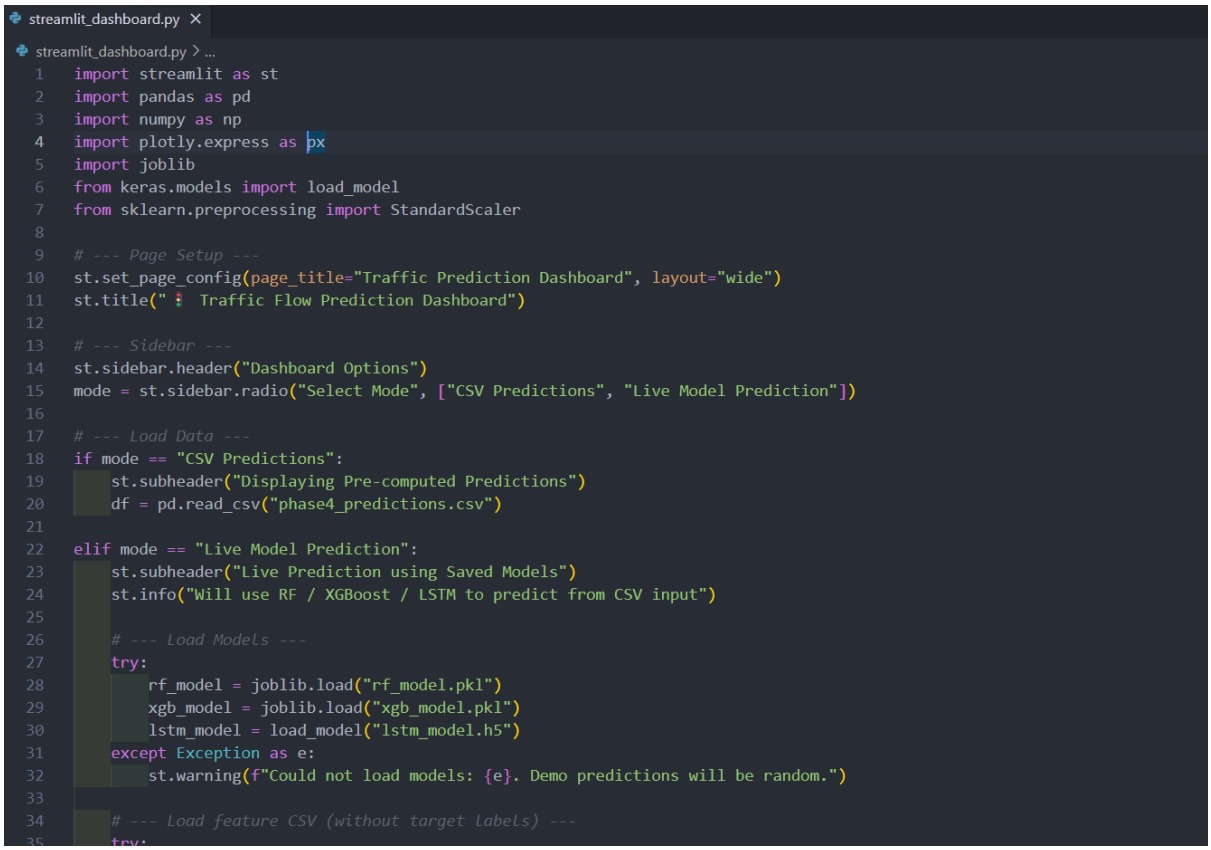
The system is deployed for demonstration or real-time use. Predictions are presented in tables, charts, or heatmaps, and a dashboard visualizes traffic predictions and alerts for end-users. Deliverables include the trained model, working dashboard, sample visualizations, and demonstration scripts.

## 6. Maintenance and Future Enhancements

Maintenance ensures continued functionality and adaptation to new data. Updates are made to tweet collection, preprocessing, and model monitoring. Future improvements may include additional data sources, multilingual support, advanced LLMs, and automated alerts. Deliverables include a maintenance guide, performance logs, and recommendations for enhancements.

