frontiers | Frontiers in Mechanical Engineering

TYPE Original Research
PUBLISHED 24 September 2025
DOI 10.3389/fmech.2025.1650918

OPEN ACCESS

EDITED BY
Xuping Zhang,
Aarhus University, Denmark

REVIEWED BY
Paolo Di Giamberardino,
Sapienza University of Rome, Italy
Viet Q. Vu,
Thai Nguyen University of Technology, Vietnam

*CORRESPONDENCE
Ilesanmi Daniyan,
iadaniyan@bellsuniversity.edu.ng

RECEIVED 24 June 2025
ACCEPTED 01 September 2025
PUBLISHED 24 September 2025

CITATION
Olusanya OO, Owosho Y, Daniyan I, Elegbede AW, Sodipo QB, Adeodu A, Phuluwa HS, Ramasu TK and Kana-Kana Katumba MG (2025) Multi-agent reinforcement learning framework for autonomous traffic signal control in smart cities.
Front. Mech. Eng. 11:1650918.
doi: 10.3389/fmech.2025.1650918

Multi-agent reinforcement learning framework for autonomous traffic signal control in smart cities

Olamide O. Olusanya[1], Yetunde Owosho[2], Ilesanmi Daniyan[3,4]*, Adedayo W. Elegbede[1], Queen B. Sodipo[1], Adefemi Adeodu[5], Humbulani Simon Phuluwa[4], Tlotlo K. Ramasu[6] and Mukondeleli Grace Kana-Kana Katumba[6]

[1]Department of Computer Engineering, Bells University of Technology, Ota, Nigeria, [2]Department of Electrical/Electronics and Telecommunication Engineering, Bells University of Technology, Ota, Nigeria, [3]Department of Mechatronics Engineering, Bells University of Technology, Ota, Nigeria, [4]Department of Industrial Engineering & Engineering Management, University of South Africa, Florida, South Africa, [5]Department of Project Management, Bells University of Technology, Ota, Nigeria, [6]Department of Industrial Engineering, Tshwane University of Technology, Pretoria, South Africa

Introduction: The increasing urbanization across the world necessitate efficient traffic management especially in the emerging economies. This paper presents an intelligent framework aimed at enhancing traffic signal management within complex road networks through the creation and evaluation of a multi-agent reinforcement learning (MARL) framework.

Methods: The research explored how Reinforcement Learning (RL) algorithms can be employed to optimize the flow of traffic, lessen bottleneck, and enhance overall transportation safety and efficiency. Additionally, the research explored the design and simulation of a typical traffic environment that is, an intersection, defined and implemented a Multi-Agent System (MAS), and developed a Multi-Agent reinforcement learning model for traffic management within a simulated environment this model leverages actor-critics and deep Q Network (DQN) strategies for learning and coordination, and performed the evaluation of the MARL model. Novel approaches for decentralized decision-making and dynamic resource allocation were developed to enable real-time adaptation to changing traffic conditions and emergent situations. Performance evaluation using metrics such as waiting time, queue length, and congestion were carried out in the SUMO simulation platforms (Simulation of Urban Mobility) to evaluate the efficiency of the proposed solution in various traffic scenarios.

Results and Discussion: The outcome of the simulation conducted in this study showed an improvement in queue management and traffic flow by 64.5% and 70.0% respectively with improvement in performance of the proposed model over the episodes. The results show that the RL model policy showed better performance compared to the baseline policy, indicating that the model learned over different episodes. The results also show that the MARL-based approach performs better for decentralized traffic control systems in both scalability and

adaptability. The proposed solution supports real-time decision-making, reduces traffic congestion, and improves the efficiency of the urban transportation system.

KEYWORDS

smart cities, multi-agent systems, reinforcement learning, traffic signal control, intelligent transportation systems, SUMO simulation

1 Introduction

The increasing urbanization across the world necessitate efficient traffic management especially in the emerging economies. Effective traffic management is one of the important components of smart cities and technology plays a significant role in achieving this (Fadila et al., 2024). Urban traffic congestion is a growing concern in the emerging economies and smart cities, leading to time wastage, energy consumption, and carbon emissions (Kong et al., 2016). Traditional traffic signal control systems often rely on fixed-timing approach, actuated controllers or rule-based algorithms which fails to adapt to real-time traffic dynamics and complexities especially in urban cities. These challenge highlight the need for a more adaptive and intelligent solution that are more precise (Zhang et al., 2023).

Advances in Artificial Intelligence (AI) especially in RL have shown substantial capability for effective, intelligent and adaptive traffic signal control by allowing systems to learn optimal policies from real-time traffic data without the need for detailed programming (Medina-Salgado et al., 2022). Traffic scenarios are usually regarded as a multi-agent system because of the different intersections and interactions between vehicles. Thus, a single agent RL may not effectively scale or coordinate traffic especially in a decentralized environment resulting in suboptimal performance However, a MARL may sufficiently model each traffic signal and intersections as independent agents and learn optimal control policies through interaction and coordination of these agents. As such it may enable adaptability, scalability, efficient coordination and real time responsiveness of traffic signal control under different scenarios especially in decentralized or urban settings. The implementation of the MARL for traffic management contributes to the goals of smart cities by enabling efficient mobility through dynamic signal adjustment in real time, facilitation of real time data-driven decision-making relating to traffic infrastructure, development of intelligent transportation systems that will meet the demand of the urban population amongst others. Thus, leveraging on the potentials of the MARL techniques can considerably promote the efficiency and sustainability of urban transportation systems.

Conventional traffic management systems have been having difficulty coping with the increasing volume of vehicles, resulting in incompetence, delays, and natural concerns. The fast advancement in the use of motor vehicles has brought about ease for individuals and altered the structure of the overall transportation system. Likewise, making it a more convenient take-away distribution. Logistics workers utilize motorcycles as a means of transportation to navigate through the streets and lanes daily, resulting in major traffic congestion issues. During instances of traffic congestion, the vehicle moves at a significantly reduced speed (Su et al., 2020). New studies when explored offer opportunities for potential future developments of smart cities based on real-life situations (Javed et al., 2022). In addition, conventional systems which could also be called traditional systems do not have a robust informed system that can handle some of the emergences of autonomous vehicles and some other intelligent transportation technologies (Hasan et al., 2020). Therefore, forms an urgent need to develop a decentralized and adaptive solution that will give the relevance expected, hence optimizing traffic flow, enhancing safety, and reducing any form of environmental impact in the smart city environment (Mishra and Singh, 2023).

This research aimed to apply reinforcement learning and multi-agent systems for autonomous traffic management in smart cities. In achieving this, the design and simulation of a traffic environment were carried out at an intersection, using "SUMO" (Simulation of Urban Mobility) stating parameters such as agents, state, action, and reward. Another objective was to define and implement a multi-agent system in the simulated environment, which involved representing the traffic lights as agents and designing state representation, action spaces, and reward structures to help optimize traffic management. Additionally, this study aims to develop a MARL model for effective traffic management. The developed model was tested in a simulated environment, using Stable- Baselines3 under a decentralised (Independent Actor-Critic) and centralised (Centralized Actor–Critic) MARL situations. The performance evaluation of the MARL model was conducted using metrics such as waiting time, queue length and congestion to assess the efficiency of the traffic flow and also prevent traffic jam.

Recent advances in autonomous driving systems and intelligent transport systems (ITS) have engineered research and innovation in the areas of deep learning, sensor fusion, multi-agent systems (MAS) as well as traffic management. Gupta et al. (2021) as well as Yeong et al. (2021) discussed the innovations resulting from machine and deep learning models, sensor fusion technologies, with emphasis on their transformative roles in real-life situations. The authors indicated the need for a robust and scalable solutions especially in a complex environment.

Boukerche et al. (2020) and Khalil et al. (2024) focused on traffic predictions and controls while evaluating statistical and machine learning models to forecast traffic and ITS applications, the use of spatial-temporal graph neural networks (STGNN) for local traffic flow was explored by Belt et al. (2023). Likewise, intelligent traffic signal control methodologies using algorithms that evolve and deep reinforcement learning (DRL) were proposed by Al-Turki et al. (2020) and Rahman (2024) highlighting the benefits such as optimization and adaptability.

Several traditional model based approaches such as autoregressive moving averages (ARIMA), exponential smoothing, regression, KNN, SVR, the Kalman filter, etc., have been employed for the analysis and prediction of traffic flow or congestion in urban cities (Williams et al., 1998; Ding et al., 2010; Li et al., 2016; Xia et al., 2016; Chang et al., 2012). However, the use of

AI-based models in traffic management and forecasting have also been reported with a higher precision and efficiency compared to other models (Kumar et al., 2013; Lv et al., 2015; Ma et al., 2015; Zhu et al., 2016; Zhao et al., 2017; Duan et al., 2019).

For instance, Zhang et al. (2023) successfully and accurately employed the integrated convolutional Long Short-Term Memory (LSTM) and the Convolutional Neural Network (CNN) for prediction of urban traffic flow and congestion.

Medina-Salgado et al. (2022) found that AI-based model such as the deep learning model, as well as the ensemble and hybrid models perform better than the traditional based models when employed for traffic management and prediction. Mystakidis et al. (2025) indicated that the linear or statistical models are usually effective in capturing linear or stationary trends but may not sufficiently capture variation or non-linear relationships typical of urban traffic scenario. ML models outperforms statistical models in capturing non-linear relationships or trends but their accuracy may be affected by overfitting. On the other hand, DL models, such as LSTM, CNN, etc., excel in capturing temporal or spatial dependencies, but may be limited by volume of datasets, and computational resources. However, the ensemble model harnesses the strength of the individual models to offer a robust performance and adaptability across different traffic management scenarios.

