IJTE, Vol. 13, No. 4, Spring 2026, pp. 2277-2287
Research Paper

Machine Learning based Distributed Traffic Signal Control

Alireza Rezaee[1,*], Amirhossein Safdari[2]

Received: 2025/05/15 *Accepted: 2025/10/11*

Abstract

Optimizing traffic flow remains a central objective in transportation research. With the continuous growth in vehicle numbers, the limited scalability of existing infrastructure, and the inherently nonlinear, dynamic, and stochastic nature of traffic systems, signal control at road junctions has become an increasingly complex control challenge. Traditional signal control methods are predominantly rule-based, designed for deterministic environments, and often fail to adapt to real-time traffic variations and unexpected disruptions. This study proposes a distributed traffic signal control framework built upon a Machine Learning (ML) paradigm utilizing Reinforcement Learning (RL). Owing to their non-model-based architecture and computational efficiency, RL-based methods exhibit strong adaptability to changing traffic conditions and robustness against environmental uncertainties. In the proposed system, each junction is independently managed by an autonomous learning agent capable of interacting with its environment, refining its control policy over time, and making localized, real-time decisions. Simulation results demonstrate the effectiveness of the proposed approach, with vehicle queue lengths and average waiting times reduced by 35% to 66.4% on roads leading to the junctions, compared to conventional rule-based systems. These findings highlight the potential of distributed, learning-enabled control strategies in achieving scalable, adaptive, and efficient urban traffic management.

Keywords: Road Traffic, Distributed Control, Machine Learning, Reinforcement Learning

---

[*] Corresponding author. E-mail: arrezaee@ut.ac.ir

[1] Associate Professor, Department of Mechatronics Engineering, School of Intelligent Systems Engineering, College of Interdisciplinary Science and Technology, University of Tehran, Tehran, Iran

[2] M.Sc., Department of Mechatronics Engineering, School of Intelligent Systems Engineering, College of Interdisciplinary Science and Technology, University of Tehran, Tehran, Iran

1. Introduction

According to the latest reports from the *World Health Organization* (WHO), approximately 1.35 million people worldwide lose their lives in traffic accidents each year, [Ahmed, S. K. et al., 2023]. This statistic highlights the critical importance of improving driving culture, vehicle safety, and road safety. Part of this safety enhancement can be achieved through the smartification of vehicles and roads, [Rezaee, A., Safdari, A., & Fazli, M. J., 2024]. Additionally, in recent decades, with the significant increase in the number of vehicles and transportation volume, urban travel times have increased, [Rizwan, P., Suresh, K., & Babu, M. R., 2016]. The slowing down of urban development arteries, reduced social patience, excessive fuel consumption, increased carbon dioxide levels, and other environmental pollutants have turned traffic into a socio-economic dilemma. As a result, proper traffic management and control have become critically important and are now on the agendas of governments, [Ata, A. et al., 2019], [Abbas, S. et al., 2022], [Bokaba, T., Doorsamy, W., & Paul, B. S., 2022].

Traffic lights are among the most familiar tools in traffic control systems; they regulate traffic flow and enhance vehicle safety at junctions. Currently, most traffic control systems rely on fixed and predefined rules that are broadly applicable as regulations but are less adaptable to local conditions and changes. However, recent studies show that adopting strategies that are real-time and responsive to environmental changes can reduce waiting times at traffic lights by 5 to 15 percent, [Balaji, P. G., German, X., & Srinivasan, D., 2010], [Alkhatib, A. A. A., et al., 2022]. For this purpose, a fuzzy logic controller was first used to manage an independent junction, [Liu, Z., 2007]. In this fuzzy controller, a macroscopic model was employed to simulate queue lengths and calculate delays. The results showed that this control system performed satisfactorily with respect to time constants. With the development of the fuzzy controller, a two-part fuzzy logic controller was introduced to manage traffic flow at junctions, [Qiao, J., Yang, N., & Gao, J., 2010]. The operation of this system was such that the first part determined whether the traffic light would operate in two-phase or four-phase mode, while the second part set the duration of the green phase. Although the use of fuzzy controllers results in a reduction in average waiting times and vehicle queue lengths, the effectiveness of such systems is inherently dependent on the quality of the rule base. These rules are typically designed by external operators based on predefined system models and remain fixed during operation, thereby limiting the controller’s ability to adapt to real-time environmental changes.

