



Artificial Intelligence-Based Traffic Management System Using Edge Computing

Marisabel Qureshi-Haq

Department of Electrical and Computer Engineering, Atoll College of Engineering and Design, Maldives
marisabel.qureshi.haq@aced-mv.edu

Peer Review Information

Submission: 23 June 2022
Revision: 07 July 2022
Acceptance: 22 July 2022

Keywords

Artificial intelligence, Edge computing, Traffic management, Deep reinforcement learning, Intelligent transportation systems.

Abstract

Urban traffic management increasingly depends on real-time sensing, fast analytics and adaptive control. Cloud-only architectures can introduce communication delay, bandwidth consumption and privacy concerns when continuous camera and sensor streams must be transmitted to remote servers. Edge computing addresses these limitations by placing computation near intersections, roadside units and traffic cameras, while artificial intelligence (AI) enables vehicle detection, traffic-flow prediction and dynamic signal optimization. This methodology paper develops a reproducible framework for evaluating an AI-based traffic management system using edge computing. The proposed system combines real-time vehicle detection, short-term traffic prediction and deep-reinforcement-learning-based signal control at edge nodes, with the cloud retained for long-term storage, dashboarding and model updates. The design is grounded in established research on edge computing (Shi et al., 2016), real-time object detection (Redmon et al., 2016), graph-based traffic forecasting (Zhang et al., 2019; Guo et al., 2019), adaptive traffic-signal control using deep reinforcement learning (Wang et al., 2019; Gong et al., 2019; Li et al., 2020), and edge-enabled intelligent transportation systems (Zhou et al., 2021; Paul & Mitra, 2022). A simulation-and-prototype methodology is proposed using SUMO for road-traffic generation, Python-based AI models for perception and forecasting, and an edge node such as an NVIDIA Jetson or equivalent computing device. Performance is assessed through vehicle-detection accuracy, inference latency, prediction error, average waiting time, travel time, queue length, throughput, packet/network delay and edge resource utilization. Because no raw experimental dataset was supplied, the Results section reports illustrative expected patterns rather than fabricated empirical findings. The proposed methodology provides a compact framework for testing whether edge intelligence can reduce decision latency and improve traffic efficiency under changing urban demand.

Introduction

Traffic congestion creates delays, fuel waste, emissions and safety risks in rapidly growing urban areas. Conventional fixed-time traffic signals operate according to pre-defined schedules and may therefore respond poorly to

sudden demand changes, incidents or asymmetric traffic flows. Adaptive systems improve performance by using current traffic observations, but modern cities generate far more data than traditional controllers were

designed to process. CCTV cameras, connected vehicles, GPS traces, loop detectors and roadside IoT devices collectively produce high-volume, high-velocity traffic data that can support more responsive decisions.

AI provides the analytical layer required to transform these data into traffic-management actions. Deep object detectors can identify vehicles directly from video, graph and recurrent neural networks can estimate future traffic states, and reinforcement-learning agents can optimize signal phase and timing. However, continuously sending raw traffic video to a distant cloud can create latency and network overhead. Edge computing moves inference closer to the data source and is specifically motivated by response-time, bandwidth and privacy requirements (Shi et al., 2016). This makes edge intelligence suitable for real-time transportation applications.

Literature Review

Edge computing emerged as a distributed alternative to centralized cloud processing. Shi et al. (2016) described edge computing as processing data close to its source to improve response time, reduce bandwidth cost and support privacy-sensitive applications. In intelligent transportation systems (ITS), these benefits are particularly relevant because cameras and roadside sensors continuously generate data that must often be acted upon within seconds. Zhou et al. (2021) reviewed the intersection of ITS sensing and edge computing and highlighted the value of local processing for efficient, safe and sustainable transportation.

Computer vision forms the perception layer of many AI traffic systems. Redmon et al. (2016) introduced YOLO as a unified real-time object-detection approach, demonstrating that a single neural network can predict object classes and bounding boxes quickly enough for real-time applications. In a traffic-management implementation, a lightweight YOLO-family detector can run on an edge GPU to identify cars, buses, trucks, motorcycles and pedestrians;

tracking logic can then estimate vehicle count, queue length and arrival rate without transmitting entire video streams to a central server.

