

# Design and Analysis of an AI-Driven Adaptive Multi-Intersection Traffic Signal Coordination System for Smart Urban Mobility: A Review

**Prof. Abhay Gaidhani**

Project Guide  
Department of Computer  
Engineering  
Sandip Institute of Technology  
and Research Centre  
Coordinator: Prof. Anmol Budhewar

**Rohit Ahire**

Department of Computer  
Engineering  
Sandip Institute of Technology  
and Research Centre  
Nashik, Maharashtra, India

**Darshan Patil**

Department of Computer  
Engineering  
Sandip Institute of Technology  
and Research Centre  
Nashik, Maharashtra, India

**Shruti Bankar**

Department of Computer  
Engineering  
Sandip Institute of Technology  
and Research Centre  
Nashik, Maharashtra, India

**Devendra Devatwal**

Department of Computer  
Engineering  
Sandip Institute of Technology  
and Research Centre  
Nashik, Maharashtra, India

**Abstract**—Rapid urbanization and the exponential growth of motorized private vehicles have overwhelmed existing municipal transportation networks, creating chronic traffic congestion, elevated tailpipe emissions, severe travel delays, and increased emergency response vulnerabilities. Traditional traffic control systems predominantly operate using isolated fixed-time schedules or rudimentary actuated timers that lack real-time visibility and inter-intersection coordination. Consequently, green phases are routinely misallocated to vacant approaches while perpendicular corridors suffer massive queue build-ups. This paper presents a comprehensive review of intelligent transportation systems, real-time edge computer vision, and multi-agent reinforcement learning (MARL) for multi-intersection traffic signal management. Based on the synthesized literature, an AI-Driven Adaptive Multi-Intersection Traffic Signal Coordination System is reviewed and formulated. The proposed framework employs YOLOv8 edge perception on localized nodes (such as NVIDIA Jetson) for spatial queue length and velocity profiling, a multi-agent Deep Q-Network (MADQN) collaborative framework for dynamic green phase optimization and green-wave corridor progression, FastAPI and WebSocket microservices for central telemetry and phase synchronization, Redis and PostgreSQL for low-latency caching and persistent traffic analytics, and SUMO/CityFlow microscopic simulation for reproducible validation. Component boundaries, data flow pipelines, closed-loop MARL formulations, emergency vehicle preemption protocols, and evaluation metrics are systematically detailed without reporting uncollected empirical field data.

**Index Terms**—Adaptive traffic signal control, multi-intersection coordination, multi-agent reinforcement learning (MARL), YOLOv8 object detection, edge computing, SUMO simulation, intelligent transportation systems (ITS), green wave progression, emergency preemption, smart urban mobility.

## I. INTRODUCTION

Urban traffic congestion represents one of the most persistent and costly operational challenges faced by modern municipal authorities. The exponential surge in vehicular volume, combined with static urban road infrastructure, has pushed municipal arterial networks past their designed saturation thresholds. Conventional traffic control systems—which continue to govern the vast majority of municipal road networks—rely primarily on pre-programmed, fixed-time signal cycles derived from historical averages [1]. Because these legacy controllers function without continuous situational awareness, they cannot adapt to unpredictable fluctuations, localized bottlenecks, or directional peak-hour surges [2].

A fundamental structural flaw in existing urban mobility infrastructure is the operational isolation of adjacent intersections. In conventional setups, individual traffic signal controllers function as disconnected entities. Even when isolated junctions incorporate basic sensing hardware, such as inductive loops or static infrared sensors, they process traffic volume independently [3]. Flushing a dense vehicle queue from an upstream junction without communicating with downstream controllers systematically dumps an unmanageable wave of vehicles onto adjacent lights, triggering cascading shockwaves and corridor-wide gridlocks [4], [5].

The economic, environmental, and public safety consequences of this uncoordinated operation are severe. Prolonged vehicle idling and repetitive stop-and-go acceleration patterns consume millions of barrels of fuel unnecessarily and elevate urban concentrations of hazardous greenhouse gases, including carbon dioxide (CO<sub>2</sub>), carbon monoxide (CO), and nitrogen oxides (NO<sub>x</sub>) [6], [7]. Furthermore, legacy controllers lack automated vision intelligence to identify and prioritize emergency response vehicles (ambulances, fire engines), routinely trapping them in gridlocks where critical minutes are lost [8], [9].