As urban transportation networks expanded, various complexities emerged, such as the stochastic nature, temporal flexibility, and numerous unpredictable factors, [Qadri, S. S. S. M., Gökçe, M. A., & Öner, E., 2020]. To address these challenges, traffic signal control strategies that rely on fixed and pre-calculated scheduling plans must evolve into flexible, real-time systems where signal timings are adjusted based on traffic flows during execution. In other words, traffic lights should make decisions based on the dynamic conditions of the environment at different junctions, rather than following a predetermined schedule plan, [Sun, G., Qi, R., Liu, Y., & Xu, F., 2024]. Traffic control systems must detect gradual variations in the traffic environment, estimate the nature of these changes, and adapt accordingly to enhance performance. Such systems must also manage unpredictable events—such as accidents—by coordinating traffic devices (e.g., traffic lights) and enabling cooperation among them. *Reinforcement Learning* (RL)–based *Machine Learning* (ML) methods provide an appropriate framework for designing and implementing these adaptive control agents. Traditional RL algorithms represent vehicle states as discrete values of traffic conditions [Bakker et al., 2010], [Abdoos, M., Mozayani, N., Bazzan, A.L., 2013], [El-Tantawy, S., Abdulhai, B., Abdelgawad, H., 2013]. As a result, the corresponding state-action matrix requires a significantly large storage space. With the increasing number of vehicles, the state space expands, leading to a decline in the efficiency and scalability of such methods.

Given the complexity and diversity of environmental conditions in urban transportation networks, the use of distributed strategies for traffic signal control enables the decomposition of the overall state space into independent and manageable subspaces [Rezaee, A., 2016]. This study aims to design a distributed RL–based urban traffic signal control system in which each junction operates independently and optimizes local traffic behavior without requiring an explicit model of the entire network. The novelty of this research lies in its architecture, which decomposes the state space into independent and controllable subspaces—effectively mitigating the curse of dimensionality.

2. Materials and Method

Intelligent Transportation Systems (ITS) represent a key component in the management of traffic within transportation networks. The implementation of such systems aims to achieve or maintain a balance between transportation supply and demand with minimal cost and high efficiency. Although various efforts have been made to automate traffic light control, RL–based ML methods offer distinct advantages over other intelligent approaches, [Eom, M., & Kim, B. I., 2020], [De Oliveira, L. F. P., Manera, L. T., & Da Luz, P. D. G., 2020].

Many intelligent traffic signal control systems require a predefined model of the environment to estimate short-term traffic flow patterns, [Reza, S. et al., 2021]. However, model-based approaches are limited to the specific conditions under which the model was originally developed. These methods often fail to adapt to unforeseen traffic dynamics, changes in road conditions, accidents, or other emergency situations that deviate from the initial assumptions. In contrast to model-based approaches, RL methods do not rely on a predefined environmental model to select actions. Instead, the relationship between *state*, *action*, and *reward* is learned through the agent's dynamic interaction with the environment, as illustrated in Figure 1.

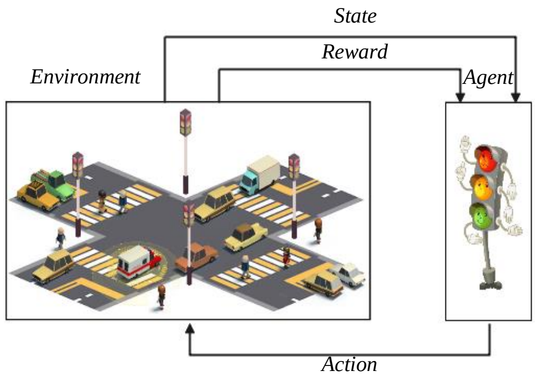

Figure 1. RL based Traffic Signal Control; The agent interacts with the environment by observing the current traffic state, selecting actions to adjust signal phases, receiving rewards based on traffic improvements, and updating its policy accordingly

One of the key advantages of RL methods is that they do not require supervision during the learning process. In contrast, supervised learning approaches—such as *Neural Networks* (NNs)—depend on a large number of input-output pairs, where each environmental state must be matched with its corresponding desired action. These samples must comprehensively cover the range of possible scenarios the agent may encounter in the environment, [Safdari, A. et al., 2024]. In RL methods, decision-making quality across various states is dynamically acquired through a trial-and-error process.

