Received 21 September 2023, accepted 6 October 2023, date of publication 8 November 2023,
date of current version 8 December 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3331317

RESEARCH ARTICLE

Collaborative Traffic Signal Automation Using Deep Q-Learning

MUHAMMAD AHMED HASSAN[1], MOURAD ELHADEF[2], AND MUHAMMAD USMAN GHANI KHAN[1]

[1]National Center for Artificial Intelligence, University of Engineering and Technology, Lahore, Lahore 54000, Pakistan
[2]Computer Science and Information Technology Department, College of Engineering, Abu Dhabi University, Abu Dhabi, United Arab Emirates

Corresponding author: Muhammad Usman Ghani Khan (usman.ghani@uet.edu.pk)

This work was supported by the Office of Research and Sponsored Program, Abu Dhabi University, United Arab Emirates.

ABSTRACT Multi-agent deep reinforcement learning (MDRL) is a popular choice for multi-intersection traffic signal control, generating decentralized cooperative traffic signal strategies in specific traffic networks. Despite its widespread use, current MDRL algorithms have certain limitations. Firstly, the specific multi-agent settings impede the transferability and generalization of traffic signal policies to different traffic networks. Secondly, existing MDRL algorithms struggle to adapt to a varying number of vehicles crossing the traffic networks. This paper introduces a novel Cooperative Multi-Agent Deep Q-Network (CMDQN) for traffic signal control to alleviate traffic congestion. We have considered innovative features such as signal state at the preceding junction, the distance between junctions, visual features, and average speed. Our CMDQN applies a Decentralized Multi-Agent Network (DMN), employing a Markov Game abstraction for collaboration and state information sharing between agents to reduce waiting times. Our work employs Reinforcement Learning (RL) and a Deep Q-Network (DQN) for adaptive traffic signal control, leveraging Deep Computer Vision for real-time traffic density information. We also propose an intersection and a network-wide reward function to evaluate performance and optimize traffic flow. The developed system was evaluated through both synthetic and real-world experiments. The synthetic network is based on the Simulation of Urban Mobility (SUMO) traffic simulator, and the real-world network employed traffic data collected from installed cameras at actual traffic signals. Our results demonstrated improved performance across several key metrics when compared to the baseline model, reducing waiting times and improving traffic flow. This research presents a promising approach for cooperative traffic signal control, significantly contributing to the efforts to enhance traffic management systems.

INDEX TERMS Reinforcement learning (RL), multi-agent deep reinforcement learning (MDRL), computer vision, deep q-network (DQN), simulation of urban mobility (SUMO), decentralized multi-agent network (DMN).

I. INTRODUCTION

Globally, improvements in living standards have led to the growth and proliferation of commercial car manufacturers. As a result, the production of cars and other vehicles is experiencing an annual surge. In burgeoning urban areas of developing nations, there is an escalating issue of traffic congestion, a daily hurdle especially prevalent in large metropolises. China's Department of Road Transportation has publicized traffic-related data pointing towards a myriad of issues. The disclosed statistics suggest that the financial toll amounts to 20% of disposable income for city dwellers or a significant reduction of 5-8% in GDP. In comparison to residents of developed regions such as Europe, inhabitants of China's 15 most populous states spend an additional 2.88 billion seconds commuting to their workplaces [1]. Addressing traffic congestion, time inefficiencies, and pollution from vehicles, two primary solutions are commonly employed. One strategy involves increasing capacity by expanding road networks, an approach

The associate editor coordinating the review of this manuscript and approving it for publication was Xinyu Du.

© 2023 The Authors. This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 License.
For more information, see https://creativecommons.org/licenses/by-nc-nd/4.0/

that often proves expensive and inadequate in adapting to rapidly evolving vehicular circumstances. A more effective alternative involves enhancing the efficiency of the existing road network. Collaborative traffic signal management stands out as one of the pivotal actions to enhance junction performance and mitigate accidents [2].

Many global transportation infrastructures presently utilize fixed traffic signal cycles, signifying that road traffic signals are adjusted in a cyclic or sequential manner. Despite the simplicity of this method, it overlooks prevailing vehicular patterns on the roads and can potentially exacerbate the frequency of traffic congestion. Urban traffic congestion has grown to be a global issue for many years, with negative effects on both the environment and the economy. For instance, traffic congestion caused 121 million dollars in economic losses and 25,396 tonnes of carbon dioxide to be produced in the US in just one year [3]. The greenhouse effect will be exacerbated by the energy consumption brought on by traffic congestion and the consequent release of greenhouse gases, including carbon dioxide. to improve climate via technological innovation by reducing excessive pollutant emissions brought on by traffic congestion. Almost everyone is impacted by traffic lights on a daily basis. Traffic signals constitute a significant element of every urban journey, exerting a direct impact on drivers, the environment, and the economy regardless of the traffic volume on a route.

Numerous Traffic Signal Timing systems (TSTs) have been proposed by traffic engineers and researchers. The purpose of a TST is to orchestrate particular traffic signals with the aim of fulfilling operational objectives across the entire network [4]. A TST typically comprises a number of junction controllers, typically the optimum values for a set of signal timing variables that reduce or optimize an objective function (such as reducing delay, lowering travel time, etc.) and enhancing the flow of traffic, or devices that regulate how the intersection's traffic signals operate, a network of communications, and either a central computer or a series of computers to run the system. The TST problem is Either implicit (time-based) or explicit collaboration between the controllers can occur, such as through communication channels. Sadly, these systems struggle to function properly when unanticipated traffic jams happen. When it comes to research, the use of an agent paradigm for TST has always attracted the attention of researchers. Agent characteristics including distribution, autonomy, and collaboration are well suited for the internet traffic domain. Regarding TST several researchers have suggested using a range of methods for signal timing, including game theory [5], neural network modeling [6], fuzzy logic [7], and the widely employed reinforcement learning (RL) [8]. The majority of agent-based methodologies, architectures, and simulations come from academia. They are therefore frequently validated on basic simulated traffic networks and ease the signal timing problem. The majority of agent-based systems that were validated using data from the real world on models of real urban areas [9] are predicated on ideas that are unable to be applied in the actual world. A promising and efficient method to reduce traffic congestion and optimize traffic flow to improve traffic signal control (TSC).

Older naive methods that relied on fixed time controllers have been replaced by traditional adaptive TSC methods that create hand-crafted rules by analyzing actual traffic data. However, they only use historical data to feed time adjustment and are not sufficiently dynamic in adjusting to a current traffic situation [10]. The technique known as Largest Queue First (LQF) has proven to be reliable in its operation, which involves activating the green light on the route with the most vehicles queued [11]. However, LQF can potentially be unfair to vehicles queued in narrower lanes that may not receive an adequate green light duration [12]. In recent years, a number of studies leveraging machine learning have been introduced for managing urban traffic signals. These incorporate methodologies such as fuzzy logic, dynamic programming, evolutionary algorithms, and neural networks to regulate traffic lights at isolated intersections with varying traffic densities. A two-stage adaptive method based on fuzzy logic was proposed in [13]. Such systems based on fuzzy logic usually formulate a rule set informed by expert knowledge, which then guides the appropriate traffic light responses based on input. In [14], a task heuristic dynamic programming approach was suggested to control traffic signals at dual junctions. Dynamic programming requires efficient solutions to navigate the computational demands and the challenge of determining transition probabilities in an operational environment as the problem's scope expands.

Reinforcement Learning (RL) is a recent study area in which traffic lights are dynamically adjusted, typically based on actual traffic flow. The attributes of RL, which include "self-learning, data-driven, and no model," make it a potential approach for environments and transportation networks in complicated urban areas. RL has been widely employed by certain communities to address challenges in the realm of cognitive science. Its particular suitability for traffic signal management arises from two key attributes: the capacity for systematic exploration and the application of trial and error methods. RL provides an accurate representation of the elements in the traffic control system: the traffic signal as the agent, the traffic itself as the state, and the operation of the traffic signal as the action. However, in some extremely complicated urban traffic scenarios, there are more states, which leads to exponential complexity when using the traditional RL because it takes so much longer to explore all state-action pairs [3], [15]. Furthermore, storing transiently learned Q-values necessitates more storage space [16].

The research and development of TSC systems, which have been greatly boosted by utilizing various deep learning techniques, are better suited to deep RL (DRL) than classical RL. In a study [17], the agent receives incentives and uses deep neural networks (DNNs), and tries to maximize the expected rewards during the interaction between traffic signals and traffic circumstances. Reference [18] proposed a model-free, Reinforcement Learning (RL) oriented architecture for traffic signal management. The goal was to optimize the traffic flow duration through the signal in an urban arterial road system, thus reducing traffic delay times. Traditional RL methods often enlarge the state space using manually crafted intersection features. While simplifying state representation can curtail the extensive state space size, this strategy may result in the loss of essential data. With the advent of deep Q-networks (DQN), Deep Reinforcement Learning (DRL) has garnered considerable interest. This technique has been employed in various other studies [19], [20], yielding positive results. However, as scenarios grow more complex, the system's memory may become saturated when inspecting a specific condition within a large table, which is time-consuming. To mitigate this issue, Deep Neural Networks (DNNs) are employed in Q-networks (QN). A Dense QN was utilized by [21] to reduce the average system-generated delay in an 8-step traffic signal management system. Deep Q-network is a sorting method in RL, incorporating the advantages of Q-learning and Artificial Neural Networks (ANN) into a singular strategy. Several research studies for traffic signal management were published [22], [23], delivering noteworthy outcomes. However, upon comparison, these studies do harbor a few limitations that need to be addressed. To begin with, all these studies [24], [25] hinge upon a single specific improvement, which may not be adequate in real-time, traffic-based scenarios. Concentrating only on a limited area and reducing traffic on roads within a specific part of a city might not bolster overall efficiency. Additionally, even though attempts have been made for wider network coverage, including [20], [26], they rely on a static statistical model.

In numerous smart cities today, the Internet of Things (IoT) is employed to improve traffic management. It enables interaction and transfer of information between many parts of the entire transportation system, such as cars, lighting, and cameras. To improve the phase of the signals at intersections, the traffic signals at TSC are interconnected. The incorporation of RL into IoT can speed up and decrease latency in TSC decision-making. For a TSC that is focused on the entire transport network, however, it is necessary for numerous intersections to operate together, extending beyond local optimization towards global optimization and discovering multiple intersections' global optimization control to an optimal or suboptimal result. The use of multi-agent RL (MARL) to manage TSC on a massive scale has shown promise. By allocating the RL method toward local agents, existing MARL approaches have attempted to accomplish global traffic signal optimization control [26], and they have had some success. The proposed research is helpful in managing and controlling the traffic flow between the junctions and making the average waiting time of every vehicle at different road junctions. The traffic signal can decide the signal patterns by analyzing its adjacent traffic signals' status and the traffic flow.

The primary contributions of the work being proposed may be encapsulated in the following assertions:

1) The introduction of an innovative, cooperative multi-agent Deep Q-network, named CMDQN, is proposed, which centres on mutual state information sharing between agents to reduce the waiting times at traffic signals.
2) The consideration of new features, such as the status of the signal at the preceding junction, the distance separating two junctions, visual features, and the average speed traversing between junctions is emphasized for the collaborative management of traffic signals.
3) The proposed multi-agent network has also been trialled with actual traffic signals, where variables such as speed, flow length, and other sensor-derived data have been collected using cameras stationed at each signal.

The paper's remaining sections are structured as follows: Section II focuses on the related work, while section III outlines the paper's methodology. Section IV offers information on experiment and evaluation. Lastly, the conclusion of the paper is described in section V.