Kong et al. (2016) employed the mobile sensor to analyse traffic congestion and the Particle Swarm Optimization (PSO) to predict traffic flow. The outcome of the study indicated that the proposed technique is accurate and stable in traffic congestion analysis and flow prediction. To improve the accuracy of traffic management, prediction and security Fadila et al. (2024) suggested the integration of the fourth industrial revolution technologies such as AI, Internet of Thing and Blockchain into the traffic management system. The authors identified some bottlenecks to this proposed solution such as users' acceptance, robustness and data availability.

While focusing on AI/ML-based treatment detection, within the transport system, cybersecurity was explored by Admass et al. (2024) while Rahman et al. (2021) worked on the challenges of ad hoc teamwork by making use of graph neural networks. MAS applications were further researched by Liu and Kohls (2010), Quallane et al. (2022), and Maldonado et al. (2024) who emphasized RL–based control, urban traffic coordination, and a standard MAS framework respectively.

Some perspectives on smart cities were also presented by Appio et al. (2019), Elassy et al. (2024), and Alfaro-Navarro et al. (2024) tackling ecosystem innovations, digital literacy, and sustainable ITS deployment. Almukhalfi et al. (2024) and Heidari et al. (2022) provided a detailed review of ML/DL roles in smart cities thereby identifying the research gaps in real-world scalability and hybrid models.

Despite various improvements and innovations, significant gaps persist: a lot of current solutions have failed to fully take advantage of the ability to decentralize, adapt, and take advantage of autonomous traffic management systems to form real-time decisions. These gaps outline the importance of more traffic systems that can sustainably improve traffic flow, advance safety, and also reduce environmental impact. Hence, this serves as an important motivation for researching the advancement of future technology to enhance the quality of human existence (Xia et al., 2023).

Additional contributions include simulation-based traffic assignment (Hui et al., 2023), AI in logistics and MaaS (Japiassu, 2024), and agent-based UTM evaluation platforms (Carramiñana et al., 2021). Environmental and policy-oriented studies by Hosseinian et al. (2024) and Jia (2021) offered insights into sustainable AV integration and emission reduction strategies. Collectively, these works emphasize the application of intelligent systems spanning AI, MAS, and RL in traffic and urban mobility management. However, challenges persist in system interoperability, real-time adaptability, and comprehensive multi-agent coordination, indicating significant avenues for future research.

Therefore, to bridge this gap, this study applied RL and Multi-Agent System (MAS) to create a decentralized, autonomous traffic management system for smart cities unlike previous methods that either relied on centralized control or static algorithms, this method utilized the strength of RL in active decision making and distributed nature of MAS to permit real-time coordination between different traffic agents (vehicles, traffic signals, and infrastructure). By stimulating urban traffic scenarios, the system can actively adjust to any changing situations thereby enabling better efficiency, scalability, and resilience in the solution that overtakes the traditional traffic management systems in environmental sustainability and effectiveness.

This study identified a significant research gap in the field of urban traffic management within smart cities. This gap primarily stems from the limitations of conventional centralized traffic management approaches. These traditional systems are characterized by a lack of agility and scalability required to handle the increasing complexities of urban traffic networks. They are described as rigid and inadequate in adjusting to various traffic patterns and demands, and they are not equipped to handle the emergence of autonomous vehicles and other intelligent transportation technologies.

Additionally, the literature review highlighted other unresolved issues and gaps in related research, such as:

1. Challenges with sensor performance and data fusion in complex urban environments underscore the need for robust calibration and real-time adaptability.
2. The need for practical implementation beyond theoretical frameworks and simulation models.
3. The persistent challenge of scalability and validation in more extensive and intricate traffic systems.
4. Deficiencies in multi-objective optimization.
5. A need for more exploration into the socio-economic implications, including equity and accessibility, and the integration of privacy and cybersecurity measures in AI-driven traffic management.

The study aimed to make contributions to knowledge, specifically by addressing the identified gap through its proposed MARL framework:

1. Introducing an innovative framework for MARL specifically for the decentralized management of urban traffic intersections.

2. Applying the concept of MAS and RL to improve traffic efficiency within the simulated environment.
3. Presenting a useful view of the application of intelligent and advanced traffic systems for the urban environment, which helps in creating better, scalable, and effective traffic management solutions.
4. Showing that this research can be applied to forthcoming advancements in self-driving traffic systems, tackling essential issues in smart cities' traffic management.

This study is significant in that it aligns with the goals of smart city development by enabling efficient mobility through dynamic signal adjustment in real time, reduction of traffic congestion, facilitation of real time data-driven decision-making relating to traffic infrastructure, development of intelligent transportation systems that will meet the demand of the urban population amongst others.

2 Methodology

2.1 Design and simulation of a typical traffic environment

The design and simulation of a typical traffic environment was done in the SUMO environment. SUMO is a tool that provides highly detailed traffic simulations with a wide range of vehicle and intersection types was utilized. Part of the goal of this work is to reduce overall vehicle waiting time at traffic lights, minimize congestion across intersections, and balance traffic load across multiple routes. It is based on the stated goal that the following parameters were defined for the simulation of the typical traffic environment which is the intersection:

1. Agents: Each traffic light at an intersection acts as an agent, or clusters of lights can form groups (e.g., agents for major intersections, arterial roads, etc.).
2. State Space: The information each agent uses to make decisions (e.g., the number of vehicles in each lane, vehicle waiting times, signal phases of neighboring intersections).
3. Action Space: The available actions for each agent (e.g., changing the traffic light phases, extending green light duration, activating turn signals).
4. Reward: A reward signal indicating the success or failure of each action, often based on the reduction of traffic congestion or vehicle wait times.

Figure 1 shows the generic model of an agent which illustrates the framework of reinforcement learning.

Thereafter, the road network was created, the network creation is a small grid of intersections for real-world maps imported from OpenStreetMap (OSM). Also, in setting up the simulation environment, a traffic flow definition was required. This entails the specification of vehicle routes, traffic volumes, and patterns that simulate peak-hour traffic, normal conditions, or random vehicle movements. Since SUMO was used in this work, TraCI (Traffic Control Interface) was used to interact with the simulation in real time.

Lastly, in the actualization of this research, Stable Baseline3 was used for the MARL algorithm to provide robust tools for reinforcement learning model training and support distributed learning which is essential for MARL setups while PettingZoo was used as Multi-Agent Extensions which provided built-in support for MARL tasks, making it easier to design agent interactions and environments. The tools and software used are:

1. Traffic Simulator: SUMO was used to stimulate urban traffic patterns and test agent behaviors in a controlled environment.'
2. Programming the Environment: Python was used as the primary language for developing the Reinforcement Learning algorithms and the libraries used were Tensorflow, Keras, and Matplotlib. Tensorflow is the core machine learning and deep learning framework used for scalable model building. The Keras was used to simplify the API for building and training the neural networks. While Matplotlib was used for the visualization of data and model performance
3. Framework Development: RLlib or custom implementations for Reinforcement Learning were used.
4. Hardware: A computer with high-performance GPUs for training deep neural networks (DNN) was used.

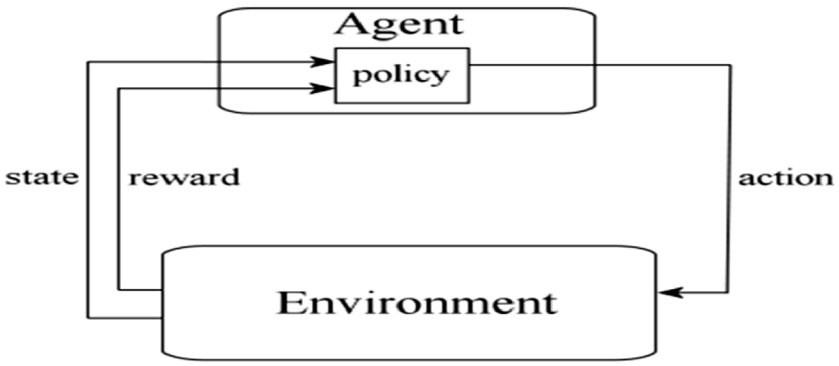

FIGURE 1
The generic model of an agent Source (Bakker and Groen, 2010).

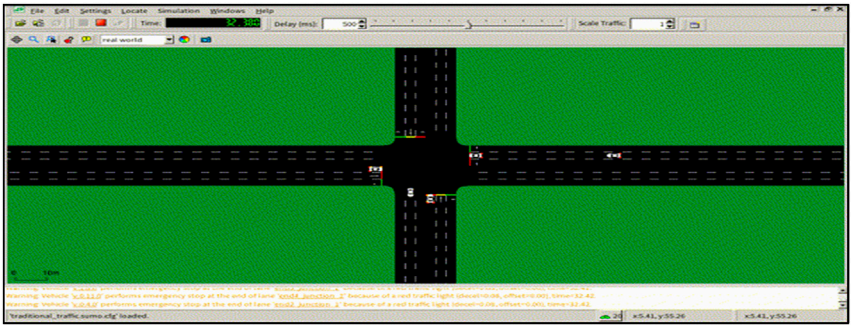

FIGURE 2
Overview of system architecture.

2.2 System Architecture overview

1. Agents: Each traffic light is denoted as an independent RL agent. These agents also monitor the local traffic conditions and develop optimal policies for the phases of the traffic light.
2. Collaborative efforts among Multi-Agents: The Agents collaborate indirectly by making use of shared traffic states to achieve comprehensive traffic optimization as shown in Figure 2.

2.3 State representation

Intersection Model: Each of the intersections has 4 arms (north, south, east, and west) each of which are divided into 20 presence cells. Out of these presence cells, 10 cells are for left-turning lanes while the other 10 cells are for straight/right-turning lanes.