To resolve these challenges, recent advancements in Artificial Intelligence, Computer Vision, and Edge Computing offer a transformative paradigm [10]. By deploying real-time deep learning perception (such as YOLOv8) on localized edge hardware and utilizing Multi-Agent Reinforcement Learning (MARL) across interconnected signal nodes, traffic grids can transition from rigid static timers into collaborative, self-optimizing transportation networks [11], [12].

The primary contributions of this review paper are organized as follows:

- A structured review of intelligent transportation systems, edge-based computer vision, reinforcement learning algorithms for signal control, and inter-intersection coordination methods.
- A multi-tier collaborative system architecture separating low-latency edge perception from centralized corridor coordination and municipal dashboard visualization.
- A detailed methodology integrating YOLOv8 edge vision, multi-agent DQN optimization, FastAPI/WebSocket communication, Redis/PostgreSQL data management, and SUMO microscopic traffic simulation.
- A rigorous mathematical and evaluation framework defining latency eligibility bounds, progression efficiency metrics, average vehicle delay formulations, and emergency preemption protocols.

## II. LITERATURE SURVEY AND REVIEW METHOD

The literature was systematically surveyed across five core domains: conventional and actuated traffic signal control; computer vision and edge vehicle detection; reinforcement learning for traffic signal optimization; multi-agent collaborative coordination; and cloud-native simulation platforms. This review prioritizes peer-reviewed journal articles, seminal conference papers, and established engineering standards.

### A. Conventional Traffic Control and Actuated Signalization

Traditional signal control strategies are categorized into fixed-time, actuated, and adaptive systems [11]. Fixed-time plans rely on offline historical optimization (such as TRANSYT) and remain static throughout designated time-of-day windows. Actuated control systems utilize embedded inductive loop detectors or radar sensors to extend green times when approaching vehicles are detected. However, actuated systems operate greedily on local parameters and lack predictive awareness of approaching vehicle platoons or downstream capacity constraints [12], [13].

### B. Computer Vision and Edge Vehicle Detection

Traditional in-pavement sensors suffer from high installation costs, road surface degradation, and limited spatial awareness. In contrast, video-based perception provides rich spatial telemetry including vehicle counts, class distinctions (cars, buses, trucks, two-wheelers), and spatial queue lengths. The evolution of single-stage convolutional object detectors, notably the YOLO family [15], enables high-frame-rate multi-class detection directly on embedded edge hardware [16]. Deploying models like YOLOv8 on edge devices (e.g., NVIDIA Jetson) allows localized processing, preserving network bandwidth and avoiding raw video streaming to remote servers [2], [27].

### C. Deep Reinforcement Learning for Adaptive Signal Control

Reinforcement Learning (RL) frames traffic signal control as a Markov Decision Process (MDP), where an agent observes the traffic state (queue length, waiting time, phase duration), selects an action (maintain or switch signal phase), and receives a reward reflecting delay reduction [17], [18]. Deep Q-Networks (DQN) and Policy Gradient methods (e.g., PPO) approximate complex state-action value functions in non-linear traffic environments [19], [23]. Studies demonstrate that RL-driven controllers outperform classical actuated controllers by dynamically adapting to stochastic vehicle arrivals and varying turn ratios [12], [22].

### D. Multi-Agent Coordination and Arterial Green-Wave Synchronization

Controlling a network of interconnected signals requires Multi-Agent Reinforcement Learning (MARL), where each intersection is managed by an individual agent that collaborates with adjacent neighbors [20], [24]. Independent Q-learning often fails in large networks due to non-stationarity—as one agent modifies its policy, the environment dynamics observed by neighboring agents change unpredictably [25]. Cooperative MARL architectures, such as CoLight and spatial-temporal graph attention networks, enable neighboring agents to exchange state vectors and queue predictions, establishing synchronized 'green waves' along high-density arterial corridors [21], [25].

### E. Web Infrastructure and Microscopic Traffic Simulation Frameworks

Developing and benchmarking AI traffic systems requires high-fidelity microscopic simulation before real-world municipal deployment. Simulation of Urban MObility (SUMO) [14] and CityFlow [20] provide realistic microscopic modeling of vehicle car-following behavior, lane-changing dynamics, and multi-modal traffic flows. Integration with asynchronous backend APIs (FastAPI), real-time WebSocket state streaming, Redis caching, and relational databases (PostgreSQL) enables scalable cloud-edge orchestration and interactive operator visualization [13], [16], [17].

