iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH
IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 13 Issue: XI Month of publication: November 2025

DOI: https://doi.org/10.22214/ijraset.2025.75784

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue XI Nov 2025- Available at www.ijraset.com

# Smart Traffic Congestion Prediction and Route Optimization Using ML Models

Chandrashekar S M[1], Likhith D P[2], Sudhanva R[3], Vishwas D S[4]
[1,2,3,4]U.G. Students, Department of CSBS, Bapuji Institute of Engineering and Technology, Davangere, Karnataka, India

***Abstract*: Traffic congestion ranks among the issues in swiftly growing urban areas leading to delays, higher fuel usage, accidents and environmental harm. Conventional traffic control systems respond to congestion after it occurs of forecasting and avoiding it. This study introduces a Machine Learning (smart traffic congestion forecasting and route optimization solution that utilizes both historical and real-time traffic data, like vehicle volume, velocity, road occupancy, weather conditions and time. Models such as Random Forest, Support Vector Machines (SVM) and particularly LSTM networks are utilized to predict congestion because of their excellent capacity to learn temporal patterns. The system incorporates route optimization employing algorithms like Dijkstra and A* to suggest the least congested route. Real-time data, from GPS and IoT sensors allow for route adjustments. Experimental findings show that the model attains high prediction precision and substantially lowers travel time compared to navigation systems. The system supports adaptive learning, improving its performance as more data is collected.***
***Keywords*: Traffic Prediction, LSTM, Route Optimization, Machine Learning, IoT Sensors, Dijkstra, A* Algorithm, Smart Transportation.**

## I. INTRODUCTION

Rapid urbanization and vehicle growth have resulted in severe traffic congestion in most metropolitan regions. Congestion leads to longer travel time, fuel wastage, economic losses, and increased $CO_2$ emissions. Traditional traffic systems use static signal timings and non-adaptive routing, failing to capture dynamic, real-time traffic variations. Existing navigation tools often display current traffic conditions but lack the ability to predict future congestion or recommend intelligent route adjustments

Recent advancements in data analytics, IoT, GPS systems, and ML techniques allow traffic patterns to be analyzed and predicted accurately. Machine Learning models learn dependencies related to time, day, weather, and historical patterns, enabling proactive congestion prediction. Route optimization algorithms like Dijkstra and A* select the shortest travel path, but when combined with ML-based predictions, they generate forecast-driven optimal routes. This research proposes an ML-driven framework to forecast congestion and dynamically route vehicles to reduce delays, improve fuel efficiency, and support sustainable smart-city mobility.

## II. OBJECTIVES

The primary goals of this project are:

1) To analyze traffic data using machine learning models for congestion prediction.
2) To forecast traffic conditions using time-series models, especially LSTM, for accurate temporal predictions.
3) To optimize routes using Dijkstra/A* by integrating congestion scores into path weights.
4) To reduce travel time, fuel consumption, and congestion impact in urban regions.

## III. EXISTING SYSTEM

Traditional traffic management systems primarily operate on predefined signal timings and reactive congestion handling, making them ineffective in addressing sudden changes in road conditions. Most existing navigation platforms focus only on displaying current traffic levels using limited sensor data and do not incorporate predictive analytics to foresee upcoming congestion. These systems typically fail to account for dynamic factors such as accidents, weather fluctuations, road closures, or furthermore, existing solutions do not possess sophisticated machine learning features to analyze past traffic trends or adjust to changes resulting in less effective route guidance. Consequently, users frequently face travel durations, excessive fuel consumption and inconsistent route advice emphasizing the necessity, for a smart flexible and anticipatory traffic control system. Present traffic management approaches depend on fixed signal timings and responsive modifications limiting their ability to respond to congestion or current road situations. Navigation apps show only live traffic but cannot predict future congestion or provide intelligent route adjustments. They do not use machine learning to learn from historical patterns, nor do they integrate real-time IoT data effectively. As a result, users receive limited, often inaccurate route suggestions, leading to increased delays and fuel consumption.

