© SEP 2025 | IRE Journals | Volume 9 Issue 3 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I3-1710940-1524

Traffic Control and Management System Using Deep Learning

ANJANA NAIR[1], TANVIR HUSSAIN[2], ANURAG HANS[3], JINAL PATEL[4], BHARAT TANK[5]
[1, 2, 3, 4, 5]*Computer Science Engineering Parul University*

Abstract—Traffic Control and Management System using Deep Learning is an Internet-of-Things (IoT)-based solution designed for the real-time optimization of urban traffic flow. The system combines IoT vehicle detection with infrared (IR) sensors and ESP32 microcontrollers that feed real-time traffic data through MQTT to a centralized traffic optimizer. Two deep-learning models, a lane classifier and a green-time regressor both implemented using CNN, compute the dynamic decision on which lane will receive the green signal and its optimum green-time duration. We used an HTML-based sensor simulator developed in Python for the training and validation of the models with realistic traffic scenarios. We developed an interactive web- based dashboard using HTML, CSS, and JavaScript to visualize the traffic status, signal timings, and performance evaluation. Experimental results reveal an increase in traffic flow following reduced waiting times, enhanced lane fairness, and an overall traffic efficiency increase, representing an intelligent analytics solution for Smart City traffic management with the potential for large-scale deployments.

Index Terms—Internet of Things (IoT), Deep learning (DL), ESP32, Message Queuing Telemetry Transport(MQTT), Sensor simulator, Lane classification, Green-time regression, Smart city.

I. INTRODUCTION

Urban traffic has long been a problem in modern metropolises, causing travel delays, wasted fuel, and emissions in the atmosphere. Conventional traffic control systems rely on fixed signal timings that fail to adapt to real-time variations in traffic demand. This often leads to problems such as unnecessary delays and lane starvation [1], [4], [11].

Recent advances in Internet of Things (IoT) devices, microcontrollers, and deep learning present possible solutions to these challenges. Infrared (IR) sensors can detect the presence of vehicles in real time, and microcontrollers, such as the ESP32, provide low-latency communication through popular protocols such as MQTT [5], [6]. Deep learning networks can estimate lane priority and the duration of the green light by learning the present traffic conditions [7]–[11]. Most of these systems are still isolated, requiring manual data entry or running only in simulated environments, disconnected from live operations [12]–[15].

To address these challenges, we introduce the Traffic Control Management System using Deep Learning, an integrated framework that combines IoT sensors, real-time deep learning optimization, and real-time visualization. A Python-based sensor simulator generates realistic traffic scenarios using virtual vehicles, creating datasets for training and validating the deep learning models. In parallel, physical IR sensors connected to ESP32 microcontrollers are used in a hardware setup where vehicle presence is emulated via hand gestures, enabling real-time system validation. A Python based traffic optimizer file uses two deep learning models, the green- time regressor and lane classifier, to decide which traffic lane should receive the green light and for how long. To facilitate the training and testing of the models, a sensor simulator generates realistic traffic scenarios, such as rush hours, normal flow, and low traffic flow. All real-time processed information and performance indicators are visualized in a web-based dashboard developed using HTML, CSS, and JavaScript, enabling traffic authorities to supervise and control intersections efficiently [16], [17].

A. *Objectives*

1) We use ESP32 microcontrollers and IR sensors for designing Traffic control system.
2) For optimizing lane priority and green-light timing we use deep learning models [7]–[11].
3) Develop a Python traffic simulator for data generation for training and testing.
4) Add MQTT real-time communication between the sensors and the optimizer.
5) Develop a web-based traffic signal dashboard with real- time data on traffic lights, active lanes, and traffic flow [16], [17].
6) Analyze the potential of the system to decrease delays, balance lanes, and enhance overall traffic conditions.

B. *Contributions*

1) Integrating IoT-based vehicle detection with deep learning for intelligent traffic signal organization.
2) Design of a lane classifier and green-time regressor for dynamically adjusting traffic flow [7]–[9].
3) Sensor simulator for generating realistic traffic data for model training.
4) Development of a web dashboard to enable real-time traffic status visualization.
5) Providing a scalable architecture applicable for future smart city traffic management applications.