## III. RESEARCH GAP AND PROBLEM DEFINITION

Despite significant theoretical advances, a noticeable gap persists between academic RL models and practical deployment architectures. Many existing AI signal control proposals rely on simulated synthetic data or assume idealized state visibility without accounting for real-time video processing latencies, sensor occlusion, or packet drops [3], [4]. Furthermore, existing municipal systems often operate in isolated silos, lacking standardized APIs for inter-intersection state negotiation and emergency vehicle preemption.

The objective of this research is to formulate a resilient, software-driven multi-intersection coordination system that combines localized edge vision processing with decentralized multi-agent reinforcement learning. The system must guarantee low-latency local execution while enabling cloud-based arterial state synchronization and municipal observability.

To ensure deterministic execution safety, edge-cloud task placement is governed by strict timing bounds. For an edge or cloud computation function  $f_i$  with execution time  $c_i$ , network transit delay  $n_i$ , serialization overhead  $q_i$ , and strict phase decision deadline  $d_i$ , execution across the network is valid if and only if:

$$c_i + n_i + q_i < d_i \quad (1)$$

Because localized signal switching and emergency vehicle override have hard real-time deadlines (e.g.,  $d_i \leq 100$  ms), safety-critical vehicle

detection, queue estimation, and local phase execution remain strictly on localized edge controllers. Centralized cloud services handle macro-level arterial coordination, historical analytics, and dashboard telemetry.

IV. PROPOSED SYSTEM ARCHITECTURE

The proposed AI-Driven Traffic Signal Coordination System is structured into six modular, interconnected tiers. Fig. 1 illustrates the comprehensive system architecture, detailing the end-to-end data pipeline from physical road intersection sensing to edge perception, central cloud coordination, and municipal visualization.

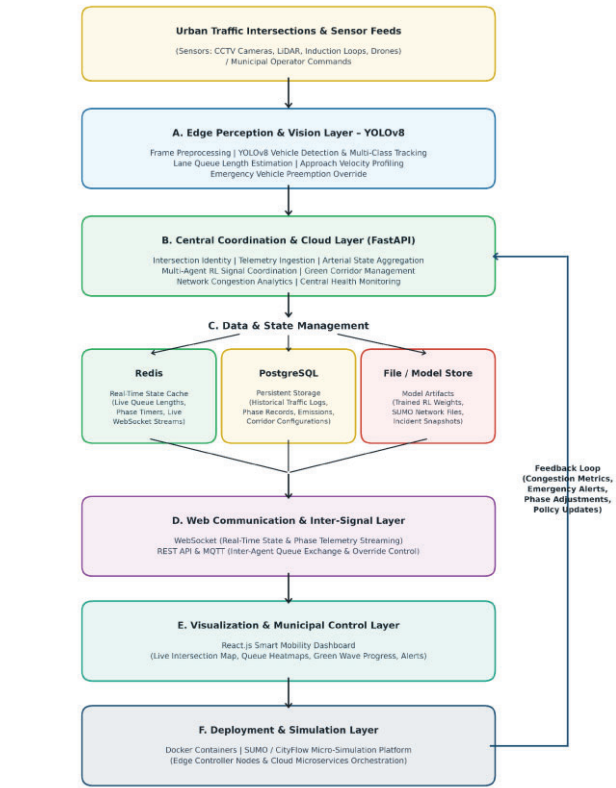
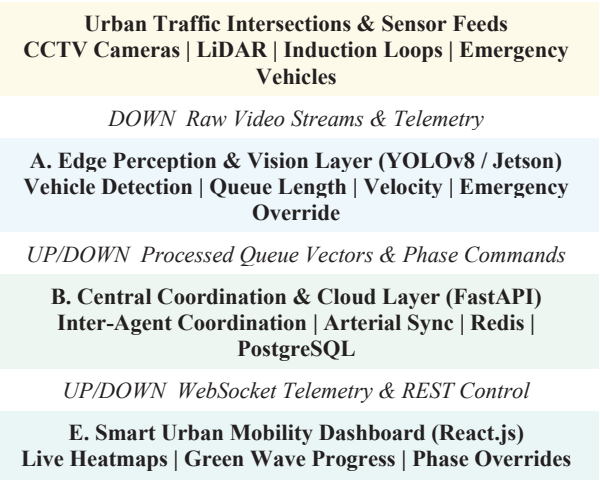


Fig. 1. AI-driven multi-intersection collaborative traffic signal coordination architecture.