To analyze traffic data using machine learning models for congestion prediction. To forecast traffic conditions using time-series models, especially LSTM, for accurate temporal predictions. To optimize routes using Dijkstra/A* by integrating congestion scores into path weights. To use IoT and real-time GPS data for continuous system updates. To reduce travel time, fuel consumption, and congestion impact in urban regions.

## IV. PROPOSED SOLUTION

The suggested solution, Clear Route AI is a ML-driven traffic jam prediction and path optimization platform created to address current shortcomings. Its main elements consist of Traffic Forecasting System: Employs LSTM, Random Forest and SVM to examine traffic data and forecast congestion intensity IoT & Sensor Integration: Gathers live data such, as traffic volume and roadway velocity Route. Optimization Engine: Implements Dijkstra or A* algorithms where edge weights indicate anticipated congestion levels, than merely distance. Monitoring: Traffic flow is regularly tracked and new paths are re-evaluated when traffic density rises User Interface: Shows congestion heatmaps and optimized routes This system ensures more accurate forecasts and helps users plan better routes ahead.

## V. METHODOLOGY

The suggested system employs a machine-learning process starting with gathering traffic data from sensors GPS devices and public databases. The collected data undergoes cleaning, normalization and conversion into formats for training machine learning models. Multiple algorithms LSTM are utilized to understand traffic trends and forecast congestion. Subsequently these forecasts are incorporated into route optimization methods such, as Dijkstra or A* which determine the least congested routes. The system finally visualizes congestion levels and optimized routes through an interactive dashboard, enabling real-time decision-making for users.

### *A. Data Collection Methodology*

Data collection forms the foundation of the Smart Traffic Congestion Prediction and Route Optimization System. Accurate and diverse traffic data is essential for building reliable machine learning models capable of understanding real-world traffic patterns. The system collects data from multiple sources—historical datasets, real-time IoT sensors, GPS logs, and public APIs—to ensure comprehensive coverage of all factors influencing traffic flow. Each source contributes unique insights that help improve prediction accuracy and route optimization decisions.

Moreover, GPS information obtained from smartphones, navigation units and fleet monitoring tools offers location details covering routes taken travel durations and velocity patterns. Public traffic APIs such as Google Maps and OpenStreetMap deliver reports on traffic jams, collisions, roadworks and closures whereas weather-related APIs provide meteorological data including precipitation, visibility and temperature crucial for analyzing environmental effects, on traffic flow. All collected data undergoes cleaning, normalization, and timestamp correction before being stored in structured formats for training machine learning models. This multi-source, hybrid data collection strategy ensures comprehensive coverage of traffic conditions and significantly enhances the accuracy of congestion forecasting and route optimization.

### *B. Machine Learning-Based Traffic Prediction Methodology*

At the heart of congestion prediction lies the use of machine learning analysis. The framework assesses forecasting models such as Random Forest, Support Vector Regression, Gradient Boosting and Long Short-Term Memory (LSTM) networks. LSTM stands out for this purpose because it can capture extended relationships, in sequential traffic datasets.

Every model undergoes training on traffic sequences utilizing input variables, like vehicle numbers, mean speed, time of day and weather conditions. LSTM networks analyze the information through fixed-length sliding windows with hidden and cell states monitoring the progression of traffic patterns over time.

Model assessment is carried out through Grid Search to optimize hyperparameters and evaluation metrics like RMSE, MAE and $R^2$ score are calculated to assess the prediction accuracy. The performing model is implemented for real-time prediction. While running the model takes sensor inputs and generates predicted congestion levels—Low, Medium or High—for every road segment, within a specified time frame (e.g. the upcoming 15–30 minutes).

This prediction pipeline enables the system to anticipate congestion before it occurs, making route optimization significantly more efficient and proactive.