Another significant advantage of RL is its adaptability—not only can it adjust to environmental changes, but such adaptation can occur through continuous and real-time interaction with the environment, a process commonly referred to as *online learning*, [Safdari, A. et al., 2025].

The primary challenge in applying RL to traffic signal control lies in the vast state space [Nguyen, T. T., Nguyen, N. D., & Nahavandi, S., 2020], [Haydari, A., & Yılmaz, Y., 2020]. Specifically, modeling the exact number of vehicles passing through each lane as the system state not only leads to a substantial increase in the number of states, but also proves impractical for real-world implementation using conventional sensors currently deployed in urban environments. Most urban areas employ electromagnetic sensors embedded at intervals in the pavement, [Fahad, M., Nagy, R., Gosztola, D., 2022]. Due to the sparse deployment and wide spacing of these sensors, precise vehicle counts cannot be obtained; instead, only estimates of traffic speed and volume are provided.

In the proposed distributed RL framework, each agent perceives its local environment using readily available traffic sensors that respond to the metallic bodies of vehicles. This design enables the agent to operate based on abstracted, aggregate traffic features rather than exact vehicle counts, thereby ensuring compatibility with existing urban infrastructure while maintaining the feasibility and scalability of the control system.

3. Machine Learning

Machine Learning (ML), a branch of *Artificial Intelligence* (AI), encompasses a set of concepts from computer science and statistics, which are used to design algorithms that enable agents to learn relationships between data and improve their learning logic over time. In simpler terms, ML suggests that solving complex problems is not merely about mimicking human behavior but about imitating how humans learn, [Safdari, A. et al., 2024]. ML algorithms can be broadly categorized into three main types based on the type of data and learning approach: *Supervised Learning*, *Unsupervised Learning*, and *Reinforcement Learning* (RL), which are discussed in the following sections.

3.1. Supervised Learning

In the *Supervised Learning* paradigm, the available dataset is *labeled*; in other words, the data is pre-labeled with the correct answers, which represent the target values. The term “*supervised*” comes from the analogy of learning “*under the supervision of a supervisor*”. Each labeled training data can have multiple features, and feature extraction methods can be used to derive relevant features for training the learning model. Ultimately, these features and labels are fed into the model, allowing the ML algorithm to generate an output. By comparing this output to the original label, the algorithm can evaluate its prediction accuracy.

3.2. Unsupervised Learning

In the *Unsupervised Learning* paradigm, algorithms operate on *unlabeled* data. These methods identify inherent structures or hidden features within the input data and group the data into distinct categories. Specifically, the model compares data instances based on their extracted features and assigns similar instances to the same *clusters*. At the end of training, multiple clusters emerge such that instances within each cluster exhibit high similarity, while instances across different clusters differ significantly. Upon encountering a new data instance, the trained model evaluates its features against the established clusters and assigns the instance to the most similar cluster.

3.3. Reinforcement Learning

Reinforcement learning (RL) is a paradigm aimed at training an *agent* to perform a specific task in an uncertain environment without requiring pre-labeled datasets or predefined features. In this approach, the agent autonomously interacts with an unknown environment and, through continuous feedback, gradually acquires experience. The agent takes actions that lead to corresponding *rewards* or *punishments*, which serve as evaluative signals. These outcomes guide the agent's behavior by reinforcing beneficial actions and discouraging suboptimal ones.

In RL, agents are typically equipped with sensors that allow them to perceive information from the environment in order to determine their current state. In general, all RL agents follow a common structure in which they understand their environment without the need for an explicit mathematical model. Agents can select suitable actions and influence the environment as a result of those actions.

The primary objective in RL problems is to maximize the state-value function ($V^{\pi}(s)$) across all defined states of the system. As depicted in Figure 2, starting from an initial state (s), the agent takes steps according to a policy (π). The goal is to choose an action ($A_t=a$) such that the expected cumulative value, resulting from a sequence of actions, is maximized and leads the agent to a new state ($S_{t+1}=s'$). Mathematically, it is expressed as:

$$V^{\pi}(s) = \mathrm{E}_{\pi}\left\{R_t \mid S_t = s\right\} = \mathrm{E}_{\pi}\left\{\sum_{t=0}^{\infty}\gamma^{t}R_{t+1} \mid S_t = s\right\} \tag{1}$$

In Eq. (1), R_t denotes the cumulative reward received starting from state ($S_t=s$), and $\gamma \in [0,1]$ represents the *discount factor* that determines the importance of future rewards. Also, E_{π} $\{\cdot\}$ represents the *Expectation* with respect to the distribution over state and reward sequences induced by the policy (π).

