sciendo

Transport and Telecommunication, 2025, volume 26, no. 3, 266-275
Transport and Telecommunication Institute, Lauvas 2, Riga, LV-1019, Latvia
DOI 10.2478/ttj-2025-0020

OPTIMIZING TRAFFIC LIGHT CONTROL USING ENHANCED DQN: MINIMIZING WAITING TIME FOR REGULAR AND EMERGENCY VEHICLES

Wissam Bouzi[1], Samia Bentaieb[2], Abdelaziz Ouamri[3]

[1,2,3]Laboratory Signals and Images, Faculty of Electrical Engineering,
Department of Electronics, University of Sciences and Technology of Oran Mohamed Boudiaf USTO-MB
Oran, Algeria, B.P. 1505, El Mnaouar – Bir el Djir
[2]Faculty of Science and Technology, Department of Electronics and Telecommunications
University of Ain Temouchent Behadj Bouchaib
Ain Temouchent, Algeria, Road of Sidi Bel Abes N101, 46000
[1]wissam.bouzi@univ-usto.dz, [2]samia.bentaieb@univ-temouchent.edu.dz, [3]abdelaziz.ouamri@univ-usto.dz

An efficient traffic management system is essential to minimize traffic problems and ensure the rapid circulation of emergency vehicles. This research proposes a new single-agent deep reinforcement-learning model using a deep Q-Network (DQN) to optimize traffic lights, aiming to reduce waiting times and increase vehicle speed, with particular emphasis on emergency vehicles. Our method incorporates a new state representation, which captures variations in vehicle density and speed that directly influence the reward structure to prioritize both traffic flow and emergency vehicle response times. The decision of the agent is enhanced by a replay memory mechanism, which ensures that experiences are effectively used in learning. The model's effectiveness was tested in a simulated environment using SUMO, showing significant improvements in traffic management compared to traditional methods. Experimental results show that our system significantly reduces average waiting times and improves emergency vehicle prioritization.

Keywords: Intelligent traffic light control, emergency vehicle,waiting time, speed, DQN

1. Introduction

Traffic congestion is one of the main problems that cause delays, rising fuel consumption, and pollution levels in urban areas. As cities become increasingly densely populated, efficient traffic solutions are required. The dynamic nature of traffic flow, particularly at busy times, often poses a problem for standard traffic control approaches, including fixed-time traffic lights. Effective traffic control at urban intersections is essential to reduce congestion, travel times, and improve road safety in general. Traditional traffic control systems, which are based on pre-programmed or actuated signals, often fail to respond to traffic conditions in real time, resulting in less-than-optimal performance. However, recent progress in artificial intelligence, particularly in Deep Reinforcement Learning (DRL), has provided promising solutions for the dynamic and adaptive control of traffic lights. This allows the rapid development of Intelligent Traffic Control Systems (ITCS) that interact with the environment to determine the best possible policies. DRL methods use real-time traffic data to make smart choices regarding the traffic flow, wait times, and emergency vehicles. This type of adaptability is especially useful in urban environments, where circulation routes vary and are unexpected. This study aims to develop a Deep Q Network (DQN) single-agent DRL model for intelligent traffic control at a single intersection. The simplicity and adaptability of the single-agent concept make it the preferred option for addressing traffic-management issues. Its easy implementation and rapid learning make it an excellent choice for practical application. Additionally, the scalability and flexibility of the single-agent technique can be increased by adding it as needed to larger traffic networks. The primary contributions of this work include:

- Designing a DQN-based traffic control model that optimizes traffic signal timings.
- Evaluating the performance of the proposed model in terms of reducing waiting times and improving the speeds of both regular and emergency vehicles.

The remainder of this paper is organized as follows. Section 1 provides an overview of relevant studies, followed by Section 2, which delves into the details of the proposed approach. Section 3 outlines the experiments and presents a detailed discussion of their results. Finally, in Section 4, we conclude the study with observations and suggestions for future research.

2. Related work

Traffic signals are a vital component of transportation systems, serving to manage traffic flow and guarantee the safety of pedestrians and cyclists. The goal is to prevent accidents and congestion. Despite minimizing travel time while maintaining safety. Recently, researchers have begun to use reinforcement learning (RL) to address traffic control issues. Two primary methods, RL and DRL, have been proposed to optimize traffic signal timing and improve traffic flow efficiency. These methods are commonly used in traffic-control research.