1. State Vector: The binary vehicle presence is encoded such that 11 is the number of vehicles present and 00 is the number of vehicles absent an example is seen as State = [1, 0, 1, . . . , 0, 1, 0].
2. Action Space: Phases of a traffic light: The agent selects one out of the four predefined light phases.
3. North- South Advance: The green light is for straight/right turns in the north/south direction
4. East-West Left Advance: The green left turn light is for east/west traffic.
5. Time Schedule: The green phase was set for 10 s while the yellow phase was set for 4 s during transitions.

2.4 Reward function

1. Objective: Reduce the total waiting time for vehicles.
2. Formula: Reward = Δ(Cumulative Waiting Time) = positive reward which is equal to a reduction in waiting time and the adverse consequence is the extended waiting period.
3. Illustration: Suppose the time (tt) is the waiting period = 500 s, thus, Time t + 1 is the waiting time = 450 s. The reward = 500–450 = +50

Furthermore, a detailed sequence of experiments was carried out using a multi-agent reinforcement learning framework within a controlled urban traffic simulation. The agents, illustrated as the traffic lights went through training and evaluation during different episodes to improve the efficiency of traffic flow. During this training, the agents developed some policies that were utilized as reward-based reinforcement also using metrics like cumulative negative rewards and cumulative delay.

4. Criteria for Evaluation: The performance of the system was scrutinized through different metrics such as queue length, total traffic congestion or throughput, and the average delay per vehicle.
5. Simulated Scenarios: The simulation involved a wide range of traffic scenarios such as dynamic flows, high-density environments, peak traffic conditions, evaluation of robustness, and the adaptability of the systems.

2.5 Set up of evaluation and system training

The experiments were carried out in a controlled urban traffic environment, carefully crafted to mirror the complexities of real-life traffic behavior within a smart city framework. The environment is made up of several intersections, each overseen by an agent symbolizing a traffic light. The complexities of the road network expose the vigorous flow of vehicles that is described as a result of the fluctuations in density and demand thereby effectively stimulating different types of urban traffic scenarios, such as peak hours, unexpected traffic congestion, and low-density flow.

1. Intersections: The interconnected nodes were where the environment was designed hence, the actions of one of the traffic lights which was collectively affected by the ones surrounding it thereby, inspiring collective decision-making.
2. Traffic Flow: The vehicles that were assigned to different travel routes were modified into real-time scenarios in other to be able to cater to peak hour congestion and off-peak periods.
3. Interactions of the Agent: each of the agents functioned individually and autonomously while exchanging some specific state variables. (e.g., queue lengths and phase timing) which was used to decentralize the coordination all through the network. This environment ensured a well-structured and even more authentic framework. That was

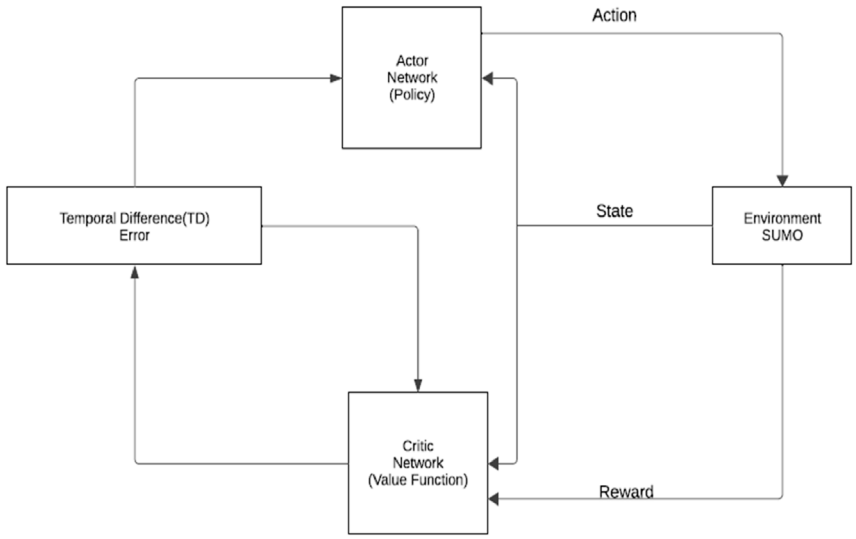

FIGURE 3
Diagram for Multi-Agent Reinforcement learning for traffic Light Control using Actot Critic Framework.

used for the training and evaluation or assessment of the developed Multi-gent Reinforcement Learning (MARL) system.

2.6 Definition and implementation of multi-agent system in the simulated environment

In defining the MARL system component, the state for each agent (traffic light) was represented. This implies that the state for each agent was captured as the key features of the traffic flow at its intersection and these features include queue lengths on each approach, average vehicle speed or waiting time, current signal phases (green, yellow, red), and status of neighboring intersections. For compactness and efficiency, these states were represented as vectors or matrices.

Next, after state representation is the action space design. Each agent had a discrete set of actions it could take. The actions typically correspond to changing traffic light phases or adjusting their durations. The action space includes:

1. Switching between different predefined signal patterns (e.g., switch from NS-green/EW-red to EW-green/NS-red).
2. Extending or reducing the duration of the current green phase by a certain number of seconds.
3. Dynamic adjustments based on real-time traffic flow.

The reward structure used in this work guided the agents toward optimal traffic management. Typical rewards include: a. Negative reward for congestion: Reward agents negatively based on the number of vehicles waiting in a queue. b. Positive reward for throughput: Reward agents positively for reducing vehicle waiting times or increasing the number of vehicles that pass through an intersection during a time step. c. Penalty for oscillation: Agents may receive a penalty for switching lights too frequently, leading to inefficiency.

Since it is a MARL, the rewards are designed both individually (local rewards for each intersection) and globally (a shared reward for all agents, reflecting system-wide performance). For implementing the Multi-Agent System, this study considered two scenarios; decentralized and centralized MARL. In decentralized MARL, each agent independently learns how to manage its local traffic, with minimal coordination between agents. Each agent focuses on its objective, such as minimizing the queue length at its intersection. Independent Actor-Critic (IAC) was utilized to achieve this.

Hence, there was minimal direct communication between agents, with each agent using its local traffic state to make decisions. While in centralized MARL, all agents trained together using a global state and reward. This led to more cooperative behavior between agents and helped to achieve a globally optimal traffic management solution. A centralized Actor-Critic (CAC) algorithm was used to achieve a centralized MARL. The “Actor” learns the policy for selecting the action while the “Critic” evaluates the quality of the policy as shown in Figure 3.

2.7 Key parameters for the training the MARL system

The key Parameters for the training the MARL system was done using the RL algorithms that are designed for traffic control.

Key parameters and configurations included:

1. Episodes: Training was carried out for 100 distinct episodes, with each episode signifying a full simulation cycle. Agents engaged with the environment at distinct intervals, utilizing feedback to enhance their traffic light control strategies.

2. The reward function was meticulously crafted to promote traffic efficiency, harmonizing both local and global objectives:
   - Penalties: Penalties represent some severe consequences imposed for protracted vehicle queue lengths, major delays, and traffic congestion.
   - Incentives: These are positive rewards implemented to promote efficiency in minimising waiting times and cumulative vehicle throughputs. The total negative rewards represents inefficiency and was used as a key performance metric during the training process.
   - Traffic Scenarios: Different traffic situations were simulated to assess the adaptability of the proposed model.
   - Baseline: This denotes steady flow of traffic signaled by a moderate vehicle density.
   - Dynamic Flow: This represents varying traffic demand to model the peak hours and off-peak periods.
   - Intense Traffic Conditions: tTis is determined by conducting a stress tests under various major congestion to evaluate its robustness.
   - Exploration vs. Exploitation: The agents employed the epsilon-greedy policy during the training process to achieve a balance between the exploration of innovative approaches and the deployment of already established policies. The exploration rate reduces gradually as agents approach optimal strategies.

The above mentioned parameters form a framework for testing the system in situations that strictly mirrored the real-world scenarios, thus, facilitating a thorough evaluation of the model's learning capabilities.

2.8 Development of multi-agent reinforcement system in the simulated environment

Stable-Baselines3 - a robust RL framework was used to build the Deep Reinforcement Learning (DRL) models. During training, the agents interacted with the environment in real time, learning from the traffic conditions.

The following characterise the training process:

1. Simulation Work Flow: The traffic data was generated and fed into the SUMO while the agents observed the state, selected actions, and received the rewards.
2. Experience Replay: It stored transitions (s, a,r,s′s, a,r,s′) employed for training the neural network in mini-batches.

The neural network architecture has 80 neurons (state features) in the input layer, while its hidden layers comprise of 5 layers with 400 neurons each. The output layer has 4 neurons (Q-values for actions). In terms of policy updates, iterative updates were used in backpropagation and the Q-learning formula.

The Deep Q Network (DQN) was utilised to train 2 models for different scenarios. Firstly, the decentralized MARL and secondly, the centralized MARL. The DQN algorithm combines the Q-learning with DNN. It is effective for simulating complex scenarios in a continuous state such as traffic management in urban settings. It also boasts of scalability to high dimensional inputs, generalization across similar scenarios, end to end learning and efficiency in complex domains thus, making it feasible for urban traffic simulation. The main essence of this algorithm is to learn the action-value function Q (s, a). The following the optimal policy (Equation 1) holds thus (Sewak, 2019).

$$Q(s.a) = (i - a)Q(s, a) + \alpha\left(\mathrm{r} + \gamma \max \mathrm{a}' Q\left(\mathrm{s}'\mathrm{a}'\right)\right) \tag{1}$$

In Equation 1, the Q Function Q (s, a) was captured twice. First, (1 - α)Q (s, a) is used to retrieve the present state-action value and to update its value, and secondly, to obtain the "target" value for the succeeding Q value (i.e., Q as in: r + γ maxa′ Q (s′, a′). Rather than successive learning from samples, the DQN uses experience relays to store previous transitions and sample them in a random manner to break the relationships between updates and employs a replay buffer to store experiences (s, a,r,s′) to stabilise training using separate target network Qθ′ with parameters θ′.