II. LITERATURE SURVEY

The study of urban traffic management has increasingly focused on intelligent and adaptive solutions to alleviate congestion and optimize traffic flow [1]. Although traditional fixed- timing systems have been widely analyzed, recent research has emphasized the integration of IoT sensors, microcontrollers, and deep learning models to dynamically manage signal timings [4], [11]. This section reviews the key approaches and technologies proposed in the literature, highlighting their contributions, limitations, and areas for further development.

A. *Advances in IoT and Sensor-based Traffic Monitoring*

Low-cost, real-time traffic detection and control is possible using inexpensive IoT hardware and sensors such as infrared (IR) sensors and ESP32 microcontrollers. While the IoT-based adaptive traffic management framework has been introduced in prior work, it was only tested in simulation, leaving open the question of whether it could be deployed in real inter- sections. Gupta and Verma [5] proposed integrating AI with autonomous vehicles for traffic optimization; however, their approach did not include low-cost IoT hardware such as the ESP32. Hence, although it is evident that IoT devices have broad applicability, the above-mentioned works are mostly simulation-based or not yet deployed on hardware.

B. *Deep Learning for Traffic Flow Optimization*

Optimizing traffic signals by predicting green-light durations and prioritizing lanes is a promising application of deep learning. Ahmed and Rahman [6] presented an AI system to alleviate congestion at busy intersections; however, their work focused only on isolated intersection scenarios, limiting its potential for wider scalability. Similarly, Lee and Kim [4] applied deep learning models to reduce urban congestion, but their approach primarily relied on prediction without real-time sensor integration. These studies demonstrate the potential of machine learning in traffic management while highlighting the gap between algorithmic design and practical hardware implementation.

C. *Research Gaps and Challenges*

Despite progress, several gaps remain:

- Real-time integration: Many systems do not effectively connect live sensor data with predictive models, limiting real-world optimization [7].
- Simulation-only focus: Many frameworks are tested only in simulations, with little or no hardware validation.
- Scalability: Few solutions address multiple intersections or varying traffic conditions, leaving system robustness an open challenge [6].

D. *Positioning of Traffic Control and Management System Using Deep Learning*

To bridge these gaps, we introduce the Traffic Control and Management System Using Deep Learning which integrates both simulation and hardware validation [5], [6]. It not only uses a Python-based sensor simulator to generate realistic virtual traffic scenarios for model training and testing, but also employs IR sensors with hand gestures connected to ESP32 microcontrollers with MQTT communication for real-time, hardware-based validation. The deep learning models, comprising a lane classifier and green-time regressor, dynamically control traffic signals based on real-time sensor input [4], [6]. The system includes two main components: a traffic simulator that generates diverse traffic scenarios to train and evaluate the models, and a web-based dashboard for real-time visualization and manual control [7]. By combining IoT sensing, edge computing, and AI-based optimization, this system bridges the gap between state-of-the-art research prototypes and deployable smart traffic management solutions.

III. METHODOLOGY

This chapter outlines the methodology for the Traffic Control and Management System Using Deep Learning. The system emphasizes real-time traffic monitoring, adaptive signal optimization, and interactive visualization. It integrates IoT sensors, ESP32 microcontrollers, MQTT communication, and deep learning models for lane classification and green-light prediction [1], [2].

A. Requirements Gathering

Requirements are categorized into three main perspectives: Users, System, and Admin [3], [4].

1. User Needs
   - Real-time updates on traffic conditions at intersections.
   - Fast response to traffic signal changes.
   - Visual display of lane traffic and signal timing.
   - Alerts for congestion and optimized route suggestions.
   - Simple and user-friendly dashboard for monitoring purposes.

2. System Requirements
   - IR sensors in each lane for vehicle detection, with vehicle presence emulated via hand gestures for hardware vali- dation.
   - ESP32 microcontrollers for low-latency data transmission using MQTT.
   - Deep learning models for lane prioritization and green- light timing.
   - Python-based traffic simulator to create realistic training and testing scenarios.
   - Web-based dashboard for traffic signals, lane usage, and performance metrics.

3. Admin Tasks
   - Authentication and access control.
   - Monitoring model performance and data integrity.
   - Updating optimization algorithms and simulation scenarios.