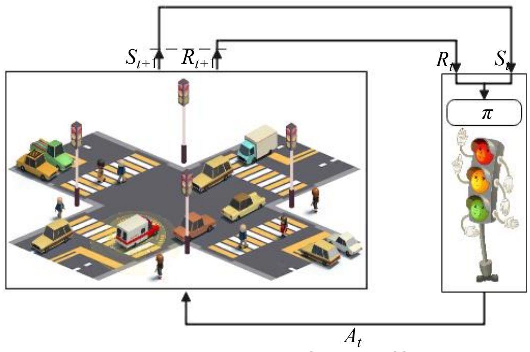

Figure 2. RL Framework for Traffic Signal Control; The traffic signal agent observes state ($S_t=s$), according to policy (π) selects action ($A_t=a$), transitions to a new state ($S_{t+1}=s'$), receives a reward ($R_t=r$), and then selects the next action *($A_t'=a'$)*

The traffic signal control, due to the continuous nature of traffic environments, is inherently a continuous control problem. Therefore, the goal is for the agent (i.e., the traffic light at the junction) to independently learn the environment model, so that any changes in the environment do not necessitate modifications in the agent's structure. Moreover, it is desired that the agent maintains its learning capability while continuously and interactively controlling the environment in real time. To address this, the On-Policy SARSA Algorithm, a method from the RL family, is employed. SARSA is based on the behavioral policy, meaning that the path followed by the agent during learning is the same as the path used for updating its knowledge. At each time step, the agent selects an action ($A_t=a$) in the current state ($S_t=s$), transitions to a new state ($S_{t+1}=s'$), receives a reward ($R_t=r$), and then selects the next action *($A_t'=a'$)*. The expected value (i.e., the Expectation) of the cumulative reward that an agent obtains by taking action ($A_t=a$) in state ($S_t=s$), and subsequently following policy (π), is given by Eq. (2):

$$Q^{\pi}(s,a) = \mathrm{E}_{\pi}\left\{R_t \mid S_t = s, A_t = a\right\} = \mathrm{E}_{\pi}\left\{\sum_{t=0}^{\infty}\gamma^t R_{t+1} \mid S_t = s, A_t = a\right\} \quad (2)$$

In Eq. (2) $Q^{\pi}(s, a)$ is the state-action value function. In RL, the state-value function ($V^{\pi}(s)$) represents only the value of a state (s) without considering the subsequent action (a); whereas, to evaluate the value of each action (a) in a specific state (s), the action-value function $Q^{\pi}(s, a)$ is used. This relationship is expressed in Eq. (3):

$$V^{\pi}(s) = \sum_{a\in \mathrm{A}(s)} \pi(a \mid s) Q^{\pi}(s,a) \quad (3)$$

As depicted in Figure 3, the $Q^{\pi}(s, a)$ for all *state-action* pairs are initially initialized to zero. The approach of initializing $Q^{\pi}(s, a)$ to zero ensures complete agent neutrality at the onset of learning, which aligns with the no prior knowledge hypothesis in autonomous learning systems design.

During each iteration, an action (a) is selected and executed based on a soft policy derived from the current $Q^{\pi}(s, a)$ estimates. Upon execution, the agent receives an immediate reward (r) and transitions to a new state (s').

Subsequently, a new action (a') is selected in state (s'), and the $Q^{\pi}(s, a)$ corresponding to the original *state-action* pair is updated using the SARSA update rule, which incorporates the observed reward (r) and the estimated value of the next *state-action* pair, as formulated in (4):

$$\begin{aligned} Q^{\pi}(s',a') &= Q^{\pi}(s,a) + \Delta Q^{\pi} \\ \Delta Q^{\pi} &= \alpha\left[r + \gamma Q^{\pi}(s',a') - Q^{\pi}(s,a)\right] \end{aligned} \quad (4)$$

In Eq. (4) $\alpha \in [0,1]$ denotes the step size and represents the learning rate.

