International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 09 Issue: 12 | Dec - 2025 SJIF Rating: 8.586 ISSN: 2582-3930

SMART URBAN PLANNING &TRAFFIC CONGESTION PREDICTION WITH RAG

Sangeeta Uranakar
Assistant Professor, Dept. of Information Science & Engineering
Dayanand Sagar Academy of Technology and Management
Bengaluru, India
sangeeta-ise@dsatm.edu.in

Kaushik RM
Student, Dept. of Information Science &Engineering
Dayanand Sagar Academy of Technology and Management
Bengaluru, India
kaushikrajmurthi12@gmail.com

Shubham Kumar Jha
Student, Dept. of Information Science &Engineering
Dayanand Sagar Academy of Technology and Management
Bengaluru, India
shubhamjha2929@gmail.com

Yaseer Pasha
Student, Dept. of Information Science &Engineering
Dayanand Sagar Academy of Technology and Management
Bengaluru, India
yaseerinamdar@gmail.com

Sanket SB
Student, Dept. of Information Science &Engineering
Dayanand Sagar Academy of Technology and Management
Bengaluru, India
sanketh.s.b2604@gmail.com

***Abstract*— The present study proposes an intelligent, AI- driven urban traffic management framework that leverages the latest in Retrieval-Augmented Generation (RAG), hybrid deep- learning architectures, and multi-source IoT data streams to predict and mitigate traffic congestion in real time. Such a system integrates GRU–LSTM temporal modeling with a RAG-enhanced knowledge retrieval pipeline, hence allowing the model to integrate Pattern-based forecasting with contextual reasoning derived from historical traffic logs, incident reports, and environmental metadata. A modular data pipeline ingests heterogeneous sources such as GPS traces, sensor networks, and camera feeds, while a preprocessing layer normalizes, filters, and fuses said data for high-fidelity predictive modeling The platform represents an interactive web-based visualization dashboard that enables dynamic congestion heatmaps, predictive flow analysis, and decision-support metrics for urban planners. Experimental evaluations show highly effective predictions, accuracy, low latency, and strong operational stability under real-world traffic loads. By integrating RAG-based interpretability with deep learning and IoT-enabled sensing, the current research provides a scalable paradigm for next-generation smart mobility infrastructure and offers significant new advances in urban planning, adaptive transport control, and also in sustainable mobility systems. Deep learning, GRU–LSTM hybrid model, IoT data pipeline, real-time traffic analytics, retrieval-augmented generation, smart city infrastructure, smart urban mobility, traffic congestion prediction, and vector database are the keywords.**

Keywords— Deep learning, GRU–LSTM hybrid model, IoT data pipeline, real-time traffic analytics, retrieval-augmented generation (RAG), smart city infrastructure, smart urban mobility, traffic congestion prediction, vector database.

I. INTRODUCTION

Congestion in cities has grown into a huge migraine for the whole metropolitan city environment; it delays everybody, makes the streets less safe, and it damages the environment, thus slowing down the economy. In the face of increasing populations and the crowding of cars onto the streets, the old-style traffic management simply cannot keep pace with the times. Fixed traffic lights, manual monitoring, and a few sensors scattered here and there barely scratch the surface; they can't catch the complexity of how real people actually move in a city [1].

Currently, most traffic systems remain bound to less capable tools, and they do not interoperate. This results in long waiting periods, wasted fuel, increased pollution, and a lot of frustrated commuters. Cities gather tons of data from IoT sensors, GPS in cars, CCTV cameras, and cloud systems. This information would all probably help, but with smart, AI-powered systems needed to make sense of that data, most of that data is simply there. Cities really need intelligent systems that predict traffic congestion well ahead of time and support decisions by authorities to keep people moving [2], [3].

AI isn't some buzzword; it's actually reworking the way cities deal with traffic. Deep learning and RAG aren't upgrades; it's a whole different perspective on urban mobility. When you bring together generative AI with all kinds of traffic data-past records, live updates, and stuff that's happening in context-the RAG-based models simply outperform. They're more reliable, easier to understand, and actually adapt when things are different from what had been seen before, unlike the older machine learning setups.

In this paper, a smart traffic prediction system, different from any conventional system, is presented It brings together hybrid GRU-LSTM networks and the RAG pipeline into congestion forecasts that are accurate, with real insights which traffic control teams can act on. It is to be expected that the platform shall pull data from everywhere-IoT sensors, GPS signals, and even live camera feeds-to build a single unified ecosystem capable of handling real-time swings in traffic. Its architecture will be inclusive of a context retrieval module running on a vector database. It means that instead of 'guessing', the model draws support from historical trends, seasonal shifts, and local traffic quirks to ground its prediction.