Figure 4 presents the neural network architecture used to supervise the learning task.

Equation 2 expresses the Q-learning update rule.

$$\mathrm{y} = \mathrm{r} + \gamma \,\mathrm{max}\mathrm{a}' \,\mathrm{Q}\theta' \left(\mathrm{s}', \mathrm{a}'\right) \tag{2}$$

where r denotes reward, γ represents discount factor, and s′ is the next state.

Equation 3 expresses the loss function for updating the parameters θ of the Q-network.

$$L(\theta) = \mathbb{E}_{(s,a,r,s')\sim D}\left[y - Q_\theta(s,a))^2\right] \tag{3}$$

where D denotes replay buffer.

For all RL systems, there is a need to balance between the exploration and exploitation scenarios. The agents will initially explore the various traffic signal techniques, however, it is expected that they should exploit the techniques that will lead to improved traffic flow over time.

Hyperparameter tuning is an important step in the training of the RL algorithms, including MARL setups such as the CAC framework. Hyperparameters are the specifications that determine how the RL model learns from the environment. It is therefore necessary to tune them to achieve optimum performance, as wrong values can lead to slow learning, poor model's performance, or suboptimal policies. The key hyperparameters in MARL include the following:

1. Learning rate: This regulates the size of updates made to the policy (actor) and value function (critic) during the training process. Usually smaller learning rate implies slower updates, resulting in a more stable learning rate but possibly slower convergence. Conversely, a larger learning rate speeds up learning rate but with the risk of instability.
2. Discount Factor: This factor determines how much future rewards are valued compared to the immediate rewards. A discount factor close to 1 implies that long-term rewards are prioritized, while a smaller value indicates that the agent focuses more on short-term rewards than long term rewards.
3. Exploration Parameters: Exploration parameters control the trade-off between exploration (experimenting new activities) and exploitation (selecting the best activity). In ε-greedy

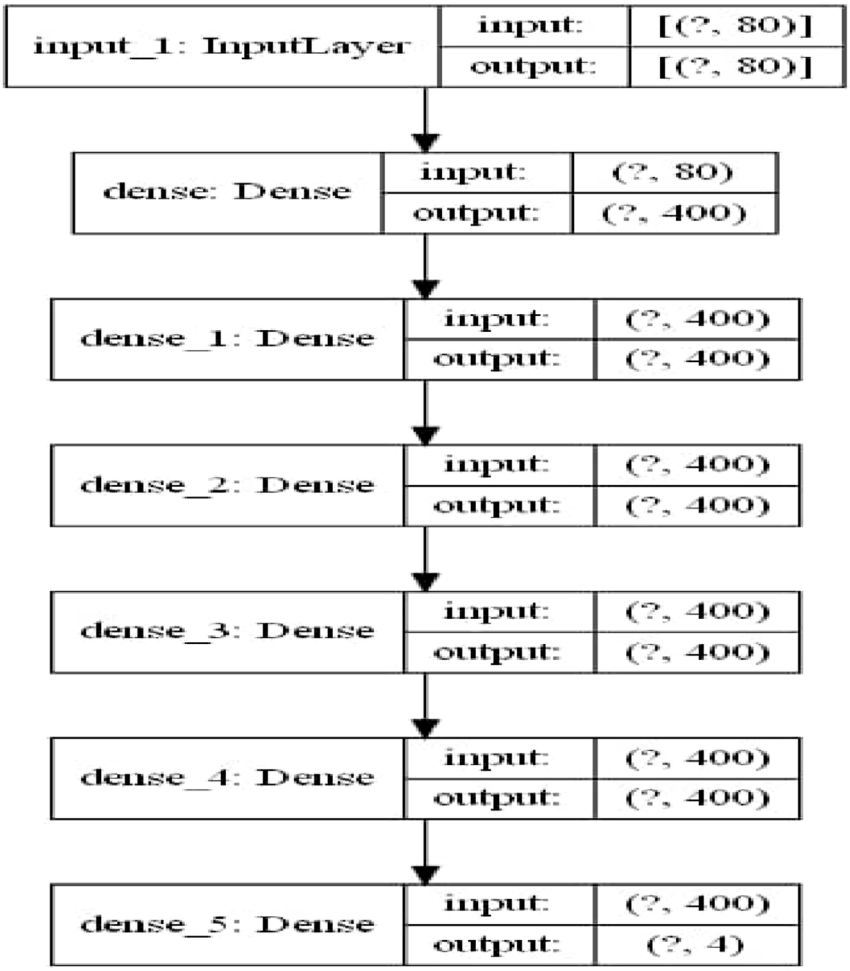

FIGURE 4
The neural network architecture used to supervise the learning task.

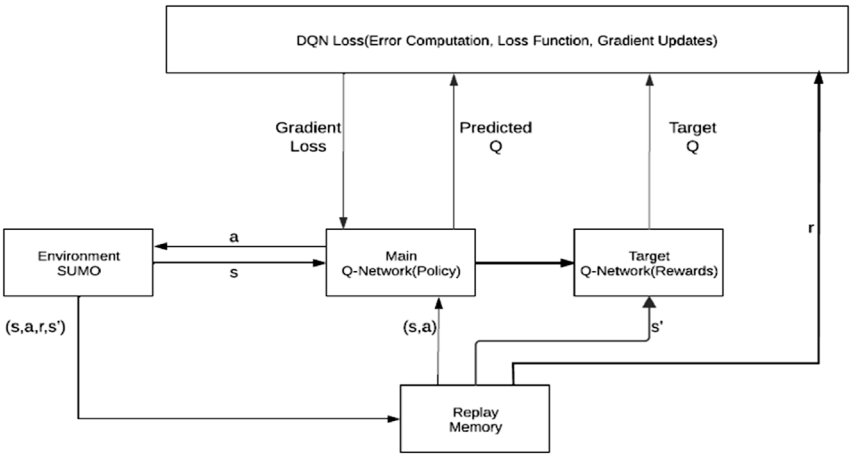

FIGURE 5
MARL for traffic Light Control using DQN.

TABLE 1 The selected hyperparameters and the justification.

Hyperparameter	Selected value	Justification
Episodes	100	A large number of episodes is usually required in MARL to enhance ease of convergence of the solution by minimizing the complexity of the model as the multi-agents learn concurrently
Episode Length (Steps per episode)	100 steps	This is to ensure a balances between adequate interaction time per episode with computational efficiency. Longer episodes might capture more dynamics of the scenarios but with the risk of instability over time
Rewards	Ranges from +50 to −50 for achieving goal, −1 per time step	This range helps in achieving stability in the learning in MARL by ensuring a cooperative behaviour among the agents
Penalties	Up to −10	This is to discourage destructive strategies
Exploration Rate (ε in ε-greedy)	from initial 1.0 → decay to 0.05	The initial exploration of 1.0 ensures sufficient state-action capture while its gradual decay to 0.05 prevents over-exploration which ensures stability of the policies
Learning Rate (α)	0.001	Usually smaller learning rate implies slower updates, resulting in a more stable learning rate but with risk of divergence (possibly slower convergence) due to non-stationary dynamics. Conversely, a larger learning rate speeds up learning rate but with the risk of instability during training
Discount Factor (γ)	0.99	MARL usually requires γ close to 1 for delayed cooperation payoffs
Batch Size	100	Mini-batches enhances smooth gradient updates while large batches improve stability but increase the memory and computation requirement
Replay Buffer Size	1e6	Large buffer is necessary for achieving stability in the off-policy MARL algorithms
Policy Update Frequency	Every 2 steps	To prevent overfitting due to rapidly changing policies

exploration, for example, ε is the probability that the agent will select a random activity rather than the best activity.

4. Batch Size: It determines the number of experiences sampled from the replay buffer during each learning step. Larger batch sizes allow more stable updates but increase memory requirements and computational complexity.
5. Replay Buffer Size: The replay buffer stores past experiences such as state, activity, reward, next state). A larger buffer permits the agent to learn from a wider range of experiences but requires more memory.
6. Target Network Update Frequency (in Actor-Critic algorithms): In methods like Deep Q-Network (DQN) or Actor-Critic, target networks are used to stabilize training.
7. Policy Update Frequency: This hyperparameter controls how often the policy (actor) is updated relative to the critic. More frequent updates to the policy can enhance a lead to quicker learning but may promote instability if the critic is not well-trained.

Figure 5 illustrates the MARL for traffic lights Control using DQN.

Table 1 presents the selected hyperparameters and the justification.

2.9 Performance evaluation

In the MARL traffic control system, congestion was evaluated by monitoring queue lengths at each intersection. If one agent reduces congestion at its intersection but creates a bottleneck at the next, the system's performance is considered poor. The evaluation focused on reducing both localized and network-wide congestion.

1. Cumulative negative reward: This metric assessed inefficiency by integrating penalties for delays, long queues, and various unfavorable traffic conditions. The analysis provided an all-inclusive perception of the system's performance, where a reduction in the negative values means improved traffic management.
2. Cumulative delay: This metric computes the cumulative delay faced by all the vehicles within the network throughout each episode. It highlights the system's ability to minimise delays and improve vehicular movements.
3. Queue lengths: To determine the congestion level, the average and peak lengths of queues at various intersections were documented. These metrics provide useful insights into the agents' ability to uniformly allocate traffic and prevent traffic jam.
4. Throughput: This is the total number of vehicles that effectively routed the network during an episode. This metric serves as a means of evaluating the overall system's efficiency.
5. Stability and convergence: This deals with the stability of policies over time and the convergence of traffic patterns. Hence, an analysis was carried out on the trends in rewards and delay metrics across episodes to assess the system's ability to achieve stable policies and successfully manage the changes in traffic patterns.

These performance metrics gave a detailed understanding of the MARL system's ability to improve traffic flow, while detecting some possible potential areas of improvement.