The distributed RL framework has two main innovations that distinguish it from previous RL-based traffic control frameworks:

- Compact State Representation

Traditional RL-based traffic control systems often model the state space using fine-grained variables such as exact vehicle counts on each lane or individual vehicle dynamics. While such detailed modeling may benefit simulation fidelity, it results in a high-dimensional state space $\mathcal{S}$ that is impractical for real-time urban deployments due to limited sensor coverage and computational constraints. In contrast, we define the state $s \in \mathcal{S}$ cumulative features that are observable via standard inductive-loop detectors, such as vehicle queue lengths q_i at each approach i of the junction. Thus:

$$s = \left[q_1, q_2, \dots q_N\right] \quad (5)$$

In Eq. (5) N denotes the number of incoming lanes or directions monitored by the agent. This compressed representation significantly reduces the effective dimensionality of the problem, mitigates the curse of dimensionality, and retains enough traffic information to support effective policy learning.

- Distributed Policy Learning via On-Policy SARSA

The proposed framework enables each traffic signal to operate as a fully autonomous learning agent, independent from global coordination or prior knowledge of traffic models. Each agent interacts solely with its local environment and employs the on-policy SARSA algorithm to learn optimal control policies through continuous real-time feedback. This design ensures that decisions are made based on localized traffic observations, such as estimated queue lengths q_i, without relying on any predefined traffic dynamics $T(s,a,s')$.

SARSA On-Policy (Traffic Signal Control)

Initialize $Q^{\pi}(s, a)$, $\forall s \in \mathcal{S}$, $a \in \mathcal{A}(s)$

Repeat (for each iteration)

Initialize s

Choose a from s using policy (π)

Repeat (for each step of iteration)

Take action a' observe r, s'

Choose a' from s' using policy (π)

$$Q^{\pi}(s',a') \leftarrow Q^{\pi}(s,a) + \Delta Q^{\pi}$$

$$\Delta Q^{\pi} = \alpha\left[r + \gamma Q^{\pi}(s',a') - Q^{\pi}(s,a)\right]$$

$$s \leftarrow s'; a \leftarrow a'$$

Until s is terminal

Figure 3. SARSA On-Policy Pseudocode

3.3.1. Traffic Control Problem

An agent must avoid processing unnecessary details when solving the traffic control problem and should instead maintain a level of perception appropriate to the task. In the simulated environment used for modeling, vehicles are represented at a microscopic level solely based on their presence, while their individual characteristics are intentionally omitted. This abstraction is justified by assuming that the agent perceives its surroundings using conventional traffic sensors that detect the metallic body of vehicles. These sensors operate similarly to metal detectors and are widely deployed in urban areas such as Tehran and Mashhad due to their simplicity, low cost, and adaptability. In this study, traffic control at a junction within a transportation network is modeled as shown in Figure 4.

Figure 4. Junction Traffic Control Modeling

Traffic flow exhibits strong analogies with fluid in pipes; accordingly, this model is characterized by stochasticity, dynamism, and time-variance. Specifically, the vehicle arrival and departure rates on each road segment are entirely random and mutually independent, reflecting the underlying traffic conditions. All approaches to the junction are assumed to have equal length and no inherent priority; what matters for the agent is the ratio of the current traffic flow on each approach to its saturation flow, rather than the geometry of the road. To simplify the process of modeling and graphical simulation, the following assumptions have been made:

- It is assumed that traffic lights at the junctions are equipped with sensors capable of detecting the length of vehicle queues on the approaching streets.
- Each street is assumed to have four lanes, with all vehicles moving in a straight line (i.e., no turning at the junctions). In a distributed RL framework, each agent independently controls a specific junction. Therefore, the assumption of straight-through navigation at each junction does not imply a geometric simplification of the network. This is because vehicle turns at adjacent junctions are modeled as vehicles traveling straight through those adjacent junctions and are subsequently managed by the corresponding agents.
- The streets are oriented in a north-south and east-west configuration. Traffic flow is assumed to be random, with a constant average rate of vehicles entering the network from the northern and western entrances, and exiting through the southern and eastern exits.

The simulation framework allows for the creation of an arbitrary number of north-south and east-west streets. Control signals for each junction, which are independently generated, consist of the status of the traffic lights and the remaining time for the signal in its current phase. Figure 5 shows a schematic representation of the designed simulator. The vehicle queue length formed at each street approaching the junction is displayed next to the junction.