2.1. Reinforcement learning

This method relies on traditional RL algorithms that are devoid of deep-learning components. It is suitable for low-dimensional state-space tasks. The study of Joo *et al*. (2020) proposed an Intelligent Traffic Signal Control (ITSC) system that employs Q-learning (QL) to optimize vehicle crossings and balance road signals. The structure of the system is adaptable and can be altered to accommodate changes at the intersection. In Kuang *et al.* (2021), an ITSC method that utilizes RL with a state reduction was developed. The model utilizes historical traffic flow and a dual-objective reward function to minimize vehicle delays and enhance the signal timing. The state and action were reduced by choosing appropriate control phases and eliminating rare states. Simulation experiments demonstrated improved intersection capacity and decreased algorithm costs. An RL-based ITSC using the n-step SARSA algorithm was employed for predictive ITSC (Kekuda *et al.*, 2021). The algorithm surpasses conventional methods, making it feasible for low-cost real-time systems and reducing the network congestion probability.

2.2. Deep reinforcement learning

DRL is a promising approach for handling high-dimensional state spaces in complex traffic scenarios, because it leverages deep learning algorithms as an approximation function to enhance the learning ability of the agent. In the study by Wei *et al.* (2018), a novel approach to ITSC was introduced, emphasizing phase-sensitive DRL. This method demonstrated superior performance compared to existing techniques through comprehensive synthetic and real-world experiments. Jang *et al.* (2018) proposed a method for integrating DRL into traffic simulation modelling to effectively solve traffic congestion in cities by adapting to changing traffic conditions and learning from experience. Kumar *et al.* (2020) developed a Dynamic Intelligent Traffic Light Control System (DITLCS) for vehicular traffic networks, considering the heterogeneity of emergencies and priority vehicles. The system operates in three phases: fairness, priority, and emergency. It uses DRL and fuzzy inference to dynamically generate traffic lights, thereby demonstrating its effectiveness and efficiency. According to the study in Zhang *et al.* (2020), a deep QL algorithm was utilized for ITSC system with partial vehicle detection. The algorithm uses a neural network to approximate the Q-value function, and aims to minimize the difference between the average travel time of commuters and the lowest average travel time. Cao *et al*. (2022) applied the DRL approach to an ITSC system for emergency vehicles. The primary objective of the system is to minimize the impact on traffic efficiency, while ensuring a quick response to emergency vehicles. Liu *et al.* (2023) applied a double-deep Q-network approach to an ITSC. The reward function was defined based on queue length, number of vehicles, and waiting time. Zhu *et al.* (2022) have compared policy-based and value-based DRL techniques to examine the use of RL for ITSC in smart cities. The authors used a policy-based DRL technique called Proximal Policy Optimization (PPO) and evaluated its performance against value-based techniques, such as Double DQN (DDQN). The findings show that PPO generates optimal policies more effectively than DDQN, particularly when variable time intervals rather than fixed intervals are used for the traffic signal phases. As a result of this adaptive strategy, there were fewer phase changes, shorter car wait times, and increased traffic efficiency. The study also investigated the PPO's adaptation to environmental and action perturbations, showing that the learning-based controller continues to operate at a high level, even in difficult situations. To enhance traffic light control, Kodagoda *et al*. (2021) suggested the use of waiting time and standard deviation of vehicle waiting times, along with the number and type of vehicles, and proposed that the current signal phase index should be included as an essential element of the state representation instead of using traditional state representations and reward functions. Choosing the following traffic signal phase from a list of safe green signal combinations is a part of the agent's action space. Rewards were determined by averaging and standardizing waiting times for regular and emergency vehicles. Tested on both synthetic and actual datasets in the SUMO simulation environment, the approach outperforms established statistical and fixed-time methods, as well as methods that ignore standard deviation. Tan *et al.* (2022) used sensor technology to collect accurate traffic status data and then applied DRL to challenging signal optimization problems. Through the integration of this exact data source, this study examines variations in state under various traffic scenarios and the correlation between the observed traffic states and traffic demand. The research involved testing the model under four distinct scenarios: extreme traffic states, entirely balanced traffic states, mildly unbalanced states, and fully balanced states. This ensures a comprehensive evaluation and adaptability. In their study, Alsheikhy *et al.* (2024) created Intelligent Adaptive Traffic Control and Management System (IATCMS), that makes use of cutting-edge methods for data collecting and processing. The system evaluates important traffic factors such as vehicle speed, density, and headway using real-time data from wireless sensor networks and cameras. Utilizing a Generative Adversarial Network (NDGAN), pertinent features are extracted, and ideal traffic signal timings are predicted, dynamically varying the lengths of red and green lights according to the traffic situation at the time. By prioritizing lanes with higher traffic congestion, this strategy helps the system reduce the average waiting time at crossings.

