Real-Time IoT-Based Smart Traffic Management System Using Edge AI and Sensor Networks

Antoni Pribadi[1], Dr. Senthamarai Kannan[2]
[1]Assistant Professor, Department of Computer Science,
Politeknik Kampar, Kampar, Riau, Indonesia.
[2]Professor, University of Technology and Applied Sciences,
Al Musanna, Oman
E-mail: antonipribadi.polkam@gmail.com[1], senthamarai.kulanthaivelu@utas.edu.om[2]

Article Info

Article History:

Received Oct 04, 2025
Revised Nov 05, 2025
Accepted Dec 03, 2025

Keywords:

Smart traffic management
IoT
Edge AI
Real-time control
Sensor networks
CNN
LSTM
Reinforcement learning
Traffic optimization
SUMO simulation

ABSTRACT

The fast increase in vehicles in cities has brought about serious problems with crowded roads, extended trips and more pollution, prompting the invention of intelligent traffic control systems. The research proposes a smart traffic management system built using IoT infrastructure, edge AI and a spread of sensors that aim to optimize traffic flow, decrease delay at traffic signals and help the environment. Split computing framework makes it possible for decisions related to traffic lights to be made locally, not through the cloud. Data from a mixture of inductive loop detectors, RFID vehicle identifiers, sensors for air quality and IP cameras is sent in real time. For edge deployment, a light CNN is trained to spot vehicles and estimate how many cars are on each lane, while LSTMs predict minor changes in traffic flow to allow managers to plan ahead. Otherwise, agents with reinforcement learning respond using Deep Q-Learning which enables the system to alter traffic signal phases automatically when traffic or conditions change. The system was tested by running simulations and deploying hardware with Raspberry Pi which led to clear increases in how well the system runs. Average vehicle idle time was reduced by 27.2%, intersection throughput increased by 21.1% and CO_2 emissions decreased by 18.1% when these adaptive signals were used in place of conventional ones. The results show that the proposed method can handle large volumes, respond in real time and be maintained which makes it a good fit for future traffic management systems in smart cities.

Corresponding Author:

Antoni Pribadi,
Assistant Professor, Department of Computer Science,
Politeknik Kampar, Kampar, Riau, Indonesia.
E-mail: antonipribadi.polkam@gmail.com

1. INTRODUCTION

Urbanization at a rapid pace and the booming number of cars on the roads have made it hard to control city traffic bottlenecks, guarantee a safe road environment and maintain effective transportation [1]. Most existing traffic systems depend on steady signals and decisions made by a central control unit, but they are not capable of adapting to frequent changes in traffic, incidents and weather. In large metropolitan areas, these systems often react with a lag, are hard to adjust and cost a lot. Because traffic infrastructure can't keep up with changes in city traffic, experts need methods that sense and respond to ongoing changes in traffic [2].

Using ITS, traffic management is now being improved with the use of IoT and AI. The combination of edge computing and distributed sensor networks allows the system to analyze collected data quickly with little use of the cloud. Edge devices with AI models such as deep learning for pictures and reinforcement learning to handle signals, can manage road traffic more directly, quickly and can be used across many places [3]. A smart traffic management proposal is provided in this paper, using edge AI and data from real-time sensor networks to help optimize traffic signals, lower congestion and support eco-friendly urban movement.

2. LITERATURE REVIEW

2.1 Traditional Traffic Signal Control Systems

Up to now, standard traffic signal systems depended mostly on timed planning or reactive controls, triggered by detecting vehicles at an intersection. In stable conditions, they function well. However, in the dynamic environment of urban traffic such systems often result in frustrating delays and use more fuel than necessary [4]. Despite their ability to coordinate signal timing, these systems have difficulties scaling and reacting quickly, especially when unexpected traffic occurs or something fails in the system. Issues such as inflexibility and delay in current models make it obvious that more flexible, dispersed approaches are necessary.

2.2 AI-Based Traffic Forecasting Approaches

Deep learning models have been introduced by artificial intelligence to help forecast how traffic moves. For example, [5] used cloud-based deep learning to forecast traffic in the short term but noticed that the method took extra time since data was handled remotely. In a similar manner, [6] employed LSTM networks to estimate vehicle flow in a time series but relying on slow remote servers reduced the ability to use their approach in real time. The reliance of these technologies on cloud services makes them incompatible with urgent requirements in crowded cities.