In line with some existing studies such as Kolat et al. (2023), Mushtaq et al. (2023) and Bie et al. (2024), the queue length and cumulative delay were selected and prioritised as crucial performance evaluation metrics because they represent general, all-inclusive measures for efficient traffic management coupled with the fact that they provide a dene and stable learning rate which align with real-world scenarios and performance goals.

3 Results

3.1 Result of the design and simulation of a typical traffic environment (scenario 1)

This research designed and simulated outcomes of a standard traffic environment which outlined the developed traffic model, emphasizing its main characteristics, and the methods it employed to simulate actual traffic scenarios. It further elaborates on the simulation configurations, encompassing the types of intersections, vehicle dynamics, and traffic signal phases utilized to create realistic and functional traffic scenarios.

Figure 6 illustrates the pattern of traffic congestion observed during various episodes and time intervals. The intensity varied from 0 to 4, where the deeper shades of blue signify longer queue lengths. It further illustrates the fluctuations in traffic load over various time periods, indicating that certain intervals encounter significant congestion (values of 3–4) whereas, others exhibit lighter traffic (values of 0–1).

Figure 6 displays the heat map for the queue length.

3.2 Result of the developed multi-agent system in the simulated environment (scenario 2)

This section illustrates the deployment of various agents within the simulated environment. It outlines the dynamics of their interactions and the method of coordination among agents. The provided heat map in Figure 6 effectively illustrates the evolution in queue lengths over different episodes as shown in Figure 7. The average queue length is the average number of vehicle on the queue due to traffic congestion. Throughout the initial 60 episodes, the system exhibited consistent and minimal queue lengths and delays. After episode 60, there are clear spikes and heightened variability in queue metric, indicating that the system faced more complex traffic scenarios and recalibrating its parameters, probably due to ineffective response to traffic congestions.

Figures 8, 9 show the cumulative delay and the reward distribution respectively which provides insights into the quality of the quality of the learned policy by the model. The visual representations in Figures 8, 9 illustrate the system's performance over various episodes. The cumulative delay is the total time the vehicle have spent waiting on the queue. Low cumulative delay implies effective system' response to traffic demands and vice versa. Throughout the initial 60 episodes, the system exhibited cumulative delay which implies a free flow of traffic. However, after episode 60, there are clear spikes and heightened variability in the cumulative delay metric, indicating that the delay due to traffic congestion and probably ineffective response of the system to traffic demands.

The reward distribution refers to the trend and features of the rewards received by an agent during training or evaluation of the RL model. The reward distribution is skewed to the left between −0.5 and 0.0 (negative rewards) and concentrated near 0 with fewer bars which implies lesser reward. This implies that the RL model improves in decision making across the different episodes although still marked with few errors. A value of 0 means optimal or neural action while −1 implies a bad action such as long queue and values ranging from 0.5 to 0 suggest a suboptimal but improving performance.

3.3 Result of developed MARL model (scenario 3)

This scenario focused on the learning process of the MARL model. Figure 9 illustrates how the agents optimized their policies over time. It elaborates more on the allocation of cumulative negative rewards within the reinforcement learning framework designed for autonomous traffic management. The horizontal axis represents the range of cumulative negative rewards whereas the vertical axis denotes the frequency of their occurrences. The distribution of rewards shows a concentration around zero, indicating that the penalties are generally low whereas instances

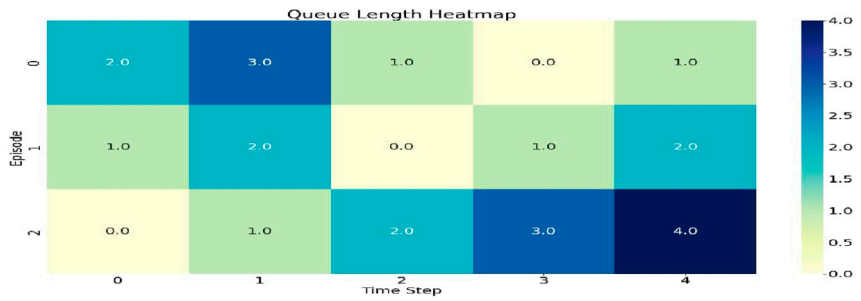

FIGURE 6
The queue length heat map.

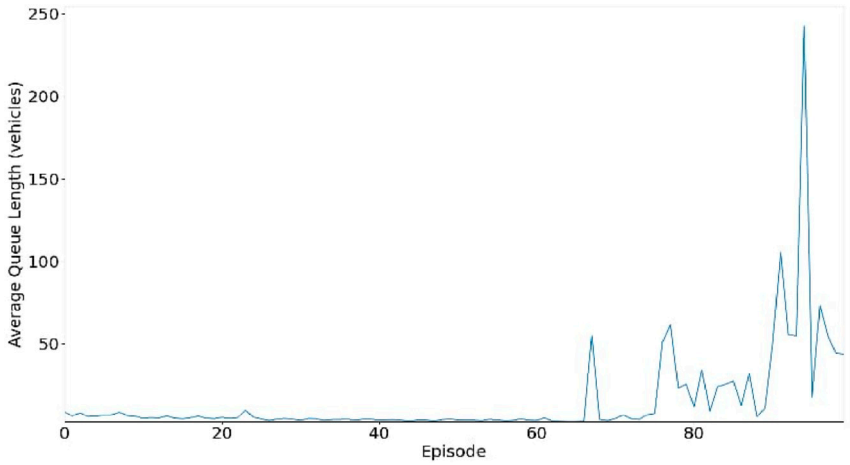

FIGURE 7
The average queue length.

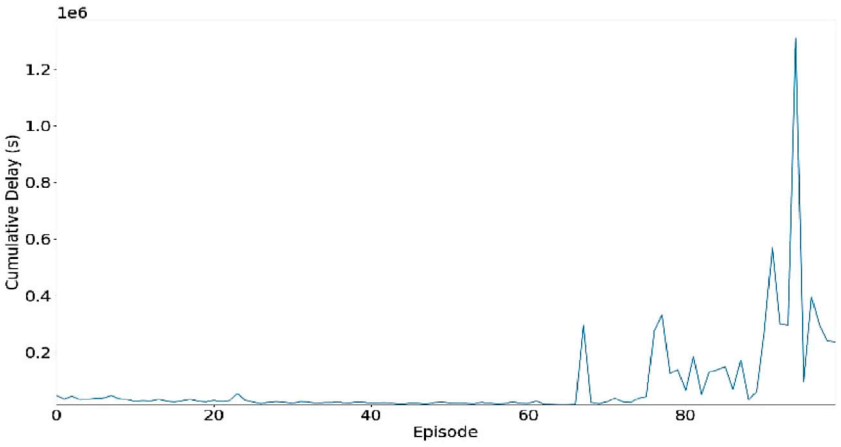

FIGURE 8
The Cumulative delay.

of higher negative rewards are less common. This indicates that the system is operating effectively, reducing negative outcomes in the majority of instances.

3.4 Result of training, analysis, and learning dynamics (scenario 4)

The graph in Figure 10 illustrates the total cumulative negative reward over episodes for two distinct simulations, labeled Run 1 and Run 2. The horizontal-axis denotes the episodes, whereas the vertical-axis denotes the cumulative penalty, where more significant negative values signify inferior performance (for instance, increased congestion or delays). At the outset, both runs exhibit consistent performance with slight fluctuations. As episodes advance, Run 2 shows notable declines (sharp negative spikes), highlighting the occurrence of considerable inefficiencies or instability within the traffic system. In contrast, Run 1 demonstrates a more consistent performance, indicating superior overall system optimization. This underscores the variations in performance of the system across various executions.

Figure 11 displays the cumulative negative reward across different episodes, which reflects the traffic system performance throughout the training process. The horizontal axis shows the episode number, while the vertical axis shows the cumulative negative reward, where more negative values imply a reduction in the system's performance (in other words, it implies heightened traffic congestion or inefficiencies.

Originally, the cumulative reward shows a steady pattern with minimal deviations, which reflects a stable and reliable system's performance. However, there are instances of a rapid decline in the reward (noteworthy spikes), which indicates instances of significant or instability or inefficiencies by the system. This behaviour depicts the system's learning process as it seeks to gain traffic control while regulating some complex situations.

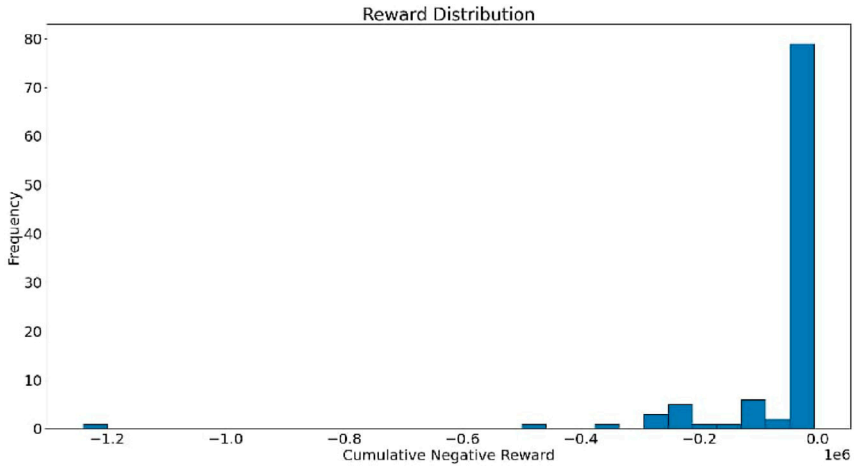

FIGURE 9
The reward distribution.