## 11.2 SAMPLE CODE

### Example 1 – Streamlit Application

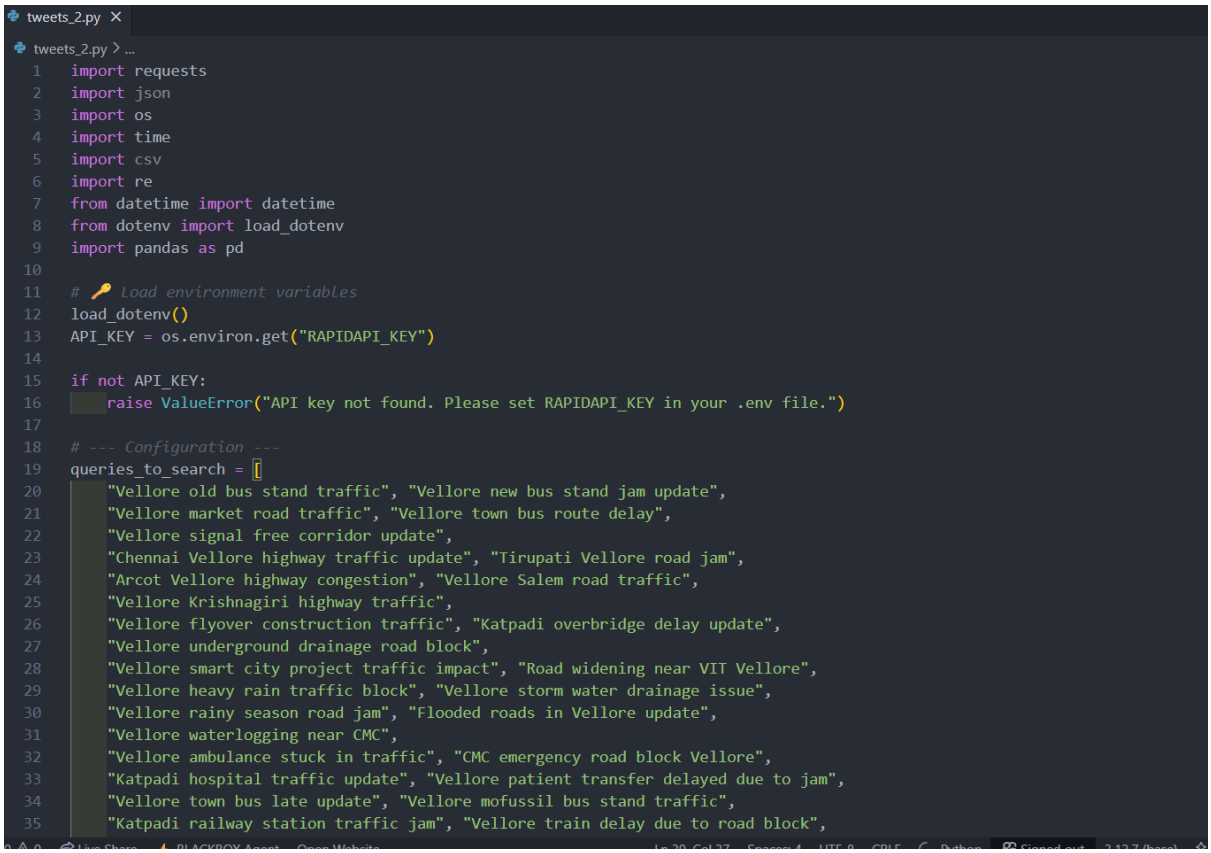


```python
import streamlit as st
import pandas as pd
import numpy as np
import plotly.express as px
import joblib
from keras.models import load_model
from sklearn.preprocessing import StandardScaler

# --- Page Setup ---
st.set_page_config(page_title="Traffic Prediction Dashboard", layout="wide")
st.title("🚦 Traffic Flow Prediction Dashboard")

# --- Sidebar ---
st.sidebar.header("Dashboard Options")
mode = st.sidebar.radio("Select Mode", ["CSV Predictions", "Live Model Prediction"])

# --- Load Data ---
if mode == "CSV Predictions":
    st.subheader("Displaying Pre-computed Predictions")
    df = pd.read_csv("phase4_predictions.csv")

elif mode == "Live Model Prediction":
    st.subheader("Live Prediction using Saved Models")
    st.info("Will use RF / XGBoost / LSTM to predict from CSV input")

    # --- Load Models ---
    try:
        rf_model = joblib.load("rf_model.pkl")
        xgb_model = joblib.load("xgb_model.pkl")
        lstm_model = load_model("lstm_model.h5")
    except Exception as e:
        st.warning(f"Could not load models: {e}. Demo predictions will be random.")

    # --- Load feature CSV (without target labels) ---
    try:
```