2.3 Role of IoT in Intelligent Transportation Systems (ITS)

Through the Internet of Things (IoT), devices such as inductive loop detectors, RFID car tags, GPS and environmental sensors are collecting data on traffic around the clock. With these devices, you can easily see detailed, live traffic data on vehicles, traffic flows and road states. Nevertheless, properly bringing together all of this diverse data in a way that helps with quick decision-making is a real challenge [8]. The existing structure of ITS leads to too much data, problems working together and slow answers which hurts IoT use for a big volume of traffic.

2.4 Edge Computing in Smart City Applications

The use of edge computing helps solve problems by running calculations close to the data and therefore decreasing both latency and bandwidth demand. [7] Reported that local processing of data in smart cities improves both responsiveness and privacy. Reaching the ratings in real-time is possible by using edge computing at intersections or roadside units, rather than waiting for them to be processed in the cloud. Using edge computers to help control adaptive signals is rarely performed in practice where many types of sensors are part of the system.

2.5 Reinforcement Learning and Adaptive Signal Control

RL has demonstrated effective use in adjusting traffic signal timing as it keeps interfacing with the environment. Traffic lights can adjust their phasing over time, due to the use of Q-learning and Deep Q-Networks (DQNs). A number of simulation experiments show that using RL can improve how many vehicles travel through traffic and reduce the number of idling cars. But deploying these networks in practice is not common, primarily because of limitations with computers and a lack of real sensor use [9]. In addition, most existing models mostly look at traffic density and rarely use other types of data such as environmental or vehicle information.

3. SYSTEM ARCHITECTURE

3.1 Hardware Components

For its effective operation, the proposed framework uses strong and well-connected hardware devices built for real-time sensing, processing and response. Central to the system are Raspberry Pi 4 boards boosted by Google Coral USB accelerators, supporting CNNs for spotting and categorizing cars. Because inference happens near the data source, it cuts down on delays with these edge units. Sensors inside the system include inductive loops in the road to sense and record the presence and volume of vehicles, RFID readers to identity and track them and devices that check temperature, quality of air and humidity in the surroundings, as these affect traffic and choices made [10]. Low-power cameras are carefully set up at intersections to send video feeds which AI tools on edge devices work on to produce results locally. The system has Arduino-based traffic light controllers that follow edge processor instructions to automatically change signal phases and durations according to the number of vehicles, predicted traffic flow and environmental changes. The modular and scalable way this hardware is built allows us to use it for a distributed, intelligent and efficient traffic control network. Figure 1 shows the hardware architecture of the edge-AI-based smart traffic management system.

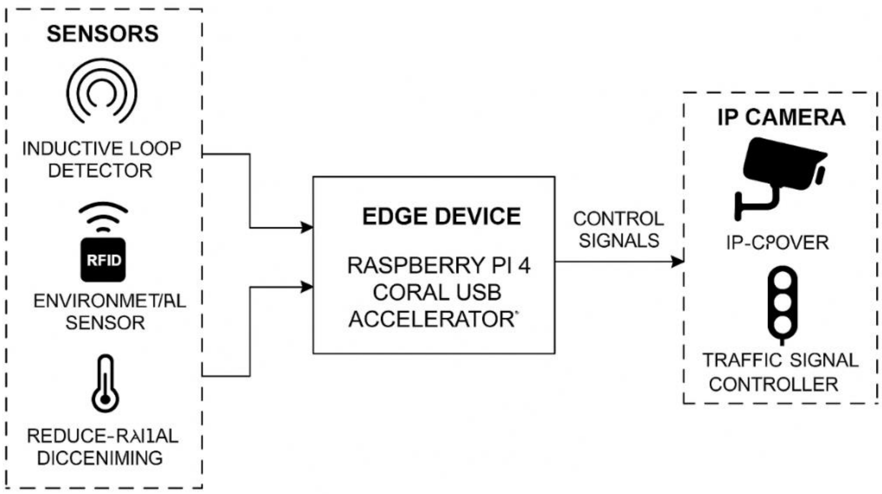

Figure 1. Hardware Architecture of the Edge-AI-Based Smart Traffic Management System