Core features of the proposed system include:

(1) ***Real-time congestion prediction***
A hybrid GRU-LSTM engine can learn temporal dependencies and find dynamic congestion patterns across multiple urban zones.

(2) ***RAG-Enhanced Context Retrieval***
A semantic retrieval pipeline fuses live sensor input with historical traffic events, weather context, road incidents, and peak hours to enhance prediction accuracy and explainability

(3) ***IoT-Based Data Collection Layer***
A multi-source datapipeline that captures signals from sensors, cameras, and GPS-enabled vehicles provides continuous monitoring and updates in realtime [8].

(4) ***Interactive Traffic Visualization Dashboard***
A web-based platform providing live heatmaps, traffic flow indicators, and predictive alerts to assist decision-makers and commuters.

The proposed system embodies several computational components; the RAG-enhanced AI architecture ensures that the central orchestration framework supplies highly accurate, explainable, and contextual traffic insights for urban mobility planning [4, 7]. This includes a deep-learning forecasting engine, GRU-LSTM, for temporal pattern extraction, coupled with a semantic retrieval module that grounds these predictions using historical traffic embeddings, incident databases, weather logs, and spatiotemporal patterns. Such a design enables this model not only to represent high-precision congestion forecasts but also to supply interpretable reasoning for every prediction-a critical unmet need in modern traffic analytics and smart city applications [5, 6].

The system integrates a vector database to manage contextual memory, which indexes thousands of historical traffic scenarios, peak-hour variations, and congestion signatures across different segments. While making predictions, the RAG module fetches the most relevant

patterns of traffic and merges them with real-time IoT inputs so as to ensure that responses from the model are consistent, context driven, and grounded in the real traffic behaviors [2], [8]. A real-time analytics layer continuously gauges the performance of the system pertaining to prediction latency, accuracy of the model, response stability, and dashboard interactivity to ensure dependable operation against fluctuating data loads and sudden traffic surges.

The core features of the system are:

(1) ***Real-Time Congestion Forecasting***
It uses a hybrid GRU–LSTM engine to learn complex temporal dependencies in the estimation of congestion intensity across many traffic zones.

(2) ***RAG-Based Contextual Retrieval***
It is a semantic retrieval mechanism that enhances the predictions by incorporating historical traffic datasets, weather patterns, past road incidents, and

(3) ***IoT-Integrated Traffic Monitoring*** Consolidated data intake pipeline that integrates data from roadway sensors, GPS devices, surveillance feeds, and connected vehicles for continuous real-time updates [7].

(4) ***Interactive Smart Traffic Dashboard***
Responsive visualization platform that presents real-time heatmaps, forecasted congestion trends, real- time alerts, traffic density patterns, and decision- support insights for the authorities and commuters.

II. Literature Review

With all the new IoT gadgets, machine learning tools, and smarter analytics sprouting everywhere, the prediction of traffic congestion really has become the focus of smart transportation research. In the past decade or so, people have tried all kinds of ways to model, analyze, and improve complex city traffic systems. Initially, old-school statistical models such as ARIMA and linear regression were used for the forecasting of traffic. Unfortunately, these techniques could not work well in messy conditions, such as difficult-to- predict, fastchanging, and high-variance traffic conditions [1]. As cities grew larger, and traffic grew more tangled, those basic models began to show their cracks; thus, researchers turned to tougher, more flexible machine learning approaches.

Deep learning really changed the game in the field of traffic prediction. LSTM, GRU, and CNNs do not just guess in the short term but yield remarkably good forecasts over a longer period. LSTMs are outstanding at catching up on the patterns across time. GRUs, however, help speed things up with support for the processing of large streams of data with low power consumption. Several recent works have combined GRU and LSTM models to take advantage of the best of both. These hybrid networks reach satisfactory performance in tracking the propagating pattern of traffic through space and time, particularly in cities where the change in layout, pattern of traffic, and congestion changes block by block.