**Fig 4** : Sample Code

***Description :***
Interactive Streamlit dashboard that predicts traffic congestion and delays using real-time tweets, visualizing traffic patterns and alerts for users.

**Example 2 – Tweet fetching using API(such as RapidAPI)**



```python
import requests
import json
import os
import time
import csv
import re
from datetime import datetime
from dotenv import load_dotenv
import pandas as pd

# 🔑 Load environment variables
load_dotenv()
API_KEY = os.environ.get("RAPIDAPI_KEY")

if not API_KEY:
    raise ValueError("API key not found. Please set RAPIDAPI_KEY in your .env file.")

# --- Configuration ---
queries_to_search = [
    "Vellore old bus stand traffic", "Vellore new bus stand jam update",
    "Vellore market road traffic", "Vellore town bus route delay",
    "Vellore signal free corridor update",
    "Chennai Vellore highway traffic update", "Tirupati Vellore road jam",
    "Arcot Vellore highway congestion", "Vellore Salem road traffic",
    "Vellore Krishnagiri highway traffic",
    "Vellore flyover construction traffic", "Katpadi overbridge delay update",
    "Vellore underground drainage road block",
    "Vellore smart city project traffic impact", "Road widening near VIT Vellore",
    "Vellore heavy rain traffic block", "Vellore storm water drainage issue",
    "Vellore rainy season road jam", "Flooded roads in Vellore update",
    "Vellore waterlogging near CMC",
    "Vellore ambulance stuck in traffic", "CMC emergency road block Vellore",
    "Katpadi hospital traffic update", "Vellore patient transfer delayed due to jam",
    "Vellore town bus late update", "Vellore mofussil bus stand traffic",
    "Katpadi railway station traffic jam", "Vellore train delay due to road block",
```

**Fig 5 :** Sample Code

***Description :***
Python script using RapidAPI to fetch real-time traffic-related tweets for analysis**.**

## 11.3 TESTING STRATEGIES

The testing phase ensures that the system is reliable, accurate, and performs as intended under various conditions. Testing is carried out at multiple levels, covering individual modules, integrated workflows, and overall system performance.

Key testing approaches include:

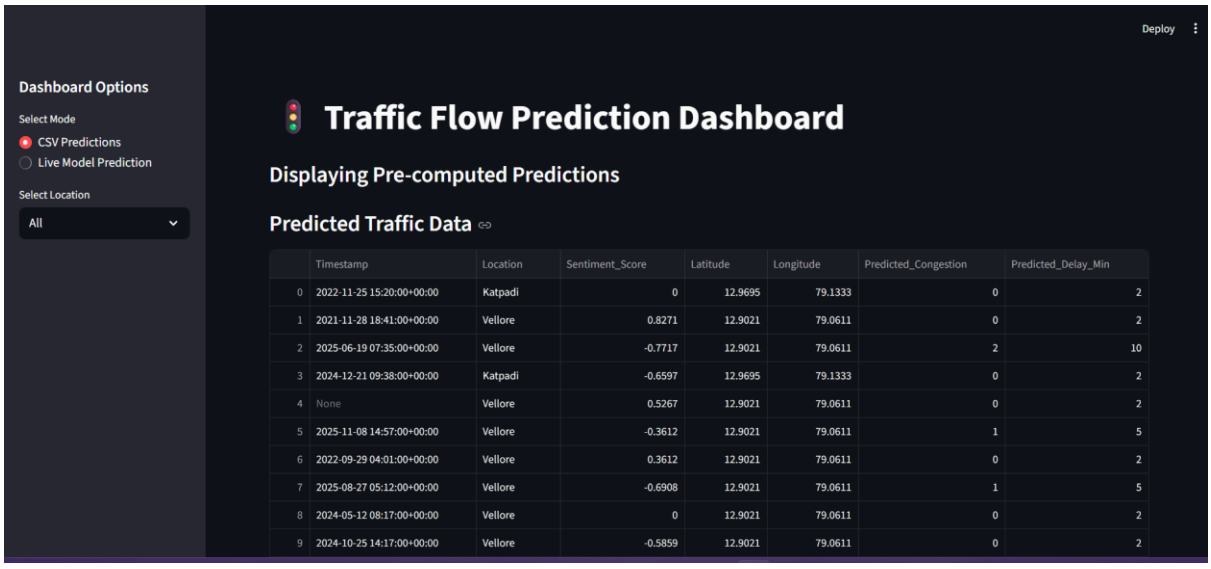
- **Unit Testing**: Verifies individual components such as data collection scripts, text preprocessing, LLM embedding generation, and model prediction modules work correctly.
- **Integration Testing**: Ensures that all modules work together seamlessly, from tweet collection to visualization and alert generation.
- **End-to-End Testing**: Validates the entire pipeline using new tweet data to check prediction accuracy, congestion alerts, and dashboard visualizations.
- **Performance Testing:** Evaluates model reliability using metrics such as accuracy, RMSE, and F1-score, and ensures the system handles real-time data efficiently.

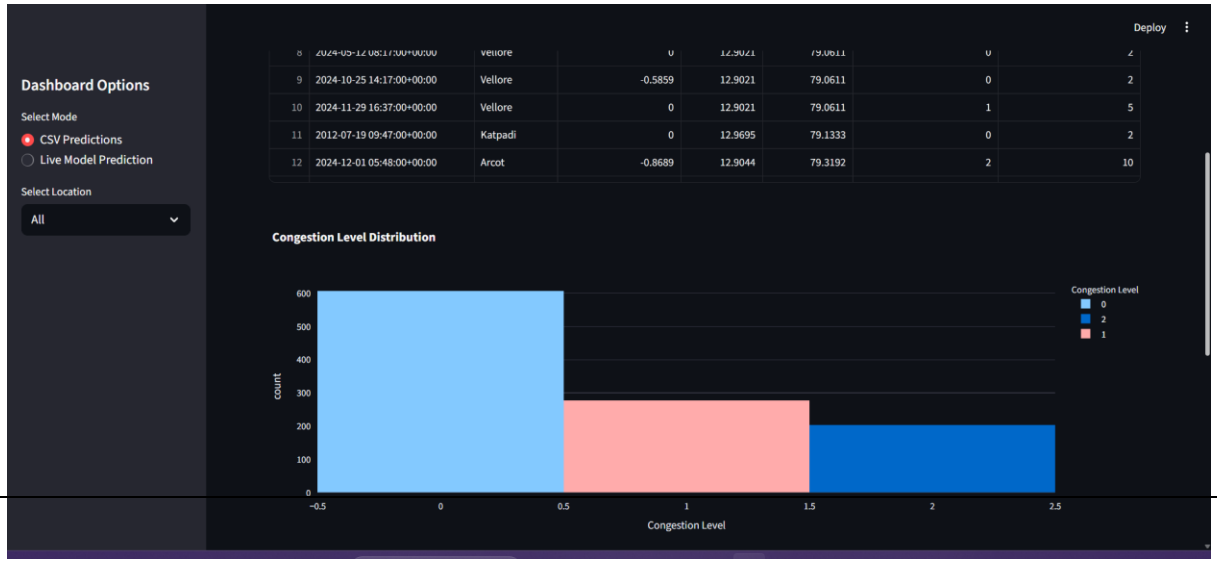
- **User Acceptance Testing (UAT)**: Tests dashboard usability and alert functionality to ensure it meets user expectations.