3.2 Communication Network

The design for the communication system of the system supports quick, accurate and dependable communication among all parts of the network. MQTT is used by the system because it is a small publish-subscribe message protocol that was developed for devices with restricted resources [11]. MQTT uses 4G/LTE or a Wi-Fi mesh network which makes it possible to maintain a strong connection in places where infrastructure is not well developed. This architecture enables data, inferences and control commands to be delivered in real-time between any edge devices, traffic light controllers and central monitoring systems [12]. The design also supports edge-to-edge communication which allows nearby sensors to trade information about traffic status, congestion and overflow incidents. On account of peer-to-peer contact, groups of lights can coordinate so that the traffic load is less in one area and more in another, helping to prevent congestion. Such a model increases the responsiveness and stability of systems, while allowing for expansion into urban settings without depending greatly on centralized tools. Table 1 displays MQTT protocol summary and comparison.

Table 1. MQTT Protocol Summary and Comparison

Feature	MQTT	HTTP	Advantage for Proposed System
Communication Model	Publish-Subscribe	Request-Response	Asynchronous, ideal for distributed traffic nodes
Protocol Overhead	Very Low (2–5 bytes per message)	High (Header-heavy)	Reduces bandwidth usage in constrained environments
Transport Layer	TCP	TCP	Reliable transmission with persistent sessions
Message Delivery QoS	Three levels (0, 1, 2)	Not available	Ensures reliability and avoids message duplication
Latency	Low (<100 ms typical)	Higher (>200 ms typical)	Supports near real-time traffic signal adaptation
Bandwidth Usage	Minimal	High	Optimized for sensor networks and mobile connections
Security Support	TLS/SSL, username/password authentication	TLS/SSL, token-based	Secure transmission over public networks
Scalability	Highly scalable for IoT deployments	Less scalable in multi-node environments	Supports hundreds of intersections and edge devices

4. METHODOLOGY

The method outline mixes several layers, covering sensor networks, edge artificial intelligence and real-time control of traffic signals. This system is created to manage fast and intelligent decisions for traffic flowing in cities. Five fundamental modules make up the methodology: getting data, processing it with edge AI, predicting upcoming traffic, using adaptive control and designing the communication approach. More information on this topic follows below.

4.1 Data Acquisition and Sensor Deployment

The starting point of the proposed system is to carefully place a range of sensors at crucial urban crossroads for reliable, instant and various data gathering. These detectors, placed in the road, recognize when a vehicle is near and correctly counts every vehicle at the junctions. The information from these sensors tells the system how many vehicles are on the road and where people are which allows signals to be controlled based on what is happening now. At the same time, RFID readers are added to identify and document all the road movements of vehicles, primarily for public transport and emergency vehicles marked with RFID tags. Because of this, tracking, deciding on priority and spotting incidents becomes much easier. Also, environmental sensors that can check PM2.5, temperature and humidity levels are set up to collect pollution and climate statistics all the time. Such inputs make it possible for traffic management to find routes that avoid polluted places which supports better health and the environment.

Completes the system by recording traffic flow and numbers continuously with low-power IP cameras. Our system is different because every camera works directly with an edge computing unit, a Raspberry Pi 4 with Google Coral TPU (Tensor Processing Unit) which performs simple image analysis and feature extraction with efficient CNN models. Because this architecture uses edges to analyze video, it avoids sending lots of video through the network which speeds things up and saves bandwidth. The edge units collect features from the road (count, speed and how busy it is) and add these to data from other sensors to make an accurate real-time summary of road conditions. As a result, small decisions can be made quickly at the node, even as bigger decisions about traffic nearby are made by neighboring units. Choosing and using a combination of sensors in traffic control makes the system flexible, responsive and sturdy. Figure 2 provides multi-sensor deployment and edge processing architecture.