3. Methodology

3.1. Traffic light control problem formalization

Effective traffic management at urban intersections presents a significant challenge in transportation engineering, which is the focus of this study. A single intersection represents the intricacies of traffic flow management. It consists of four junctions, each with three incoming lanes and one outgoing lane. Eight phases of traffic lights were defined, as depicted in Figure 1, which illustrates the organization of traffic lights for one phase. RL is an effective method for training agents to navigate and make decisions in intricate environments such as ITLC. By interacting with the environment, the agent learns the optimal policy and uses trial and error to maximize the objective reward function. The RL agent explores the environment to make better decisions at each time step. It has been used in many domains such as telecommunications (Sherif *et al.*, 2024), and plays a crucial role in transportation (Ballis and Dimitriou, 2020), in robotics (Hohenfeld *et al.*, 2024) and game playing (Lample and Chaplot, 2017). Several algorithms are used in RL, including Q-learning, policy gradient methods, and Monte Carlo methods. Deep Q-Networks DQN is a widely used algorithm that utilizes a deep neural network to approximate the Q-value function, which maps states to action values.

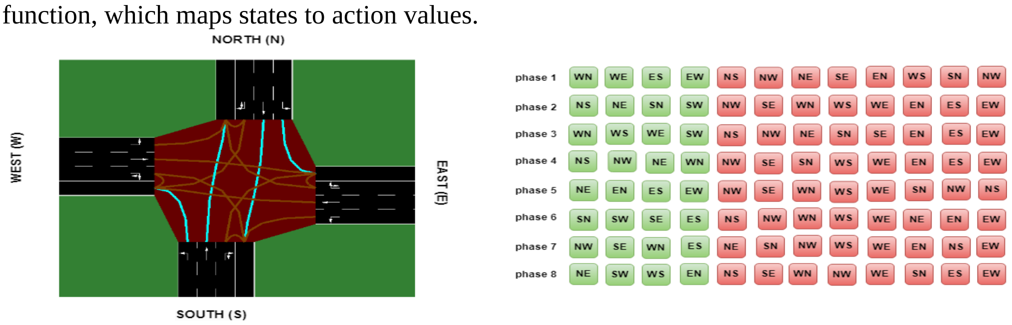

Figure 1. Traffic light organization for phase one

3.2. DQN algorithm

The Deep Q-Network (DQN) method combines deep learning with reinforcement learning to address the sequential decision-making challenges. This is a variant of the Q-learning algorithm, which represents a fundamental RL technique. The aim of this algorithm is to learn an optimal policy that maximizes the total expected reward by estimating a Q function denoted as Q(s, a), which signifies the cumulative reward resulting from action (a) taken in state (s), following policy π. The DQN method uses a Neural Network (NN) to approximate the Q-function. The NN accepts the state as an input and yields a Q-value for all the feasible actions. The Q-value indicates the expected future reward for each action in a given state. Training a neural network involves supervised learning, which requires continually updating the target values to align them with the observed rewards. The objective of the training is to minimize the gap between the predicted and target Q-values. To calculate the target Q-values, the Bellman equation is employed, which explains how the current Q-value relates to the expected future reward, as expressed in Equation 1.

$$Q(s,a) = R + \gamma * \max\big(Q(s',a')\big). \tag{1}$$

This equation indicates that (R) denotes the immediate reward gained after executing action (a) in state (s). (s') is the resulting state and (a') denotes the action. The discount factor γ (Gamma), determines the significance of future rewards by balancing the trade-off between the immediate and long-term rewards. The term $\max(Q(s',a'))$ represents the maximum target Q-value for the next state (s′), reflecting the best possible future reward the agent can expect by taking the optimal action (a′).

During the training process, the DQN algorithm uses experience replay to break temporal correlations and stabilize learning by saving experience. The loss function, denoted as (L), quantifies the difference between the predicted Q-values and the target Q-values, and is employed to train the neural network. This function is expressed in Equation 2.

$$L = (Q(s,a) - (R + \gamma * \max(Q(s',a'))))^2. \tag{2}$$

3.3. Parameter modelling

As a first step in designing our model based on deep reinforcement learning, we define the reward, action, and state in our method.

a) State:

The state representation in our ITSC system is crucial for capturing the necessary information regarding the traffic conditions at the intersection. The following elements are utilized to construct the state.

- The queue length represents the number of vehicles in each lane N, with the maximum capacity of vehicles per lane.
- Waiting time, denoted by W, is calculated as the sum of all vehicle waiting times divided by the number of vehicles, as shown in Equation 3.

  $$W = \frac{\sum_{i=1}^{N} w_i}{N}, \tag{3}$$

  where $\sum_{i=1}^{N} w_i$ represents the sum of waiting times for vehicles in the lane.
- The maximum emergency waiting time, denoted as MWT, is the longest duration that an emergency vehicle can take before leaving the intersection. This is defined in Equation 4.

  $$MWT = max\ (w_1, w_2, \ldots., w_n), \tag{4}$$

  where *n* denotes the total number of emergency vehicles in a specified lane.
- The average speed of regular vehicles, denoted as S_{reg}, is computed as the sum of the speeds of all regular vehicles divided by the number of regular vehicles, as presented in Equation 5.