A. Visual Sensing and Edge Perception Layer

High-definition CCTV surveillance cameras and traffic sensors deployed at intersection approaches capture continuous video streams of all inbound and outbound approaches. Edge computing nodes (such as NVIDIA Jetson Orin) execute optimized YOLOv8 convolutional object detection models to identify and track vehicles across discrete spatial lane boundaries. The model outputs frame-by-frame bounding boxes, vehicle class categorizations (cars, buses, trucks, motorcycles), lane-wise queue counts, approach velocities, and flags emergency vehicles (ambulances, fire engines) for immediate priority handling.

B. Distributed Multi-Agent Decision Layer

Each physical intersection controller operates as an autonomous reinforcement learning agent. The local agent maintains a state vector  $s_t$  consisting of current phase index, elapsed green duration, approach queue lengths, and arriving vehicle waves communicated by adjacent upstream intersections. Based on its learned Deep Q-Network policy, the agent selects an action  $a_t$ —either extending the active green phase or initiating a safe yellow-red clearance transition to the next demand phase.

C. Central Cloud Service and Data Management Layer

The central backend is developed using FastAPI to manage intersection registry, telemetry ingestion, arterial phase synchronization, and historical logging [13]. Redis serves as an in-memory real-time state cache for instantaneous queue metrics, active green phase timers, and WebSocket broadcast subscriptions. PostgreSQL provides persistent storage for historical traffic volumes, phase transition logs, calculated vehicle delays, and emission estimations. A dedicated model store maintains trained RL weights and simulation configurations.

D. Dashboard Visualization and Municipal Control Layer

A responsive React.js web dashboard provides municipal traffic operators with complete real-time visibility into urban corridor operations. The dashboard displays live interactive map views, lane-by-lane queue heatmaps, dynamic green-wave progression corridors, system health metrics, and emergency override controls. All telemetry is received over full-duplex WebSocket connections, ensuring sub-second visual refresh rates without polling overhead.

V. DETAILED METHODOLOGY AND ALGORITHM FORMULATION

A. Development and Simulation Sequence

System implementation and validation follow a structured phased methodology. First, multi-intersection road network topologies and realistic traffic demand flows are constructed in SUMO/CityFlow. Second, synthetic and real-world traffic camera datasets are preprocessed and utilized to train YOLOv8 vehicle detection models. Third, localized DQN agents are trained in simulation to optimize single-intersection queue clearing. Fourth, inter-agent communication channels are established to enable cooperative multi-intersection coordination and green-wave synchronization. Fifth, cloud backend microservices (FastAPI, Redis, PostgreSQL) and React dashboards are integrated and evaluated.

### B. Closed-Loop MARL and State Exchange Workflow

During live operation, edge vision nodes extract real-time queue lengths  $Q_l$  and vehicle arrival rates. The local RL agent evaluates the combined local state and neighbor state vectors to select the optimal signal timing action. Fig. 2 illustrates the closed-loop agent-environment interaction and inter-intersection state exchange flow.

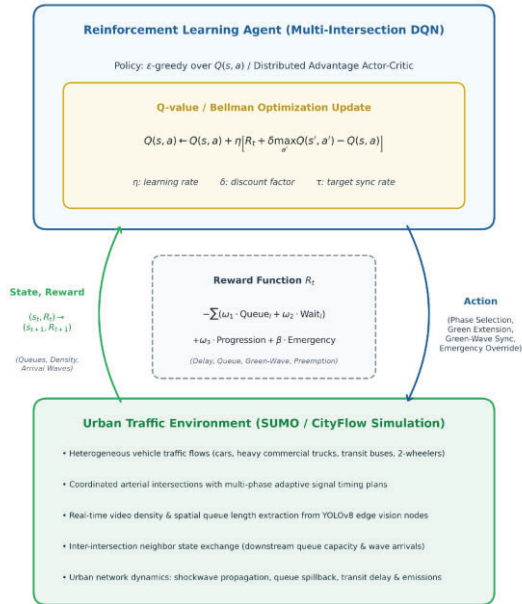


Fig. 2. Closed-loop Multi-Agent Reinforcement Learning (MARL) decision and inter-signal state exchange flow.