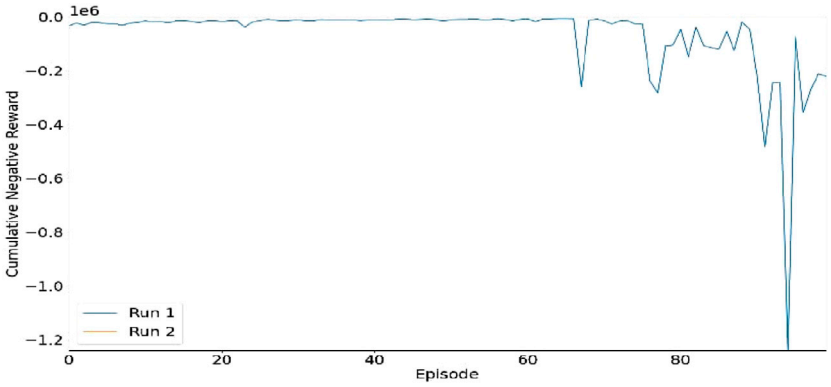

FIGURE 10
The cumulative Negative Reward (Run 1).

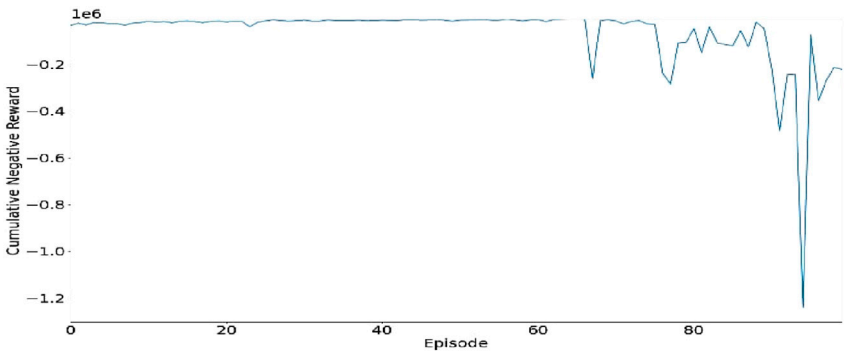

FIGURE 11
Cumulative negative reward (Run 2).

Figure 12 shows the total negative reward gathered over the episodes. It shows the system's performance during the training process. The horizontal axis represents the episode number, while the vertical axis shows the cumulative negative reward. Lower values implies a decline in the system's performance, due to delays or heightened traffic congestion.

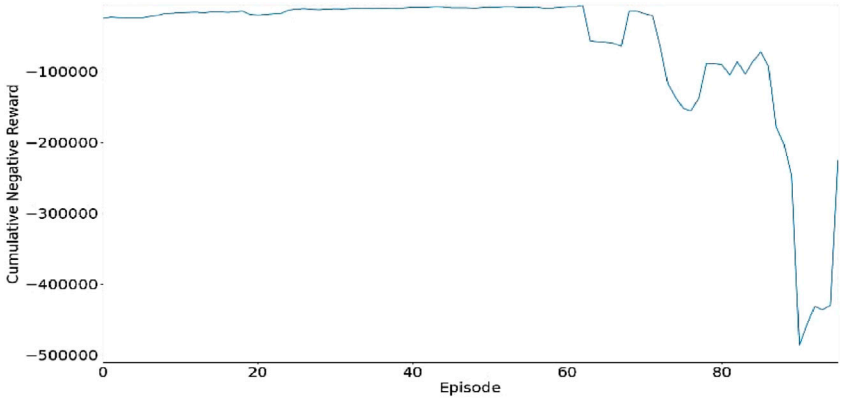

FIGURE 12
Cumulative Negative Reward over different episodes.

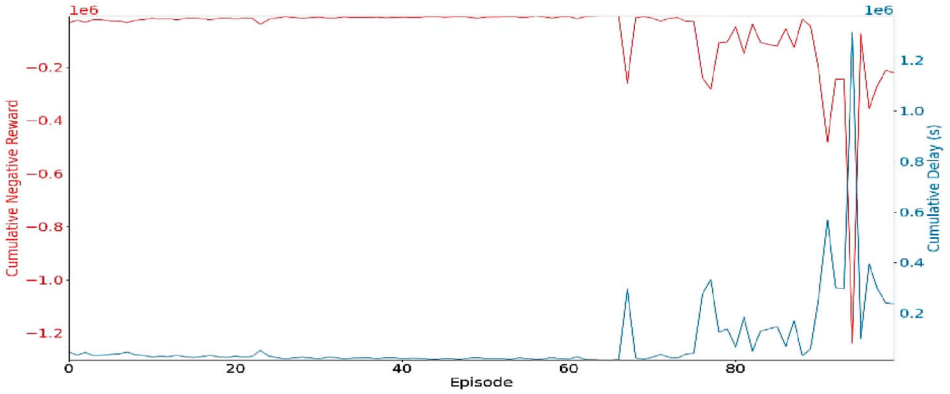

FIGURE 13
Cumulative Negative Reward for baseline policy and RL learned policy.

Firstly, the cumulative negative reward depicts a fairly stable, showing only some slight variations, which indicates a relatively steady performance. However, as the training progresses, there are evidences of diminishing performances between episodes 60 and 90). This decline in performance represents instances when the system was faced significant limitations. The variations observed highlight the dynamic features of the learning process as the system responds and regulates various traffic situations.

Figure 13 compares the cumulative negative reward for the baseline policy and the RL model's learned policy across different episodes. The red curve represents the cumulative negative reward for the baseline policy, showing the stability in the initial episodes of the system while exhibiting substantial variations in succeeding ones, signifying challenges encountered in adjusting to the intricacy of the environment. The blue curve represents the cumulative delay of the RL learned policy, which shows substantial increase in the initial episodes as a result of inefficiencies due to heightened traffic congestion but later dropped below the red line as the training progresses implying a better performance compared to the baseline policy, indicating that the model that learned over different episodes.

3.5 Performance evaluation of the developed MARL model (scenario 5)

Figure 14 further shows the advancement in the learning of the MARL model. The distribution of rewards tends towards smaller negative values, indicating improved performance of the model.

This scenario assesses the efficacy and performance of the MARL model in traffic management. The analysis of the metrics conducted in this study includes cumulative delay, throughput, and average queue length. The delay plot demonstrates the gradual alleviation of traffic congestion over time. The comprehensive assessment of the performance is depicted using various metrics. Figure 14 illustrates a relationship where increased delays correspond with more negative rewards. The analysis of the runs in Figures 10–12 illustrates model consistency and reliability.

Analysis of performance metrics indicates:

- The beginning of a stable phase (episodes 0–60) characterized by minimal delays and short queue lengths.
- A demanding phase (episodes 60–80) characterized by heightened variability.

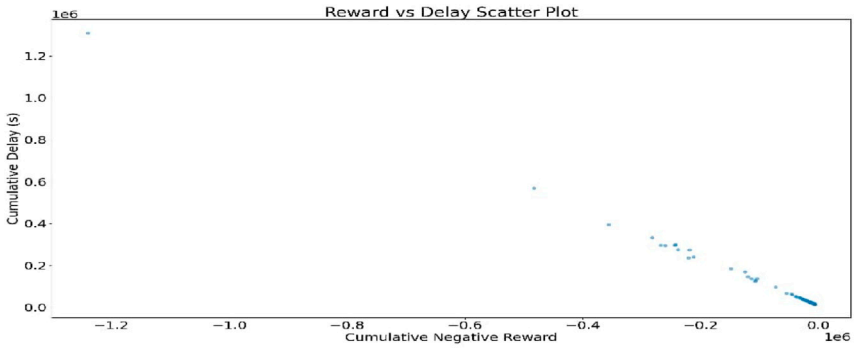

FIGURE 14
The reward vs. delay-scatter plot.

TABLE 2 Systems performance under different evaluation metrics.

Scale	Halting vehicles	Queue Time(s)	Queue length (m)	Speed (m/s)	Training Time/Rollout(s)
5 × 5	4.76 ± 0.92	0.35 ± 0.02	6.15 ± 0.18	19.20 ± 0.31	12.45
10 × 10	9.82 ± 1.45	0.42 ± 0.03	8.73 ± 0.22	17.85 ± 0.25	42.18
15 × 15	15.37 ± 2.68	0.55 ± 0.04	10.89 ± 0.27	15.48 ± 0.38	95.72

TABLE 3 Performance evaluation of the MARL model summary.

Category	Metric	Value/Observation	Analysis
Queue Management	Maximum Queue Length	4 vehicles	Peak congestion periods show the highest queue formation
	Minimum Queue Length	0 vehicles	During low-traffic periods
	Average Queue Length	<50 vehicles	Maintained during stable operation periods
Traffic Flow	Congestion Patterns	0–4 vehicles	Effectively captured varying traffic intensities
	Peak Congestion Periods	3–4 vehicles	Represented by dark blue regions in the heat map
	Low Traffic Periods	0–1 vehicles	Represented by light colors in the heat map
MARL Performance	Cumulative Negative Rewards	−0.2 × 10^{6}	Stabilization point for the system
	System Stability Period	Episodes 0–60	Shows effective baseline performance
	Challenge Period	After Episode 60	Peak performance challenges emerged
System Efficiency	Throughput	Variable	Consistent flow maintained except during peak congestion
	Average Waiting Time	Correlated with rewards	Direct relationship with negative rewards
	Inter-agent Coordination	Observable	Visible in delay patterns
Adaptive Response	Traffic Condition Response	Dynamic	Demonstrated through queue length variations
	Peak Congestion Handling	Limited	System shows stress during extreme scenarios
	Normal Operation	Efficient	Effective management under standard conditions

- The concluding adaptation phase (episodes 80+) illustrating the system's reaction to intricate traffic scenarios.

The result indicates that the MARL model performs well under standard traffic scenarios (initial 60 episodes), yet encounters difficulties in high congestion situations, demonstrating its ability to adapt. The system exhibits learning behaviour via reward mechanisms; however, enhancements are needed in managing peak traffic scenarios.

Table 2 shows the system's performance under different evaluation metrics.