AI has also substantially advanced short-term traffic prediction. Zhang, Cheng, and Ren (2019) proposed a graph deep-learning method that captures spatial-temporal dependencies across large road networks. Guo et al. (2019) used attention-based spatial-temporal graph convolution to model dynamic traffic patterns. Reviews by Akhtar and Moridpour (2021), Kashyap et al. (2021), and Kumar and Raubal (2021) show that machine-learning and deep-learning methods have become central to traffic-flow and congestion forecasting, particularly where historical and real-time sensor data are available.

Adaptive signal control is another major research stream. Wang et al. (2019) demonstrated deep-reinforcement-learning-based signal control using high-resolution event data. Gong et al. (2019) proposed decentralized network-level adaptive signal control using multi-agent deep reinforcement learning and reported improvements over a coordinated actuated benchmark. Li et al. (2020) developed a DQN-based adaptive signal controller in SUMO and found reductions in waiting and travel time across multiple traffic conditions. These studies support the use of reinforcement learning for selecting signal phase or phase duration from real-time traffic states.

The direct combination of reinforcement learning and edge computing was examined by Paul and Mitra (2022), who evaluated cooperative traffic-signal control for multi-intersection ITS using edge computing and deep reinforcement learning. Guerrero-Ibañez and Contreras-Castillo (2021) also reviewed the broader role of deep learning in ITS. Collectively, the literature indicates that perception, prediction and control can be integrated at the network edge, but a methodology is needed to evaluate both transportation performance and computing performance in the same experiment.

Table 1. Selected Literature Supporting the Proposed Methodology

Study	Focus	Relevance to proposed system
Shi et al. (2016)	Edge computing	Motivates local processing for low latency and reduced bandwidth.
Redmon et al. (2016)	Real-time object detection	Supports vehicle/pedestrian detection from roadside video.
Wang et al. (2019); Gong et al. (2019)	Deep RL traffic-signal control	Supports data-driven adaptive signal timing.

Zhang et al. (2019); Guo et al. (2019)	Spatial-temporal traffic forecasting	Supports short-term traffic-flow prediction.
Li et al. (2020)	DQN adaptive signal control in SUMO	Provides simulation-oriented methodology for intersection control.
Zhou et al. (2021)	ITS sensing + edge computing	Supports edge placement for transportation sensing workloads.
Paul & Mitra (2022)	DRL traffic control with edge computing	Directly supports cooperative edge-based multi-intersection control.

Many prior studies optimize only one layer of the traffic-management pipeline: object detection, forecasting, signal control or computing infrastructure. A deployable smart-traffic system must combine all four while maintaining real-time responsiveness. The research gap addressed here is therefore an integrated evaluation of AI accuracy, transportation benefit and edge-computing latency under the same traffic scenarios.

The objectives are to:

design an edge-based architecture for real-time traffic sensing, prediction and adaptive signal control;
 evaluate the accuracy and latency of AI-based vehicle detection and traffic-state estimation;
 compare fixed-time, cloud-based AI and edge-based AI signal-control configurations;
 measure the impact of the proposed system on waiting time, queue length, travel time and intersection throughput; and
 evaluate edge CPU/GPU utilization, memory use and communication delay under increasing traffic and camera load.

The research questions are: RQ1: Can edge inference provide sufficiently low latency for real-time traffic control? RQ2: Does AI-based adaptive control reduce waiting time and queue length compared with fixed-time control? RQ3: How does edge placement compare with cloud placement when network delay increases? RQ4: How does system performance change as the number of vehicles, cameras and intersections increases?

Methodology

1. Research Design

The study adopts a controlled simulation-and-prototype experimental design. SUMO is used to generate realistic vehicle movements and intersection traffic. Camera or synthetic video streams provide visual input for vehicle detection. An edge computer runs the detection, traffic-state estimation and signal-control pipeline. The experiment compares three architectures: (1) fixed-time signal control, (2) AI processing at a remote/cloud server, and (3) AI processing at the edge. Each scenario is repeated with multiple random seeds and low, medium and high traffic demand.