II. RELATED WORK

The timing of traffic signals has typically been approached as an optimization problem, which entails determining the best values for a number of signal timing parameters in order to minimize an objective function such as automobile travel time or delay. Methods based on Reinforcement Learning (RL) and Deep Reinforcement Learning (DRL) possess a number of attractive attributes. RL, for instance, is a learning approach rooted in environmental interactions and is goal-driven. On the other hand, deep learning showcases considerable capabilities for nonlinear approximation and hierarchical feature extraction. In this section, we initially review RL and DRL-based methods for traffic signal control. Subsequently, we scrutinize existing challenges and deliberate on the catalysts behind them. The urban TSC challenge, which is typically considered a conventional multi-agent system, has been developed via the optimization of a single intersection to several intersections. Numerous studies proposing RL-based traffic signal control systems primarily concentrate on single-traffic intersections [27], [28], [29]. With an increasing number of intersections, the state space grows exponentially, making the representation of all possible actions for each state impractical. Consequently, extending traditional tabular-based RL classifiers to multiple intersections presents a significant challenge. To address this issue, multi-agent RL-based methods have been developed for adaptive signal control in regional traffic scenarios [30]. The control strategy of these algorithms is bifurcated into two distinct categories: independent and integrated control mode. In this context, the Traffic Signal Control (TSC) serves as an agent at each intersection. The traffic light is controlled by each agent, and online collaboration is required. The multi-agent system's main objective is to lessen traffic congestion. The use of computing at the edge and IoT equipment makes traffic control easier. In a more expansive traffic system, a multi-agent network often serves as a representation [31]. Each agent is tasked with overseeing the traffic light at a specific intersection, with its state estimations based solely on local data from the intersection, like average queue lengths. However, this strategy, grounded in an independent approach, does not account for the influence of neighboring intersections. TSC is viewed as a "agent" across junctions with the capacity to make decisions. The current traffic state S_t and the reward R_t originate by examining the real-time flow of traffic [32]. The agent chooses and carries out the appropriate actions (to modify or maintain the lights) in accordance with the current situation. The agent then monitors how the action affects the intersection's traffic in order to determine the modified traffic status $S_{t+}1$ and a new reward R_{t+1}. The agent assesses the action and refines methods until it reaches the best "state and action". Real-time RL in TSC enables the learning of intermediary actions and incentives related to long-term traffic circumstances (such as the waiting time and the overall number of vehicles in a line). Three distinct representations are included in RL models: (1) State relates to the variables that influence decisions, such as the time of the wait, the speed, and the positioning of the vehicles in a lane. (2) Examples of actions include traffic phase divisions and phases. (3) Travel delay, as well as vehicle travel time, are two examples of rewards that indicate if state-action combinations are adequate [16]. In the integrated mode, agents employ various methods to coordinate signal control actions with their counterparts. As described in [33], two types of agents were designed: central and peripheral. The peripheral agents employ the Longest Queue First (LQF) approach to manage traffic signals and aid the central agent by providing proportional traffic flow data. The central agent then formulates value functions based on traffic patterns in its immediate vicinity. Thus, coordination is confined to the central agent, while the peripheral agents operate independently. A coordinated traffic signal management strategy based on cooperative models was put forward in [34]. Here, neighboring agents communicate with each other to understand the local state. However, the max-plus method, which is computationally intensive and susceptible to local optima, was used to ascertain the optimal collective action. In light of the remarkable performance of Deep Q-Networks (DQN), recent studies have begun employing RL algorithms for cooperative traffic signal control [35], [36]. QL is the foundation of one of the initial RL contributions to control traffic signals [37]. In [38], a CNN classifier was trained using QL with experience replay and discrete traffic state encoding. An auto encoder methodology [39] predicated on deep stacking has been integrated into Reinforcement Learning to forecast the optimal Q-value at a single intersection. In this encoding approach, the count of vehicles in a line and the associated reward are employed to represent traffic conditions and the variation in queues across roads traversing in orthogonal directions, respectively. A value-based DRL model that was derived from the DQN [28], in this study ideal action-value function was expressed using Bellman's equation. DNNs are used to build the traffic signal controller to control traffic signals in various real-world decision-making issues. Reference [39] Learning the Q-function of the RL method to control traffic signals involved stacking auto-encoders. The inputs of agents can be efficiently compressed and archived in this way. CNN was utilized in various research [3]. Reward and policy interpretation were both taken into consideration when [10] designed the DQN-embedded system to control traffic signals.

In their work, [38] utilized Deep Convolutional Neural Networks (CNN) to extract details regarding vehicle position and speed, as well as to estimate the optimal Q-value. This developed Deep Reinforcement Learning (DRL) agent was then trained for single intersection traffic control using Q-Learning and experience replay. While the algorithm displayed enhanced performance, its stability was undermined by potential correlations between possible action states and target values. To rectify this instability issue, [39] implemented a target network strategy. Furthermore, [38] observed that the majority of prior RL studies relating to traffic characteristics failed to accurately encapsulate the diversity inherent in real-world traffic scenarios. Instead, they opted to use video footage of an intersection to achieve this. Recently, a multi-agent deep Reinforcement Learning approach was employed by [40] to manage the signals at several straightforward intersections, excluding any left turns. Independent QL (IQL) over A2C with constrained communications was developed by [41]. The state first includes local observations. Then, each local agent autonomously learns its own policy, allowing for the modeling of other agents. The local agents are divided among the global Q-function. In scenarios involving only two agents, a Q-function is learned for low resource situations and is applied to other challenges [40]. The optimal coordinated collective action is eventually learned using the max-plus algorithm across various intersections. While the max-plus method is utilized in cooperative multi-agent systems, it does not guarantee convergence to the optimal result. Each traffic signal was regarded as a separate agent, and the round-robin technique was employed to determine the length of every agent's phases by multi-agent QL [42]. Based on the cost feedback signals received from its neighboring agents, every agent's Q-factor is changed. Numerous state-of-the-art studies using the RL technique for traffic control systems are presented in Table 1. The DQN model is either single-agent or multi-agent, as indicated by the second column in Table 1, in which the DQN method of RL modeling is mentioned in the third column. The fourth, fifth, and sixth columns describe the RL model's definitions of states, actions, and rewards, respectively. According to Table 1, the majority of studies concentrated on the queue length, which reveals the number of vehicles in the traffic path. The green and traffic phases are two frequently used actions, indicating, accordingly, the directions in which vehicles are permitted to pass the intersection simultaneously as well as the time for green lines to be modified. Waiting and traffic delays are both well-known rewards. Multi-agent DQN was studied by [16] in a Malaysian city. Reference [43] offered a three-state definition depending on the number of vehicles entering the green route, the number of vehicles backed up in the red direction, and the length of the line. This work's extension used various on-policy and off-policy RL methods presented in [44]. There were four reward operations: the number of stops, the number of stops immediately, and the total delay.

The fusion of RL and DNNs results in the creation of Deep RL (DRL) models, which exhibit enhanced more robustness the automatic feature and the traffic behavior insights extraction [45]. RL Policy controllers trained with DRL offer consistent and efficient inference duration, and as the intricacy of the problem space become broader, their computational complexity also escalates. Another benefit associated with DRL controllers lies in their robustness, a consequence of the abundant training data supplied by simulated environments that can offer massive variational data easily. RL is widely being adoptive in various domain such as hydraulic system controller [46], games [47], and traffic signal automation [45]. The DRL algorithms, which leverage DNNs, seek to learn meaningful embeddings of intersection network states [48]. These state embeddings play a pivotal role in shaping more effective signal control policies for the Traffic Intersection Network, encompassing intersections, traffic signals, road lanes, and vehicles [49]. Furthermore, the DDRL framework confronts the intricacies of the moving-target challenge, a result of the interplay among multiple control agents to adapt optimal Policies for the control network. In this context, rewards for an agent become contingent upon the actions of other agents within the network, thereby deviating from the MDP policy of static reward distribution. This departure from traditional MDP policies renders the convergence of learning objectives unattainable and compromises learning stability [50]. Compounded by a network environment characterized by partially observable features, agents within the DRL framework lack complete access to the state space and coordination is hindered due to the mutual dependence of one agent's actions on the actions of others.

While some studies have utilized real-world environment datasets to enhance agent performance, consequently improving traffic signal control, further exploration is needed concerning the cooperative management of traffic signals at different intersections. This could potentially reduce the average delay and waiting times for vehicles. In traditional traffic signal management systems, RL agents typically consider only the current intersection and determine the most effective course of action based on environmental factors such as queue length and delay time. Furthermore, in real-life situations, the time taken by a vehicle to traverse the distance between two adjacent signals can fluctuate due to various environmental influences such as elevated temperatures, precipitation, and fog. Cumulatively, modern traffic signal management systems necessitate more proficient RL agents that can collaboratively control signals and reduce the vehicle's average delay and waiting times across the entirety of a journey, not just at individual intersections. In the proposed study, we implement an RL-based DQN framework for cooperative traffic signal management, aiming to decrease the waiting time primarily and eventually reduce the average delay time, and sequence length of vehicles at intersections.

III. METHODOLOGY

Taking inspiration from multi-agent systems [48], [52] for traffic control, we have created a model for multi-agent signal automation (MSA) that utilizes a decentralized multi-agent network (DMN) to address waiting time issues for each intersection within the network. The DMN allows these individual intersections to work together as a collaborative multi-agent network (CMN), contributing by sharing state information to reduce the waiting of each signal as well as of the overall signal network. To create the DMN, we have employed a Markov Game abstraction, which is similar to the multi-agent network [53], enabling nodes in the DMN to collaborate by sharing state information and providing the path with the shortest waiting time. This collaborative network is also modelled as a DMN, similar to the one depicted in [52].

A. PROBLEM FORMULATION

Inspired from the related studies [52], we have modelled the multi-intersection traffic control as an n-agent ($n \geq 2$) partially observable MDP using tuple N, $A^n_{i=0}$, S,$R^n_{i=0}$, T, O,r, π, λ, where N is the number of decentralized agents, A is the action space of agent t(i=0,…,n), S is the state space, $R_i : S \times A \rightarrow R$ is the reward of ith agent, $T : S \times A \times S \rightarrow [0, 1]$ is the transition function, O is the global observation space, and $\pi_I = O_i \times A_i \rightarrow [0, 1]$ that works for maximizing the aggregated discounted reward $R_i = \sum_{t=r}^{T} \lambda^{t-1} r_i^t$ where $\lambda \epsilon$ [0, 1] is the discount factor.

1) AGENT ACTION SPACE

For each time-step, each signal of the MSA has a discrete action set, and the intersection can choose an action from this action space [X, X]. Potential action space is given in Equation 1:

$$a_i = \left\{\left(NSG, l_t^1\right), \left(NSLG, l_t^2\right), \left(EWG, l_t^3\right), \left(EWLG, l_t^4\right)\right\} \quad (1)$$

where, $\left(NSG, l_t^1\right)$ is North–South Green light, $\left(NSLG, l_t^2\right)$ is North-South Left Green light, $\left(EWG, l_t^3\right)$ is East–West Green light, and $\left(EWLG, l_t^4\right)$ is East-West Left Green light that MSA agent n turns on for the duration l_t^n (n = 1, 2, 3, 4) at time-step t. This action space aggregates all the potential combinations

TABLE 1. Commonly used RL-based QL and DQN approach works.