4. Results and Discussion

To investigate distributed traffic signal control based on ML using the SARSA algorithm, a graphical simulation environment was developed, as illustrated in Figure. 5. The vehicle queue lengths on each road approaching a junction are visually displayed adjacent to the corresponding junction. The maximum detectable queue length, as constrained by the sensor capabilities, is limited to 31 vehicles, which corresponds to approximately 150 meters.

The simulation results, considering the configuration settings and the average travel time through the network, are presented in Tables 1 and 2 for each individual road segment. All junction controllers were configured with identical parameter settings.

According to Tables 3 and 4, increased vehicle density at a junction prompts the traffic signal controller to correctly detect congestion and allocate extended green-phase duration, thereby reducing queue length and consequently decreasing waiting time. Quantitative results demonstrate significant improvements over conventional fixed-rule traffic signals, with the proposed distributed framework achieving average queue length reductions of 44.5% for east-west approaches (Table 3) and 66.4% for north-south approaches (Table 4). This improvement stems from each agent's autonomous learning capability, which leverages local sensory data without reliance on global traffic models. Such efficiency in unpredictable, dynamic environments is enabled by the online, interactive learning mechanism of RL, which continuously adapts control policies through real-time environmental feedback.

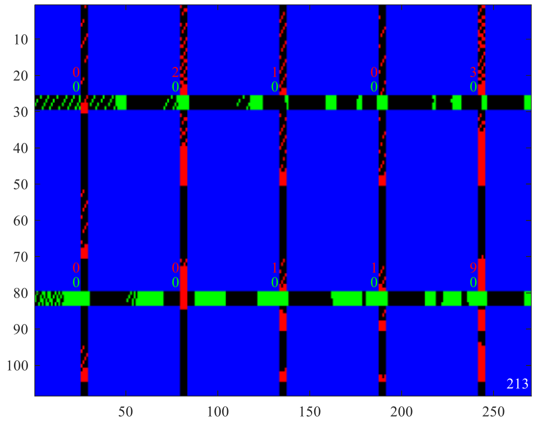

Figure 5. Schematic of the Designed Traffic Signal Control Software

Tabel 1. East-West Street Traffic Flow

Average Navigation ($\bar{x}$)					
Street Num. (N)	1	2	3	4	5
Traffic Rate $\times 10(\frac{\text{Car}}{\text{hour}})$	400	1000	600	400	1200
Travel Time [s]	81	85.5	82.8	81.7	86.7

Tabel 2. South-North Street Traffic Flow

Average Navigation ($\bar{x}$)		
Street Num. (N)	1	2
Traffic Rate $\times 10(\frac{\text{Car}}{\text{hour}})$	600	1200
Travel Time [s]	220.1	226.4

Tabel 3. East-West Street Traffic Flow

Average Reduction ($\downarrow\bar{x}$)					
Street Num. (N)	1	2	3	4	5

Vehicle Que. Reduction (%)	35	42.5	38	36.2	44.5
Wait Time Reduction [s]	21	25.5	22.8	21.7	26.7

Tabel 4. South-North Street Traffic Flow

	Average Reduction ($\downarrow\bar{x}$)	
Street Num. (*N*)	1	2
Vehicle Que. Reduction (%)	60.1	66.4
Wait Time Reduction [s]	100.1	106.4

Figure 6 illustrates the upward trend of the cumulative reward received by each traffic-signal agent, reflecting progressive learning under the SARSA-based distributed control framework. Concurrently, Figure 7 depicts the increasing vehicle exit rates on both east–west and north–south approaches, demonstrating the gradual alleviation of congestion and the effectiveness of the proposed control scheme in smoothing traffic flow over time. The observed increase in cumulative reward and vehicle exit rates is a direct consequence of the agent's adaptive learning behavior under the SARSA algorithm. Due to On-Policy, the agents explore various signal phase durations. Over time, as the agents experience state (*s*)–action (*a*)–reward (*r*) transitions, they learn to favor signal switching patterns that reduce queue lengths and improve throughput. These results confirm that the distributed RL approach not only enhances the agent's decision-making capabilities but also yields tangible improvements in network-level performance metrics, underscoring its potential for real-world traffic signal management applications.