  $$S_{reg} = \frac{\sum_{i=1}^{N_{reg}} s_i}{N_{reg}}, \tag{5}$$

  where s_i is the speed of the i-th regular vehicle and N_{reg} is the number of regular vehicles.
- The average speed of emergency vehicles, denoted as S_{emg}, is calculated as the sum of the speeds of all emergency vehicles divided by the number of emergency vehicles, as stated in Equation 6.

  $$S_{emg} = \frac{\sum_{i=1}^{N_{emg}} s_i}{N_{emg}}, \tag{6}$$

  where s_i is the speed of the i-th emergency vehicle and N_{emg} is the number of emergency vehicles.
- A one-hot encoding vector, referred as (A), signifies the state of the signal, with values ranging from 1 to 8. This indexing system provides an agent with a comprehensive overview of vehicle movements.

b) Action:

To regulate the signal, we implemented actions corresponding to potential phase changes of the traffic light. After a 12-seconds interval, the agent selects the next phase. If the agent's current phase is the same as the next phase, it will remain in the same state for an additional 12 seconds. Alternatively, the agent will transit to a new phase, preceded by a 3-second yellow signal.

c) Reward:

The reward function serves as a directive signal for the agent, providing feedback on the efficiency of its actions. In the proposed ITSC system, the reward function is formulated using Equation 7.

$$R = 50 - \left[\left(W_{reg} + \alpha * \left(60 - S_{reg}\right)\right) + \left(\beta * W_{emg} + \alpha * \left(60 - S_{emg}\right)\right)\right]. \quad (7)$$

The α and β parameters allow the adjustment of the reward function to achieve specific objectives. By modifying these parameters, the system can vary the importance placed on minimizing waiting times and speed penalties, as well as prioritizing emergency vehicle responses. Coefficient α controls the trade-off between waiting time (W_{reg}) and speed (S_{reg}) for regular vehicles, while β governs the emphasis on emergency vehicle prioritization, as indicated by (W_{emg}) and (S_{emg}). In this study, we set α = 0.5 and β = 1. Both coefficients can be fine-tuned based on the system requirements.

d) Neural Network Architecture Choice:

To determine the best strategy, we investigated several neural network designs in our study on traffic light control optimization using DRL. Through rigorous experimentation, we discovered in our use case that a simple architecture of only dense layers was the best performance when compared with other more sophisticated models such as Convolutional Neural Networks (CNN) and Long Short Term Memory (LSTM) networks.

Given the nature of our features, we construct fully connected layers in the context of DQN. Naturally, the traffic data in our case were tabulated, collecting different aspects, such as average waiting times, queue lengths, and vehicle speeds. Because dense layers, often referred as fully linked layers, can learn complex patterns and relationships between all input features without assuming any particular spatial or temporal organization, they are especially well suited for tabular data. The efficiency and simplicity of dense layers' make them perfect for traffic management scenarios when making real time decisions. This simplicity ensures that the model provides predictions very rapidly with bare necessities, which allows traffic signals to be adjusted in the next phases. CNN, on the other hand, is specifically designed for spatial data, such as images, and LSTM is built to operate on sequential data involving long-term dependencies. Both CNN and LSTM run more complex calculations, which adds a higher inference time and increases the computational overhead. In addition, dense layers are simpler in their architecture, and the risk of overfitting is minimized compared to other models. We strike a balance between model complexity and generalization ability using dense layers, ensuring a strong performance in a variety of traffic situations.

e) Learning optimal policy process:

DQN learning is run after every environmental signal change on a batch of 256 experiences that include interactions between the agent and environment (state-action-reward triads). The process to learn optimal policy is presented in Algorithm 1.

Algorithm 1: DQN Learning Process for Traffic Control

Input: A batch of 256 experiences, each containing:
State (s), Action(a), Reward (R), Next state (s′), Discount factor (γ), Learning rate ((l_r)
Output: Optimized Q-network for traffic control
for each sample (s_t, a_t, R_t) in the experience batch **do**
Step 1: Compute Expected Q-value Q (s_t, a_t):
Compute the expected Q-value for the current state s_t and action a_t based on the agent's current knowledge.
Step 2: Calculate Target Q-value Q_{target} (s_t , a_t):
$Q_{\text{target}}(s_t, a_t) = R_{t+1} + \gamma \max_{a \in A} Q(s_{t+1}, a)$
Step 3: Calculate the Error E:
$E = Q(s_t, a_t) - Q_{\text{target}}(s_t, a_t)$
Step 4: Update the Q-value Q (s_t, a_t):
$Q(s_t, a_t) = Q(s_t, a_t) + l_r \cdot E$