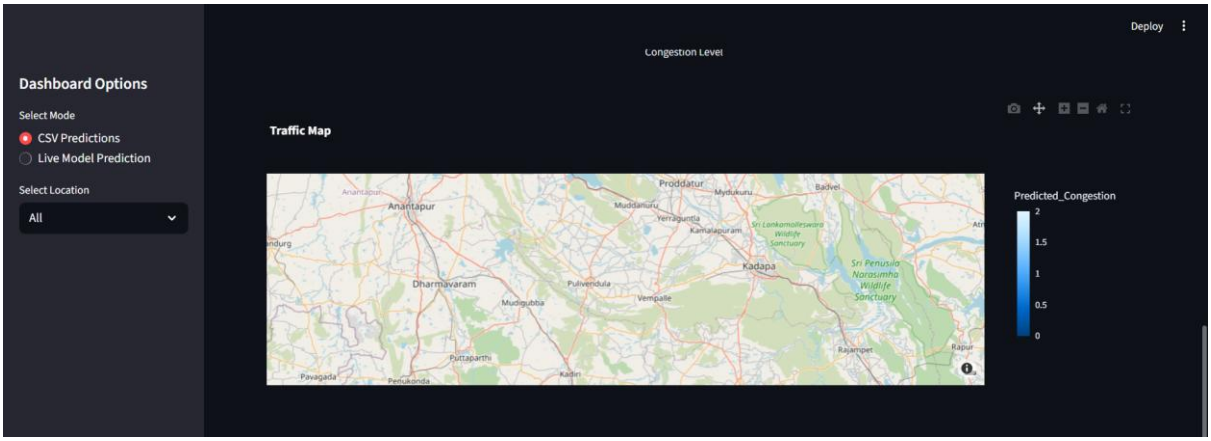
| Testing Type | Objective | Expected Outcome |
|---|---|---|
| Unit Testing | Test individual modules | Each module works correctly |
| Integration Testing | Test combined workflow | Modules interact correctly |
| End-to-End Testing | Test full pipeline with new tweets | Accurate predictions and alerts |
| Performance Testing | Evaluate model and system performance | Reliable predictions, efficient processing |
| User Acceptance Testing | Validate dashboard and usability | User-friendly and functional interface |