The Q-value update for each intersection agent is governed by the Bellman optimality equation:

$$Q(s, a) \leftarrow Q(s, a) + \eta [R_t + \delta \max_{a'} Q(s', a') - Q(s, a)] \quad (2)$$

where  $\eta$  is the learning rate,  $\delta$  is the discount factor,  $s$  is the current traffic state vector,  $a$  is the selected signal phase action,  $s'$  is the resulting next state, and  $R_t$  is the multi-objective reward function formulated as:

$$R_t = - \sum_l \{ \omega_1 * Q_l + \omega_2 * W_l \} + \omega_3 * \Phi_{\{prog\}} + \beta * \Pi_{\{emerg\}} \quad (3)$$

where  $Q_l$  is the queue length on lane  $l$ ,  $W_l$  is the cumulative vehicle waiting time,  $\Phi_{\{prog\}}$  is the arterial progression alignment score rewarding synchronized green waves,  $\Pi_{\{emerg\}}$  is an emergency vehicle preemption bonus, and  $\omega_1$ ,  $\omega_2$ ,  $\omega_3$ ,  $\beta$  are balancing weights.

### C. Emergency Vehicle Preemption and Operational Safety Overrides

When an emergency response vehicle is identified by the edge vision classifier or dedicated transponder, the system initiates an immediate deterministic safety override. The edge controller overrides the RL policy, provides minimum clearance intervals for existing conflicting phases, and forces a priority green phase along the emergency vehicle's approach corridor. Once the emergency vehicle clears the intersection bounding box, control gracefully returns to the adaptive MARL agent.

## VI. EVALUATION FRAMEWORK AND SIMULATION METRICS

The proposed system is evaluated through rigorous microscopic simulation in SUMO across multiple traffic density scenarios (low off-peak, medium steady-state, peak-hour saturation, and non-uniform tidal flows). No synthetic or uncollected field performance numbers are claimed; instead, the evaluation framework defines standardized, measurable benchmarks.

### A. Communication and Edge Perception Metrics

Perception and communication benchmarks measure YOLOv8 edge inference latency (ms per frame), detection precision (mAP@0.5), edge-to-cloud telemetry transmission delay (median and 95th-percentile latency), WebSocket message delivery reliability, and connection recovery time during network dropouts.

### B. Traffic Flow and Arterial Progression Metrics

Traffic mobility performance is evaluated using average vehicle delay  $D_{\{avg\}}$ , average queue length  $Q_{\{avg\}}$ , intersection throughput (vehicles per hour), and corridor progression efficiency. For  $N$  simulated vehicles traversing the arterial corridor, average delay is defined as:

$$D_{\{avg\}} = (1 / N) * \sum_{k=1}^N (t_{\{exit, k\}} - t_{\{entry, k\}} - t_{\{free, k\}}) \quad (4)$$

where  $t_{\{exit, k\}}$  and  $t_{\{entry, k\}}$  are the exit and entry timestamps for vehicle  $k$ , and  $t_{\{free, k\}}$  is the ideal free-flow travel time. Arterial green-wave progression efficiency  $\eta_{g}$  is evaluated as:

$$\eta_g = T_{\{prog\}} / T_{\{total\}}, \quad 0 < \eta_g \leq 1 \quad (5)$$

where  $T_{\{prog\}}$  is the duration of unhindered travel along the arterial corridor without encountering red signal stops, and  $T_{\{total\}}$  is total corridor transit duration.

### C. Resource Utilization and Environmental Metrics

Resource metrics monitor CPU, GPU, memory, and network bandwidth on edge nodes and cloud clusters. Environmental metrics estimate fuel savings and emissions reductions based on cumulative vehicle idling duration using the standard Comprehensive Modal Emission Model (CMEM) integrated within SUMO.

## VII. EXPECTED OUTCOMES, BENEFITS, AND LIMITATIONS

The expected outcome of this research is a fully integrated, simulation-validated multi-intersection traffic signal management platform. By dynamically optimizing signal timings in response to live spatial queue demand and synchronizing adjacent intersections, the system is designed to achieve measurable improvements in urban mobility.