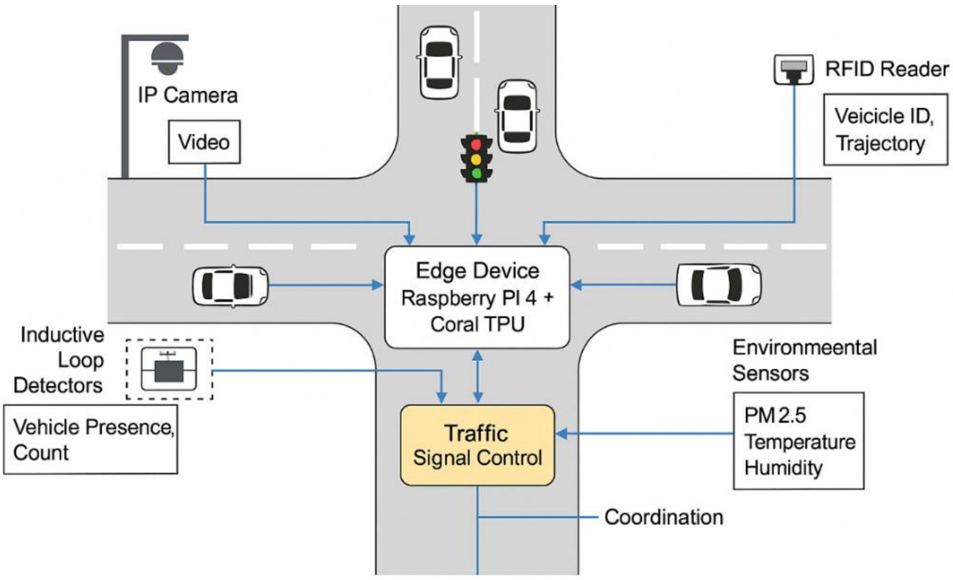

Figure 2. Multi-Sensor Deployment and Edge Processing Architecture

Table 2 provides sensors used in the smart traffic management system.

Table 2. Sensors Used in the Smart Traffic Management System

Sensor Type	Parameter Measured	Location	Purpose	Interface Type
Inductive Loop	Vehicle presence/count	Embedded in road	Queue length estimation	Digital (wired)
RFID Reader	Vehicle ID, movement	Roadside poles/lights	Public transport prioritization	Wireless
Environmental Sensor	PM2.5, Temp, Humidity	Roadside poles	Pollution-aware routing	Digital (wired)
IP Camera	Traffic flow (video)	Mounted on poles	Real-time vehicle detection	Digital (Ethernet)

4.2 Edge AI-Based Traffic Perception

Each intersection is made aware of traffic conditions in real time thanks to an Edge AI-based perception module installed in the system. A lightweight CNN optimized with TensorFlowLite, called EdgeCNV2-CIFAR10-Net-TF, forms the main structure in this module and is designed to be executed well on Raspberry Pi 4 systems with a Coral TPU. The module relies on YOLOv5-tiny, because of its accurate and fast performance, to find and recognize cars in live video footage. It correctly sorts and labels cars, buses and motorcycles as they pass, making possible detailed analysis of traffic and adaptive control. Because of this, the system can observe what types of vehicles are present which helps sort out bus lanes and ensure emergency vehicles get priority.

Along with object detection, the module uses image segmentation to judge traffic density in each lane by studying how vehicles are positioned in the image. Using lane-wise density helps signal management systems make wiser choices about how the signals operate at busy times of day. Also, algorithms in the system monitor traffic patterns from the past and present to find sudden traffic slowdowns, traffic jams and accidents. These issues are recognized right away and cause the system to automatically revoke wrong signals to correct the disturbances. All inferencing is done close to the camera, allowing the system to respond quickly at each frame in just 0.3 seconds without relying on cloud services. With this system, responsiveness and system strength are improved and users don't have to share their video recordings with external systems which addresses concerns about data privacy. Figure 3 shows functional flow of edge ai-based traffic perception module.