Table 3 Presents the performance evaluation of the MARL model across different grid scales while Table 4 compared the

TABLE 4 Comparison of the MARL model with the baseline model or policies.

Component	Description of erformance	Why it outperforms the baseline model
Decentralized Decision-Making	This enables each agent to learn policies based on feedbacks or local observations and feedback, thereby minimizing dependency on a central controller	Baseline policies have fixed-time control or centralized heuristic making it difficult to adapt rapidly to dynamic local variations
Agents coordination	This is achieved via rewards, communication, or joint learning thus, agents align actions to reduce negatives like queues or penalties	Baselines treat nodes as independent entities thus resulting in conflict in decisions
Reward Formulation	The rewards and incentives are targeted at reducing cumulative delay and queue lengths	Baseline rules may maximize throughput thereby creating inequities, long waits or traffic bottlenecks
Exploration–Exploitation Balance	The balance between exploration and exploitation ensures gents discover novel strategies before converging	Baseline policies such as static policies lack improvement over time because of its non-dynamic nature
Scalability	MARL can scale to many agents without exponential increase in state space	A centralized baseline becomes computationally complex with the introduction of many agents
Learning in Non-Stationary Environments	Agents adapt to traffic dynamics such as changes in traffic flows or network	Baseline models are inflexible
Replay and Policy Stabilization	The use of replay buffers, target networks, and shared experiences reduce training instability	Baseline lacks self-improvement over time due to its fixed nature

outcomes of the MARL model with the baseline model or policies. The outcome of the study shows that the MARL outperforms the baseline because it makes independent decisions in a coordinated, adaptive, and goal driven manner. The MARL combines the strategies of decentralized control, cooperative reward shaping, and dynamic adaptation to reduced queue lengths and delays, unlike the baseline models. However, while the MARL model demonstrates superior performance over the baseline models, some challenges may include scaling to larger urban environments which may present some computational, real-time, and data-related limitations. These challenges can be addressed by ensuring efficient training of the model, deployment of edge computing and transfer learning and by ensuring the development of a scalable features and architectures amongst other.

The outcome of the simulation conducted in this study showed an improvement in queue management and traffic flow by 64.5% and 70.0% respecitively with improvement in performance of the proposed model over the episodes. These findings agree significantly with some existing literature such as Sewak (2019), Liu and Kohls (2010) as well as Bakker and Groen (2010), that RL model can learn over episodes and improve in performance. Similar to the findings in this study on the deployment of RL model for effective traffic control, Kolat et al. (2023) also reported that the use of reinforcement learning for traffic control using the Q-deep learning algorithm minimised fuel usage and average travel time by 11% and 13% respectively. Mushtaq et al. (2023) also reported that the use of MARL specifically the Multi-Agent Advantage Actor-Critic (MA2C) for enhancing the flow of autonomous vehicles on road networks resulted in 38% improvement in the management of traffic scenarios. Zeynivand et al. (2022) achieved 7.143% improvement in queue length using MARL while Bie et al. (2024) found that that the integration of the spatiotemporal graph attention network (SGAN) into the MARL model to form a hybrid model improved traffic flow patterns in terms of the reduction in the average vehicle delays and stops, as well as increase in the travel speeds when compared to other baseline models or algorithms.

4 Conclusion

The effectiveness of this study is in the ability to meet the goals by creating and simulating a dynamic traffic environment, applying the MAS, and also formulating a MARL model to manage traffic. In other to improve overall efficiency, the MARL system exhibited the capacity to optimize traffic flow by reducing cumulative delays and queue lengths. Highlighting the system's applicability for practical implementations in the urban traffic system management, the model's resilience and flexibility across diverse traffic scenarios were validated with the performance. The outcome of the simulation conducted in this study showed an improvement in queue management and traffic flow by 64.5% and 70.0% respecitively with improvement in performance of the proposed model over the episodes.

The results show reward distribution is skewed to the left between −0.5 and 0.0 (negative rewards) and concentrated near 0 with fewer bars which implies lesser reward. This implies that the RL model improves in decision making across the different episodes although still marked with few errors. Furthermore, the RL model policy showed better performance compared to the baseline policy, indicating that the model that learned over different episodes. The result also indicates that the MARL model performs well under standard traffic scenarios (initial 60 episodes), yet encounters difficulties in high congestion situations, demonstrating its ability to adapt. Hence, this study contributes to the development of intelligent transportation systems that enhance safety, and efficiency in urban environments. The outcomes of this may assist in the development of smart city with effective traffic management system. This study is limited o he use of RL, future studies may compare the outcome of this study with the performance of ensemble model under similar traffic scenarios.

Data availability statement

The original contributions presented in the study are included in the article/supplementary material, further inquiries can be directed to the corresponding author.

Author contributions

OO: Methodology, Software, Supervision, Writing – original draft, Conceptualization, Formal Analysis, Data curation, Visualization, Resources, Investigation, Writing – review and editing, Validation, Project administration. YO: Writing – original draft, Methodology, Conceptualization, Data curation, Software, Visualization, Investigation, Resources, Validation, Formal Analysis, Project administration, Writing – review and editing. ID: Conceptualization, Investigation, Writing – original draft, Data curation, Software, Resources, Formal Analysis, Visualization, Project administration, Validation, Writing – review and editing, Methodology. AE: Methodology, Software, Writing – original draft, Investigation, Supervision, Resources, Formal Analysis, Data curation, Visualization, Conceptualization, Project administration, Validation. QS: Project administration, Conceptualization, Methodology, Writing – review and editing, Validation, Data curation, Investigation, Writing – original draft, Software, Formal Analysis, Resources, Visualization. AA: Data curation, Methodology, Formal Analysis, Project administration, Validation, Funding acquisition, Resources, Visualization, Software, Writing – review and editing. HP: Data curation, Methodology, Formal Analysis, Project administration, Validation, Funding acquisition, Resources, Visualization, Software, Writing – review and editing. TR: Data curation, Methodology, Formal Analysis, Project administration, Validation, Funding acquisition, Resources, Visualization, Software, Writing – review and editing. MK-KK: Data curation, Methodology, Formal Analysis, Project administration, Validation, Funding acquisition, Resources, Visualization, Software, Writing – review and editing.

Funding

The author(s) declare that financial support was received for the research and/or publication of this article. The authors acknowledge Tshwane University of Technology, Pretoria, South Africa for providing funding support for the publication of this article.

Acknowledgments

The authors acknowledge Bells University of Technology, Ota, Nigeria where this study was conducted.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Generative AI statement

The author(s) declare that no Generative AI was used in the creation of this manuscript.

Any alternative text (alt text) provided alongside figures in this article has been generated by Frontiers with the support of artificial intelligence and reasonable efforts have been made to ensure accuracy, including review by the authors wherever possible. If you identify any issues, please contact us.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

References

Admass, W. S., Munaye, Y. Y., and Diro, A. A. (2024). Cyber security: state of the art, challenges and future directions. *Cyber Secur. Appl.* 2 (October 2023), 100031. doi:10.1016/j.csa.2023.100031

Al-Turki, M., Jamal, A., Al-Ahmadi, H. M., Al-Sughaiyer, M. A., and Zahid, M. (2020). On the potential impacts of smart traffic control for delay, fuel energy consumption, and emissions: an NSGA-II-based optimization case study from Dhahran, Saudi Arabia. Sustain. *Switz.* 12 (18), 7394–22. doi:10.3390/SU12187394

Alfaro-Navarro, J. L., López-Ruiz, V. R., Huete-Alcocer, N., and Nevado-Peña, D. (2024). Quality of life in the urban context, within the paradigm of digital human capital. *Cities* 153 (June), 105284. doi:10.1016/j.cities.2024.105284

Almukhalfi, H., Noor, A., and Noor, T. H. (2024). Traffic management approaches using machine learning and deep learning techniques: a survey. *Eng. Appl. Artif. Intell.* 133 (PB), 108147. doi:10.1016/j.engappai.2024.108147

Appio, F. P., Lima, M., and Paroutis, S. (2019). Understanding Smart Cities: innovation ecosystems, technological advancements, and societal challenges. *Technol. Forecast. Soc. Change*, 142, 1–14. doi:10.1016/j.techfore.2018.12.018

Bakker, B., and Groen, F. C. A. (2010). "Traffic light control by multiagent reinforcement learning systems traffic light control by multiagent reinforcement learning systems," in *Studies in computational intelligence* (Berlin: Springer). doi:10.1007/978-3-642-11688-9

Belt, E. A., Koch, T., and Dugundji, E. R. (2023). Hourly forecasting of traffic flow rates using spatial temporal graph neural networks. *Procedia Comput. Sci.* 220, 102–109. doi:10.1016/j.procs.2023.03.016

Bie, Y., Ji, Y., and Ma, D. (2024). Multi-agent deep reinforcement learning collaborative traffic signal control method considering intersection heterogeneity. *Transp.* Res. Part C Emerg. Technol. 164, 104663. doi:10.1016/j.trc.2024.104663

Boukerche, A., Tao, Y., and Sun, P. (2020). Artificial intelligence-based vehicular traffic flow prediction methods for supporting intelligent transportation systems. *Comput. Netw.* 182 (August), 107484. doi:10.1016/j.comnet.2020.107484

Carramiñana, D., Campaña, I., Bergesio, L., Bernardos, A. M., and Besada, J. A. (2021). Sensors and communication simulation for unmanned traffic management. *Sensors* 21 (3), 927–929. doi:10.3390/s21030927

Chang, H., Lee, Y., Yoon, B., and Baek, S. (2012). Dynamic near-term traffic flow prediction: system-oriented approach based on past experiences. *IET Intell. Transp. Syst.* 6, 292–305. doi:10.1049/iet-its.2011.0123