B. Technology Stack

- Backend: Python and Flask for traffic optimization, MQTT communication, and simulation services [5], [6].
- Frontend: HTML, CSS, and JavaScript for an interactive dashboard.
- Data Storage: MySQL/PostgreSQL for logs and CSV/JSON for simulation data.
- IoT Components: IR sensors for vehicle detection and ESP32 microcontrollers for sending data via MQTT.
- Deep Learning Models: Lane classifier and green-time regressor for adaptive signal control [11], [12].
- Visualization: Chart.js for real-time traffic metrics and lane usage data.

C. System Design

- Data Input and Processing: IR sensors in each lane detect vehicles in real time. ESP32 microcontrollers transmit this data to the backend via MQTT, ensuring low-latency and reliable communication. The system pre- processes the data by filtering noise, handling missing readings, and aggregating vehicle counts for each lane .In parallel, a Python-based traffic simulator generates virtual traffic scenarios for model training and validation [5], [6].
- Traffic Optimization: Deep learning models, including a lane classifier and green-time regressor, determine the lane that should receive the green signal. They also predict the optimal green-light duration. These models adapt continuously by learning from real-time data and simulated scenarios, improving traffic flow and reducing waiting times [11], [12].
- Simulation Support: A Python-based traffic simulator generates varied scenarios, such as rush hours, normal traffic flow, and low traffic conditions. Simulated datasets are used to train and validate the deep learning models, ensures their robustness before deployment in live inter- sections [16], [17].
- Dashboard Visualization: The web-based interactive dashboard displays lane traffic, signal status, and key performance metrics, including average waiting time, lane fairness. Alerts for congestion and visualizations like graphs help authorities quickly understand and manage traffic conditions [9], [10].
- Communication and Coordination: MQTT enables efficient communication between multiple ESP32 microcontrollers and the backend, allowing the system to coordinate signals across several intersections, thus optimizing overall traffic flow [7], [8].
- Security and Data Integrity: Authentication, encryption, and audit logs safeguard the system against unauthorized access, maintain data integrity, and ensure ac- countability in traffic signal management [16], [17].

D. Rationale and Implementation

The system design emphasizes real-time traffic

© SEP 2025 | IRE Journals | Volume 9 Issue 3 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I3-1710940-1524

management rather than complex model architectures. IoT sensors combined with ESP32 microcontrollers ensure accurate and low-latency data transmissions. Deep learning models provide actionable and interpretable outputs that enable the adaptive control of traffic signals. The Python-based simulator allows safe model training and testing without disrupting live inter- sections [13], [14].

Furthermore, the modular design of the system supports scalability and flexibility. Components can be independently updated, such as retraining models with new traffic data or integrating additional scenarios. The dashboard serves as a centralized interface for traffic authorities, simplifying monitoring and control while maintaining system transparency [4], [15].

IV.SYSTEM ARCHITECTURE

A. Use Case Analysis

There are three main stakeholders in the system:

- Users (commuters and drivers): Benefit indirectly from reduced waiting times, smoother traffic flows, and optimized signal timings.
- System: Collects real-time data from IoT sensors, processes it using deep learning models, and generates adaptive signal-control decisions.
- Administrators (traffic authorities): Monitor the dash- board, track performance metrics, and intervene in exceptional circumstances when required.

This setup establishes a complete cycle of sensing, processing, decision-making, and monitoring to ensure efficient traffic management [16], [17].

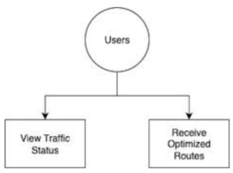

Fig. 1. Use Case Diagram showing interactions among Users.

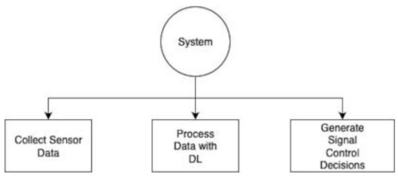

Fig. 2. Use Case Diagram showing interactions among System.

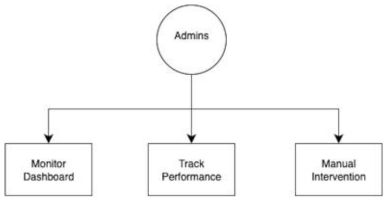

Fig. 3. Use Case Diagram showing interactions among Administrators.