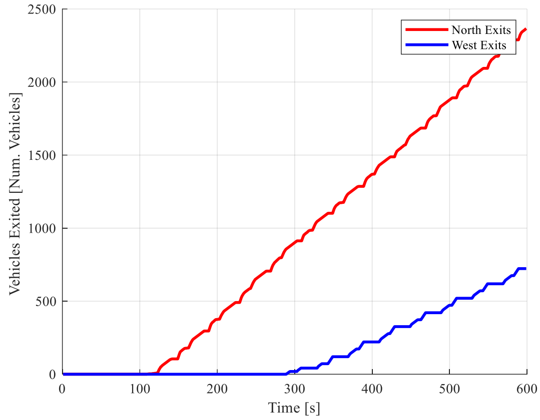

Figure 6. The reward accumulated by the traffic light agent for junction over time

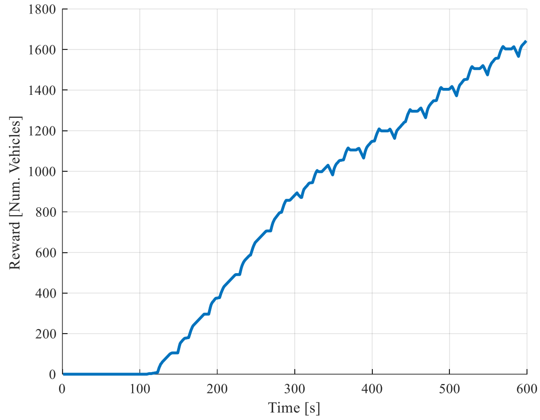

Figure 7. The number of vehicles exiting the junctions over time

5. Conclusion

Urban traffic control and the reduction of vehicle waiting time at junctions remain critical challenges in modern transportation systems. The limited adaptability of conventional computational algorithms in handling dynamic traffic conditions motivated this research to explore the potential of intelligent and learning-based systems. *Reinforcement Learning* (RL), as a *Machine Learning* (ML) approach capable of online decision-making and real-time interaction with the environment—even in the absence of an explicit model—was adopted as the core strategy in this study. The traffic signal control problem was effectively reformulated as an interpretable RL task, demonstrating how a complex control challenge can be addressed through adaptive decision-making. Simulation results confirmed the capability of the proposed method in improving traffic flow by optimizing the scheduling of traffic lights at junctions. Each traffic signal acted as an autonomous agent, interacting independently with its environment and learning to regulate the timing of its respective junction through continuous feedback. The increasing trend in the received rewards signified the agent's learning progress over time. Additionally, the growing number of vehicles exiting in both the east-west and north-south directions reflected the gradual alleviation of congestion and enhanced traffic management efficiency across the network.

Future research directions include extending the proposed framework to large-scale, heterogeneous urban traffic networks with increased topological complexity and multimodal interactions. Integrating pedestrian dynamics, public transportation schedules, and priority-based vehicle flows (e.g., emergency or transit vehicles) would further enrich the decision space and enhance practical applicability. Moreover, coupling the learning agents with real-time data streams from Vehicular Ad hoc Networks (VANETs), roadside infrastructure, and edge-based sensing units can significantly improve the responsiveness and robustness of the system under non-stationary and uncertain traffic conditions, ultimately facilitating deployment in smart city ecosystems.

6. References

– Ahmed, S. K., Mohammed, M. G., Abdulqadir, S. O., El-Kader, R. G. A., El-Shall, N. A., Chandran, D., ... & Dhama, K. (2023). Road traffic accidental injuries and deaths: A neglected global health issue. Health science reports, 6(5), e1240.

– Rezaee, A., Safdari, A., & Fazli, M. J. (2024). Automatic Path Planning for GMAW Welding on Locomotive Bogie in Low Contrast by Welding Robot. International Journal of Robotics, Theory and Applications, 10(1), 22-31.

– Rizwan, P., Suresh, K., & Babu, M. R. (2016, October). Real-time smart traffic management system for smart cities by using Internet of Things and big data. In 2016 international conference on emerging technological trends (ICETT) (pp. 1-7). IEEE.

– Ata, A., Khan, M. A., Abbas, S., Ahmad, G., & Fatima, A. (2019). Modelling smart road traffic congestion control system using machine learning techniques. Neural Network World, 29(2), 99-110.