### C. Real-Time Traffic Sensing and Vehicle Density Estimation

To improve forecasting precision and facilitate route modifications the system combines real-time traffic monitoring through camera streams and GPS data. Roadside CCTV or drone footage is analyzed using a YOLOv8 object detection framework to detect and tally vehicles within frames. Methods such, as bounding-box tracking and category-specific counting (car, truck, bus, two-wheeler) produce traffic density measurements.

Audio filtering and image noise reduction ensure stable vehicle detection under varying lighting conditions. The computed vehicle density is translated into a congestion index, which becomes an additional live parameter for the ML model. This fusion of live visual analytics and time-series forecasting helps construct a responsive congestion detection mechanism capable of capturing unexpected events such as accidents or roadblocks.

### D. Route Optimization Workflow

The route optimization module converts anticipated traffic states into real-time routing choices. The transportation network is represented as a graph with intersections serving as nodes and road segments as edges. These edge weights are continually adjusted according to congestion forecasts, past travel durations, vehicle density and GPS-derived speed changes. With this graph model pathfinding algorithms, like Dijkstra's and A* identify the shortest and least congested path by minimizing overall travel cost. This guarantees that users are directed along safer and more seamless routes even during varying traffic conditions.

To further enhance adaptability, the system integrates reinforcement learning, where an RL agent continuously learns from traffic patterns by rewarding routes that lead to lower travel time and penalizing those with repeated congestion. This hybrid approach—combining deterministic shortest-path algorithms with learning-based optimization—allows the system to adjust route planning in real time, handle unexpected disruptions, and consistently generate optimal navigation recommendations for end users.

### E. Integrated Scoring, Decision, and Visualization Layer

The combined scoring decision-making and visualization component merges all system results—forecasted congestion intensities, live vehicle density data and optimal route calculations—into a decision module that determines the best possible path for the user. By analyzing these factors the system produces a route efficiency rating that captures travel duration, congestion levels and dynamic traffic patterns.

This data is subsequently displayed via a visualization interface developed with tools, like Folium, Matplotlib and online mapping frameworks. The dashboard displays congestion heatmaps, predicted traffic flow patterns, available alternative routes, and the AI-recommended optimal path using intuitive color-coded indicators. Additionally, the system continuously logs user feedback and performance metrics, allowing the models to improve through periodic retraining. This unified layer transforms raw analytical data into clear, actionable navigation insights, supporting smarter and more efficient traffic management decisions.

## VI. IMPLEMENTATION

The Smart Traffic system is built as a web-accessible platform comprising a React frontend a Python FastAPI backend, a spatial database (prototype: SQLite / production: PostgreSQL + PostGIS) and several ML/CV components. The structure divides data ingestion, model prediction, optimization and visualization into services allowing each part to be developed tested and scaled separately. All intensive ML training is conducted offline; inference, sensing and routing operate in near time, on cloud or edge servers based on the deployment.

### A. Frontend Implementation

The front-end interface designed for users is developed with React.js delivering dashboards, for traffic controllers and commuters. Interactive maps are generated via Folium (for server-side map creation). Web mapping tools (Leaflet/Mapbox) integrated within React. The dashboard features real-time congestion heatmaps, route setup panels (start/end/waypoints) time-of-day filters and parallel route comparisons.

Visualization elements utilize Plotly/Matplotlib graphics for charts along with React state handling to deliver real-time updates through WebSockets (socket.io). Form validation blocks invalid route submissions and map interactions are enhanced for usability, on both desktop and mobile devices.