Ding, Q., Wang, X. F., Zhang, X. Y., and Sun, Z. (2010). Forecasting traffic volume with space-time ARIMA model. *Adv. Mater. Res.* 156–157, 979–983. doi:10.4028/www.scientific.net/amr.156-157.979

Duan, Z., Zhang, K., Chen, Z., Liu, Z., Tang, L., Yang, Y., et al. (2019). Prediction of city-scale dynamic taxi origin-destination flows using a hybrid deep neural network combined with travel time. *IEEE Access* 7, 127816–127832. doi:10.1109/access.2019.2939902

Elassy, M., Al-Hattab, M., Takruri, M., and Badawi, S. (2024). Intelligent transportation systems for sustainable smart cities. *Transp. Eng.*, 16, 100252. doi:10.1016/j.treng.2024.100252

Fadila, J. N., Wahab, N. H. A., Alshammari, A., Aqarni, A., Al-Dhaqm, A., and Aziz, N. (2024). Comprehensive review of smart urban traffic management in the context of the fourth industrial revolution. *IEEE Access* 12, 196866–196886. doi:10.1109/access.2024.3509572

Gupta, A., Anpalagan, A., Guan, L., and Khwaja, A. S. (2021). Deep learning for object detection and scene perception in self-driving cars: survey, challenges, and open issues. *Array* 10 (December 2020), 100057. doi:10.1016/j.array.2021.100057

Hasan, U., Whyte, A., and Jassmi, H.Al. (2020). *A review of the transformation of road transport systems: are we ready for the next step in artificially intelligent sustainable transport ? from the public public*, 1–21.

Heidari, A., Navimipour, N. J., and Unal, M. (2022). Applications of ML/DL in the management of smart cities and societies based on new trends in information technologies: a systematic literature review. *Sustain. Cities Soc.* 85 (July), 104089. doi:10.1016/j.scs.2022.104089

Hosseinian, S. M., Mirzahossein, H., and Guzik, R. (2024). Sustainable integration of autonomous vehicles into road networks: ecological and passenger Comfort Considerations. *Sustain. Switz.* 16 (Issue 14), 6239. doi:10.3390/su16146239

Hui, C. X., Dan, G., Alamri, S., and Toghraie, D. (2023). Greening smart cities: an investigation of the integration of urban natural resources and smart city technologies for promoting environmental sustainability. *Sustain. Cities Soc.* 99 (October), 104985. doi:10.1016/j.scs.2023.104985

Japiassu, N. (2024). AI-powered logistics and mobility as a Service (MaaS): driving the future of autonomous vehicles and smart transportation. doi:10.13140/RG.2.2.22609.75366

Javed, A. R., Shahzad, F., ur Rehman, S., Zikria, Y. B., Razzak, I., Jalil, Z., et al. (2022). Future smart cities: requirements, emerging technologies, applications, challenges, and future aspects. *Cities*, 129, 103794. doi:10.1016/j.cities.2022.103794

Jia, S. (2021). Economic, environmental, social, and health benefits of urban traffic emission reduction management strategies: case study of Beijing, China. *Sustain. Cities Soc.* 67 (December 2020), 102737. doi:10.1016/j.scs.2021.102737

Khalil, R. A., Safelnasr, Z., Yemane, N., Kedir, M., Shafiqurrahman, A., and Saeed, N. (2024). Advanced learning technologies for intelligent transportation systems: Prospects and challenges. *IEEE Open J. Veh. Technol.* 5, 397–427. doi:10.1109/OJVT.2024.3369691

Kolat, M., Kővári, B., Bécsi, T., and Aradi, S. (2023). Multi-agent reinforcement learning for traffic signal control: a cooperative approach. *Sustainability* 15 (4), 3479. doi:10.3390/su15043479

Kong, X., Xu, Z., Shen, G., Wang, J., Yang, Q., and Zhang, B. (2016). Urban traffic congestion estimation and prediction based on floating car trajectory data. *Future Gener. Comput. Syst.* 61, 97–107. doi:10.1016/j.future.2015.11.013

Kumar, K., Parida, M., and Katiyar, V. K. (2013). Short term traffic flow prediction for a non-urban highway using artificial neural network. *Procedia Soc. Behav. Sci.* 104, 755–764. doi:10.1016/j.sbspro.2013.11.170

Li, L., He, S., Zhang, J., and Ran, B. (2016). Short—term highway traffic flow prediction based on a hybrid strategy considering temporal—spatial information. *J. Adv. Transp.* 50, 2029–2040. doi:10.1002/atr.1443

Liu, I. A. C., Kohls, T. U. A. G., and Urbanik, T. (2010). Reinforcement learning-based multi-agent system for network traffic signal control. *IET Intell. Transp. Syst.* 4 (July 2009), 128–135. doi:10.1049/iet-its.2009.0070

Lv, Y., Duan, Y., Kang, W., Li, Z. X., and Wang, F. (2015). Traffic flow prediction with big data: a deep learning approach. *IEEE Trans. Intell. Transp. Syst.* 16, 865–873. doi:10.1109/TITS.2014.2345663

Ma, X., Tao, Z., Wang, Y., Yu, H., and Wang, Y. (2015). Long short-term memory neural network for traffic speed prediction using remote microwave sensor data. *Transp. Res. Part C* 54, 187–197. doi:10.1016/j.trc.2015.03.014

Maldonado, D., Cruz, E., Abad Torres, J., Cruz, P. J., and Gamboa Benitez, S. D. P. (2024). Multi-agent systems: a survey about its components, framework and Workflow. *IEEE Access* 12 (April), 80950–80975. doi:10.1109/ACCESS.2024.3409051

Medina-Salgado, B., Sánchez-DelaCruz, E., Pozos-Parra, P., and Sierra, J. E. (2022). Urban traffic flow prediction techniques: a review. *Sustain. Comput. Inf. Syst.* 35, 100739. doi:10.1016/j.suscom.2022.100739

Mishra, P., and Singh, G. (2023). Energy management systems in sustainable smart cities based on the internet of energy: a technical review. *Energies* 16 (19), 6903. doi:10.3390/en16196903

Mushtaq, A., Haq, I. U., Sarwar, M. A., Khan, A., Khalil, W., and Mughal, M. A. (2023). Multi-agent reinforcement learning for traffic flow management of autonomous vehicles. *Sensors* 23 (5), 2373. doi:10.3390/s23052373

Mystakidis, A., Koukaras, P., and Tjortjis, C. (2025). Advances in Traffic Congestion Prediction: an overview of emerging techniques and methods. *Smart Cities* 8 (1), 25. doi:10.3390/smartcities8010025

Quallane, A. A., Bakali, A., Bahnasse, A., Broumi, S., and Talea, M. (2022). Fusion of engineering insights and emerging trends: intelligent urban traffic management system. *Inf. Fusion* 88 (July), 218–248. doi:10.1016/j.inffus.2022.07.020

Rahman, N. B. A. (2024). Adaptive traffic signal control in smart cities through deep reinforcement learning: an intelligent infrastructure perspective. *Appl. Res. Artif. Intell. Cloud Comput.* 7 (5), 1–12. Available online at: https://researchberg.com/index.php/araic/article/view/195.

Rahman, H., Abdel-aty, M., and Wu, Y. (2021). A multi-vehicle communication system to assess the safety and mobility of connected and automated vehicles. *Transp. Res. Part C* 124 (November 2020), 102887. doi:10.1016/j.trc.2020.102887

Sewak, M. (2019). "Deep reinforcement learning Frontier of AI," in *Deep reinforcement learning*. 1st Edition (Singapore: Springer).

Su, Y., Liu, X., Li, X., and Oh, K. (2020). Research on traffic congestion based on system dynamics: the case of chongqing, China. *Complexity* 2020, 1–13. Article ID 6123896. doi:10.1155/2020/6123896

Williams, B. M., Durvasula, P. K., and Brown, D. E. (1998). Urban freeway traffic flow prediction: application of seasonal autoregressive integrated moving average and exponential smoothing models. *Transp. Res. Rec.* 1644, 132–141. doi:10.3141/1644-14

Xia, D., Wang, B., Li, H., Li, Y., and Zhang, Z. (2016). A distributed spatial-temporal weighted model on mapreduce for short-term traffic flow forecasting. *Neurocomputing* 179, 246–263. doi:10.1016/j.neucom.2015.12.013

Xia, L., Semirumi, D. T., and Rezaei, R. (2023). A thorough examination of smart city applications: exploring challenges and solutions throughout the life cycle with emphasis on safeguarding citizen privacy. *Sustain. Cities Soc.* 98 (July), 104771. doi:10.1016/j.scs.2023.104771

Yeong, D. J., Velasco-Hernandez, G., Barry, J., and Walsh, J. (2021). Sensor and sensor fusion technology in autonomous vehicles: a review. *Sensors* 21 (6), 2140. doi:10.3390/s21062140

Zeynivand, A., Javadpour, A., Bolouki, S., Sangaiah, A. K., Ja'fari, F., Pinto, P., et al. (2022). Traffic flow control using multi-agent reinforcement learning. *J. Netw. Comput. Appl.* 207, 103497. doi:10.1016/j.jnca.2022.103497

Zhang, K., Chu, Z., Xing, J., Zhang, H., and Cheng, Q. (2023). Urban traffic flow congestion prediction based on a data-driven model. *Mathematics* 11 (4075), 4075–20. doi:10.3390/math11194075

Zhao, Z., Chen, W., Wu, X., Chen, P. C. Y., and Liu, J. (2017). LSTM network: a deep learning approach for short-term traffic forecast. *IET Intell. Transp. Syst.* 11, 68–75. doi:10.1049/iet-its.2016.0208

Zhu, Z., Peng, B., Xiong, C., and Zhang, L. (2016). Short-term traffic flow prediction with linear conditional Gaussian Bayesian network. *J. Adv. Transp.* 50, 1111–1123. doi:10.1002/atr.1392