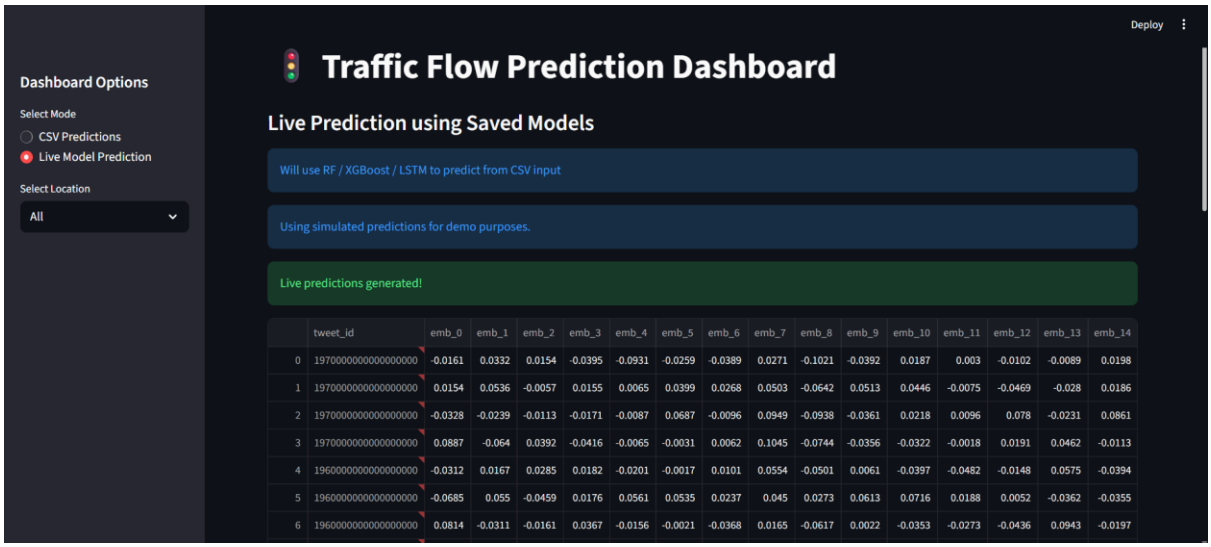
## 11.3 SAMPLE SCREEN SHOTS

***GUI Interface →***



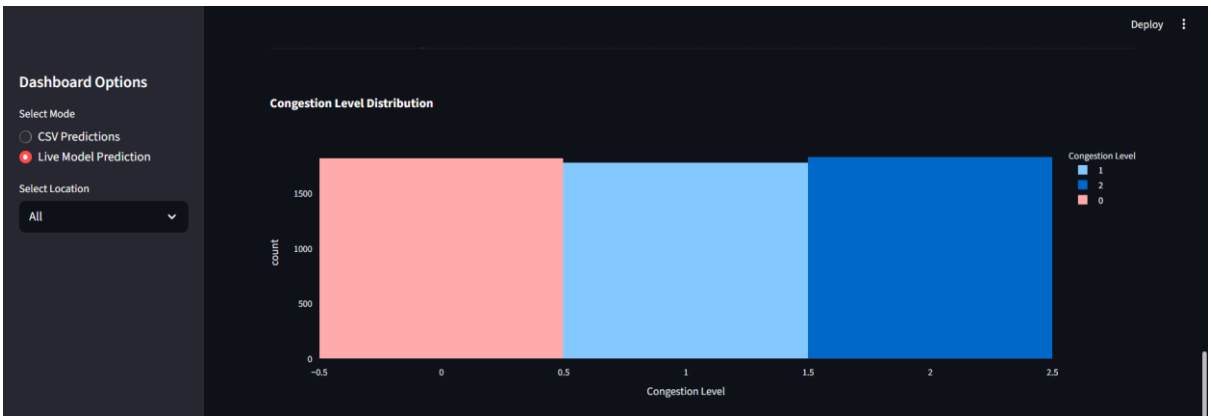






| | tweet_id | emb_0 | emb_1 | emb_2 | emb_3 | emb_4 | emb_5 | emb_6 | emb_7 | emb_8 | emb_9 | emb_10 | emb_11 | emb_12 | emb_13 | emb_14 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1970000000000000000 | -0.0161 | 0.0332 | 0.0154 | -0.0395 | -0.0931 | -0.0259 | -0.0389 | 0.0271 | -0.1021 | -0.0392 | 0.0187 | 0.003 | -0.0102 | -0.0089 | 0.0198 |
| 1 | 1970000000000000000 | 0.0154 | 0.0536 | -0.0057 | 0.0155 | 0.0065 | 0.0399 | 0.0268 | 0.0503 | -0.0642 | 0.0513 | 0.0446 | -0.0075 | -0.0469 | -0.028 | 0.0186 |
| 2 | 1970000000000000000 | -0.0328 | -0.0239 | -0.0113 | -0.0171 | -0.0087 | 0.0687 | -0.0096 | 0.0949 | -0.0938 | -0.0361 | 0.0218 | 0.0096 | 0.078 | -0.0231 | 0.0861 |
| 3 | 1970000000000000000 | 0.0887 | -0.064 | 0.0392 | -0.0416 | -0.0065 | -0.0031 | 0.0062 | 0.1045 | -0.0744 | -0.0356 | -0.0322 | -0.0018 | 0.0191 | 0.0462 | -0.0113 |
| 4 | 1960000000000000000 | -0.0312 | 0.0167 | 0.0285 | 0.0182 | -0.0201 | -0.0017 | 0.0101 | 0.0554 | -0.0501 | 0.0061 | -0.0397 | -0.0482 | -0.0148 | 0.0575 | -0.0394 |
| 5 | 1960000000000000000 | -0.0685 | 0.055 | -0.0459 | 0.0176 | 0.0561 | 0.0535 | 0.0237 | 0.045 | 0.0273 | 0.0613 | 0.0716 | 0.0188 | 0.0052 | -0.0362 | -0.0355 |
| 6 | 1960000000000000000 | 0.0814 | -0.0311 | -0.0161 | 0.0367 | -0.0156 | -0.0021 | -0.0368 | 0.0165 | -0.0617 | 0.0022 | -0.0353 | -0.0273 | -0.0436 | 0.0943 | -0.0197 |