Reference/Year	Scale	RL-based Algorithm	State	Action	Reward
[46]/2014	Multi-Agent	Q-Learning	• Queue Length • Arriving Vehicles • Cumulative Delay	Green Phases	Cumulative Delay
[44]/2014	Multi-Agent	Q-Learning	• Queue Length	Traffic Phases	Stage Costs
[39]/2016	Single Agent	Q-Learning	• Queue Length	Signal Phase	Vehicle Delay
[40]/2016	Single Agent	DQN	• Queue length • Average speed • Waiting time	Right ways Phase duration	Traffic Delay
[29]/2018	Single Agent	DQN	• Turn Conditions	Signal Phase	System Delay
[3]/2018	Single Agent	DQN	• Vehicles Matrix • Queue Length	Phase DurationWaiting Time	Queue Length
[10]/2018	Single Agent	DQN	• Waiting Time • Vehicle Number • Waiting Time	Traffic Phases	Vehicle Number
[17]/2020	Multi-Agent	DQN	• Vehicle Speed	Traffic Phases	Travel Delay
[53]/2020	Multi-Agent	DQN	• Queue Length • Arriving Vehicles	Signal Phase	Queue Length

TABLE 2. Parameters of synthetic traffic network.

Parameters	S1, S2, S3
Traffic flow unit	14 vehicle/lane/hour
Multiple of flow unit	[1, 3, 5, 7, 9, 7, 5, 3, 1]
Vehicle length	4.0 m
Grid cell length	4.0
Maximum vehicle speed (v_{max})	56
Maximum acceleration	1.8 m/s^2
Maximum Deceleration	3.5 m/s^2
Lower green light duration limit (G_{min})	60 s
Upper green light duration limit (G_{max})	120
Yellow light duration (L_{yellow})	4.60 s

of individual actions available for each agent of the MSA. Duration of $l_t^n \epsilon [L_{min}, L_{max}]$, and the duration of yellow light is set to $l_{yellow} = V_{max}/a_{dec}$. Rest of the related parameters are given in Table 2.

2) OBSERVATION SPACE O

Each agent partially observes the agent state s ϵ S as their observation o ϵ O. At time step t, i_{th} agent state is defined as o_i^t, and the agent space state has three blocks: Nodes feature set F_t^X, an adjacency matrix M_t^A, and an intersection mask M_t^I. These blocks are computed as follows:

1) **Node features set** There are three node types named: intersection, traffic flow and road, denoted as $f_i^X = (F^I, F^T, F^R)$ and each of the nodes has various attributes as shown in Table 3. Traffic flow node is the core for global objective, and each of it's attribute is calculated as follows:

a) N_s^T is the traffic flow position for the intersection signal i, which is defined as a categorical variable, and ''one hot'' encoding is used to mention the signal position. Traffic flow position denoted based on signal of the each intersection position as follows: East signal [1, 0, 0, 0], West signal [0, 1, 0, 0], North signal [0, 0, 1, 0] and South signal [0, 0, 0, 1]. Each traffic flow has its own intersection locator vector that is updated by each DGN.

b) Actual speed of the incoming traffic flow is estimated as: $V_{es} = V_{historic} + V_v$, where Vhistoric is the speed estimation of the traffic flow corresponding to different time phases, V_v is the speed variation determined from the historical speed variation.

c) N_L^T refers to traffic flow location on the road at a given time step t. There is no sensory information about the traffic flow in between intersections, so, this location is inferred from the available information about the traffic flow and intersection environment as given in Equation 2.

$$N_L^T = P_i n - (V_{es} \times g) \tag{2}$$

Overall, feature map F^X for the time step t is the matrix of size $N \times 11$ where 11 is the total number of feature including intersection (F^I), traffic flow (F^T) and road (F^R). Feature vector of size 11 is vertically stacked for all intersections.

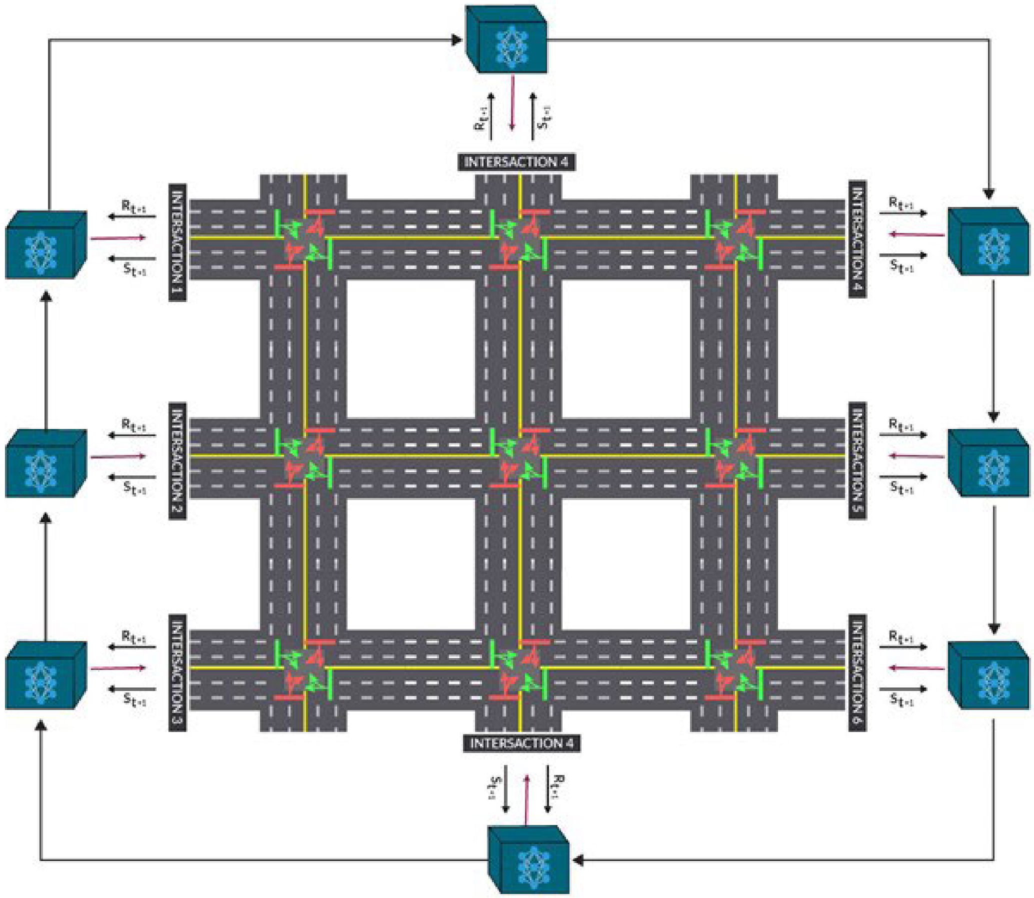

FIGURE 1. Traffic intersection network.

TABLE 3. Feature set of different node types.

Node type	Attributes
Intersection (F^I)	$Ison_{NSG}, Ison_{NSLG}, Ison_{EwG}, Ison_{EWLG}$, distance from previous intersection (P_in)
Traffic flow (F^T)	Intersection's signal position (N^s_T), Real-time speed (V_{es}), location on road
Road (F^R)	Length, average traffic speed, vehicle count

It is represented in Equation 3.

$$F_t^X = [f_i^X]_{i=0}^N \epsilon R^{N\times 11} \tag{3}$$

2) **Adjacency** At, a adjacency matrix M_t^A, is a binary N x N matrix that links the information dependency between and among intersections, where N is the number of intersections in the MSA. All the intersections of MSA are linked with each other through a graphical structure that builds a global collaborative network where each of the intersections share information with DMN. Connecting intersections make the policy decision implicit in the fusion block.
3) **Intersection mask** M_t^I, intersection mask, filters the intersection graph embeddings after the DMN fusion block. Mask vector is a binary vector of length N where 1s for the included intersection and 0s for the filtered-out intersection embeddings.

3) REWARD FUNCTION

Our MSA system utilizes two types of reward functions: intersection reward and network-wide reward function. The network-wide reward function is used to evaluate the collaborative performance of the DMNs, while the intersection reward function is used to enhance the reward by minimizing the waiting time and increasing the throughput at each signal. In order to achieve these objectives, each agent collaborates with neighboring nodes to increase its own reward and contribute to the rewards of others. Drawing inspiration from the principles of frequent measurability and spatial decomposability [54], we have defined a local reward function, r, for each intersection, which enables us to achieve these goals in a highly efficient and effective manner.

$$r_i = -\sum_{j\epsilon N_i} \left(q_{(i,j)}^{(t+l_t^n)} + W_{(i,j)}^{(t+l_t^n)} \right) \tag{4}$$

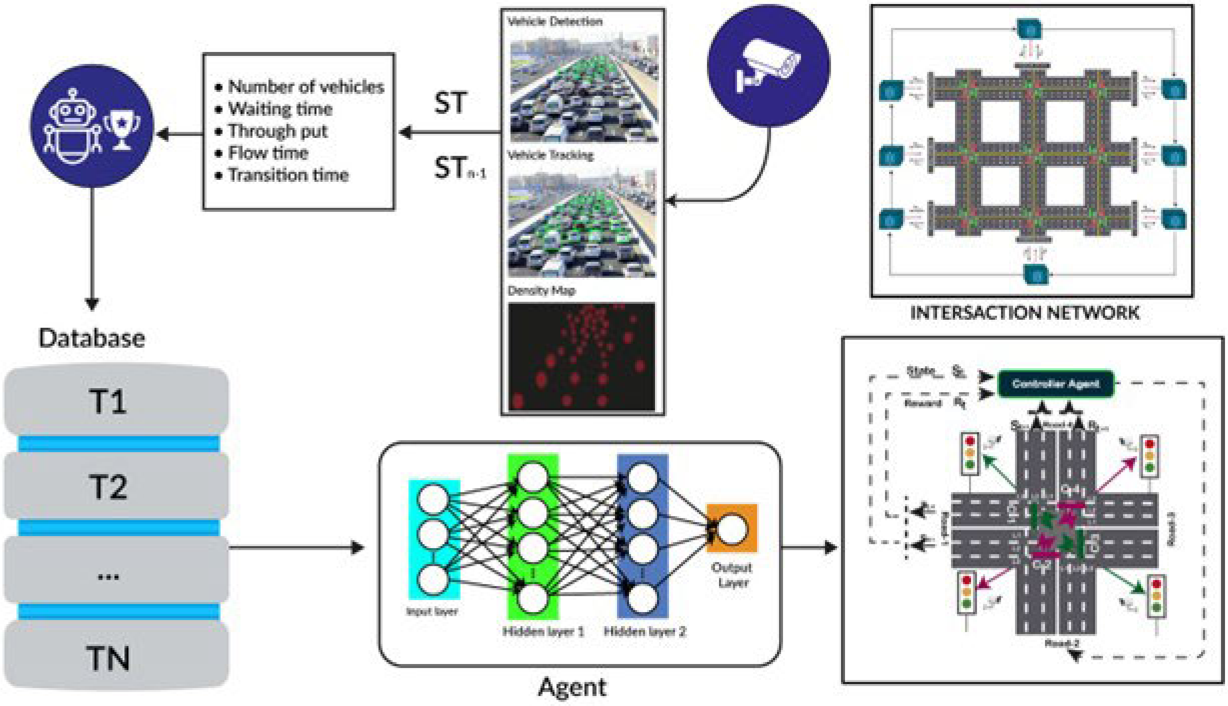

FIGURE 2. Workflow illustration of the proposed traffic control system. Camera input including flow length and flow time is fed to the agent where the Q-Network model determines the next state of the signal given the current state. The vehicle count of the flow is stored to build the simulation if desired.

In Equation 4 N_i refers to the set of intersection agents i's neighbors, l_t^n is the action duration at each intersection, $q_{(i,j)}^{(t+l_t^n)}$ is the queue length of the waiting vehicle at each intersection, and $W_{(i,j)}^{(t+l_t^n)}$ is the aggregated delay of traffic flow. Each agent in the network contributes to achieving a higher reward function for the subsequent agent, and this contribution is aggregated into MSA.