B. Data Flow Diagram (DFD)

The logical flow of the data begins with the IR sensors detecting vehicles in each lane. These readings are transmitted by ESP32 microcontrollers via the MQTT protocol to the backend. The backend preprocesses the data, applies deep learning models (lane classifier and green-time regressor), and generates optimized signal timings. Finally, the processed out- puts are displayed on the dashboard, where administrators can review the traffic density, system decisions, and performance metrics in real time.

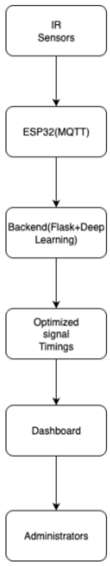

Fig. 4. Data Flow Diagram (DFD) illustrating the flow of traffic data.

C. System Layers

The architecture is modular and divided into four layers:

- IoT Layer: Vehicle presence is simulated using hand gestures over infrared (IR) sensors connected to ESP32 microcontrollers or simulated usind python simulator.
- Communication Layer: The MQTT protocol ensures low-latency, reliable data transfer between intersections and the backend.
- Processing Layer: A Flask-based backend preprocesses the data, applies the deep learning models, and determines the signal-control

© SEP 2025 | IRE Journals | Volume 9 Issue 3 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I3-1710940-1524

decisions.

- Application Layer: A web-based dashboard (HTML, CSS, JavaScript) visualizes the traffic flow, signal timings, and system alerts.

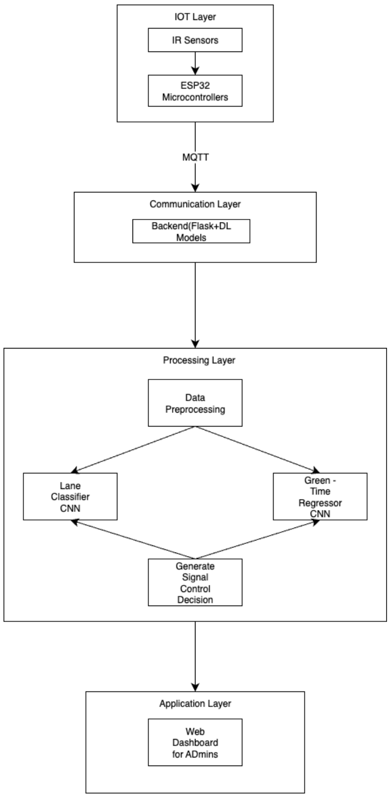

Fig. 5. System Architecture Layer.

D. AI Decision Pipeline

The decision-making pipeline operates as follows:

1) Data Collection: Vehicle counts is captured by IR sensors or through python based simulator
2) Preprocessing: Noise is filtered, missing data are handled, and inputs are aggregated for each lane.
3) Lane Classification: The lane classifier predicts which lane should receive the next green signal [11].
4) Green-Time Regression: The regressor estimates the optimal green-light duration for that lane [12].
5) Signal Control: The decision is executed in real time and visualized on the dashboard by administrators.

E. Key Features

- Scalability: Supports the simultaneous management of multiple intersections.
- Adaptability: Deep learning models are continuously updated with real-time and simulated traffic data.
- Transparency: The dashboard ensures clear visualization and auditability of traffic decisions.
- Security: Authentication, encryption, and audit logging preserve the reliability and integrity of the system.

V.SIMULATION

This project includes a simulation module to allow testing the adaptive traffic control strategies in a controlled environment [1], [2]. It can evaluate the lane classification and green-time regression (implemented with CNN) deep learning models across traffic situations [3], [4]. It produces authentic vehicle flow data without affecting real intersections [5], [6]. The scenario is also a dataset for model training and testing so that the system can adapt to diverse traffic situations [7], [8].