| | tweet_id | emb_0 | emb_1 | emb_2 | emb_3 | emb_4 | emb_5 | emb_6 | emb_7 | emb_8 | emb_9 | emb_10 | emb_11 | emb_12 | emb_13 | emb_14 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1970000000000000000 | -0.0161 | 0.0332 | 0.0154 | -0.0395 | -0.0931 | -0.0259 | -0.0389 | 0.0271 | -0.1021 | -0.0392 | 0.0187 | 0.003 | -0.0102 | -0.0089 | 0.0198 |
| 1 | 1970000000000000000 | 0.0154 | 0.0536 | -0.0057 | 0.0155 | 0.0065 | 0.0399 | 0.0268 | 0.0503 | -0.0642 | 0.0513 | 0.0446 | -0.0075 | -0.0469 | -0.028 | 0.0186 |
| 2 | 1970000000000000000 | -0.0328 | -0.0239 | -0.0113 | -0.0171 | -0.0087 | 0.0687 | -0.0096 | 0.0949 | -0.0938 | -0.0361 | 0.0218 | 0.0096 | 0.078 | -0.0231 | 0.0861 |
| 3 | 1970000000000000000 | 0.0887 | -0.064 | 0.0392 | -0.0416 | -0.0065 | -0.0031 | 0.0062 | 0.1045 | -0.0744 | -0.0356 | -0.0322 | -0.0018 | 0.0191 | 0.0462 | -0.0113 |
| 4 | 1960000000000000000 | -0.0312 | 0.0167 | 0.0285 | 0.0182 | -0.0201 | -0.0017 | 0.0101 | 0.0554 | -0.0501 | 0.0061 | -0.0397 | -0.0482 | -0.0148 | 0.0575 | -0.0394 |
| 5 | 1960000000000000000 | -0.0685 | 0.055 | -0.0459 | 0.0176 | 0.0561 | 0.0535 | 0.0237 | 0.045 | 0.0273 | 0.0613 | 0.0716 | 0.0188 | 0.0052 | -0.0362 | -0.0355 |



## [12]SUMMARY:

The project focuses on traffic flow prediction and congestion analysis using social media data, specifically tweets from the Vellore region. The system collects real-time traffic-related tweets via APIs, preprocesses them by cleaning, filtering for English, and geotagging, and generates embeddings using lightweight LLMs such as BERT or DistilBERT. These embeddings are combined with temporal and location metadata to create structured inputs for predictive models. Machine learning and deep learning models—including LSTM for time-series analysis and XGBoost or Random Forest for structured data—are trained to predict traffic density, congestion levels, and estimated delays.