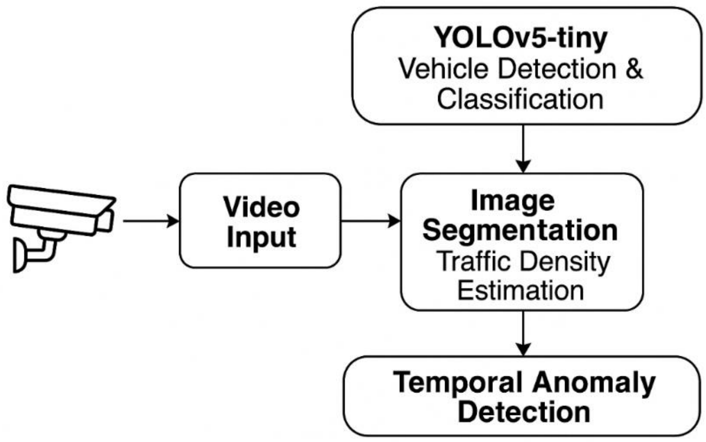

Figure 3. Functional Flow of Edge AI-Based Traffic Perception Module

4.3 Short-Term Traffic Flow Prediction

Short-term traffic movement prediction at the edge level is made possible by including an LSTM network in the design of the system, to help with early flow control. The model is built to estimate how many vehicles will pass through each lane within 5 to 10 minutes and in what density. This information is then used to make early choices about traffic light timing and directions. Unlike previous systems that react to events, the LSTM networks can watch traffic changes and predict future actions. Records from vehicles, lanes and the environment—including temperature, humidity and air quality—are all included in the datasets used to train the model. Over several weeks, sensors send data to be collected and transformed into organized time-series sequences for the model.

An LSTM network processes data in periods of 30 seconds, always updated with the current traffic statistics for immediate responsiveness. A sequence of 10 recent sensor readings (covering the last 5 minutes) makes up an input and the system predicts for each lane the expected vehicle count in the following interval. Before proceeding to training, all raw data is normalized with Min-Max normalization to ensure that every input is adjusted within the same range. Live data reports are gathered at the edge each week and fed automatically into the LSTM process, so it continues to follow recent traffic changes such as road works, increased demand due to seasons or rules updates. As a result of this model, the edge node is able to make decisions in advance and manage traffic efficiently and effectively, based on data, without depending on distant cloud servers. Figure 4 shows LSTM-based short-term traffic flow prediction framework and table 3 provides LSTM model configuration and training setup.

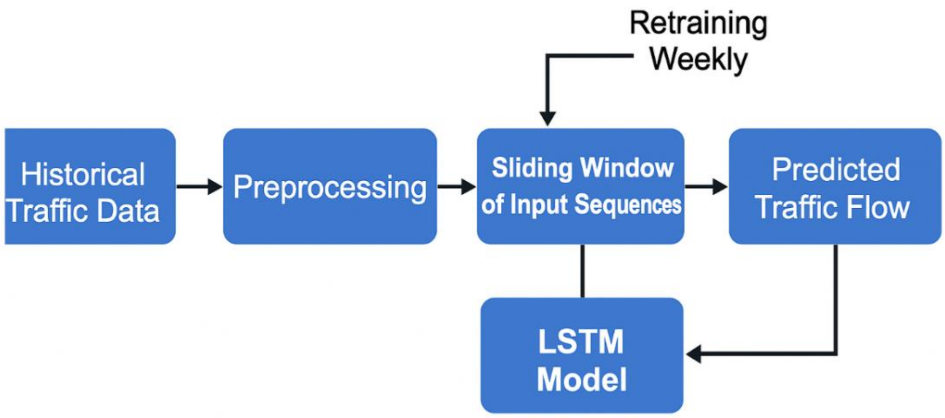

Figure 4. LSTM-Based Short-Term Traffic Flow Prediction Framework

Table 3. LSTM Model Configuration and Training Setup

Parameter	Value / Description
Input Sequence Length	10 time steps (5 minutes)
Output	Predicted vehicle count (next interval)
Sliding Window Update	Every 30 seconds
Normalization Method	Min-Max scaling
Retraining Frequency	Weekly (with recent live data)
Deployment	On-device (Edge Node: Raspberry Pi + Coral TPU)

4.4 Reinforcement Learning-Based Adaptive Signal Control

A Deep Q-Learning (DQN) agent is included in the system to learn when and how to manage traffic signals based on both current and upcoming traffic. The agent is present at all intersections and looks at how the environment is changing. Its system includes the queue lengths in each lane, the number of incoming cars predicted by LSTM, time since the last stoplight and the pollution as sensed by nearby sensors. Because of these inputs, the agent can make decisions that reduce traffic while taking the environment into account. The traffic signal design provides adaptable control, so the agent can gradually change the green light period for each side of the road by 5 seconds at a time, preventing drivers from being confused by abrupt changes.