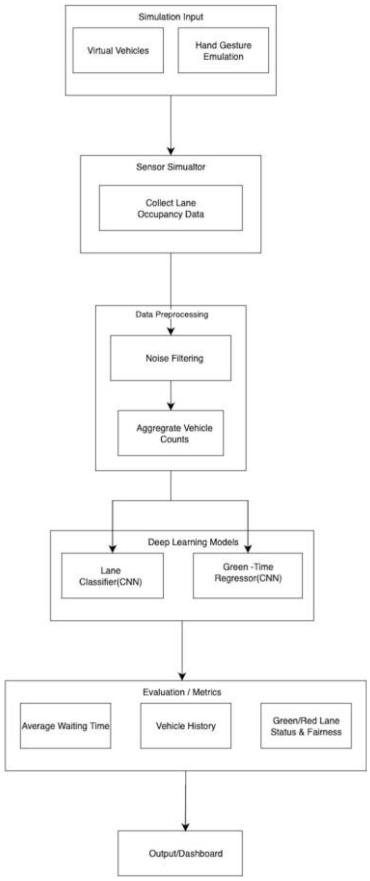

Fig. 6. Simulation Framework showing how traffic scenarios are generated, processed by models, and evaluated.

A. Simulation Framework

- A Python-based Framework reflection of traffic on a multi-lane intersection [9], [10].
- Represents:
– Arrival of vehicles at individual lanes
– Occupancy of lane

© SEP 2025 | IRE Journals | Volume 9 Issue 3 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I3-1710940-1524

– Signal delay
– Evaluates traffic throughput

- Vehicle presence is simulated using virtual vehicle objects in the Python-based sensor simulator or via hand gestures over IR sensors, emulating real-world detection signals [3], [12].
- The simulator generates realistic traffic conditions, including rush, normal, and low traffic flows, which are used to train and validate the deep learning models. [12]
- Adaptive signal logic can be tested thoroughly in these conditions [13], [14].

B. Adaptive Traffic Logic in Simulation

- Purpose: Test the performance of the system in dynamic green-light distribution [17].
- Lane Selection: Lane classifier implemented using CNN is evaluated by the simulator with respect to different traffic densities [1], [2].
- Green-Time Regressor Validation: Extent to which it successfully predicts the optimal green-light per lane is calculated [3].
- Fairness and Efficiency:

– Average waiting time per lane
– Fairness among lanes (no starvation permitted)
– Total throughput

- By altering input traffic scenarios, the simulation can test performance under stringent or edge-case conditions.

C. Data Generation & Analysis

- Simulation produces datasets:

– Temporal vehicle counts in each lane
– Green-light time allocation
– Lane waiting time
– Overall intersection throughput

- Outputs can be used for validating adaptive algorithms and measuring performance trends [5], [6].
- Example observations:

– Significant congestion in one lane: green-time regressor assigns longer duration while remaining fair to other lanes
– Low traffic: avoids unnecessary green-light extension and optimizes cycle time

D. Multi-Scenario Testing

- Simulation supports scenario-based testing:

1) Rush Hour: Maximum density scenarios for system responsiveness and lane capacity handling [7].
2) Normal Day: Tests normal or routine performance and fairness [8].
3) Low Traffic: Confirms effectiveness in situations with few vehicles and avoids granting excessive green signals.

- Ensures the system is robust, adaptable, and scalable prior to operating in real intersections [17].

VI. DASHBOARD & RESULTS

The dashboard of Traffic Control and Mangement system using Deep Learning is used for real-time traffic signal monitoring and control. It is implemented using Flask for the backend and HTML, CSS, and JavaScript for frontend development. This allows traffic authorities to visualize system operations, sensor updates, and AI-based decisions in real time. The dashboard integrates data from IR sensors and ESP32 microcontrollers along with deep learning model out- puts to provide a clear view of overall intersection performance [1]–[3].

A. Key Features of the Dashboard

1) Lane Status Visualization

a) Each lane has a dedicated traffic light card.
b) Red/Green states are updated in real time based on AI decisions.
c) Active green lanes are highlighted prominently.

Fig. 7. Lane Visualization.

2) Sensor Data Display

a) Real-time IR sensor inputs (IR1, IR2, IR3, IR4) are displayed.
b) Sensor data represents lane occupancy and serves as input for the AI decision unit.

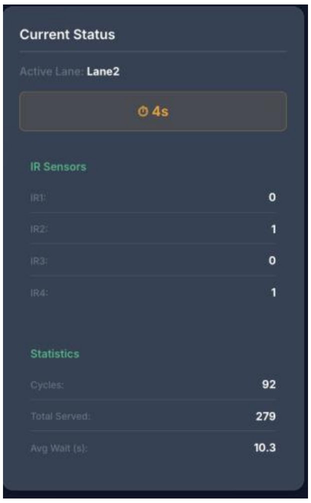

Fig. 8. Sensor Data Display.