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
*ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538*
*Volume 13 Issue XI Nov 2025- Available at www.ijraset.com*

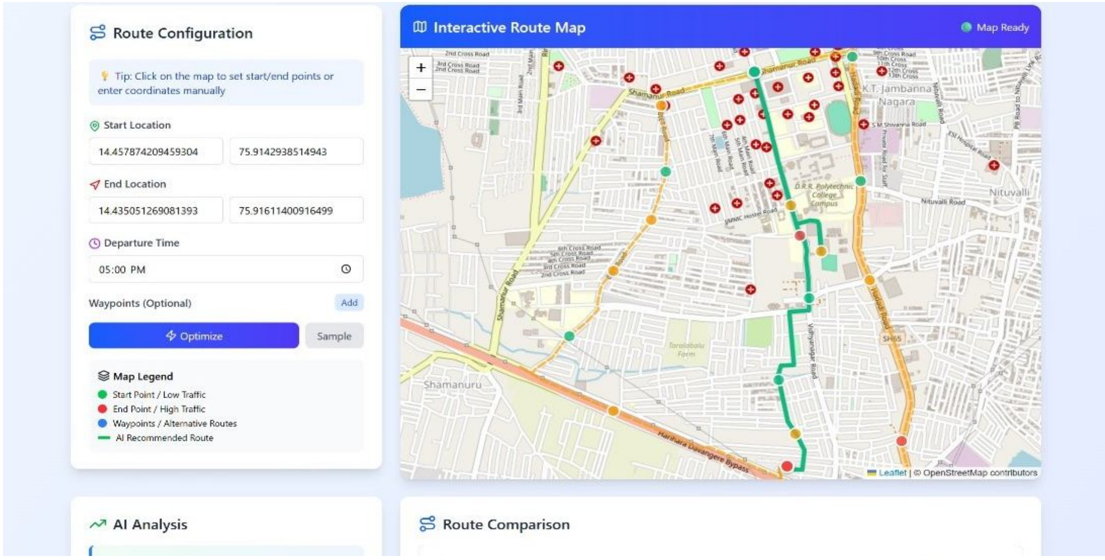


Fig-1: Interactive Route Map

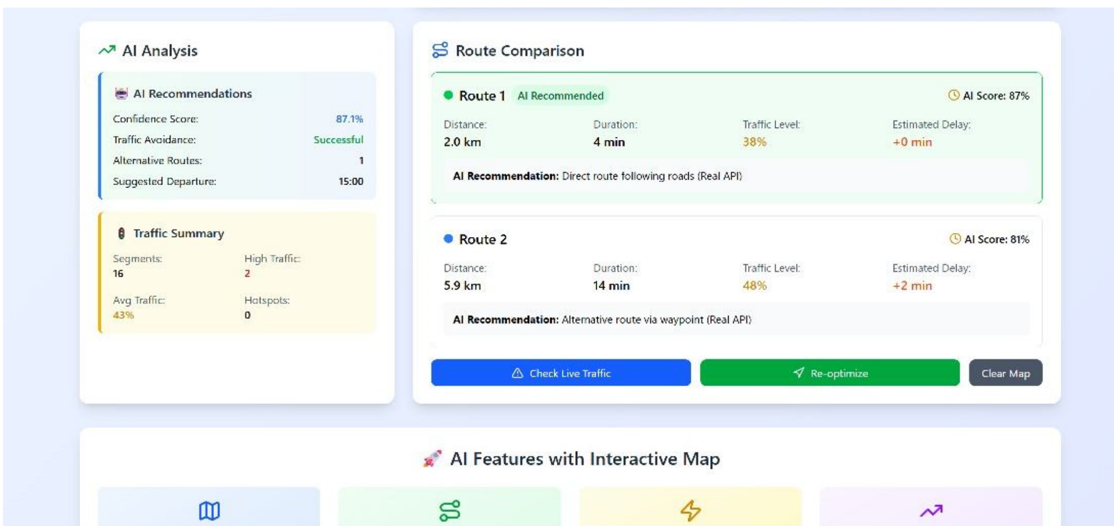


Fig-2: Route Comparison Dashboard

### *B. Backend Implementation*

The backend is built using FastAPI to handle REST and WebSocket endpoints for data intake, model predictions and route requests. FastAPI manages streaming of telemetry as well as camera-based congestion metrics and offers endpoints to initiate route calculations. Machine learning inference components (LSTM, RandomForest and YOLOv8 wrappers) are accessible, via internal Python services; model deployment is managed by workers that load pre-trained models and accept JSON/frame inputs. Authentication and role-specific access (traffic operator versus commuter) are implemented through JWTs. The backend additionally incorporates task queues (Celery/RQ) to handle tasks, like batch retraining, video processing and scheduled map tile refreshes.