B. DEEP Q-NETWORK

Similar to [55], Our traffic signal system is designed to be adaptable through Reinforcement Learning (RL), which receives real-time information about traffic density through Deep Computer Vision. The main objective of the system is to optimize traffic flow through signal intersections by enabling controllers of adjacent signals to communicate with each other and coordinate to minimize the stop time for traffic at these intersections. Our system assumes that each traffic intersection is isolated and can control traffic flow without affecting neighboring intersections. Fig.1 depicts the workflow of our proposed control system. A camera installed on the road detects and counts the number of vehicles, and this data is used to analyze the intersection and build a simulation for neural network training. The model takes in vehicle flow lengths, vehicle flow times (i.e. the maximum stop times of vehicles at each signal), and the current state of the signal (i.e. traffic light) as input. The model then decides the next state of the signal while optimizing the stop time for traffic flow.

In order to optimize traffic control at intersections, we propose an RL-based method, as shown in Fig. 2. The intersection in question has four roads, each with its own separate signal unit. To gather information about traffic flow, stop time, and pass time, four cameras (C-1, C-2, C-3, and C-4) are installed on the four sides of the intersection. Each signal unit is managed by four agents, responsible for applying an action (A_t) to change the current state of the signal. At each time step, the controller agent receives the current state and current reward and determines the next state and future reward. When the method is first applied, the controller agent observes the present state of the intersection, S_t and determines the appropriate action, A_t, to be taken. Once the action has been applied to the signal, the intersection is updated to S_{t+1}, and the controller agent receives a reward, R_t, based on the effectiveness of the applied action. This reward mechanism helps to ensure that the correct action is selected for each situation.

1) ENVIRONMENTAL INFORMATION

To ensure that we are able to obtain real-time data on vehicle count, stop time, and flow duration, we have chosen to utilize a pre-trained Yolo object detection [56] for the detection and counting of vehicles in the traffic flow. This model consists of 53 convolutional layers, which enable it to extract features from a given frame. YOLO is capable of detecting multiple objects in real time, making it an ideal choice for our needs. When a traffic signal is activated, the model detects and counts the number of vehicles in the flow until the signal changes again. This count enables us to determine how many vehicles have passed through the signal at a specific time step. We then feed this information, along with the stop time and flow duration, as initial inputs to our RL model, allowing us to determine the appropriate action needed to minimize the stop time for traffic flow.

2) DEEP Q-NETWORK

The proposed controller for traffic control utilizes a combination of Deep Q-Network (DQN) [36] and Reinforcement Learning (RL) methods. DQN is a combination of Convolutional Neural Network (CNN) [57] and Q-learning [58], and is particularly effective at processing higher dimensional and large-sized inputs, such as traffic images. As deep Computer Vision is accurate about traffic count and flow time, DQN is a reasonable choice for this application, allowing us to avoid the need to measure and track flow length as input to the controller. DQN is a suitable choice for traffic control because it eliminates the need for environment simulation and learns the optimal policy for traffic control directly from the physical environment. The DQN receives the current signal state (St) along with traffic flow information and determines the next action (A_t). Once the decided action is applied to the traffic signal, the current signal state changes, and the controller agent receives the new signal state (S_{t+1}) and a reward (R_t), indicating the effectiveness of the last action. Because the traffic signal problem is sequential in nature, the objective of the controller is to select an action at each time step that maximizes the future reward. This cumulative future reward is determined by the state-action-value function, known as the Q-function. The optimal Q-function is determined through the use of DQN as shown in Fig. 3.

$$Q^{\star}(S_t, A_t) = max(\delta)E[R_t + \rho R_{t+1} + \rho^2 R_{t+2} + \ldots \mid \delta] \tag{5}$$

In Equation 5 discount factor ρ which indicates the difference between the immediate and future, $\rho = P(a \mid s)$in the policy of the controller agent which is the conditional probability of each action to be chosen against a specific state. Q-function in the form of the Bellman equation, assuming that the controller chose the action with the highest state value in each time step.

$$Q^{\star}(S_t, A_t) = E[R_t + \rho max(A_t) Q^{\star}(S_{t+1}, A)] \tag{6}$$

In DQN, the action-value function is an approximation in the form of CNN having parameters θ, $Q(S, A; \theta) \approx Q^{\star}(S, A)$. A CNN is trained to approximate the Q-function by minimizing the loss function $L(\theta)$ as represented in 7.

$$L(\theta) = E[(W_t - Q((S_t, A_t; \theta))^2] \tag{7}$$

where $W_t = (R_t - max(A_t)Q^{\star}(S_{t+1}, A'; \theta))$ is the network's target. RL methods when combined with nonlinear approximation functions become insatiable due to the correlation of interaction data between environment and agent, and correlation between target value and action-value function. To mitigate the problem of instability, we introduce the experience replay and period update of the network. Agent's experience $e = (S_t, A_t, R_{t+1}, S_{t+1})$ during interactions with the environment is saved in memory, called replay memory (D_t=e_t,e_{t-1},e_{t-2},…,$et - n + 1$, n is the memory size memory, and previous interactions ($(S_k, A_k, R_k, S_{k+1}) \sim U(D)$) feed to CNN. This mechanism of experience-replay reduces the correlation between the training samples. Different Q-network whose parameters (θ) parameters are updated with a lower frequency with respect to the actual Q-network (θ). Loss function updating at iteration i is represented in Equation 8:

$$L_j(\theta_j) = E(S_k, A_k, R_k, S_{k+1}) \sim U(D)[R + \rho max\acute{a} Q(S_{k+1}, \acute{A}; \theta_i) - Q(S_t, A_t; \theta_i))^2] \tag{8}$$

The target parameters (θ) are only updated the Q-network parameters (θ) at C steps and remains unchanged in between.

C. PROPERTY ANALYSIS

The proposed algorithm centers around a traffic control model leveraging the principles of multi-agent systems. Taking cues from multi-agent systems for traffic control, our approach introduces a novel model MSA which uses a DMN to reduce waiting times at individual intersections. By allowing intersections to collaborate via a CMN, they share state information, thus optimizing the waiting times not only for individual signals but for the overall signal network. This decentralized collaboration is realized through a Markov Game abstraction, a concept that echoes the dynamics of multi-agent networks. This represents the multi-intersection traffic control using a partially observable MDP with several key parameters including the number of agents, action spaces, state spaces, and the reward of each agent, among others. The agent action space essentially breaks down the possible actions an intersection signal might take at any given time-step, such as switching the North-South or East-West lights. Our model's observation space provides each agent with a partial view of the agent state. This is divided into node features for intersection, traffic flow, and road, an adjacency matrix, and an intersection mask. These features and matrices ensure that the agent has all the information it requires to make an informed decision.

Regarding the reward mechanism, the MSA system we propose has dual reward functions: one for individual intersections and one network wide. The intent is to enhance intersection performance while ensuring overall network optimization. Each agent's reward function is influenced by the queue length of waiting vehicles and the aggregated delay of traffic flow. By drawing from established principles of frequent measurability and spatial decomposability, the reward function is optimized for efficacy. Adopting methodologies similar to earlier works, our traffic signal system is adaptive through RL. It harnesses real-time traffic data, optimizing flow through signal intersections by allowing adjacent signals to coordinate. A crucial feature is the use of Deep Computer

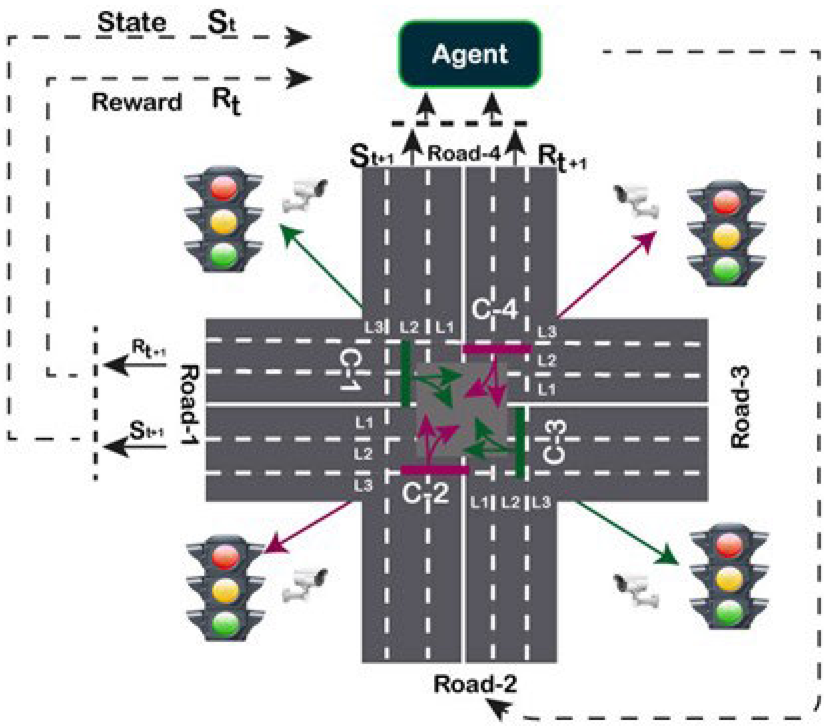

FIGURE 3. Visualization of the intersection where the RL agent interacts at discrete time step t = (1, 2, . . . , n), here n is the number of time steps. The objective of the controller agent is to minimize the stop time of the flow.

Vision to feed real-time traffic density data into the system. This RL-based control model involves cameras positioned at intersections feeding data to a DQN agent, which then determines the next state of traffic signals, all with the intent of minimizing vehicle stoppage times. Environmental data acquisition through a pre-trained YOLO that detects and tracks the vehicle flow count and then DQN predict the next action for the signal.

D. TRANSFERABILITY AND GENERALIZATION

Transferability and generalization of traffic signal policies across different scenarios is very vital for any signal automation system. RL in traffic signal automation offers potential in optimizing traffic flow, yet grapples with transferability and generalization challenges. Diverse traffic patterns, influenced by temporal, urban-rural, and cultural factors, question RL's universal applicability. Environmental unpredictabilities, such as weather or local events, further complicate models' adaptability. Each traffic intersection, with only a limited view of the entire system, faces the issue of partial observability. The vastness and complexity of state and action spaces in traffic scenarios pose significant learning hurdles for RL models. Non-stationarity of traffic environments necessitates frequent retraining, while the risk of model overfitting remains pertinent. Additionally, the intricacy of formulating an effective reward function can inadvertently skew RL agent behavior. In larger urban frameworks, the demand for multi-agent coordination introduces challenges in ensuring cohesive agent behaviors. Safety implications of direct RL deployment in real-world traffic scenarios and the existing simulation-to-real gap further underscore the need for meticulous evaluation and calibration.

Our proposed MSA) algorithm offers a robust framework for traffic signal automation. Key to its efficacy is the DMN which enables scalable, autonomous operation of intersections while maintaining inter-sectional communication. Complementing this is the CMN, which promotes adaptability by sharing intersectional state data, fostering a collective response to diverse traffic scenarios. The model's realism is heightened through the adoption of a partially observable MDP, reflecting the limited visibility characteristic of real-world intersections. Integrated node features provide a holistic environmental perspective, guiding decisions. Dual reward structures further ensure local and network-wide optimization. The integration of the DGN facilitates nuanced traffic pattern recognition, while real-time data input from the YOLO object detection system enhances adaptability. Continuous learning mechanisms underpin the model's evolving competence.