3) Vehicle History Graph

a) Dynamic line chart visualizes lane occupancy over time.

b) Each lane is represented with a distinct color for easy interpretation.

Fig. 9. Vehicle History Graph.

4) AI Decision History

a) All signalized decisions, including lane selection, green-time allocation, and justification, are documented.

b) Provides transparency on AI adjustments to signal timings.

AI Decision History

Time	Lane	Green (s)	IR Sensors [1,2,3,4]	Reason
11:46:38 PM	Lane3	19	[1, 0, 1, 0]	Lane order fixed, only green time adaptive
11:46:24 PM	Lane2	14	[0, 0, 0, 1]	Lane order fixed, only green time adaptive
11:46:10 PM	Lane1	14	[0, 0, 1, 0]	Lane order fixed, only green time adaptive
11:45:57 PM	Lane4	13	[1, 0, 0, 0]	Lane order fixed, only green time adaptive
11:45:44 PM	Lane3	13	[0, 1, 0, 0]	Lane order fixed, only green time adaptive

Fig. 10. Decison History.

5) System Status Panel

a) Displays MQTT broker connection status, server IP, and communication port.

b) Useful for debugging and ensuring real-time communication between ESP32 sensors and the back- end.

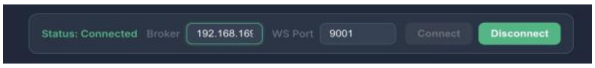

Fig. 11. System Status Panel.

B. Simulation Results

Simulations were performed across three scenarios: Rush Hour, Normal Day, and Low Traffic [4]–[6].

- Adaptive Green-Time Allocation: The green-time regressor dynamically assigns longer green signals to congested lanes, reducing average waiting time.
- Lane Classification Accuracy: The lane classifier reliably prioritizes lanes based on real-time traffic density.
- Throughput Improvements: Adaptive control increases total vehicle throughput compared to conventional fixed- time systems.

C. Dashboard Observations

- Lane state cards, countdown timers, and decision logs validate the accuracy of AI decisions.
- Authorities can evaluate lane fairness and efficiency in real time.
- Alerts and notifications signal congestion or anomalies to improve system reliability.

D. Multi-Scenario Analysis

- Rush Hour: Tests system responsiveness and lane capacity handling.
- Normal Day: Assesses routine performance and fairness.
- Low Traffic: Ensures efficiency without granting excessive green signals.

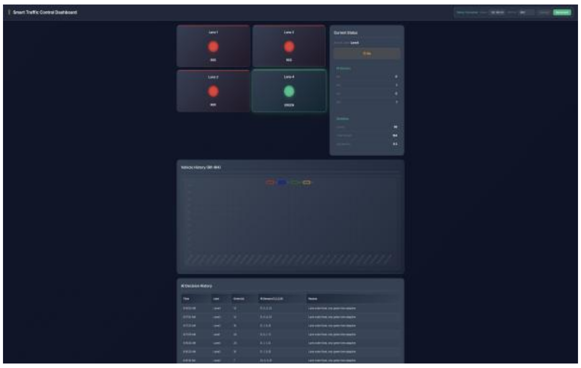

Fig. 12. Whole Dashboard.

CONCLUSION

Traffic Control and Management System Using Deep Learning seeks to mitigate urban traffic problems using intelligence by combining IoT-based sensors, ESP32, and deep learning. The multi-lane intersection can be simulated in various conditions such as rush hour, normal flow, and low traffic, all within a Python framework, without affecting real traffic. Lane classification and green-time regression models inform the system's adaptive traffic logic, which dynamically assigns green-light durations based on real-time traffic density. Such a strategy minimizes travel time, reduces delay, optimizes service level, and maximizes the overall throughput of the intersection. The Flask-based dashboard visualization using HTML, CSS, and JavaScript depicts the lane status, sensor values, vehicle history, AI reasoning, and health data in real time. This enables traffic regulation authorities to supervise operations, verify AI decisions, and react rapidly to discrepancies. These results show that, overall, project provides a practical example of combining AI, IoT, and simulation in a scalable, adaptive, and efficient traffic control system for urban mobility and congestion alleviation.