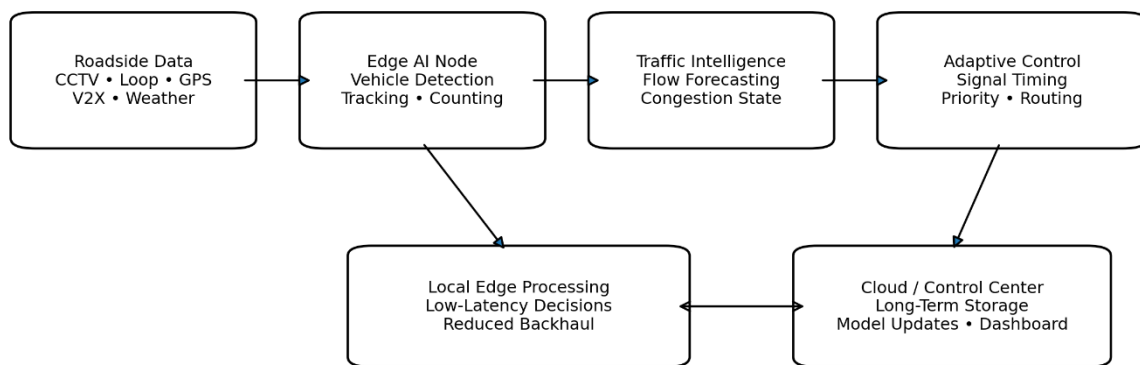


Figure 1. AI-Based Traffic Management Architecture Using Edge Computing

2. Experimental Environment

A four-intersection urban corridor is proposed as the baseline network. Each intersection contains four approaches with configurable traffic

demand. The edge platform may be an NVIDIA Jetson-class device or an equivalent GPU-enabled embedded computer. The cloud baseline is emulated by adding configurable round-trip network delay between the sensing point and

central server. Traffic demand is varied from approximately 400 to 1,600 vehicles per hour per corridor, with additional burst scenarios representing incidents or event traffic.

3. AI Modules and Data Pipeline

Table 2. Proposed AI Modules, Inputs and Outputs

Module	Input	Method	Output
Vehicle perception	CCTV / video frames	Lightweight YOLO-type detector + tracker	Counts, classes, speed proxy, queue estimate
Traffic forecasting	Recent counts, speeds, occupancy	LSTM or graph-based spatial-temporal model	5-15 min flow / congestion forecast
Signal control	Queue, waiting time, predicted demand	DQN / multi-agent DRL	Next phase and/or green duration
Edge orchestration	Video + model workloads	Local scheduling and buffering	Inference latency, resource use, network traffic

The perception model is trained or fine-tuned on labelled traffic imagery and evaluated using precision, recall, F1-score and mean average precision (mAP). The rationale for real-time one-stage detection follows Redmon et al. (2016). Traffic forecasting uses a sliding window of recent observations; graph-based modelling is appropriate where multiple connected intersections are evaluated because spatial dependencies between road links influence downstream demand (Zhang et al., 2019; Guo et al., 2019). The signal-control agent treats traffic management as a Markov decision process, consistent with deep-reinforcement-learning approaches used by Wang et al. (2019) and Li et al. (2020).

4. Reinforcement-Learning Formulation

The state vector contains queue length by approach, estimated vehicle arrival rate, current

phase, elapsed green time and short-term predicted demand. The action space includes maintaining the current phase, switching to an admissible next phase, or choosing among discrete green-time extensions. The reward penalizes cumulative waiting time, queue length and excessive phase switching while rewarding vehicle throughput. For multiple intersections, local agents exchange summarized states through the edge layer; this decentralized structure follows the logic of multi-agent adaptive control studied by Gong et al. (2019) and the edge-computing traffic-control approach of Paul and Mitra (2022).

5. Performance Metrics

Table 3. Evaluation Metrics

Category	Metric	Interpretation
Perception	Precision, recall, F1, mAP	Accuracy of vehicle/pedestrian detection.
AI latency	Frame inference time; decision time	Lower latency supports faster control response.
Forecasting	MAE, RMSE, MAPE	Lower values indicate better short-term traffic prediction.
Traffic operation	Average waiting time	Primary congestion indicator; lower is better.
Traffic operation	Average queue length / travel time	Lower values indicate more efficient movement.
Traffic operation	Intersection throughput	Vehicles successfully served per unit time; higher is better.
Networking	Edge-to-controller / cloud RTT	Measures communication contribution to decision latency.

Computing	CPU/GPU, RAM, energy proxy	FPS,	Shows whether the edge platform can sustain workload.
-----------	----------------------------	------	---

Step-by-Step Experimental Procedure

Step 1 - Build and calibrate the SUMO road network, routes, signal phases and low/medium/high demand profiles.