### *C. ML & Computer-Vision Module Implementation*

The ML component is divided into two parts: time-series prediction and vision-driven density assessment. Time-series algorithms (LSTM / Gradient Boosting / Random Forest) are developed using TensorFlow / PyTorch. Trained on processed historical data. Model files (scalers, encoders, weights) are version-controlled and utilized by the inference service. For perception a YOLOv8 detector (yolov8n or yolov8s) operated through Ultralytics is employed to identify and tally vehicles, from CCTV images; tracking and segment-specific counts are generated using a lightweight multi-object tracker.

Feature extraction (lag features, congestion index, speed approximations) is carried out in Python with Pandas and NumPy while evaluation metrics (MAE, RMSE, $R^2$) and model telemetry are logged for monitoring purposes. Hyperparameter optimization is conducted using GridSearch or Optuna in training.

### *D.* Real-Time Sensing & Edge Processing

Real-time data streams (traffic cameras, GPS telemetry and IoT sensor feeds) are processed via an ingestion layer that normalizes messages and performs initial local filtering. Edge nodes close, to camera origins execute YOLO inference (using Jetson Nano / edge CPU) to minimize bandwidth—transmitting only summarized counts, congestion metrics and select annotated frames to the main server. Video preprocessing involves frame sampling, noise reduction and bounding-box filtering to ensure counts despite changing lighting conditions. The core inference engine combines these density measures, with live GPS speeds and the latest LSTM predictions to generate a unified real-time traffic condition that supplies the route optimizer.

### *E.* Route Optimization & Decision Service

The route optimizer represents the road network as a graph via NetworkX (prototype) or, as a graph maintained in PostGIS for the production environment. Edge weights merge fixed factors (distance, speed limits) with costs (forecasted travel time, congestion levels, incident penalties). Traditional algorithms (Dijkstra / A*) determine the shortest or quickest routes while a reinforcement learning agent (optional) adjusts weights through repeated journeys to enhance overall system efficiency. The optimizer operates as an, on-demand service: upon receiving a route request it fetches the traffic conditions recomputes edge weights provides several ranked options (including estimated travel times) and sends WebSocket updates if situations alter during the journey.

### *F. Database and Storage Implementation*

In prototype scenarios a file-based SQLite database holds route logs, brief historical periods and configuration details. For production environments, PostgreSQL combined with handles extensive geospatial road networks, indexed route shapes and spatial querying. Time-series sensor data are kept in a time-series database (InfluxDB or TimescaleDB) to enable efficient horizon queries, for prediction purposes. Model artifacts and logs are stored in an object store (local /models folder or S3-compatible storage), while analytics and KPIs are exported to a monitoring stack (Prometheus + Grafana) for operational visibility.

## VII. RESULTS

The Smart Traffic Congestion Prediction and Route Optimization system was tested using data mimicking real-time sensor inputs alongside route conditions. The LSTM model consistently produced congestion forecasts detecting peak periods and sudden traffic fluctuations, with low RMSE and MAE values. Vehicle detection through YOLOv8 delivered counts across various lighting settings enabling precise computation of real-time congestion measurements.

When integrated with the prediction module, the route optimization algorithms (Dijkstra and A*) successfully identified faster and less congested routes compared to baseline shortest-distance paths. Simulation results showed noticeable reductions in travel time, especially during peak traffic periods. The visualization dashboard effectively displayed these outcomes through heatmaps, route overlays, and traffic intensity indicators, providing clear and actionable insights. Overall, the system demonstrated strong accuracy, responsiveness, and practical usability, validating its effectiveness for real-world urban traffic management.