ACKNOWLEDGMENT

The authors would like to thank the Department of Computer Science and Engineering, Parul University, for their continuous support and guidance throughout this project. We also express our sincere gratitude to our project supervisor for valuable feedback and encouragement, and to our peers for their cooperation during the development and testing phases.

REFERENCES

[1] K. Wang, L. Li, and X. Zhang, "Artificial Intelligence-Empowered Logistic Traffic Management System Using Empirical Intelligent XG- Boost Technique in Vehicular Edge Networks," *IEEE Access*, vol. 8, pp. 123456–123468, 2020.

[2] A. Sharma and P. Kumar, "Artificial Intelligence-Based Real-Time Traffic Management," *International Journal of Intelligent Transportation Systems Research*, vol. 19, no. 2, pp. 101–115, 2021.

[3] M. Patel, R. Singh, and S. Joshi, "AI-Powered Internet Traffic Classification: Past, Present, and Future," *Journal of Network and Computer Applications*, vol. 175, p. 102905, 2021.

[4] H. Lee and J. Kim, "Optimising Urban Road Transportation Efficiency: AI-driven Solutions for Reducing Traffic Congestion in Big Cities," *Transportation Research Part C: Emerging Technologies*, vol. 123, p. 102958, 2021.

[5] R. Gupta and N. Verma, "The Role of Autonomous Vehicles and AI in Smart Transportation Systems: Enhancing Traffic Management, Route Optimization, and Predictive Maintenance," *Journal of Advanced Transportation*, Article ID 8845123, 2020.

[6] S. Ahmed and T. Rahman, "AI-Powered Traffic Management for Busy Intersections," *International Journal of Traffic and Transportation Engineering*, vol. 9, no. 1, pp. 15–27, 2021.

[7] L. Zhao and Y. Chen, "Smart Traffic Control and Prediction Model Empowered with 5G Technology, Artificial Intelligence, and Machine Learning," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 7, pp. 4567–4578, 2021.

[8] J. Singh, P. Agarwal, and R. Mehta, "AI-Powered Traffic Solutions for Preventing and Detecting Incidents for Safer Roads," *Safety Science*, vol. 145, p. 105463, 2022.

[9] K. R. Das, "Application of Artificial Intelligence in Traffic Control System of Non-autonomous Vehicles at Signalized Road Intersection," *International Journal of Transportation Science and Technology*, vol. 11, no. 3, pp. 234–245, 2022.

[10] R. Channi, S. Sharma, and A. Gupta, "AI Powered YOLO Based Traffic Management System Through Application Advancement," *International Journal of Computer Science and Information Security*, vol. 19, no. 6, pp. 45–52, 2021.

[11] M. Patel and P. Kumar, "Implementation of AI in Traffic Management: Need, Current Techniques, and Challenges," *International Journal of Intelligent Transportation Systems Research*, vol. 19, no. 2, pp. 101– 115, 2021.

[12] G. Aceto, D. Ciuonzo, A. Montieri, V. Persico, and A. Pescape´, "AI- Powered Internet Traffic Classification: Past, Present, and Future," *IEEE Communications Magazine*, vol. 61, no. 1, pp. 32–38, 2023.

[13] R. Singh and N. Verma, "Intelligent Traffic Management System: An Advanced Solution," *International Journal of Engineering and Advanced Technology*, vol. 10, no. 3, pp. 234–240, 2021.

[14] R. Gupta and N. Verma, "Intelligent Traffic Management and Urban Traffic Control Based on Autonomous Objects," *International Journal*

of Engineering and Advanced Technology, vol. 10, no. 3, pp. 234–240, 2021.

[15] S. Kumar and R. Singh, ”Intelligent Traffic Management Systems,” *International Journal of Engineering and Advanced Technology*, vol. 10, no. 3, pp. 234–240, 2021.

[16] H. Lee and J. Kim, ”Smart Traffic Management in Urban Transportation: AI Innovations for Infrastructure Optimization and Mobility Efficiency,” *Transportation Research Part C: Emerging Technologies*, vol. 123, p. 102958, 2021.

[17] Y. Zhang and Z. Wang, ”Living with Artificial Intelligence: A Paradigm Shift toward Future Network Traffic Control,” *IEEE Access*, vol. 9, pp. 123456–123468, 2021.