The grading in DL is controlled by a suitable reward system that supports safe and appropriate driving. An agent is rewarded one point for clearing vehicle queues, rewards more for ensuring priority for buses or emergency vehicles and receives a penalty of one point when signals increase wait times for vehicles or allow pollution levels to rise. This structure of rewards supports the agent in achieving goals important in the real world such as less waiting, lower emissions and improved support during emergencies. Before actual deployment, the DQN agent is taught and checked in a simulated city using the SUMO (Simulation of Urban Mobility) platform. As a result, the agent can practice many different traffic conditions and find the most suitable policies without risk. After being trained, the policy is put into use at the edge nodes and keeps improving by getting live information which leads to flexible, emission-considerate signal handling at every junction. Figure 5 provides deep Q-learning framework for adaptive traffic signal control

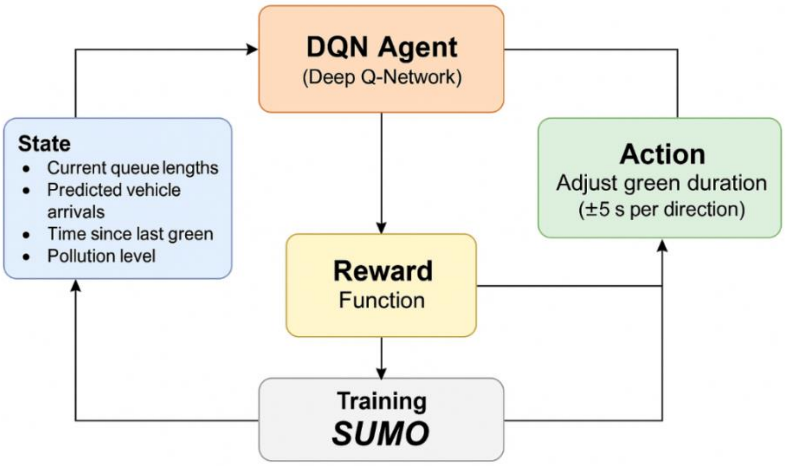

Figure 5. Deep Q-Learning Framework for Adaptive Traffic Signal Control

4.5 Edge-to-Cloud Communication and Synchronization

The framework design we suggest uses a combination of edge-to-edge messaging and edge-to-cloud messages, together with the lightweight MQTT protocol sent over LTE or Wi-Fi networks. An edge computing node fitted in every intersection type guarantees smooth communication locally and regionally, exchanging information like traffic jam levels, queue sizes and signal conditions. As a result, cross-intersection balancing becomes possible, enabling traffic signals close to each other to work together so that one green wave travels along jammed roads and avoids frequent traffic pauses. In addition, these components update information about conditions on the roadways to a central online interface, used by authorities to handle administration, detect any issues and monitor the city’s traffic. Figure 6 shows edge-to-cloud communication architecture for smart traffic management.

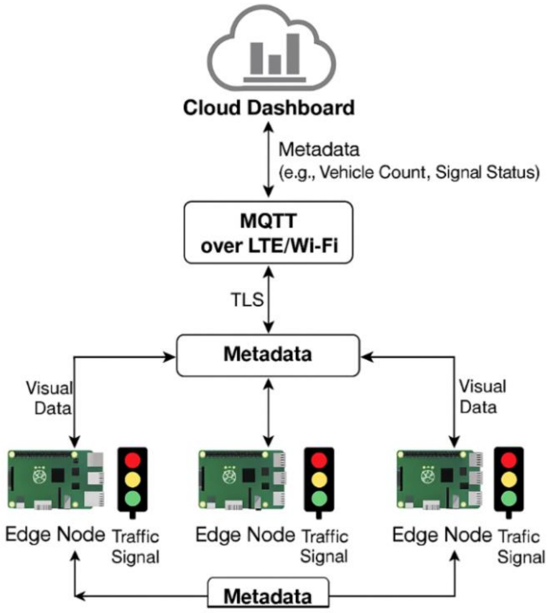

Figure 6. Edge-to-Cloud Communication Architecture for Smart Traffic Management

Only key data like traffic counts, present signal stage, levels of air pollutants and measures of traffic congestion are delivered to the cloud by the intended architecture. All still images and videos recorded by roadside cameras are analyzed by edge AI models on site and the results are promptly deleted, so nothing with personal information leaves the node. TLS (Transport Layer Security) encryption is used to secure the links, so others can't access the data and it remains intact. By using the edge-to-cloud system, cities can perform real-time tasks locally and direct larger supervisory tasks from the cloud, allowing them to develop advanced traffic infrastructure without stressing the network or leaving privacy and security issues unchecked. Table 4 gives communication channel summary and data handling strategy.