IV. EXPERIMENT AND EVALUATION

A. SETUP

Our proposed CMDQN traffic controller is trained on SUMO [59] traffic simulator, and evaluated on the SUMO-based synthetic network and Real-world traffic network where sensory information comes from the cameras installed at each signal. The required sensory information inputs from the simulator are obtained using SUMO Application Programming Interface (API) and fed to the controller network that is coded in Python. TCN receives flow length as the state and the mean reduction in the intersection delay as the reward [6]. It uses a ϵ-greedy action selection policy and refers to the reverse function against the number of visits against a particular state. This policy decreases after every 50 simulation interactions in an exponential fashion. We run the experiment on a machine having 128GB RAM and Nvidia GeForce RTX 4080 GPU on Ubuntu. As the SUMO simulator is slower than our controller network, our controller network is set to update its weights once against each time step of the SUMO.

CMDQN has 3 convolutional layers with the filter size is 3, 3, and 5 having a stride of 2, 2 and 4 respectively. We used a 1 filter after every 3×3 convolutional layer, it reduces the number of feature maps without reducing the size of the feature maps. The last convolutional layer is followed by the 2 fully connected layers with 64 and 34 hidden units, respectively. We set the red, yellow and green light times to 20, 4, and 4 respectively across all four intersection signals. Memory size for the experience-replay is set to 50000 so that information of longer time can be stored to observe the change over a longer period of time and used to optimize the policy. When the allocated memory becomes fully occupied then it starts replacing the interactions based on the first-in-first-out strategy. We have a batch size of 8 for the network training with the RMSprop optimizer having momentum of 0.2 and the learning rate is set to 0.0000. The discount parameter is set to 0.75 for the training of the STC network.

B. TRAINING MECHANISM

Models are trained and evaluated on the three synthetic traffic scenarios across three high-fidelity traffic scenarios 2 × 2, 3 × 3 and 4 × 4 inbound traffic lanes on each road. These traffic scenarios featured varying road network structures and traffic signal programs:

1) Traffic scenario (S1): Structured with 2 phases and 2 × 2 inbound lanes.
2) Traffic scenario (S2): Designed with 4 phases and 3 × 3 inbound lanes.
3) Traffic scenario (S3): Configured with 4 phases and 4 × 4 inbound lanes.

For scenarios (S1) and (S2), a single inbound lane is dedicated to permissive and protected left turns respectively. Conversely, in scenario (c), two inbound lanes are reserved for protected left turns. In all scenarios, the intervals for change, clearance, and minimum green are set to T_y = 5 seconds, T_r = 3 seconds, and T_g, min = 15 seconds respectively. Each approach spans 500 meters. Given that the vehicle size and inter-vehicle gap in SUMO are 6 and 3 meters, we used cells of 10 meters in CMDQN, and a detection range of 300 meters to accommodate up to 30 connected vehicles per lane.

SUMO simulations, lasting for 5000 seconds, are used as episodes, with randomly generated traffic demand fostering a heterogeneous traffic environment. Each episode gives a random traffic penetration rate within the range [0, 1] and random traffic flows for insertion qe within [10, 2000] vehicles per hour for each entry approach ϵ. The traffic demand complied with a Poisson process with parameter $\Lambda_\epsilon = q_\epsilon^{-1}$. 5000. During the learning phase, the CMDQN agent underwent training for 6 million timesteps, equivalent to 50 hours on a GPU, in each scenario following an ϵ-greedy policy. Following this, in the deployment phase, the CMDQN agents are then evaluated in each scenario with the trained neural network weights, adhering to the optimal policy that was learned.

The ϵ-greedy action selection policy in CMDQN effectively balances exploration and exploitation during learning. This choice addresses the exploration-exploitation trade-off challenge in our SUMO-based CMDQN traffic controller. It strikes a balance between exploiting current knowledge (exploitation) and exploring new actions (exploration). Under ϵ-greedy, the traffic controller network (TCN) considers both optimal and random actions, vital for reinforcement learning. Initially encouraging randomness for varied experience, it shifts towards exploitation as learning progresses, favoring optimal actions. We implement a decreasing ϵ value over time. This policy aids CMDQN's training, gradually reducing randomness as experience accumulates. Incorporating this policy ensures systematic exploration and exploitation, enhancing CMDQN's adaptability in diverse traffic scenarios, evident in our research on synthetic and real-world networks.

C. SYNTHETIC NETWORK

When comparing the Coordinated Multi-Agent Deep Q-Network (CMDQN) with the baseline model, several contrasting trends emerge across different performance metrics. Starting with the intersection flow time, which measures the duration it takes for vehicles to navigate through an intersection, CMDQN records a higher value of 174.22 seconds compared to the baseline model's 87.09 seconds. This suggests that under CMDQN, vehicles are taking more time to cross intersections. The network flow time, denoting the time needed for traffic to traverse the complete network of intersections, exhibits a similar pattern. CMDQN again performs at a slower pace with a figure of 357.64 seconds, compared to the baseline model's more brisk 218.39 seconds. Consequently, the flow of traffic throughout the entire network is less speedy when the CMDQN model is employed. However, the tables turn when considering the single flow time, where CMDQN significantly outperforms the baseline. With a value of 38.12 seconds versus the baseline's 97.81 seconds, CMDQN drastically reduces the time needed for an individual vehicle to journey through the network. In terms of mean waiting

TABLE 4. Performance comparison of CMDQN with the baseline non-cooperative traffic controller.

Comparison matrix	CMDQN	Baseline
Intersection flow time (sec)	174.22	87.09
Network flow time (sec)	357.64	218.39
Single flow time (sec)	38.12	97.81
Mean waiting time steps	2.17	3.24
Mean flow length (vehicle count)	51.62	23.19
Flow length (std)	19.2	38.41

time steps, CMDQN again excels, boasting fewer steps (2.17) compared to the baseline model's 3.24. This indicates that under CMDQN, vehicles spend less time in idle wait at intersections. A significant divergence is seen in the mean flow length, measured in vehicle count, where CMDQN (51.62 vehicles) greatly exceeds the baseline model (23.19 vehicles). This implies that CMDQN allows more vehicles to successfully traverse intersections. Lastly, the standard deviation in flow length presents a smaller variability in CMDQN (19.2) than in the baseline model (38.41). This might indicate a more consistent and predictable traffic flow when utilizing CMDQN. Taken as a whole, while CMDQN might seem to lag in the intersection and network flow times, it shines in other crucial metrics. This suggests a more efficient and steady approach to traffic flow management compared to the baseline model. We run our test on 60 different simulations to analyze the performance of both networks under different circumstances. The number results of the comparison are presented in Table 4.

D. REAL-WORLD NETWORK

To evaluate the performance of CMDQN on the real-world, we have used the recorded traffic scenario using the SafeCity Lahore traffic and camera infrastructure. In real-world traffic scenarios, either most of the vehicles do not have sensors to provide the location and other sensory information or do not share the information with any main controller, while simulators like SUMO provides can provide information about each vehicle. Hence it seems unpractical to deploy the traffic signal automation framework in real-world scenarios. We have recorded the real-world traffic scenario comprised of 4 traffic intersections with 3 lanes on each side road. We have used the Yolov5 for vehicle detection from the stream received from the high-definition cameras installed on each roadside. We have compared the CMDQN with the baseline non-cooperative DQN and static signaling time. This comparison aimed to highlight the potential benefits of CMDQN in managing traffic flow at busy intersections, even in the face of the practical challenges present in real-world applications.

A comparison among the three models: CMDQN, Baseline, and Static Signaling, reveals some interesting trends. CMDQN registers the highest intersection flow time, clocking in at 214.22 seconds, significantly higher than the 180 seconds of Static Signaling and far outstripping the Baseline's 67.09 seconds. This suggests that vehicles under the CMDQN model are spending the most time at intersections. Similarly, CMDQN has the highest network flow time at 297.04 seconds, compared to the Baseline model at 188.81 seconds. This implies that it takes traffic a long time to navigate the entire network when managed by the CMDQN model. Data for network flow time under Static Signaling is not provided. Interestingly, even with the longer flow times, CMDQN boasts the lowest mean waiting time at just 7.17 seconds. This is lower than the baseline's mean waiting time of 9.24 seconds and significantly better than Static Signaling, which logs a hefty waiting time of 120 seconds. The discrepancy suggests that the CMDQN model, despite a longer total flow time, is more efficient at reducing idle time for vehicles. The Baseline model outperforms the others in terms of mean flow length, handling an average of 43.25 vehicles. This is slightly more than CMDQN's average of 39.62 vehicles and significantly less than Static Signaling's average of 58.28 vehicles. This suggests that the Baseline model tends to handle more vehicles on the road network, on average, than the CMDQN model. Lastly, regarding the standard deviation of flow length, Static Signaling has the highest variability at 25.41 vehicles, followed by Baseline at 17.01 vehicles. CMDQN exhibits the least variation, with a standard deviation of just 8.82 vehicles as shown in Table 5. In conclusion, CMDQN may present longer flow times, but it ensures a lower mean waiting time, indicating effective vehicle flow management compared to the other models. Conversely, while the Baseline model accommodates more vehicles on average, it also displays a higher variability. Static Signaling, despite facilitating a larger flow length, results in a substantially higher mean waiting time and higher flow length variability.

E. COMPARISON NETWORKS

1) DASMC

DASMC [60] comprises two principal components: control of virtual platoons and regulation of traffic flow. The control of virtual platoons is a distributed control strategy that transforms the two-dimensional motion of vehicles into a one-dimensional virtual platoon to facilitate intersection functions. The control of these virtual platoons is accomplished via a distributed adaptive sliding mode controller, capable of managing unknown parameters and disruptions in vehicle dynamics. Regulation of traffic flow, on the other hand, is a central control approach that oversees the influx of vehicles at the intersection to avert traffic bottlenecks. This regulation is executed with a model predictive controller that aims to minimize the total travel time for all vehicles at the intersection.

2) QT-CDQN

QT-CDQN [52] is a cooperative deep Q-learning algorithm that enhances learning efficiency by employing Q-value transfer. In the QT-CDQN model, a multi-intersection traffic

TABLE 5. Performance comparison of CMDQN with the baseline non-cooperative traffic controller.

Comparison matrix	CMDQN	Baseline	Static Signaling
Intersection flow time (sec)	214.22	67.09	180
Network flow time (sec)	297.04	188.81	340
Mean waiting time (sec)	7.17	9.24	120
Mean flow length (vehicle count)	39.62	43.25	58.28
Flow length (std)	8.82	17.01	25.41

network is conceptualized as a multi-agent reinforcement learning framework. Here, each agent is responsible for controlling a single intersection. The agents learn collectively to diminish the total travel duration for all vehicles across the network. To approximate the Q-function for every agent, QT-CDQN uses a deep Q-network. The Q-function offers a mapping from a state to a specific action value, providing an estimate of the utility of performing a particular action in a given state. QT-CDQN incorporates Q-value transfer to augment the speed of learning. This mechanism facilitates the transfer of knowledge from one agent to another, enhancing their learning process. In QT-CDQN, the Q-values of neighboring agents are included in the Q-network's loss function, expediting its learning.

3) MOA3CG

MOA3CG [48] which is developed based on a multi-step return technique and an off-policy asynchronous advantage actor-critic (A3C) algorithm. The algorithm models a multi-intersection traffic network as a graph and uses a deep neural network to approximate the Q-function, effectively learning traffic signal control through a process of trial and error. MOA3CG utilizes techniques such as multi-step return, off-policy learning, and asynchronous learning to improve learning efficiency, sample efficiency, and scalability, respectively.