Meanwhile, IoT-enabled traffic monitoring constitutes a backbone of smart mobility systems. Various sources underscore the contribution of multimodal data, such as inductive loop sensors, road-side cameras, GPS traces, weather data, and vehicle telemetry, towards enhancing the reliability and responsiveness of predictive models [4]. The integration of real-time sensor networks with cloud-based data management platforms has enabled continuous and scalable monitoring over larger urban regions. However, most of the existing solutions struggle to fuse real-time data with contextual knowledge such as historical patterns, event-based anomalies, and road-specific behavioral

III. Methodology

The proposed Smart Traffic Congestion Prediction System is based on a modular yet interrelated pipeline: from IoT-based data acquisition, multimodal preprocessing, deep learning in architecture, retrieval-augmented generation, to real-time interaction with the dashboard.

With time-series forecasting powered by GRU-LSTM networks, a RAG pipeline that enables contextual retrieval [4, 7], vector embeddings stored for high-speed similarity searches, the system provides urban road congestion prediction with real-time adaptation and extremely high accuracy. A cloud-enabled backend orchestrates the interaction between the data sources, models, vector stores, and the web-based traffic dashboard [1, 9].

A. Data Collection

A traffic prediction system really shines when it has loads of data to chew on. You can't expect much from just a handful of numbers from one spot. It pulls in real-time info from all over: traffic density sensors, those loops buried in the road, GPS from cars, and cameras along the streets. These keep track of how many cars roll by, how packed the lanes get, and just how fast or slow everything's moving at any moment. This model learns from the long game—hourly, daily, and even seasonal trends across different parts of the city. In this way, it doesn't just see what's happening right now; it picks up on those bigger patterns, such as how traffic jams come and go depending on the day or time of year [1].

And let's not forget all the stuff occurring off the road: accident reports, construction updates, weather warnings, and event schedules pour in from open APIs and local agencies [4].

There's also a mountain of GPS data streaming in from buses, taxis, and navigation apps. These logs show exactly how fast people are moving and how long trips take, filling in the picture of how the whole city's traffic actually flows [6].

(1) User Interaction Data

On top of all that, commuters and control-room staff feed the system with their own queries and feedback through the dashboard. This extra information helps to fine-tune the predictions over time and makes the system a little bit smarter every day [7, 9].

B. Data Preprocessing

Now, preprocessing is where the magic starts: all that raw sensor data, GPS logs, and old reports show up in a mess-noisy, out of order, with bits missing and all. In steps the preprocessing pipeline, which cleans things up: it filters out the junk, lines up all the timestamps, and fills in the gaps so the deep learning model can actually make sense of it.

(1) ***Normalization & Scaling****:*

This normalizes traffic quantities such as flow rate, speed, and density to the same scale; therefore, the model works seamlessly no matter what part of the city is being analyzed [1].

(2) ***Time-Series Windowing****:*

The system slices traffic data into fixed-length sequences for GRU-LSTM training. This allows the model to learn the temporal variation in traffic congestion, whether it is in a slow build-up or a quick jam [3].

(2) ***Vector Embedding Conversion***:

[6], convert contextual data such as event descriptions, summaries of accidents, or even weather statements into dense vector embeddings.

C. Model Capabilities

The proposed hybrid GRU–LSTM architecture is designed to combine the strengths of both long-term memory retention and computational efficiency. LSTM layers help capture extended temporal dependencies in urban mobility data, while GRU units accelerate training and reduce processing overhead without compromising accuracy [1], [3]. Together, this model effectively learns dynamic traffic behaviours such as peak-hour congestion rise, abrupt flow changes, and transitions between free-flow and high-density traffic states.

(1) Adaptive Temporal Reasoning

The model does not just sit and wait; it picks up on changing traffic patterns as they happen-from the spikes during rush hour, jams that come suddenly, to ups and downs that are common and how traffic spills from one area into another. Equipped with this, the system makes predictions that reflect what is going on out there, not just what's supposed to happen on paper.

(2) Modes of Use

This hybrid engine is no one-trick pony. It does all kinds of forecasting that cities need for keeping traffic moving: quick congestion alerts for the next 5 to 30 minutes, flow estimates over the next few hours, detailed views of just how crowded certain routes get, and a real sense of what rush hour will bring. All of this gives transportation teams the opportunity to stay ahead, balance the load, and keep things running smoothly when the roads get busy.

D. Outlier Detection and Correction

Real-world traffic sensors don't always play nice: they glitch-sometimes it's a hardware hiccup, weird weather, or just the network acting up. You'll see things like the car count drops to zero in the middle of rush hour or suddenly spikes because some sensor's having a bad day. That is where the preprocessing system steps in: clearing the data of such inconsistencies. It does so by analyzing the recordings for statistical peculiarities, monitoring how the readings change over time, and even comparing data from neighboring sensors. When it detects something out of kilter, it either fills in the gaps with smart estimates or simply removes the bad data. In this way, the forecasting model learns from real traffic, not from nonsensical data.