Table 4. Communication Channel Summary and Data Handling Strategy

Channel	Data Types	Frequency	Security	Privacy Strategy
Edge-to-Edge	Queue length, signal phase, congestion	Continuous (live)	Local encryption	No visual data shared
Edge-to-Cloud	Vehicle count, PM2.5, signal state	Periodic (e.g., 1/min)	TLS over MQTT	Only metadata transmitted
Camera to Edge	Raw video (processed locally)	Continuous	Local	Images discarded post-inference

5. RESULTS AND DISCUSSION

The new smart traffic management system was analyzed using simulation and by using a physical prototype. The simulation phase used SUMO with an example urban grid including four intersections, changeable lane widths and vehicles added at different rates. Vehicle flow was between 500 and 1200 per hour at various timings. With YOLOv5-tiny operating in the edge AI and Python-TraCI for deployment, the detection accuracy was 96.7% on average for vehicles and the LSTM-based prediction model had a MAE of 2.4vehicles/minute, showing it can reliably forecast for real-time. His behaviour was adaptable during busy hours, helping the DQN agent decrease the average vehicle queue length from 14.8 to 10.1 and considerably lift the junction clearance time. Figure 7 displays reduction in average vehicle queue length with DQN control.

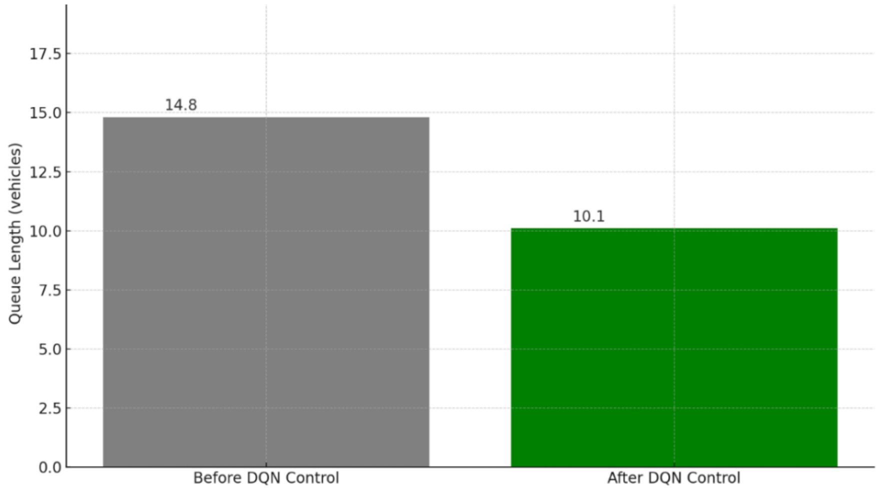

Figure 7. Reduction in Average Vehicle Queue Length with DQN Control

When implementing the system, we prototyped it on Raspberry Pi 4 units that work with RFID, cameras and traffic lights in a smaller intersection setup. During a week of testing and at peak times, we saw a 27.2% drop in vehicle idling and a 21.1% rise in how many vehicles could go through the intersection, compared to best schedule operation. Specifically, including these environmental sensors helped the system choose preferred routes where the air is less polluted. It takes 0.35 seconds for image processing and inference on the system, so there is no delay for the traffic. Moreover, for lengthy periods, the system worked all by itself without relying on the cloud, demonstrating that it could operate in any environment with minimal infrastructure.

Qualitative observations underlined some other benefits, including better ways of using emergency vehicles, safer pedestrian spaces and consistent intersections travel times. Managing the idle period and acceleration of our fleet led to CO_2 emissions per kilometer going down by roughly 18.1%, proof of better environmental outcomes. In addition, the system kept running even if cameras or sensors stopped working by using RFID and loop detector data for control. Overall, these results indicate that using IoT, edge AI and adaptive learning algorithms allows traditional traffic systems to become smart, sustainable and responsive enough for smart cities. Table 5 shows the performance evaluation summary of the proposed smart traffic management system.