3.4. Deep Q network architecture

The DQN architecture presented in Figure 2 contains two main branches: a traffic state and a phase index branch. The first traffic state branch presented by the matrix M_s used two dense layers with 64 and 32 filters, each with Rectified Linear Unit (ReLU) activation, to process the features of the incoming traffic. The second-phase index branch, depicted in vector A, interprets one-hot encoded traffic light phase data using two dense layers with 32 and 16 filters and ReLU activation. Then, the output branches are concatenated, allowing the network to understand the relationship between the current state and available traffic-light phases. The output of the concatenation layer is passed through a dense layer of 16 units with ReLU activation. Finally, we used a dense layer of eight units (no activation) to generate Q-values for the eight different traffic-light operations. In a DQN, the final layer typically avoids the activation function. This ensures that the network can predict Q-values over a larger range, incorporating both positive rewards and negative penalties involved in different traffic light control actions. The principal activation functions and layers are as follows:

- Fully connected layer: Fully linked layers allow the model to learn sophisticated patterns by combining dissimilar features and connecting all neurons in one layer to all the neurons in the next layer. When tabular data are available, this strategy is effective, enabling us to understand the complex details of the traffic data patterns.
- Activation Function: Deep neural networks are highly dependent on the Rectified Linear Unit (ReLU) activation function. This function produces nonlinearity by reducing negative input values to zero and maintaining positive values. This minimizes the problem of the vanishing gradient and promotes faster convergence during training.
- Concatenation layer: Concatenation layers are used to merge features extracted from different branches of the network.

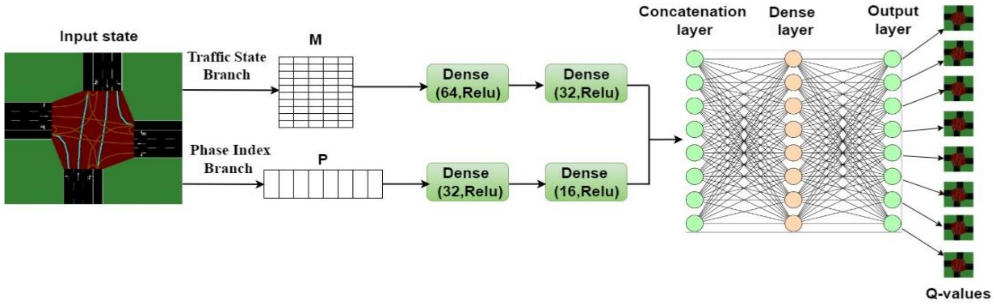

Figure 2. Architecture of Deep Q Network

4. Result and discussion

Our goal is to improve the assessment of RL algorithms, primarily focusing on DQN to minimize waiting time and prioritize emergency vehicles. Our main objective is to determine how well these algorithms optimize the ITSC to reduce delays and improve the efficiency of intersections. We performed our experiments on an open-source platform Simulation of Urban Mobility (SUMO), with the advantage of simulating various traffic scenarios. SUMO's advanced simulation capabilities enabled us to reproduce the geometric layout, lane configuration, and traffic flow dynamics that closely defined our simulated traffic network.

After presenting the training protocol, we used the parameters listed in Table 1 for the experimental configuration. These parameters are the learning rate, discount factor, replay memory capacity, batch size, and target network update frequency. These parameters must be defined because they play an important role in the manner in which our reinforcement-learning algorithm works inside SUMO and in how it learns. The training stage consists of tuning the hyperparameters to maximize the performance of the RL agent. The key hyperparameters were discount factor (γ), minimum epsilon (ε_{min}), epsilon decay (ε_{decay}) learning rate (l_r), and the replay memory capacity. Fine-tuning these parameters means that we can control and regulate the real ratio of exploration to exploitation, thereby driving robust policy learning.

Table 1. Hyperparameters and parameters for training the DRL agent

Variable	Value	Description
γ	0.999	Discount factor
ε_{min}	0.01	Minimum epsilon
ε_{decay}	0.0002	Epsilon decay rate
l_r	0.001	Learning rate
τ	0.125	Target network update frequency
Replay Memory Capacity	10000	Capacity of replay memory
Replay Batch Size	256	Size of replay batches
Steps per Episode	5000	Number of steps per episode

To evaluate the performance of DQN control strategies, we examined key measures, such as average waiting time, average speed to regular and emergency vehicles, and reward. We aimed to determine the benefits of ITLC regulation by comparing the performance of the DQN architecture to standard fixed-time techniques and other RL techniques in terms of lowering the traffic waiting time and enhancing overall the travel speeds.

Figure 3 shows the history of the average waiting time for a regular vehicle over a number of actions, which refers to the number of times a specific scenario was run in a simulation. Initially, the average waiting time for regular vehicles is high, approximately 3000 seconds, owing to the initial learning phase of the system and the difficulty of the traffic scene. However, as the number of actions increases, the waiting time decreases considerably, reaching much lower and more stable levels. The reduction in waiting time after 2000 actions, to almost 0 second until 8000 actions proves that the DQN-based intelligent traffic control system learns and adapts successfully over time.