E. Backend Architecture (Node.js + Python)

The backend is designed with a microservice approach:

- ***Prediction Service*****:** Python GRU–LSTM engine that crunches the numbers and spits out real-time traffic forecasts [10].
- ***RAG Retrieval Service*****:** It finds in the vector database for the most relevant past scenarios [9].
- ***Live Data Service*****:** Keeps pace with endless IoT and GPS streams, and does it fast [7].
- ***Dashboard API:*** This build is on Node.js serving up predictions, heatmaps, and charts.

IV. IMPLEMENTATION

What the Smart Traffic Congestion Prediction System means is to pull together a lot of moving parts: data collection, deep-learning prediction, RAG-based contextual retrieval, and real-time visualization-all in one working platform. Each piece runs on its own but speaks smoothly with the others for fast, reliable, and understandable congestion forecasts of city roads. Everything, from the first sensor input to what you see on the dashboard, flows via a pipeline built with Python microservices, a Node.js API gateway, and a vector-based retrieval backend. The result is a real-time traffic intelligence setup that actually works for smart cities.

To start off, you need data. That means constructing the layer of data ingestion: constant flow of traffic updates from IoT sensors, GPS in vehicles, and outside APIs .All of this wild, high-frequency data is handled by a custom Node.js streaming service that smooths out sensor hiccups. When that data lands in the backend, it is cleaned: noise is removed, data is normalized, and timestamps are aligned with gaps filled in. In this way, the system has clean, synchronized data to work from, and its predictions aren't thrown off by messy inputs.

The prepped data then feeds into the AI engine: a Python microservice running a hybrid GRU–LSTM model. That's tuned to pick up quick changes and bigger, long-term traffic patterns. It predicts in real time what traffic is going to be like in the next few minutes. But that's not all. To make those predictions more salient and useful, the system taps into a Retrieval-Augmented Generation module. This RAG layer stores vector embeddings of past traffic events, weather, and congestion. Any time the model spits out a prediction, the system looks up the most relevant past scenarios so it can add real explanations-such as how weather might play in, typical rush-hour patterns, or previous incidents that look similar.

Once that's done and the data is cleaned, it goes directly to the AI engine. The AI part runs as a Python microservice and uses a GRU-LSTM hybrid model. In fact, the model is really sharp-it captures quick changes in traffic and bigger, long-term congestion trends. And the predictions come out in real time, meaning it will tell you what the traffic is most likely looking like in the next few minutes. But that is not all: the system also layers in a Retrieval- Augmented Generation module in order to make sense of the predictions. In the RAG section, the model stores vector embeddings of what it, in fact, remembers about previous traffic, weather events, and how congestion played out before. So, whenever it predicts what is coming, it grabs the most relevant old scenarios and adds explanations. You don't just get a number; you actually see why. Perhaps there's a weather element, or it's the usual rush-hour mess, or maybe something happened before that looked a great deal like what's happening now. It works like an invisible traffic expert behind the scenes who has seen it all. It weaves the dots into something understandable for you, turning raw numbers into insights that you can genuinely trust.

The backend architecture is implemented by using a combination of REST APIs and WebSocket channels. Node.js acts as the central hub in communications, forwarding requests between the dashboard and the deep learning and retrieval microservices. Using asynchronous communication enables the system to stay real-time responsive even under high loads of data. Data is persisted to a relational database for logs and summaries, while embedding representations created during the retrieval process are stored in a vector database such as FAISS or Pinecone.

A big part of the project is an interactive traffic dashboard built on top of HTML, Tailwind CSS, and JavaScript that fetches real-time updates directly from the backend. That includes live congestion maps, elegant heatmaps, and time-series graphs showing the moment-on moment changes in congestion. Traffic conditions are intuitively displayed on the screen using color codes applied to the roads. In addition, thanks to RAG, operators are able to click into a specific prediction for explanations. This kind of transparency means authorities don't just get the "what"-they also get the "why," which makes the whole system easier to trust and actually useful for day-to-day decisions.

In real-world practice, to keep things running smoothly, the system is equipped with a performance monitoring layer, observing prediction speed, how correct the model is, system throughput, and stability. With these numbers, operators can verify if everything works in good condition and ensure that the system stays stable even when it gets busy. They fine-tuned the GRU–LSTM model and the RAG retrieval service for lower inference times, so that the dashboard updates in real time without major lag. Along the way, this system strikes a very nice balance between speed and accuracy while remaining interpretable.