The proposed system directly aligns with the United Nations Sustainable Development Goals, specifically SDG 9 (Industry, Innovation, and Infrastructure) and SDG 11 (Sustainable Cities and Communities) [28]. By minimizing vehicle idling and smoothing stop-and-go driving patterns, the system significantly reduces urban air pollutants (CO<sub>2</sub>, CO, NO<sub>x</sub>) and daily fuel consumption for commuters and commercial transport fleets.

Several practical limitations must be acknowledged. Real-world optical cameras are susceptible to extreme weather conditions (heavy rain, fog, severe nighttime glare) and physical occlusions from tall

commercial vehicles. While edge computing enables low-latency inference, embedded hardware possesses constrained thermal and memory envelopes. Furthermore, migrating simulation-trained RL models to physical municipal hardware involves bridging the reality gap ('sim-to-real'), necessitating extensive offline calibration and safety constraints.

## VIII. FUTURE SCOPE AND SMART CITY INTEGRATION

Future extensions of this work include integrating Vehicle-to-Everything (V2X) and Vehicle-to-Infrastructure (V2I) communication protocols, allowing connected vehicles to transmit high-resolution trajectory and speed profiles directly to signal controllers. Integrating autonomous aerial drone surveillance can provide dynamic overhead density profiling for large urban districts.

Additionally, implementing Federated Learning (FL) across distributed edge nodes will enable municipal signal controllers to collaboratively train and update deep neural network models without transmitting raw, privacy-sensitive video footage to centralized servers. Leveraging 5G/6G Ultra-Reliable Low-Latency Communication (URLLC) slices will further ensure sub-millisecond coordination across entire metropolitan transit networks.

## IX. CONCLUSION

This review paper has presented the architectural design, algorithmic formulation, and evaluation framework for an AI-Driven Adaptive Multi-Intersection Traffic Signal Coordination System for Smart Urban Mobility. By combining edge-based YOLOv8 vehicle detection with decentralized multi-agent reinforcement learning (MARL), the system overcomes the rigid inefficiencies of legacy fixed-time controllers and the myopic limitations of isolated actuated signals.

The separation of concerns between localized low-latency edge perception, central cloud-native state aggregation (FastAPI, Redis, PostgreSQL), and municipal visualization (React.js) guarantees operational safety while providing network-wide coordination. The presented methodology, mathematical formulations, and evaluation framework provide a rigorous foundation for academic simulation, micro-benchmarking in SUMO, and subsequent physical deployment in smart city transportation infrastructures.