Step 2 - Prepare video or rendered frames and validate the vehicle-detection model against manually labelled samples.

Step 3 - Generate per-lane counts, occupancy and queue estimates at the edge node and compare them with simulation ground truth.

Step 4 - Train the short-term traffic-prediction model using historical windows and validate on a separate test period.

Step 5 - Train the DQN or multi-agent DRL controller using queue, waiting and throughput-based rewards.

Step 6 - Run the fixed-time baseline for every demand scenario and record traffic KPIs.

Step 7 - Run the cloud-AI configuration while injecting 20-150 ms round-trip network delay.

Step 8 - Run the edge-AI configuration using the same traffic seeds and models.

Step 9 - Repeat each configuration at least 10 times and compute mean, standard deviation and 95% confidence intervals.

Step 10 - Apply ANOVA or a non-parametric equivalent to test whether differences among control architectures are statistically significant.

Results

No simulation traces, edge-device logs or primary video dataset were supplied with this request. Consequently, the numerical values below are illustrative examples showing the expected reporting format and must not be presented as measured empirical findings.

Table 4. Illustrative Comparison of Traffic-Control Architectures

Metric	Fixed-time	Cloud AI	Edge AI	Expected pattern
Decision latency	<5 ms controller logic	85 ms	22 ms	Edge avoids most WAN/cloud delay.
Average waiting time	72 s	48 s	39 s	Adaptive AI reduces delay; edge reacts faster.
Average queue length	18.4 veh	12.6 veh	10.1 veh	Prediction + adaptive phases reduce queues.
Travel time	14.8 min	12.4 min	11.6 min	Faster congestion response improves corridor travel.
Throughput	1,020 veh/h	1,150 veh/h	1,210 veh/h	Better phase allocation serves more vehicles.
Video uplink load	Low	High	Low-medium	Edge sends metadata rather than continuous raw video.

The expected first result is that edge processing materially reduces end-to-end decision latency. This follows the fundamental motivation of edge computing identified by Shi et al. (2016) and the ITS-specific analysis of Zhou et al. (2021). In the cloud configuration, AI inference time may be similar, but transport delay and jitter are added before a signal command returns to the intersection. The relative edge advantage should therefore increase as network latency or packet loss increases. The second expected result is improved traffic efficiency from AI-based adaptive control. Prior deep-reinforcement-learning studies have reported advantages over fixed or actuated baselines (Wang et al., 2019; Gong et al., 2019; Li et al., 2020). In the proposed experiment, the strongest gains are expected during asymmetric or rapidly changing traffic demand, when fixed-time plans cannot

redistribute green time quickly. Under very light traffic, differences may be smaller because the network is uncongested. The third expected result concerns forecasting. Graph-based and other deep-learning approaches are designed to capture temporal and spatial dependencies in road networks (Zhang et al., 2019; Guo et al., 2019). Accurate short-term prediction should allow the controller to react before queues fully develop. Prediction error is therefore expected to be negatively associated with control benefit: inaccurate forecasts can lead to premature or unnecessary phase extensions. Perception accuracy also constrains overall system performance. False negatives underestimate queue demand, while false positives may waste green time. A lightweight detector should therefore be selected by considering both mAP and edge inference speed. Redmon et al. (2016)

demonstrated the importance of real-time object detection, but actual deployment must evaluate the speed-accuracy trade-off on the selected edge hardware.

Discussion

The methodology integrates transportation and computing performance rather than treating them separately. A traffic-management algorithm may reduce waiting time in simulation but still be unsuitable for deployment if inference is too slow, network dependence is excessive or edge hardware becomes overloaded. Conversely, a very fast edge model is not useful if perception or forecasting errors lead to poor traffic decisions. The evaluation therefore combines AI accuracy, latency, resource utilization and road-network outcomes. Deep-learning reviews in ITS emphasize the growing role of data-driven forecasting and decision support (Guerrero-Ibañez & Contreras-Castillo, 2021; Akhtar & Moridpour, 2021; Kashyap et al., 2021; Kumar & Raubal, 2021). The proposed architecture extends that logic by positioning AI near intersections, enabling local autonomy even when the cloud link is degraded. The cloud remains useful for citywide analytics, long-term retraining and historical storage, creating a hybrid rather than purely edge-only system.