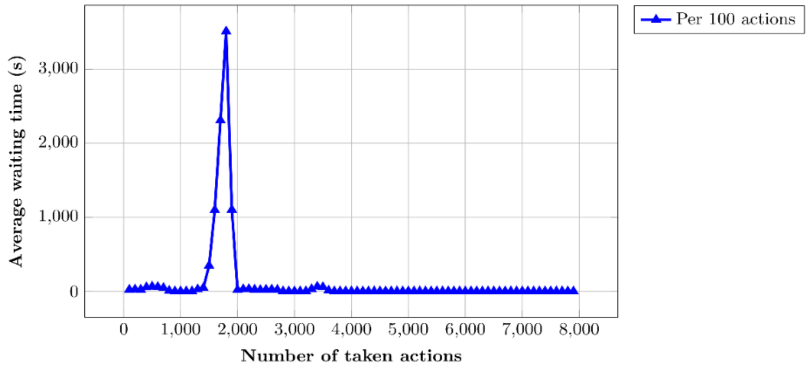

Figure 3. Average waiting time of regular vehicles

Figure 4 shows the average waiting time over the number of taken actions. The waiting time of emergency vehicles starts to increase then decreases to reach 0 second as the number of actions increases, indicating that the DQN model adapts to prioritize emergency vehicles more effectively over time.

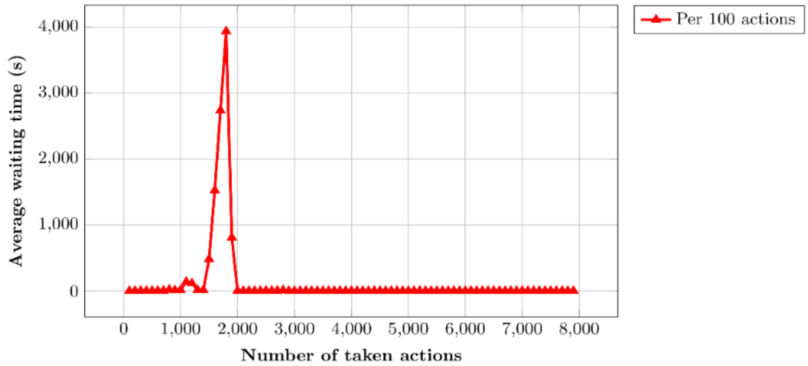

Figure 4. Average waiting time of emergency vehicles

The average speed feature of regular vehicles provides information on the consistency and reliability of the traffic light control models.

In Figure 5, the average speed of regular vehicles changes slightly but remains relatively stable at approximately 57-59 seconds throughout the 8000 taken actions, which indicates that our DQN-based model maintains a reasonable speed for regular vehicles.

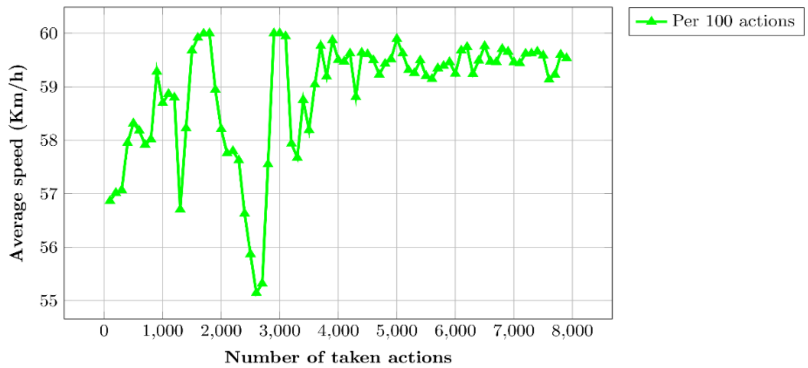

Figure 5. Average speed of regular vehicles

Figure 6 presents the average speed of emergency vehicles against the number of taken actions. As shown in this graph, the average speed of emergency vehicles increases significantly from 60 to 70 seconds when the number of actions taken reaches 8000. This proves that our DQN-based intelligent traffic control system effectively prioritizes emergency vehicles and enables them to maintain higher speeds over time.

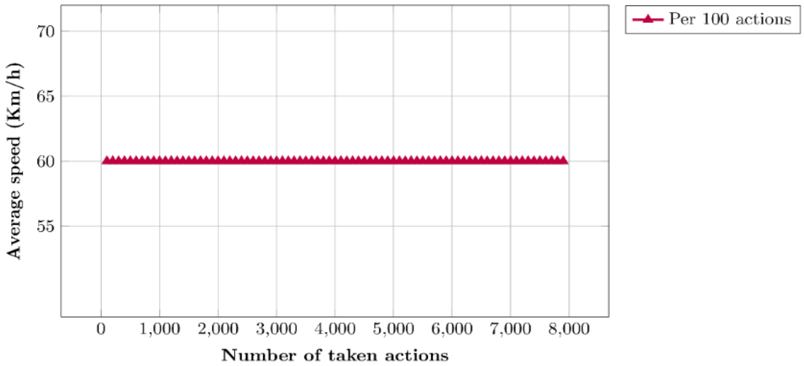

Figure 6. Average speed of emergency vehicles

The stable trend indicates that the system effectively prioritizes emergency vehicles, maintaining a consistent speed throughout the simulation and allowing them to pass through intersections without delays, ensuring timely responses.