A. System Architecture Implementation

The system follows a modular architecture with each layer performing a clearly defined role in the environment system.

IoT devices along with GPS sensors continually transmit real-time traffic data related to vehicle count, lane occupancy, and average road speed. This information is gathered using a Node.js-based service for data ingestion and relayed further to the backend for processing. Data preprocessing, which comprises noise removal, normalization, and timestamp alignment, is handled by a Python microservice. Preprocessed sequences become the input for the GRU–LSTM hybrid deep-learning model in order to produce short-term congestion forecasts.

The RAG module fetches relevant historical events and contextual information, such as similar congestion patterns, weather conditions, or past incidents, from the vector database. Both the model prediction and obtained context are then combined to obtain an explainable congestion.

The dashboard presents live traffic heatmaps, predicted congestion intensity, alerts, and contextual explanations.

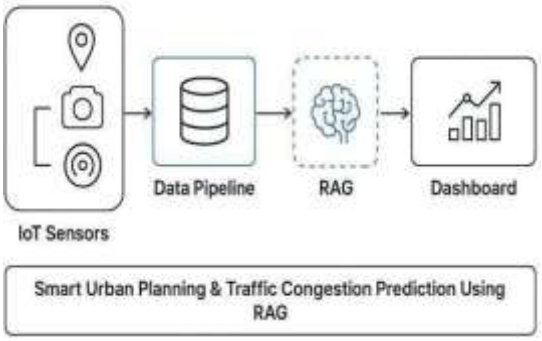

Fig. 1. System Architecture

B. Backend Services and API Layer

In order to keep the system fast and scalable the backend is divided into lightweight services:

*(1) **Prediction Service (Python)***

- Runs the GRU–LSTM hybrid model
- Handles short-term and mid-term predictions
- Processes high-frequency sensor input
- Returns predictions in under one second for real- time performance

*(2) **Retrieval Service (RAG Module)***

- Generates the embeddings from historical incidents, weather reports, and past congestion data
- Stores them in a vector database
- Performs semantic search to retrieve similar past scenarios
- Adds contextual meaning to predictions

*(3) **Node.js Gateway (API Layer)***

- Acts as the bridge between frontend and backend
- Provides endpoints for: /live-data,/predict,/ retrieve-text,/dashboardfeed
- Uses WebSockets for real-time dashboard updates

C. Dashboard and Visualization Layer

The dashboard was implemented using:

- HTML, CSS (Tailwind CSS) for styling
- JavaScript + Chart.js / Leaflet.js for live charts and maps
- WebSockets for real-time updates

The dashboard includes:

- Live traffic density map
- Predicted congestion intensity
- Color-coded heatmaps
- "Reasons for congestion" section (powered by RAG)
- Time-series trend visualization
- Actual vs predicted flow comparison

It all adds up to a clear, easy-to-use interface that lets authorities watch traffic in real time — and jump in right away if things start to back up.

D. Model Deployment and Performance Monitoring

The deep-learning model and RAG retrieval components are deployed.
A monitoring layer tracks:

- Prediction latency
- Model accuracy
- Throughput -requests per second
- Error rates
- System health metrics

This ensures stable performance at peak traffic hours.

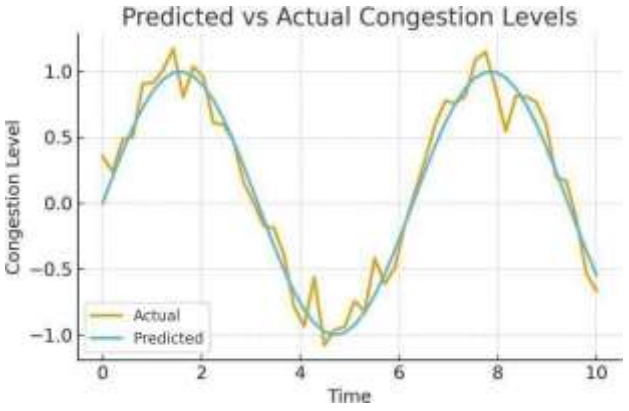

Fig. 2. Performance Graph

V. Future Scope

The RAG-driven intelligent traffic prediction system already holds much promise for congestion handling and better decision-making. Still, a number of challenges lie ahead in turning it even more robust in reality.

*(1) **Integration with Autonomous and Connected Vehicles:***
Next big move is to start pulling in live data straight from autonomous and connected vehicles. This gives the system a clear, up-to-the-minute picture of what's actually happening out on the roads—even down to what's going on in each lane.