Conclusion

This methodology paper proposes an integrated experimental framework for an artificial-intelligence-based traffic management system using edge computing. The system combines real-time perception, short-term traffic forecasting and adaptive signal control at roadside or intersection-level edge nodes. The evaluation compares fixed-time control, cloud-based AI and edge-based AI under identical traffic conditions and measures both transportation outcomes and computing/network performance. The principal hypothesis is that edge AI will reduce control-response latency and network bandwidth demand while preserving the traffic-efficiency advantages of modern AI methods. Deep reinforcement learning is expected to reduce waiting time and queue length under dynamic demand, while local perception and forecasting provide the state information required for responsive control. The methodology is intentionally modular: alternative object detectors, forecasting models or reinforcement-learning algorithms can be substituted while keeping the same evaluation protocol. Future empirical work should validate simulation results on a physical intersection testbed or roadside edge device, include

night/rain/occlusion conditions for computer vision, examine cybersecurity and privacy, and assess energy consumption during continuous inference. Federated or collaborative learning across intersections may also reduce the need to centralize raw traffic data. Even without these extensions, the proposed design provides a clear step-by-step basis for assessing whether edge intelligence can improve real-time urban traffic management.

References

- Shi, W., Cao, J., Zhang, Q., Li, Y., & Xu, L. (2016). Edge computing: Vision and challenges. *IEEE Internet of Things Journal*, 3(5), 637-646. <https://doi.org/10.1109/JIOT.2016.2579198>
- Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 779-788. <https://doi.org/10.1109/CVPR.2016.91>
- Wang, S., Xie, X., Huang, K., & Zeng, J. (2019). Deep reinforcement learning-based traffic signal control using high-resolution event-based data. *Entropy*, 21(8), 744. <https://doi.org/10.3390/e21080744>
- Gong, Y., Abdel-Aty, M., Cai, Q., & Rahman, M. S. (2019). Decentralized network level adaptive signal control by multi-agent deep reinforcement learning. *Transportation Research Interdisciplinary Perspectives*, 1, 100020. <https://doi.org/10.1016/j.trip.2019.100020>
- Zhang, Y., Cheng, T., & Ren, Y. (2019). A graph deep learning method for short-term traffic forecasting on large road networks. *Computer-Aided Civil and Infrastructure Engineering*, 34(10), 877-896. <https://doi.org/10.1111/mice.12450>
- Guo, S., Lin, Y., Feng, N., Song, C., & Wan, H. (2019). Attention based spatial-temporal graph convolutional networks for traffic flow forecasting. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01), 922-929. <https://doi.org/10.1609/aaai.v33i01.3301922>
- Li, D., Wu, J., Xu, M., Wang, Z., & Hu, K. (2020). Adaptive traffic signal control model on intersections based on deep reinforcement learning. *Journal of Advanced Transportation*, 2020, 6505893. <https://doi.org/10.1155/2020/6505893>

Zhou, X., Ke, R., Yang, H., & Liu, C. (2021). When intelligent transportation systems sensing meets edge computing: Vision and challenges. *Applied Sciences*, 11(20), 9680. <https://doi.org/10.3390/app11209680>

Kumar, N., & Raubal, M. (2021). Applications of deep learning in congestion detection, prediction and alleviation: A survey. *Transportation Research Part C: Emerging Technologies*, 133, 103432. <https://doi.org/10.1016/j.trc.2021.103432>

Akhtar, M., & Moridpour, S. (2021). A review of traffic congestion prediction using artificial intelligence. *Journal of Advanced Transportation*, 2021, 8878011. <https://doi.org/10.1155/2021/8878011>

Guerrero-Ibañez, J., & Contreras-Castillo, J. (2021). Deep learning support for intelligent

transportation systems. *Transactions on Emerging Telecommunications Technologies*, 32(3), e4169. <https://doi.org/10.1002/ett.4169>

Kashyap, A. A., Raviraj, S., Devarakonda, A., Nayak K, S. R., Santhosh, K. V., & Bhat, S. J. (2021). Traffic flow prediction models - A review of deep learning techniques. *Cogent Engineering*, 8(1), 2010510. <https://doi.org/10.1080/23311916.2021.2010510>

Paul, A., & Mitra, S. (2022). Deep reinforcement learning based cooperative control of traffic signal for multi-intersection network in intelligent transportation system using edge computing. *Transactions on Emerging Telecommunications Technologies*, 33(11), e4588. <https://doi.org/10.1002/ett.4588>