An interactive GUI has been developed using Streamlit, allowing users to visualize predictions in real-time, view tweet-based traffic alerts, and explore patterns via charts and dashboards. At this stage, the system successfully integrates data collection, preprocessing, feature engineering, and predictive modeling, with real-time visualization providing a clear demonstration of traffic insights. Deliverables completed include the labeled tweet dataset, LLM embedding pipeline, trained models with performance evaluation, and a functional dashboard interface. The project is now ready to proceed to optimization, final integration, testing, and deployment.

## REFERENCES:


[1] Guo, X., Zhang, Q., Jiang, J., Peng, M., Zhu, M., & Yang, H. F. (2024). Towards explainable traffic flow prediction with large language models. *Communications in Transportation Research*, *4*, 100150.

[2] Navarro-Espinoza, A., López-Bonilla, O. R., García-Guerrero, E. E., Tlelo-Cuautle, E., López-Mancilla, D., Hernández-Mejía, C., & Inzunza-González, E. (2022). Traffic flow prediction for smart traffic lights using machine learning algorithms. *Technologies*, *10*(1), 5.

[3] Zhu, L., Yu, F. R., Wang, Y., Ning, B., & Tang, T. (2018). Big data analytics in intelligent transportation systems: A survey. *IEEE transactions on intelligent transportation systems*, *20*(1), 383-398.

[4] Mounica, B., & Lavanya, K. (2022). Real time traffic prediction based on social media text data using deep learning. *Journal of mobile multimedia*, *18*(2), 373-391.

[5] Khajeh Hosseini, M., & Talebpour, A. (2019). Traffic prediction using time-space diagram: a convolutional neural network approach. *Transportation Research Record*, *2673*(7), 425-435

[6] Sayed, S. A., Abdel-Hamid, Y., & Hefny, H. A. (2023). Artificial intelligence-based traffic flow prediction: a comprehensive review. *Journal of Electrical Systems and Information Technology*, *10*(1), 13.

[7] Fan, Y., Yeh, C. C. M., Chen, H., Wang, L., Zhuang, Z., Wang, J., ... & Zhang, W. (2023, September). Spatial-temporal graph sandwich transformer for traffic flow forecasting. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases* (pp. 210-225). Cham: Springer Nature Switzerland.

[8] Yu, B., Yin, H., & Zhu, Z. (2017). Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. *arXiv preprint arXiv:1709.04875*.

[9] Cui, Z., Henrickson, K., Ke, R., & Wang, Y. (2019). Traffic graph convolutional recurrent neural network: A deep learning framework for network-scale traffic learning and forecasting. *IEEE Transactions on Intelligent Transportation Systems*, *21*(11), 4883-4894.

[10] Masri, S., Ashqar, H. I., & Elhenawy, M. (2025). Large language models (llms) as traffic control systems at urban intersections: A new paradigm. *Vehicles*, *7*(1), 11.

[11] Ahmed, S. F., Kuldeep, S. A., Rafa, S. J., Fazal, J., Hoque, M., Liu, G., & Gandomi, A. H. (2024). Enhancement of traffic forecasting through graph neural network-based information fusion techniques. *Information Fusion*, *110*, 102466.

[12] Wang, Y., He, Z., & Hu, J. (2020). Traffic information mining from social media based on the MC-LSTM-Conv model. *IEEE Transactions on Intelligent Transportation Systems*, *23*(2), 1132-1144.

[13] Zhang, M., & Zhao, W. (2025). Traffic Flow Prediction Based on Large Language Models and Future Development Directions. In *ITM Web of Conferences* (Vol. 70, p. 01008). EDP Sciences.

[14] Zhao, Y., Luo, X., Wen, H., Xiao, Z., Ju, W., & Zhang, M. (2024). Embracing large language models in traffic flow forecasting. *arXiv preprint arXiv:2412.12201*.

[15] Melhem, W., Abdi, A., & Meziane, F. (2024, November). Traffic Detection and Forecasting from Social Media Data Using a Deep Learning-Based Model, Linguistic Knowledge, Large Language Models, and Knowledge Graphs. In *16th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management*. 92277.