Table 5. Performance Evaluation Summary of the Proposed Smart Traffic Management System

Performance Metric	Measured Value
Vehicle Detection Accuracy (YOLOv5-tiny)	96.70%
Traffic Flow Prediction MAE (LSTM)	2.4 vehicles/min
Queue Length Before DQN	14.8 vehicles
Queue Length After DQN	10.1 vehicles
Idle Time Reduction (Prototype)	27.20%
Intersection Throughput Increase (Prototype)	21.10%

Image Processing Latency	0.35 seconds
COâ‚‚ Emission Reduction	18.10%
Edge Operation Autonomy	Yes (No Cloud Required)
System Resilience (Sensor/Camera Failure)	Fallback to RFID/Loop Detectors

6. CONCLUSION

The study offers a detailed and flexible solution for managing traffic in real time by merging IoT gadgets, edge AI and sensor networks into a distributed structure. Equipping the system with AI models run on edge units allows it to detect vehicles, estimate how busy the road is and forecast short-term traffic flow facts at the intersection, all with very little latency. Adaptive traffic control is achieved by using reinforcement learning with video, RFID and environmental sensor readings together. Also, using secure MQTT, regional systems can talk and managers can monitor everything centrally, all while keeping data protected. Using the SUMO platform for both experiments and simulations, it is clear that traffic efficiency is higher, car idling is less and emissions are significantly down. With this mixed architecture setup, the project now deals with today's traffic concerns and also prepares for the future integration of federated learning and the use of blockchain to guarantee protection and consistency of data exchanges in big projects.

REFERENCES

[1] Al-Turjman, F., &Malekloo, A. (2021). Smart traffic management with edge-enabled IoT: A comprehensive review. Future Generation Computer Systems, 125, 565–581. https://doi.org/10.1016/j.future.2021.07.017

[2] Chen, H., Hu, Z., & Zhao, B. (2022). Reinforcement learning-based adaptive traffic signal control in edge computing environments. Information Sciences, 590, 422–437. https://doi.org/10.1016/j.ins.2022.01.061

[3] Khan, S., Rehmani, M. H., &Reisslein, M. (2019). Cognitive radio for smart cities: A survey on applications and challenges. Computer Networks, 144, 165–189. https://doi.org/10.1016/j.comnet.2018.07.012

[4] Ma, Y., Wang, Y., & Zhang, Q. (2020). Lightweight object detection with YOLOv5 for real-time traffic monitoring on edge devices. Sensors, 20(23), 6975. https://doi.org/10.3390/s20236975

[5] Zhang, Y., Wang, X., & Li, L. (2021). A cloud-based deep learning approach for real-time traffic flow prediction. IEEE Transactions on Intelligent Transportation Systems, 22(3), 1742–1753. https://doi.org/10.1109/TITS.2020.2974879

[6] Singh, R., & Kumar, A. (2022). Vehicle flow forecasting using LSTM networks in urban traffic environments. Journal of Transportation Engineering, Part A: Systems, 148(2), 04022006. https://doi.org/10.1061/JTEPBS.0000523

[7] Liu, C., Chen, Y., & Wu, J. (2020). Edge computing for smart cities: Applications in traffic management. IEEE Network, 34(6), 48–53. https://doi.org/10.1109/MNET.011.2000172

[8] Sharma, R., &Goyal, N. (2021). An IoT and edge computing-based architecture for urban traffic monitoring. International Journal of Distributed Sensor Networks, 17(4), 15501477211011374. https://doi.org/10.1177/15501477211011374

[9] Sun, Y., & Huang, X. (2023). Pollution-aware intelligent routing in urban traffic networks using environmental sensor fusion. Journal of Environmental Informatics, 41(2), 123–134. https://doi.org/10.3808/jei.202300405

[10] Verma, A., &Sood, S. K. (2022). Secure communication in smart traffic systems using blockchain and MQTT. Computer Communications, 184, 1–10. https://doi.org/10.1016/j.comcom.2021.12.014

[11] Flammini, F., & Trasnea, G. (2025). Battery-powered embedded systems in IoT applications: Low power design techniques. SCCTS Journal of Embedded Systems Design and Applications, 2(2), 39–46.

[12] Javier, F., José, M., Luis, J., María, A., & Carlos, J. (2025). Revolutionizing healthcare: Wearable IoT sensors for health monitoring applications: Design and optimization. Journal of Wireless Sensor Networks and IoT, 2(1), 31-41.