4) CVLIGHT

CVLight [61] substantially minimizes traffic gridlock and boosts traffic efficiency. CVLight visualizes the traffic network as a graph, where individual intersections represent nodes, and the connections between these intersections symbolize the edges of the graph. The state of the graph is derived from the traffic situation at each intersection, such as the queue of vehicles at every entry point. The actions within the graph constitute the green signal durations for each intersection. The graph reward is negatively correlated to the overall travel time for all vehicles across the network. CVLight develops proficiency in controlling traffic signals via a process of iterative learning, beginning with a random policy, and then continually improving this policy based on the outcomes observed through graph interaction. CVLight employs a deep neural network to estimate the Q-function, which projects a specific state to an action value. This action-value denotes the potential benefits of taking a specific action within a given state. CVLight adopts a decentralized reinforcement learning (RL) strategy to acquire control over traffic signals. In this decentralized RL approach, each intersection learns its own policy independently. The intersections share traffic information with each other, enabling them to coordinate their actions and enhance the overall performance of the system.

F. PERFORMANCE COMPARISON

In the comparison of various models' average waiting time, as illustrated by the plotted curves in Figure 4, CMDQN notably outperforms the others. The trends demonstrated by CMDQN, QT-CDQN, and ICVLight are remarkably close, especially during the initial training stages, each hitting respective peaks at 18.44, 28.33, and 29.80 within distinct time steps. MOA3CG depicts a gradual ascent until 2600, followed by a steep surge until 3330, ending with a declining pattern. ICVLight displays a sharp uptick prior to simulation time 1970, decreases modestly until around 3510, and then maintains a consistent level. The peaks of average waiting time for CMDQN, QT-CDQN, and ICVLight occur at different simulation periods, yet they remain lower compared to DASMC and Metalight. After reaching their apex, CMDQN and QT-CDQN curves flatten, suggesting frequent stops at intersections due to traffic flow. As the simulation time advances, resulting in increased traffic in the network, the WT curves for QT-CDQN, DASMC, and ICVLight initially rise before reaching a consistently high level, a stark contrast to CMDQN. This pattern suggests that as simulation time progresses, vehicles are increasingly likely to halt at intersections and wait for green lights. Conversely, the CMDQN model exhibits a peak during the simulation time interval 1850-2200, followed by a swift downturn and maintenance of a notably lower level relative to the other models. This indicates that traffic is frequently encountering green lights, reducing waiting times significantly. Moreover, the CMDQN curve touches zero several times during the simulation period, reflecting instances when waiting was unnecessary. These observations suggest that CMDQN is efficiently communicating and successfully achieving our objective of creating a signal-free corridor once traffic flow has halted at the initial intersection.

S3 is the most complex traffic network among all that include the S1 and S2 traffic networks, so we have analyzed the performance in detail. Thus, the following parts about the S1 and S2 analyze the diversion of trends from S3. This tells the behavior of all models in different levels of complexity of traffic networks. CMDQN has surpassed the other model in all the evaluation metrics plotted in (Figure 5-6). Performance

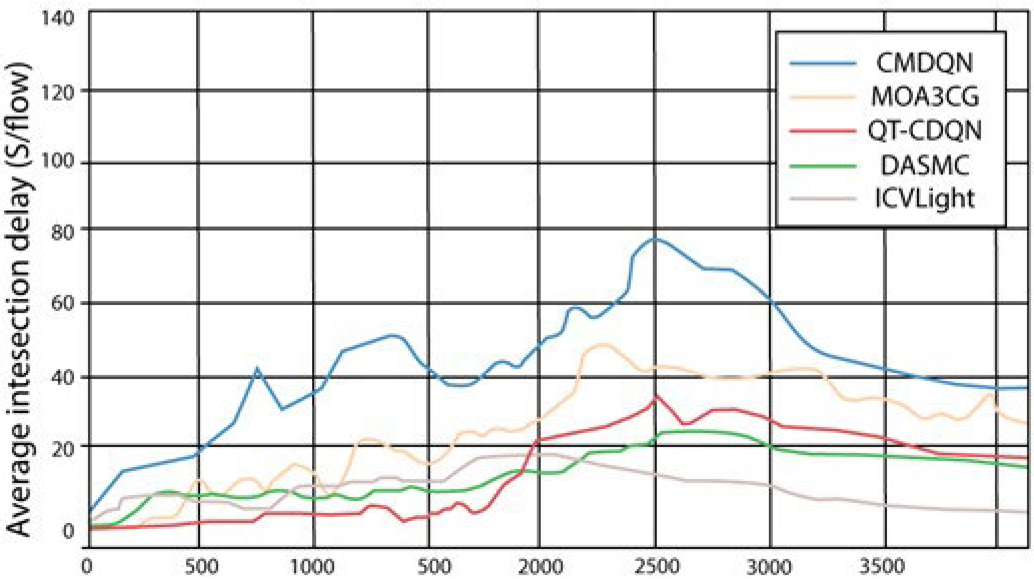

FIGURE 4. Average waiting time comparison for traffic scenario S3.

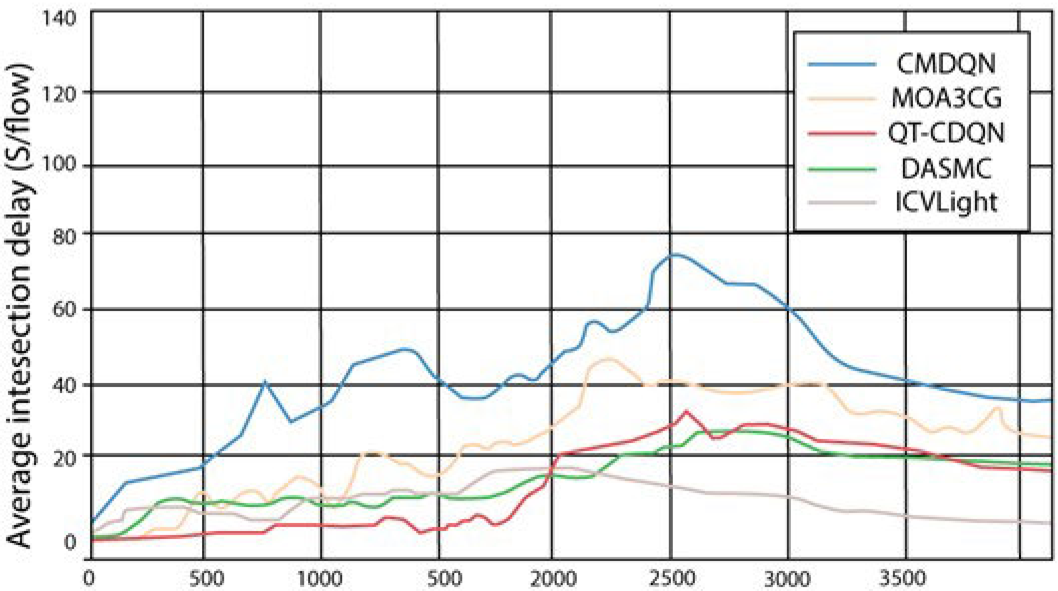

FIGURE 5. Average waiting time comparison for traffic scenario S2.

of CMDQN is relatively better on the S1 as compared to S2 and S3 because S1 is much simpler than others. As shown in Figure 7-9(a) it can be seen that waiting time for S1 and S2 becomes zero most of the simulation time which shows that traffic flows have the signal free corridor.

Mean episode delay (MED) is notably reduced with the application of CMDQN compared to other models as shown in Fig 7. This delay is averaged across varying traffic inflow rates, such as 0.1, 0.5, 0.7, and 0.9. At the peak inflow rate of 0.5 (Traffic flow/min), the MED for CMDQN is significantly lower, clocking in at 26.23, compared to CGB-MATSC, QT-CDQN, and DASMC, which exhibit 61.55, 69.5, and 102.70 MED respectively. Lower MEDs contribute to decreased average waiting and travel times. A core objective in this context is to minimize the MED, which in turn impacts both waiting and travel times. If the inflow rate is decreased, for instance to 0.1 (Traffic flow/min), the MED consequently reduces to 19. Fewer vehicles per minute results in less required coordination between intersections, leading to most traffic receiving the green light. However, ICVLight and MOA3CG appear to struggle with adapting their learning to new and fluctuating traffic flows. Their generalization capabilities are thus deemed minimal.

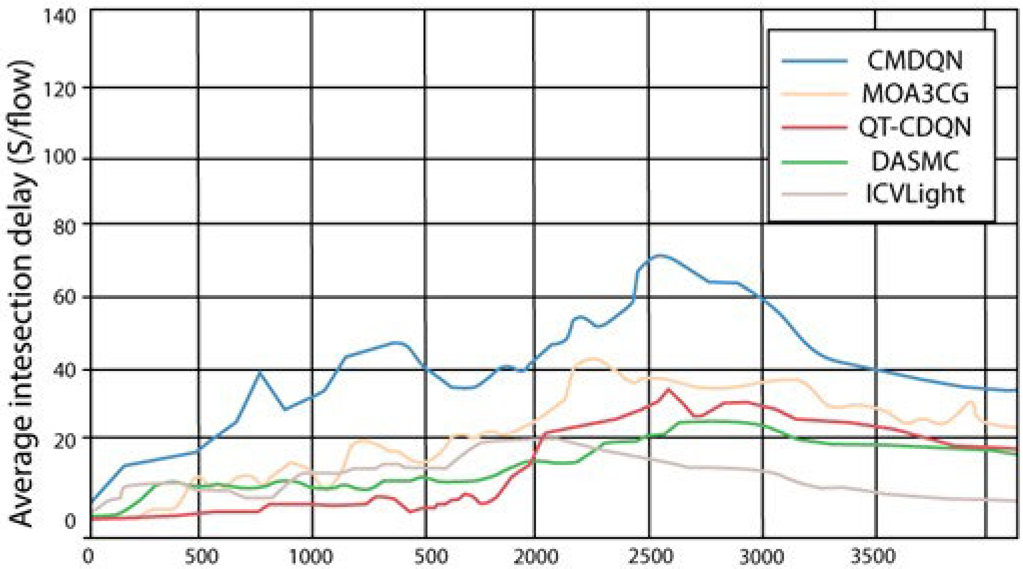

FIGURE 6. Average waiting time comparison for traffic scenario S1.

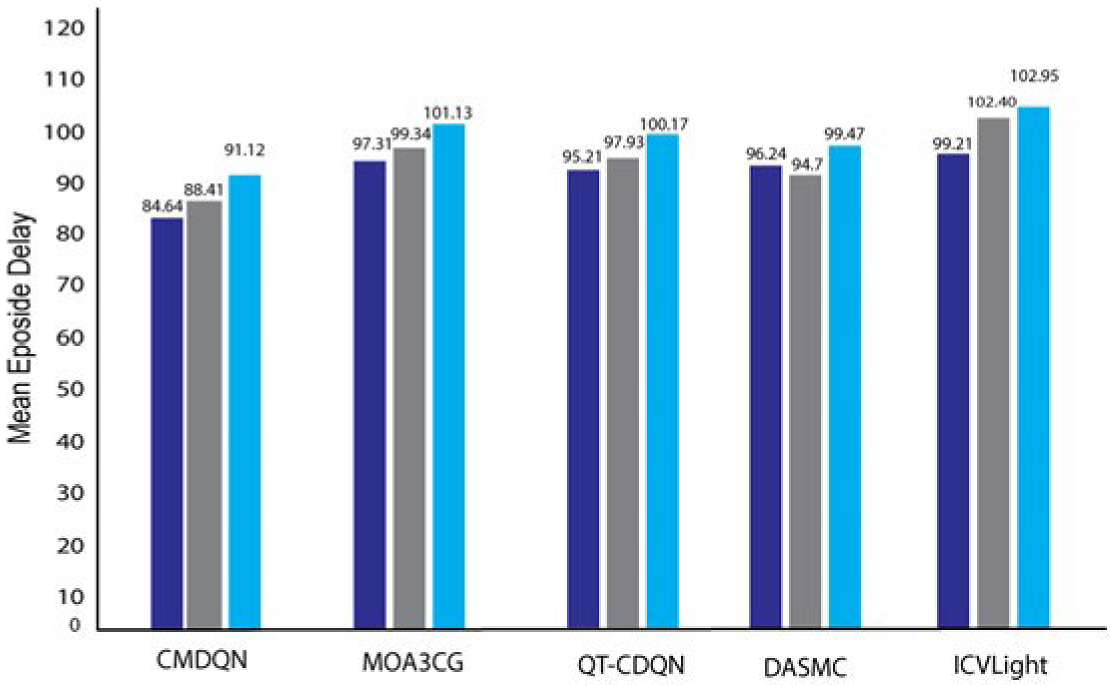

FIGURE 7. Comparison of mean episode delay.