Figure 7 depicts the reward history for the DQN model and shows an overall increasing trend. The rewards start to be negative, indicating poor initial performance and penalties taken by the agent at the beginning of the training. The model becomes increasingly better at rewards, reaching 10 positive rewards as the number of actions increased. This indicates that the system learns and improves its performance over time.

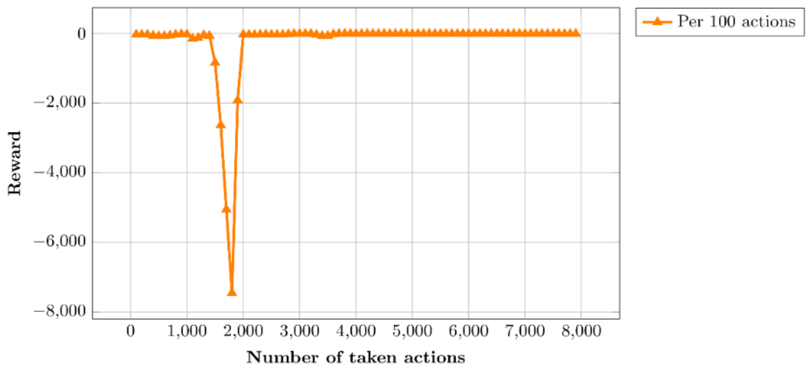

Figure 7. Reward history

The proposed algorithm is compared to several established approaches as shown in Table 2. This study focuses on the primary objectives, used strategies, and results obtained, with a particular focus on the average waiting time. The results demonstrate the effectiveness of our strategy in reducing traffic congestion and improving traffic flow efficiency on a single agent compared to existing methods.

Table 2. Comparative evaluation of traffic light control methods

Method	**Year**	**Average Waiting Time (s)**	**Emergency Vehicle Waiting Time (s)**
Kodagoda *et al.* (2021)	2021	43.51	5.06
Zhu *et al.* (2022)	2022	58,38	-
Alsheikhy *et al.* (2024)	2024	56.5	-
Proposed Approach	2025	30	0

From the comparison, it is evident that the proposed method outperforms other state-of-the-art methods in terms of reducing the average waiting time of vehicles. Specifically, the proposed approach achieves 30 seconds average waiting time, which is substantially lower than the 58.38 seconds reported by Zhu *et al.* (2022), the 43.51 seconds by Kodagoda *et al.* (2021), and the 56.5 seconds by Alsheikhy *et al.* (2024).

In addition to reducing the average waiting time for all vehicles, the proposed approach effectively eliminates the waiting time for emergency vehicles, achieving a waiting time of 0 seconds. This is a critical improvement over the 5.06 seconds waiting time for emergency vehicles reported by Kodagoda *et al.* (2021). Other studies have not reported specific waiting times for emergency vehicles, thus highlighting a potential gap in their evaluation.

5. Conclusion

This study proposes a novel approach to traffic signal control based on a DQN model to reduce traffic congestion while enhancing the flow of emergency vehicles in urban areas. The model efficiently balances general traffic and emergency vehicle priorities using a novel state representation that considers vehicle density and speed fluctuations. An essential part of our methodology is the use of a fully connected layer-based DQN architecture, which has been chosen because it is simpler and uses less processing power than more sophisticated architectures, even though CNN and LSTM can be used to extract rich features, not all tabulated features or data types benefit from their use. Furthermore, they have high memory and processing requirements, which may be problematic for real-time traffic-management systems. Experimental results in the SUMO simulation environment demonstrate that the proposed method outperforms traditional traffic signal control strategies in terms of reducing the average waiting times for regular vehicles and improving the response times for emergency vehicles. These results highlight the potential of reinforcement-learning-based approaches in enhancing urban mobility and safety. While these results are promising, the model remains limited to simulated environments and does not incorporate pedestrian flows or handle complex multi emergency scenarios. Future research will focus on extending the framework to multi agent coordination across multiple intersections, incorporating pedestrian activity and contextual features into the state representation, and validating the model using real-world traffic data to assess its performance in live environments.