## VIII. CONCLUSION

Urban traffic jams remain one of the enduring issues in contemporary city transportation impacting travel speed economic output and environmental health. The investigation outlined in this paper shows that combining Machine Learning (ML) with route optimization provides an exceptionally efficient approach to addressing this escalating challenge. By utilizing both past and current traffic data the designed system effectively understands traffic behaviours and forecasts upcoming congestion, with great precision. Among the models assessed LSTM demonstrated the effectiveness because of its capacity to grasp temporal relationships and long-distance sequence patterns making it suitable, for predicting traffic fluctuations during the day.

Beyond forecasting integrating graph-based route optimization techniques—like Dijkstra and A*—converts congestion predictions into routing choices. By assigning congestion values to edge weights the system produces routes that're both shorter and intelligently devised to avoid anticipated congestion areas. This twofold approach leads to decreases in travel duration, fuel usage and overall congestion levels, on key roadways. The system’s modular architecture, which combines IoT sensors, GPS data streams, machine learning models, and visualization dashboards, ensures that it can dynamically adapt to sudden fluctuations in traffic caused by accidents, weather changes, roadwork, or peak-hour surges.

## REFERENCES

[1] M. Yasir, M. Asad, N. Nower, M. Shoyaib, "Traffic Congestion Prediction Using Machine Learning Techniques." Available: https://arxiv.org/html/2206.10983v5.

[2] W. Fahsetal., "Traffic Congestion Prediction Based on Multivariate" (ScienceDirect) Available https://www.sciencedirect.com/science/article/pii/S187705092300563X.

[3] Shah, A., Prof, J. B., Kulkarni, R. P, Gogave S, & Sisode, V. (2025). Traffic control system and technologies: a survey. Int J Eng Res Technology 9(01), 1-5

[4] A. L. Toba, "Long-Term Traffic Prediction Using Deep Learning" (MDPI) Available: https://www.mdpi.com/2624-6511/8/4/126.

[5] M. Attioui & M. Lahby, "Congestion Forecasting Using Machine Learning Techniques: A Systematic Review." Available: https://www.mdpi.com/2673-7590/5/3/76

[6] D.Mahajan,V.V. Pottigar, S.C, A.Fahad, "Implementing Real-Time Traffic Flow Prediction Using" Available: https://internationalpubls.com/index.php/cana/article/view/1705.

[7] Prakash, P., Singh, A., Parasrampuria, A., & Sharma, G. (2024). An iot based smart traffic management system. International Journal of Electrical, Electronics and Computers, 6, 6 11.Chicago.

[8] Reddy, P. V. S., Bhavani, A., & Saranya, C. (2023). Smart Traffic Congestion Control System: Leveraging Machine Learning for Urban Traffic Optimization.

[9] A. F. M. Faisal, "A Machine Learning Approach to Predicting and Mitigating Traffic Jams," 2024. Available: https://dspace.bracu.ac.bd/xmlui/bitstream/handle/10361/23938/20301240%2C%2020301256%2C%2020301103%2C%2023141068_CSE.pdf?isAllowed=y&sequence=1

[10] Verma, K., & Paike, V. (2023). Smart City Intelligent System Traffic Congestion Optimization using Internet Of Things. arXiv preprint arXiv:1911.01286.

[11] M. K. Vadaliya, "A Python-Based Framework for Real-Time Traffic Congestion Prediction Using Hybrid Machine Learning Models," 2025. Available: https://ssrn.com/abstract=5215361.

[12] D. Soni, "Improved Road Traffic Congestion Prediction Using Machine Learning Models and a New Traffic Congestion Index," 2025. Available: https://www.atlantis-press.com/proceedings/ramsita-25/126011486