Standard deviation (Std) of episode delay or S3 as illustrated in scenarios Figure 8, CMDQN significantly surpasses both QT-CDQN and ICVLight. Not only are both means considerably lower, but the disparity in standard deviations is also noteworthy, with CMDQN with QT-CDQN demonstrating lower std compared to the ICVLight algorithms. This is validated by the probability distribution curves, which exhibit condensed peaks centred around lower values for CMDQN with QT-CDQN, in contrast to the broader value ranges shown by both DASMC ICVLight and MOA3CG. These curves appear more diffuse and cover wider value areas. Consequently, this suggests that, when compared to actuated methods, CMDQN exhibits higher resilience in response to a wide array of traffic scenarios. This is especially pertinent for intricate intersections with fluctuating demands and S3 scenarios.

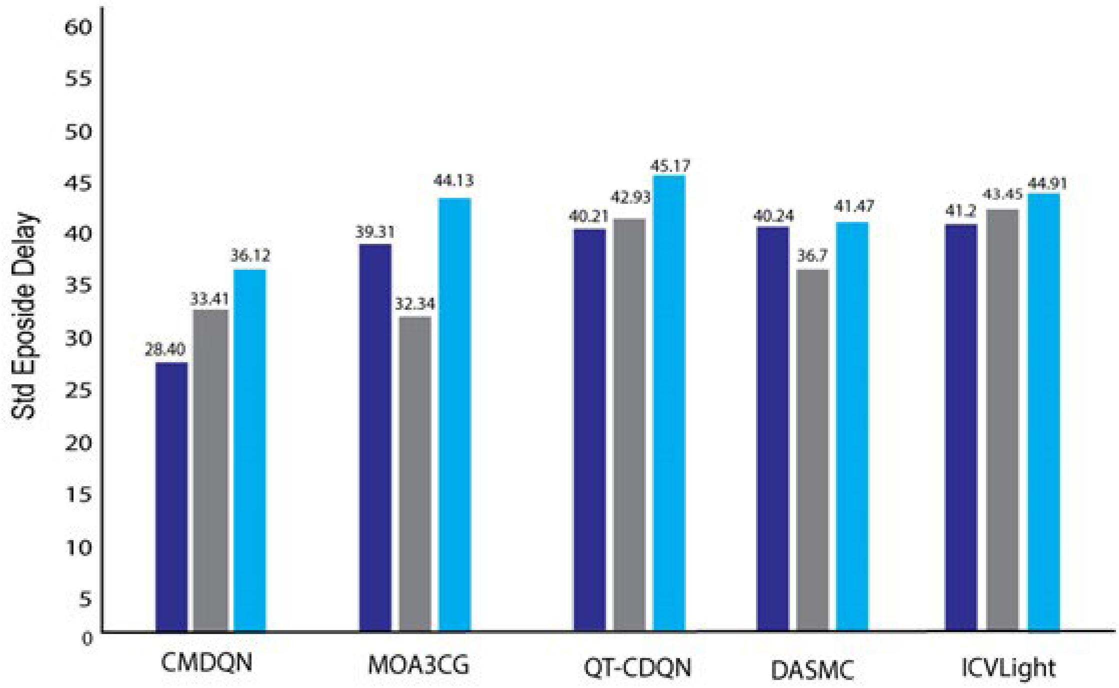

FIGURE 8. Comparison on Std episode delay.

V. CONCLUSION

In conclusion, the research paper presents a novel approach to traffic signal automation, leveraging a decentralized multi-agent network (DMN) model termed Multi-Agent Signal Automation (MSA). The core of the model utilizes two types of reward functions and a Deep Q-Network (DQN), to enhance the traffic flow and reduce the waiting times at intersections. The study draws inspiration from multiple multi-agent systems to create this collaborative multi-agent network (CMN) which effectively shares state information and minimizes the waiting times of signals across the network. The use of a Markov Game abstraction aids in the creation of the DMN and allows nodes to work together, providing the shortest waiting time path. In the model, the intersection and network-wide reward functions help optimize traffic flow by enabling controllers of adjacent signals to communicate and coordinate, thereby minimizing the stop time for traffic at intersections. The system also uses a reward mechanism to select the most efficient action for a given situation. The methodology employs a combination of DNN and Q-learning through a DQN, making it a reliable choice for traffic control. Its ability to learn the optimal policy for traffic control directly from the physical environment, without the need for environment simulation, is noteworthy. Moreover, the model was extensively tested using the SUMO traffic simulator across different scenarios. Despite CMDQN recording longer intersection and network flow times compared to the baseline model, it significantly outperformed in metrics like single flow time, mean waiting time steps, mean flow length, and standard deviation in flow length, indicating its superior efficiency and predictability in traffic flow management. The model was further tested and validated using real-world traffic scenarios from the SafeCity Lahore traffic and camera infrastructure. Despite real-world limitations like lack of sensor-equipped vehicles or sharing of sensor information with the main controller, CMDQN performed remarkably well when compared to non-cooperative DQN and static signaling time. The study concludes that CMDQN is a promising solution for managing traffic flow efficiently at busy intersections. However, practical challenges in real-world applications need to be addressed for large-scale implementation. Nonetheless, the model faces constraints in terms of scalability and performance, including its responsiveness to non-uniform traffic patterns and the decreasing likelihood of a more-signal free pathway as traffic volume and intersections rise. Future work will aim to overcome above mentioned constraints and develop a centralized information sharing and decentralized execution model potentially focusing on network-wide information sharing through centralized graphical information sharing network to provide signal-free pathway to priority side traffic flow.

REFERENCES

[1] L. Zhihui, C. Qian, Z. Yonghua, and Z. Rui, "Signal cooperative control with traffic supply and demand on a single intersection," *IEEE Access*, vol. 6, pp. 54407–54416, 2018.

[2] Y. Meng, L. Li, F.-Y. Wang, K. Li, and Z. Li, "Analysis of cooperative driving strategies for nonsignalized intersections," *IEEE Trans. Veh. Technol.*, vol. 67, no. 4, pp. 2900–2911, Apr. 2018.

[3] X. Liang, X. Du, G. Wang, and Z. Han, "Deep reinforcement learning for traffic light control in vehicular networks," 2018, *arXiv:1803.11115*.

[4] X. Zhang, X. Fan, S. Yu, A. Shan, S. Fan, Y. Xiao, and F. Dang, "Intersection signal timing optimization: A multi-objective evolutionary algorithm," *Sustainability*, vol. 14, no. 3, p. 1506, Jan. 2022.

[5] A. L. C. Bazzan, "A distributed approach for coordination of traffic signal agents," *Auto. Agents Multi-Agent Syst.*, vol. 10, no. 2, pp. 131–164, Mar. 2005.

[6] K.-H. Chao, R.-H. Lee, and M.-H. Wang, "An intelligent traffic light control based on extension neural network," in *Proc. Int. Conf. Knowl.-Based Intell. Inf. Eng. Syst.*, 2008, pp. 17–24.

[7] Y. Bi, D. Srinivasan, X. Lu, Z. Sun, and W. Zeng, "Type-2 fuzzy multi-intersection traffic signal control with differential evolution optimization," *Expert Syst. Appl.*, vol. 41, no. 16, pp. 7338–7349, Nov. 2014.

[8] A. L. C. Bazzan, D. de Oliveira, and B. C. da Silva, "Learning in groups of traffic signals," *Eng. Appl. Artif. Intell.*, vol. 23, no. 4, pp. 560–568, Jun. 2010.

[9] I. Dusparic, J. Monteil, and V. Cahill, "Towards autonomic urban traffic control with collaborative multi-policy reinforcement learning," in *Proc. IEEE 19th Int. Conf. Intell. Transp. Syst. (ITSC)*, Nov. 2016, pp. 2065–2070.

[10] H. Wei, G. Zheng, H. Yao, and Z. Li, "IntelliLight: A reinforcement learning approach for intelligent traffic light control," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Jul. 2018, pp. 2496–2505.

[11] R. Wunderlich, C. Liu, I. Elhanany, and T. Urbanik, "A novel signal-scheduling algorithm with quality-of-service provisioning for an isolated intersection," *IEEE Trans. Intell. Transp. Syst.*, vol. 9, no. 3, pp. 536–547, Sep. 2008.

[12] J. Wu, D. Ghosal, M. Zhang, and C.-N. Chuah, "Delay-based traffic signal control for throughput optimality and fairness at an isolated intersection," *IEEE Trans. Veh. Technol.*, vol. 67, no. 2, pp. 896–909, Feb. 2018.

[13] W. Yang, L. Zhang, Z. He, and L. Zhuang, "Optimized two-stage fuzzy control for urban traffic signals at isolated intersection and paramics simulation," in *Proc. 15th Int. IEEE Conf. Intell. Transp. Syst.*, Sep. 2012, pp. 391–396.

[14] L. Cao, B. Hu, X. Dong, G. Xiong, F. Zhu, Z. Shen, D. Shen, and Y. Liu, "Two intersections traffic signal control method based on ADHDP," in *Proc. IEEE Int. Conf. Veh. Electron. Saf. (ICVES)*, Jul. 2016, pp. 1–5.

[15] H. Xu, C. Zhang, J. Wang, D. Ouyang, Y. Zheng, and J. Shao, "Exploring parameter space with structured noise for meta-reinforcement learning," in *Proc. 29th Int. Joint Conf. Artif. Intell.*, Jul. 2020, pp. 3153–3159.

[16] F. Rasheed, K.-L.-A. Yau, and Y.-C. Low, "Deep reinforcement learning for traffic signal control under disturbances: A case study on Sunway city, Malaysia," *Future Gener. Comput. Syst.*, vol. 109, pp. 431–445, Aug. 2020.

[17] A. Zhu, F. Chen, H. Xu, D. Ouyang, and J. Shao, "Empowering the diversity and individuality of option: Residual soft option critic framework," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 34, no. 8, pp. 4816–4825, Aug. 2023.

[18] P. G. Balaji, X. German, and D. Srinivasan, "Urban traffic signal control using reinforcement learning agents," *IET Intell. Transp. Syst.*, vol. 4, no. 3, p. 177, 2010.

[19] A. Salkham, R. Cunningham, A. Garg, and V. Cahill, "A collaborative reinforcement learning approach to urban traffic control optimization," in *Proc. IEEE/WIC/ACM Int. Conf. Web Intell. Intell. Agent Technol.*, Dec. 2008, pp. 560–566.

[20] D. Li, J. Wu, M. Xu, Z. Wang, and K. Hu, "Adaptive traffic signal control model on intersections based on deep reinforcement learning," *J. Adv. Transp.*, vol. 2020, pp. 1–14, Aug. 2020.