References

1. Alsheikhy, A.A., Said, Y.F. and Shawly, T. (2024) A novel intelligent smart traffic system using a deep-learning architecture. *Multimedia Tools and Applications*, 83(13), 1-14. DOI:10.1007/s11042-023-17003-3.
2. Ballis, H. and Dimitriou, L. (2020) Evaluation of reinforcement learning traffic signalling strategies for alternative objectives: Implementation in the network of Nicosia, Cyprus. *Transport and Telecommunication Journal*, 21(4), 295–302. DOI:10.2478/ttj-2020-0024.
3. Cao, M., Li, V.O. and Shuai, Q. (2022) A gain with no pain: Exploring intelligent traffic signal control for emergency vehicles. *IEEE Transactions on Intelligent Transportation Systems*, 23(10), 17 899–17 909. DOI:10.1109/TITS.2022.3159714.
4. Hohenfeld, H., Heimann, D., Wiebe, F. and Kirchner, F. (2024) *Quantum deep reinforcement learning for robot navigation tasks*. DOI:10.48550/arXiv.2202.12180.
5. Jang, I., Kim, D., Lee, D. and Son, Y. (2018) An agent-based simulation modeling with deep reinforcement learning for smart traffic signal control. In: *Proceedings of International Conference on Information and Communication Technology Convergence (ICTC),* Jeju, October 2018. IEEE, 1028–1030. DOI: 10.1109/ICTC.2018.8539377.
6. Joo, H., Ahmed, S.H. and Lim, Y. (2020) Traffic signal control for smart cities using reinforcement learning. *Computer Communications*, 154(1), 324–330. DOI:10.1016/j.comcom.2020.03.005.
7. Kekuda, A., Anirudh, R. and Krishnan, M. (2021) Reinforcement learning based intelligent traffic signal control using n-step sarsa. In: *Proceedings of International Conference on Artificial Intelligence and Smart Systems (ICAIS)*, Coimbatore, March 2021. IEEE, 379–384. DOI: 10.1109/ICAIS50930.2021.9395942.
8. Kodagoda, D., Perera, D., Seneviratne, G. and Kumarasinghe, P. (2021) Minimize traffic congestion with emergency facilitation using deep reinforcement learning. In: *Proceedings of the 21st International Conference on Advances in ICT for Emerging Regions (ICter)*, Colombo, December 2021. IEEE, 55–62. DOI: 10.1109/ICter53630.2021.9774792.
9. Kumar, N., Rahman, S.S. and Dhakad, N. (2020) Fuzzy inference enabled deep reinforcement learning-based traffic light control for intelligent transportation system. *IEEE Transactions on Intelligent Transportation Systems*, 22(8), 4919 - 49281. DOI:10.1109/TITS.2020.2984033.

10. Kuang, L., Zheng, J., Li, K. and Gao, H. (2021) Intelligent traffic signal control based on reinforcement learning with state reduction for smart cities. *ACM Transactions on Internet Technology (TOIT)*, 21(4), 1–24. DOI:10.1145/3418682.
11. Lample, G. and Chaplot, D.S. (2017) Playing FPS games with deep reinforcement learning. In: *Proceedings of the AAAI Conference on Artificial Intelligence*, San Francisco, February 2017. Palo Alto: AAAI Press, 31(1). DOI: https://doi.org/10.1609/aaai.v31i1.10827.
12. Liu, X.-Y., Zhu, M., Borst, S. and Walid, A. (2023) *Deep reinforcement learning for traffic light control in intelligent transportation systems*. DOI:10.48550/arXiv.2302.03669.
13. Sherif, H., Ahmed, E. and Kotb, A.M. (2024) Energy-efficient and accelerated resource allocation in O-RAN slicing using deep reinforcement learning and transfer learning. *Cybernetics and Information Technologies*, 24(3), 132–150. DOI:10.2478/cait-2024-0029.
14. Tan, J., Yuan, Q., Guo, W., Xie, N., Liu, F., Wei, J. and Zhang, X. (2022) Deep reinforcement learning for traffic signal control model and adaptation study. *Sensors*, 22(22), 8732. DOI:10.3390/s22228732.
15. Wei, H., Zheng, G., Yao, H. and Li, Z. (2018) Intellilight: A reinforcement learning approach for intelligent traffic light control. In: *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, London, August 2018. New York: Association for Computing Machinery, 2496–2505. DOI:10.1145/3219819.3220096.
16. Zhang, R., Ishikawa, A., Wang, W., Striner, B. and Tonguz, O.K. (2020) Using reinforcement learning with partial vehicle detection for intelligent traffic signal control. *IEEE Transactions on Intelligent Transportation Systems*, 22(1), 404–415. DOI: 10.1109/TITS.2019.2958859.
17. Zhu, Y., Cai, M., Schwarz, C.W., Li, J. and Xiao, S. (2022) Intelligent traffic light via policy-based deep reinforcement learning. *International Journal of Intelligent Transportation Systems Research,* 20(3), 734–744. DOI:10.1007/s13177-022-00321-5.