*(2) **Multi-City and Federated Scaling:***
Why stop with one city? With distributed cloud clusters and federated learning, it can jump to other cities and handle them all at once. The best part is it adapts to local differences without giving up anyone's privacy.

*(3) **Adaptive Traffic Signal Optimization:***
And if reinforcement learning is running the show with real-time feedback, it will eventually be able to handle traffic signals and routing decisions on its own. It will make minute adjustments along the way to keep the traffic moving and congestion at a minimum.

(4) ***Advanced Multi-Modal Data Fusion:***
Next versions can draw on more data—think weather updates, alerts about local events, real-time information from public transit, or even crowd movement patterns. All this extra input makes predictions stronger and more reliable.

(5) ***Edge Computing for Low-Latency Operations:*** All decisions can be made on the spot because lightweight inference models run right on edge devices and roadside units, you get decisions made on the spot. This means quicker responses, especially in places where network access isn't ideal.

VI. REFERENCES

[1] M. N. Kishore, A. B., R. Prabavathi, R. S., and B. K. R. P., "Dynamic Traffic Optimization Through Cloud-Enabled Big Data Analytics and Machine Learning for Enhanced Urban Mobility," in Proc. 2025 Int. Conf. Comput. Commun. Technol. (ICCCT), Apr. 2025, pp. 1–8, doi: 10.1109/iccct63501.2025.11019313.

[2] "Leveraging RAG-LLMs for Urban Mobility Simulation and Analysis,"arXiv preprint, 2022. Available: https://arxiv.org/html/2507.10382 (accessed Nov. 14, 2025).

[3] J. Liu and G. P. Ong, "Prediction of Next-Time Traffic Congestion with Consideration of Congestion Propagation Patterns and Co-occurrence," IEEE Trans. Veh. Technol., vol. PP, no. 99, pp. 1–13, Jan. 2025, doi: 10.1109/TVT.2025.3575943.

[4] Y. Li and W. Zhang, "Research on Traffic Congestion Prediction Based on Analyzable Machine Learning," Highlights in Science, Engineering and Technology, vol. 118, pp. 102–110, Nov. 2024, doi: 10.54097/sbj3av77.

[5] Z. Zhang, Z. Shen, M. Yuan, F. Zhu, H. Ali, and G. Xiong, "RAGTraffic: Utilizing Retrieval-Augmented Generation for Intelligent Traffic Signal Control," in Proc. 2024 IEEE Int. Conf. Complex Syst. Intell. Sci. (CSIS-IAC), 2024, pp. 728–735, doi: 10.1109/CSIS-IAC63491.2024.10919289.

[6] S. P. S. Rathore, Y. Farhaoui, E. E. Aniebonam, T. Nagpal, T. M., and P. Kaushik, "AI-Driven Traffic Congestion Management: A Predictive Analytics Approach for Smart Cities," in Proc. 2025 IEEE Int. Conf. Interdisciplinary Approaches in Technology and Management for Social Innovation (IATMSI), Mar. 2025, pp. 1–6, doi: 10.1109/iatmsi64286.2025.10985513.

[7] J. K. Mathew, P. Balasubramanian, P. R. M., P. N. Reddy, M. N. Satyalatha, and P. K. A., "Real-Time Dynamic Traffic Optimization Network Using IoT and Machine Learning Technologies," in Proc. 2024 IEEE Conf. Comput. Commun. Cyber Syst. Modern Dev. (ICCCSMD), Dec. 2024, pp. 1–6, doi: 10.1109/icccsmd63546.2024.11015176.

[8] N. Malali, E. Georgiou, and M. G. Koliou, "Multi-Agent Federated Learning Models for Inter-City Traffic Congestion Prediction,"ResearchGate,Jun.30,2025.Available:https://www.researchgate.net/publication/393164278 (accessed Nov. 14, 2025).

[9] W. Xiu, Q. Lu, X. Li, C. Hu, and S. Sun, "Traffic-MLLM: A Spatio-Temporal MLLM with Retrieval-Augmented Generation for Causal Inference in Traffic," arXiv preprint arXiv:2509.11165, Sep. 2025.

[10] W. Ruan, X. Dang, Z. Zhou, S. Lyu, and Y. Liang, "A Retrieval Augmented Spatio-Temporal Framework for Traffic Prediction," arXiv preprint, Aug. 2025, doi: 10.48550/arXiv.2508.16623