– Abbas, S., Khan, M. A., Athar, A., Shan, S. A., Saeed, A., & Alyas, T. (2022). Enabling smart city with intelligent congestion control using hops with a hybrid computational approach. The Computer Journal, 65(3), 484-494.

– Bokaba, T., Doorsamy, W., & Paul, B. S. (2022). A comparative study of ensemble models for predicting road traffic congestion. Applied Sciences, 12(3), 1337.

– Balaji, P. G., German, X., & Srinivasan, D. (2010). Urban traffic signal control using reinforcement learning agents. IET Intelligent Transport Systems, 4(3), 177-188.

– Alkhatib, A. A., Maria, K. A., AlZu'bi, S., & Maria, E. A. (2022). Smart traffic scheduling for crowded cities road networks. Egyptian Informatics Journal, 23(4), 163-176.

– Liu, Z. (2007). A survey of intelligence methods in urban traffic signal control. IJCSNS International Journal of Computer Science and Network Security, 7(7), 105-112.

– Qiao, J., Yang, N., & Gao, J. (2010). Two-stage fuzzy logic controller for signalized intersection. IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans, 41(1), 178-184.

– Qadri, S. S. S. M., Gökçe, M. A., & Öner, E. (2020). State-of-art review of traffic signal control methods: challenges and opportunities. European transport research review, 12, 1-23.

– Sun, G., Qi, R., Liu, Y., & Xu, F. (2024). A dynamic traffic signal scheduling system based on improved greedy algorithm. PLoS one, 19(3), e0298417.

– Bakker, B., Whiteson, S., Kester, L., & Groen, F. C. (2010). Traffic light control by multiagent reinforcement learning systems. In Interactive Collaborative Information Systems (pp. 475-510). Berlin, Heidelberg: Springer Berlin Heidelberg.

– Abdoos, M., Mozayani, N., & Bazzan, A. L. (2013). Holonic multi-agent system for traffic signals control. Engineering Applications of Artificial Intelligence, 26(5-6), 1575-1587.

– El-Tantawy, S., Abdulhai, B., & Abdelgawad, H. (2013). Multiagent reinforcement learning for integrated network of adaptive traffic signal controllers (MARLIN-ATSC): Methodology and large-scale application on downtown Toronto. IEEE transactions on intelligent transportation systems, 14(3), 1140-1150.

– Rezaee, A. (2016). Control of Road Traffic Using Learning Classifier System.

– Eom, M., & Kim, B. I. (2020). The traffic signal control problem for intersections: a review. European transport research review, 12, 1-20.

– De Oliveira, L. F. P., Manera, L. T., & Da Luz, P. D. G. (2020). Development of a smart traffic light control system with real-time monitoring. IEEE Internet of Things Journal, 8(5), 3384-3393.

– Reza, S., Oliveira, H. S., Machado, J. J., & Tavares, J. M. R. (2021). Urban safety: an image-processing and deep-learning-based intelligent traffic management and control system. Sensors, 21(22), 7705.

– Safdari, A., Zarafshan, P., Alipour, K., Tarvirdizadeh, B., & Fang, G. (2024, December). Machine Learning based Multiple Impedance Control of a Space Free-Flying Robot. In 2024 12th RSI International Conference on Robotics and Mechatronics (ICRoM) (pp. 001-007). IEEE.

– Safdari, A., Mansouri, A., Fazli, M. J., Zarafshan, P., Alipour, K., & Tarvirdizadeh, B. (2025, February). PID Controller Design for a Mechatronic System using Lion Optimization Algorithm. In 2025 Fifth National and the First International Conference on Applied Research in Electrical Engineering (AREE) (pp. 1-7). IEEE.

– Nguyen, T. T., Nguyen, N. D., & Nahavandi, S. (2020). Deep reinforcement learning for multiagent systems: A review of challenges, solutions, and applications. IEEE transactions on cybernetics, 50(9), 3826-3839.

– Haydari, A., & Yılmaz, Y. (2020). Deep reinforcement learning for intelligent transportation systems: A survey. IEEE Transactions on Intelligent Transportation Systems, 23(1), 11-32.

– Fahad, M., Nagy, R., & Gosztola, D. (2022). Pavement sensing systems: literature review. Civil and Environmental Engineering Reports, 18(2), 603-630.