[21] C.-H. Wan and M.-C. Hwang, "Adaptive traffic signal control methods based on deep reinforcement learning," in *Book Intelligent Transport Systems for Everyone's Mobility*. Singapore: Springer, 2019, pp. 195–209.

[22] H. Ge, Y. Song, C. Wu, J. Ren, and G. Tan, "Cooperative deep Q-learning with Q-value transfer for multi-intersection signal control," *IEEE Access*, vol. 7, pp. 40797–40809, 2019.

[23] X. Liang, X. Du, G. Wang, and Z. Han, "A deep reinforcement learning network for traffic light cycle control," *IEEE Trans. Veh. Technol.*, vol. 68, no. 2, pp. 1243–1253, Feb. 2019.

[24] S. S. Mousavi, M. Schukat, and E. Howley, "Traffic light control using deep policy-gradient and value-function-based reinforcement learning," *IET Intell. Transp. Syst.*, vol. 11, no. 7, pp. 417–423, Sep. 2017.

[25] Y. Chin, N. Bolong, S. S. Yang, and K. T. K. Teo, "Q-learning based traffic optimization in management of signal timing plan," *Int. J. Simul., Syst., Sci. Technol.*, vol. 12, no. 3, pp. 29–35, 2011.

[26] A. Daeichian and A. Haghani, "Fuzzy Q-learning-based multi-agent system for intelligent traffic control by a game theory approach," *Arabian J. Sci. Eng.*, vol. 43, no. 6, pp. 3241–3247, Jun. 2018.

[27] S. Touhbi, M. A. Babram, T. Nguyen-Huu, N. Marilleau, M. L. Hbid, C. Cambier, and S. Stinckwich, "Adaptive traffic signal control: Exploring reward definition for reinforcement learning," *Proc. Comput. Sci.*, vol. 109, pp. 513–520, Jan. 2017.

[28] C. Wan and M. Hwang, "Value-based deep reinforcement learning for adaptive isolated intersection signal control," *IET Intell. Transp. Syst.*, vol. 12, no. 9, pp. 1005–1010, Nov. 2018.

[29] P. Ha-Li and D. Ke, "An intersection signal control method based on deep reinforcement learning," in *Proc. 10th Int. Conf. Intell. Comput. Technol. Autom. (ICICTA)*, Oct. 2017, pp. 344–348.

[30] P. Mannion, J. Duggan, and E. Howley, "An experimental review of reinforcement learning algorithms for adaptive traffic signal control," in *Autonomic Road Transport Support Systems*. Springer, 2016, pp. 47–66.

[31] M. Abdoos, N. Mozayani, and A. L. C. Bazzan, "Traffic light control in non-stationary environments based on multi agent Q-learning," in *Proc. 14th Int. IEEE Conf. Intell. Transp. Syst. (ITSC)*, Oct. 2011, pp. 1580–1585.

[32] Q. Wu, J. Wu, J. Shen, B. Du, A. Telikani, M. Fahmideh, and C. Liang, "Distributed agent-based deep reinforcement learning for large scale traffic signal control," *Knowl.-Based Syst.*, vol. 241, Apr. 2022, Art. no. 108304.

[33] I. Arel, C. Liu, T. Urbanik, and A. G. Kohls, "Reinforcement learning-based multi-agent system for network traffic signal control," *IET Intell. Transp. Syst.*, vol. 4, no. 2, p. 128, 2010.

[34] L. Kuyer, S. Whiteson, B. Bakker, and N. Vlassis, "Multiagent reinforcement learning for urban traffic control using coordination graphs," in *Proc. Joint Eur. Conf. Mach. Learn. Knowl. Discovery Databases*, Antwerp, Belgium, Sep. 2008, pp. 656–671.

[35] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, "Playing Atari with deep reinforcement learning," 2013, *arXiv:1312.5602*.

[36] V. Mnih, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[37] B. Abdulhai, R. Pringle, and G. J. Karakoulas, "Reinforcement learning for true adaptive traffic signal control," *J. Transp. Eng.*, vol. 129, no. 3, pp. 278–285, May 2003.

[38] W. Genders and S. Razavi, "Using a deep reinforcement learning agent for traffic signal control," 2016, *arXiv:1611.01142*.

[39] L. Li, Y. Lv, and F.-Y. Wang, "Traffic signal timing via deep reinforcement learning," *IEEE/CAA J. Autom. Sinica*, vol. 3, no. 3, pp. 247–254, Jul. 2016.

[40] E. Van der Pol and F. A. Oliehoek, "Coordinated deep reinforcement learners for traffic light control," in *Proc. Learn., Inference Control Multi-Agent Syst. (NIPS)*, vol. 8, 2016, pp. 21–38.

[41] T. Chu, J. Wang, L. Codecà, and Z. Li, "Multi-agent deep reinforcement learning for large-scale traffic signal control," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 3, pp. 1086–1095, Mar. 2020.

[42] K. J. Prabuchandran, H. Kumar A. N, and S. Bhatnagar, "Multi-agent reinforcement learning for traffic signal control," in *Proc. 17th Int. IEEE Conf. Intell. Transp. Syst. (ITSC)*, Oct. 2014, pp. 2529–2534.

[43] S. El-Tantawy and B. Abdulhai, "An agent-based learning towards decentralized and coordinated traffic signal control," in *Proc. 13th Int. IEEE Conf. Intell. Transp. Syst.*, Sep. 2010, pp. 665–670.

[44] S. El-Tantawy, B. Abdulhai, and H. Abdelgawad, "Design of reinforcement learning parameters for seamless application of adaptive traffic signal control," *J. Intell. Transp. Syst.*, vol. 18, no. 3, pp. 227–245, Jul. 2014.

[45] H. Mukhtar, A. Afzal, S. Alahmari, and S. Yonbawi, "CCGN: Centralized collaborative graphical transformer multi-agent reinforcement learning for multi-intersection signal free-corridor," *Neural Netw.*, vol. 166, pp. 396–409, Sep. 2023.

[46] Z. Yao, X. Liang, G.-P. Jiang, and J. Yao, "Model-based reinforcement learning control of electrohydraulic position servo systems," *IEEE/ASME Trans. Mechatronics*, vol. 28, no. 3, pp. 1446–1455, Jun. 2023.

[47] B. Lian, W. Xue, F. L. Lewis, and T. Chai, "Inverse reinforcement learning for multi-player noncooperative apprentice games," *Automatica*, vol. 145, Nov. 2022, Art. no. 110524.

[48] S. Yang, B. Yang, H.-S. Wong, and Z. Kang, "Cooperative traffic signal control using multi-step return and off-policy asynchronous advantage actor-critic graph algorithm," *Knowl.-Based Syst.*, vol. 183, Nov. 2019, Art. no. 104855.

[49] F.-X. Devailly, D. Larocque, and L. Charlin, "IG-RL: Inductive graph reinforcement learning for massive-scale traffic signal control," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 7, pp. 7496–7507, Jul. 2022.

[50] R. Zhang, R. Leteurtre, B. Striner, A. Alanazi, A. Alghafis, and O. K. Tonguz, "Partially detected intelligent traffic signal control: Environmental adaptation," in *Proc. 18th IEEE Int. Conf. Mach. Learn. Appl. (ICMLA)*, Dec. 2019, pp. 1956–1960.

[51] C. Chen, "Toward a thousand lights: Decentralized deep reinforcement learning for large-scale traffic signal control," in *Proc. AAAI Conf. Artif. Intell.*, vol. 34, no. 4, 2020, pp. 3414–3421.

[52] T. A. Haddad, D. Hedjazi, and S. Aouag, "A deep reinforcement learning-based cooperative approach for multi-intersection traffic signal control," *Eng. Appl. Artif. Intell.*, vol. 114, Sep. 2022, Art. no. 105019.

[53] Y. Liu, W. Wang, Y. Hu, J. Hao, X. Chen, and Y. Gao, "Multi-agent game abstraction via graph attention neural network," in *Proc. AAAI Conf. Artif. Intell.*, vol. 34, 2020, pp. 7211–7218.

[54] A. J. Singh, A. Kumar, and H. C. Lau, "Learning and exploiting shaped reward models for large scale multiagent RL," in *Proc. Int. Conf. Automated Planning Scheduling*, vol. 31, 2021, pp. 588–596.

[55] Z. Tong, H. Chen, X. Deng, K. Li, and K. Li, "A scheduling scheme in the cloud computing environment using deep Q-learning," *Inf. Sci.*, vol. 512, pp. 1170–1191, Feb. 2020.

[56] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," 2018, *arXiv:1804.02767*.

[57] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, no. 6, pp. 84–90, May 2017.

[58] H. Van Hasselt, A. Guez, and D. Silver, "Deep reinforcement learning with double Q-learning," in *Proc. AAAI Conf. Artif. Intell.*, vol. 30, 2016, pp. 1–13.

[59] L. Codeca and J. Härri, "Monaco sumo traffic (most) scenario: A 3D mobility scenario for cooperative its," in *SUMO 2018—Simulating Autonomous and Intermodal Transport Systems* (EPiC Series in Engineering), vol. 2. Germany: EasyChair, 2018, pp. 43–55.

[60] A. Zhou, S. Peeta, M. Yang, and J. Wang, "Cooperative signal-free intersection control using virtual platooning and traffic flow regulation," *Transp. Res. C, Emerg. Technol.*, vol. 138, May 2022, Art. no. 103610.

[61] Z. Mo, W. Li, Y. Fu, K. Ruan, and X. Di, "CVLight: Decentralized learning for adaptive traffic signal control with connected vehicles," *Transp. Res. C, Emerg. Technol.*, vol. 141, Aug. 2022, Art. no. 103728.

MUHAMMAD AHMED HASSAN is currently pursuing the Ph.D. degree in computer engineering with the University of Engineering and Technology, Lahore. He is also a Team Lead with the Intelligent Criminology Research Laboratory, National Center of Artificial Intelligence. He has over six years of experience in the field of machine learning, deep learning, computer vision, and embedded systems. He developed many AI based solution for educational and industrial sector. His research interests include machine learning, deep learning, computer vision, embedded systems, and pattern recognition in visual data.

MOURAD ELHADEF received the B.Sc., M.Sc., and Ph.D. degrees in computer science from Institut Supérieur de Gestion, Tunisia, and the Ph.D. degree in computer science from the University of Sherbrooke, Sherbrooke, QC, Canada. He is currently a Professor of computer science with the College of Engineering, Abu Dhabi University, United Arab Emirates. He has over 50 peer-reviewed articles and conference proceedings to his credit. His current research interests include failure tolerance and fault diagnosis in distributed, wireless, ad-hoc networks, cloud computing, artificial intelligence, and security. He is on the Editorial Boards of several major conferences and journals, including IEEE Transactions on Parallel and Distributed Systems and the *Journal of Parallel and Distributed Computing*.

MUHAMMAD USMAN GHANI KHAN has over 18 years of research experience specifically in the areas of image processing, computer vision, bioinformatics, medical imaging, computational linguistics, and machine learning. He is currently the Director of the Intelligent Criminology Laboratory under the Center of Artificial Intelligence. He is also the Director and the Founder of five research labs, including the Computer Vision and Machine learning Laboratory, the Bioinformatics Laboratory, the Virtual Reality and Gaming Laboratory, the Data Science Laboratory, and the Software Systems Research Laboratory. He is a well groomed-Teacher and a Mentor for subjects related to artificial intelligence, machine learning, and deep learning. He recorded freely available video lectures on YouTube for courses of bioinformatics, image processing, data mining and data science, and computer programming.