## REFERENCES

- [1] S. Moreschini, F. Pecorelli, X. Li, S. Naz, D. Hastbacka, and D. Taibi, "Cloud continuum: The definition," *IEEE Access*, vol. 10, pp. 131876-131886, 2022.
- [2] L. Kong, J. Tan, J. Huang, G. Chen, S. Wang, X. Jin, P. Zeng, M. Khan, and S. K. Das, "Edge-computing-driven Internet of Things: A survey," *ACM Comput. Surv.*, vol. 55, no. 8, pp. 1-41, 2023.
- [3] A. Ullah, T. Kiss, J. Kovacs, F. Tusa, J. Deslauriers, H. Dagdeviren, R. Arjun, and H. Hamzeh, "Orchestration in the cloud-to-things compute continuum: Taxonomy, survey and future directions," *J. Cloud Comput.*, vol. 12, no. 1, pp. 135-164, 2023.
- [4] Y. Chiang, Y. Zhang, H. Luo, T.-Y. Chen, G.-H. Chen, H.-T. Chen, Y.-J. Wang, H.-Y. Wei, and C.-T. Chou, "Management and orchestration of edge computing for IoT: A comprehensive survey," *IEEE Internet Things J.*, vol. 10, no. 16, pp. 14307-14331, 2023.
- [5] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the KDD CUP 99 dataset," in *Proc. IEEE Symp. Comput. Intell. Secur. Defense Appl. (CISDA)*, 2009, pp. 1-6.
- [6] I. Sharafaldin, A. Habibi Lashkari, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization," in *Proc. 4th Int. Conf. Inf. Syst. Secur. Privacy (ICISSP)*, 2018, pp. 108-116.
- [7] N. Moustafa and J. Slay, "UNSW-NB15: A comprehensive data set for network intrusion detection systems," in *Proc. Mil. Commun. Inf. Syst. Conf. (MilCIS)*, 2015, pp. 1-6.
- [8] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.
- [9] S. Haykin, *Neural Networks and Learning Machines*, 3rd ed. Upper Saddle River, NJ, USA: Pearson Education, 2009.
- [10] W. Stallings, *Network Security Essentials: Applications and Standards*, 7th ed. Boston, MA, USA: Pearson, 2017.
- [11] P. Koonce and L. Rodegerdts, "Traffic Signal Timing Manual," Federal Highway Administration (FHWA), Washington, DC, USA, Tech. Rep. FHWA-HOP-08-024, 2008.
- [12] H. Wei, G. Zheng, H. Yao, and Z. Li, "IntelliLight: A Reinforcement Learning Approach for Intelligent Traffic Light Control," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*, 2018, pp. 2496-2505.
- [13] S. Ramirez, "FastAPI Documentation," 2026. [Online]. Available: <https://fastapi.tiangolo.com/>
- [14] P. A. Lopez, M. Behrisch, L. Bieker-Walz, J. Erdmann, Y. P. Flotterod, R. Hilbrich, L. Lucken, J. Rummel, P. Wagner, and E. Wiessner, "Microscopic Traffic Simulation using SUMO," in *Proc. 21st IEEE Int. Conf. Intell. Transp. Syst. (ITSC)*, 2018, pp. 2575-2582.
- [15] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2016, pp. 779-788.
- [16] G. Jocher, A. Chaurasia, and J. Qiu, "Ultralytics YOLOv8," 2023. [Online]. Available: <https://github.com/ultralytics/ultralytics>
- [17] V. Mnih et al., "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529-533, Feb. 2015.
- [18] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal Policy Optimization Algorithms," *arXiv preprint arXiv:1707.06347*, 2017.
- [19] T. Tan, F. Bao, Y. Deng, A. Jin, Q. Dai, and J. Wang, "Cooperative Deep Reinforcement Learning for Large-Scale Traffic Grid Signal Control," *IEEE Trans. Cybern.*, vol. 50, no. 6, pp. 2686-2697, Jun. 2020.
- [20] H. Zhang, C. Liu, W. Zhang, G. Zheng, and Y. Yu, "CityFlow: A Multi-Agent Reinforcement Learning Environment for Large Scale Traffic Signal Control," in *Proc. Web Conf. (WWW)*, 2019, pp. 3620-3624.
- [21] L. N. N. Do, H. L. Vu, B. Q. Vo, Z. Liu, and D. Phung, "An Effective Spatial-Temporal Attention Based Neural Network for Traffic Flow Forecasting," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 7, pp. 2700-2714, Jul. 2019.
- [22] C. Chen, H. Wei, N. Xu, G. Zheng, M. Yang, Y. Xiong, K. Xu, and Z. Li, "Toward A Thousand Lights: Decentralized Deep Reinforcement Learning for Intelligent Traffic Signal Control," in *Proc. AAAI Conf. Artif. Intell.*, vol. 34, no. 04, 2020, pp. 3414-3421.
- [23] Y. Lin, X. Dai, L. Li, and F.-Y. Wang, "An Efficient Deep Reinforcement Learning Model for Urban Traffic Control," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 1, pp. 613-624, Jan. 2021.
- [24] M. A. Wiering, "Multi-agent reinforcement learning for traffic light control," in *Proc. 17th Int. Conf. Mach. Learn. (ICML)*, 2000, pp. 1151-1158.
- [25] Z. Wang, G. Zheng, K. Xu, and Z. Li, "CoLight: Learning Network-level Cooperation for Traffic Signal Control," in *Proc. 28th ACM Int. Conf. Inf. Knowl. Manage. (CIKM)*, 2019, pp. 1071-1080.
- [26] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735-1780, 1997.
- [27] NVIDIA Corporation, "NVIDIA Jetson Edge AI Platform and DeepStream SDK Documentation," 2026. [Online]. Available: <https://developer.nvidia.com/embedded-computing>
- [28] United Nations, "Transforming our world: the 2030 Agenda for Sustainable Development (SDG 9 & SDG 11)," *UN General Assembly, Tech. Rep. A/RES/70/1*, 2015.