smart cities

MDPI

Article

An Intelligent IoT Based Traffic Light Management System: Deep Reinforcement Learning

Shima Damadam [1], Mojtaba Zourbakhsh [1], Reza Javidan [1,*] and Azadeh Faroughi [2]

[1] Computer Engineering and IT Department, Shiraz University of Technology, Shiraz 71557-13876, Iran
[2] Computer Engineering and IT Department, University of Kurdistan, Sanandaj 66177-15175, Iran
* Correspondence: javidan@sutech.ac.ir

Abstract: Traffic is one of the indispensable problems of modern societies, which leads to undesirable consequences such as time wasting and greater possibility of accidents. Adaptive Traffic Signal Control (ATSC), as a key part of Intelligent Transportation Systems (ITS), plays a key role in reducing traffic congestion by real-time adaptation to dynamic traffic conditions. Moreover, these systems are integrated with Internet of Things (IoT) devices. IoT can lead to easy implementation of traffic management systems. Recently, the combination of Artificial Intelligence (AI) and the IoT has attracted the attention of many researchers and can process large amounts of data that are suitable for solving complex real-world problems about traffic control. In this paper, we worked on the real-world scenario of Shiraz City, which currently does not use any intelligent method and works based on fixed-time traffic signal scheduling. We applied IoT approaches and AI techniques to control traffic lights more efficiently, which is an essential part of the ITS. Specifically, sensors such as surveillance cameras were used to capture real-time traffic information for the intelligent traffic signal control system. In fact, an intelligent traffic signal control system is provided by utilizing distributed Multi-Agent Reinforcement Learning (MARL) and applying the traffic data of adjacent intersections along with local information. By using MARL, our goal was to improve the overall traffic of six signalized junctions of Shiraz City in Iran. We conducted numerical simulations for two synthetic intersections by simulated data and for a real-world map of Shiraz City with real-world traffic data received from the transportation and municipality traffic organization and compared it with the traditional system running in Shiraz. The simulation results show that our proposed approach performs more efficiently than the fixed-time traffic signal control scheduling implemented in Shiraz in terms of average vehicle queue lengths and waiting times at intersections.

Keywords: adaptive traffic signal control; intelligent transportation systems; internet of things; artificial intelligence; machine learning; multi-agent reinforcement learning

check for updates

Citation: Damadam, S.; Zourbakhsh, M.; Javidan, R.; Faroughi, A. An Intelligent IoT Based Traffic Light Management System: Deep Reinforcement Learning. *Smart Cities* **2022**, *5*, 1293–1311. https://doi.org/10.3390/smartcities5040066

Academic Editor: Pierluigi Siano

Received: 15 August 2022
Accepted: 22 September 2022
Published: 27 September 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

In the age of the Internet of Things (IoT), there are complex systems and many interconnected devices that produce large amounts of data. In recent years, Artificial Intelligence (AI) has been used to handle these devices and data and to supply intelligent control for complex scenarios due to its capacity and ability to manage complex tasks. Numerous studies have attempted to assess the efficacy of artificial intelligence and the Internet of Things (IoT) in relation to Intelligent Transportation Systems (ITS), which are crucial to the development of smart cities and the quality of people's daily lives [1–3].

On the other hand, traffic in urban areas is constantly increasing and the resulting congestion is a major concern for transportation management. One of the most important goals of research in the field of transportation at the global level is to optimize traffic flow. According to a report by the U.S. Department of Transportation [4], many cities face challenges in controlling traffic flows and reducing congestion. Thus, many algorithms and approaches have been proposed to solve traffic signal control challenges. Many cities use

a fixed-time traffic signal control system that works according to a predefined schedule. Nevertheless, they are not optimal because they are not affected by the traffic conditions, and traffic changes over time and is not constant. Hence, this method cannot adapt to the dynamics and changes in the environment and results in traffic [5].

With the advent of sensors (for example, loop detectors, radar, cameras, etc.) at intersections, it has become possible to implement Adaptive Traffic Signal Controllers (ATSC) which aim to optimize traffic flows by accommodating signal timing based on real-world traffic conditions. ATSC methods such as the Split, Cycle, Offset, Optimization Technique (SCOOT) [6], and the Sydney Coordinated Adaptive Traffic System (SCATS) [7], which are both centralized, have been used in many cities around the world to reduce congestion. However, these methods have some problems: they need many sensors, networks of computers for implementation, and a control center with a human operator to manage. Considering the fact that implementation and preservation costs are too high, it is not an optimal solution to deploy in metropolitan cities [8]. Optimization Policies for Adaptive Control (OPAC) [9] and the Real-Time Hierarchical Optimized Distributed Effective System (RHODES) [10], which are decentralized with wrapped computation [1] and a PRODYN algorithm [11], are similar methods that have been used for traffic control, but they have high computing costs. Although these controllers can change their phase duration or sequence, they cannot do it in real-time and dynamically. Adaptive approaches can potentially improve performance, but they are difficult to develop [12]. Reference [13] compares these methods and discusses their advantages and disadvantages.

IoT, on the other hand, is an environment and a platform that links people, devices, and computers by exchanging data through machine-to-machine or machine-to-human interaction [14], which alone has many complex systems, network infrastructures, and a large number of interconnected devices that generate vast amounts of data. In order to not only handle these devices and traffic data but also to control complicated scenarios intelligently, Machine Learning (ML) methods as AI techniques have been widely used in recent years due to their ability to perform complex tasks [1]. The ML methods, such as neuro-fuzzy [15], immune network algorithms [16], neural networks [17], and genetic algorithms [18] have been used for ATSC in recent years. Nevertheless, these methods require a lot of computational costs.

On the other hand, Reinforcement Learning (RL) is an effective ML approach among all the ML algorithms that has the powerful advantage of learning from experience and is used in the design of ATSC [19,20].

Although many Reinforcement Learning algorithms have been used to design ATSC, centralized RL-based control methods are not suitable when there are numerous intersections, because the state of all intersections must be collected as a general state and an action must be applied at each intersection. This will increase the delay and, in turn, the state space increases exponentially. Therefore, model training becomes difficult in such a complex situation. This is why a decentralized method is needed [21]. By combining ML and the decentralized nature of IoT, it is important to measure the usefulness of Machine Learning-based ITS because the ITS depends on people's daily lives and is one of the most crucial aspects of creating a smart city [1].

The purpose of this research is to apply a combination of ML and IoT approaches to provide an intelligent traffic signal control solution for multiple intersections. This is done by using Reinforcement Learning techniques where the RL agent learns the best control policy through collaboration with the environment. The observations of each intersection are exchanged with its neighboring intersection in a distributed way to obtain the optimal global schedule for the whole system [1]. Moreover, due to the limited communication among agents, the exchange of information between intersections becomes difficult. Therefore, a method of observations and fingerprints of neighboring agents is used to stabilize each local agent's learning [22]. To evaluate the effectiveness of the implemented algorithm, we conducted numerical simulations for two synthetic intersections by simulated data and a real-world map of Shiraz City (from the Open Street Map (Openstreetmap. [Online].

Available: https://www.openstreetmap.org (accessed on 1 January 2022)) (OSM)) with real-world traffic data received from the transportation and municipality traffic organization. The implementation is done on an opensource simulation platform, SUMO (Simulation of Urban MObility) [23]. The simulation results indicated that our proposed approach performs more efficiently than the fixed-time traffic signal control scheduling implemented in Shiraz in terms of vehicle average queue lengths and waiting times at intersections.

The contributions of this article are as follows:

- IoT technology and Machine Learning methods have been used to make intelligent traffic light control systems at Shiraz City intersections.
- Real traffic data have been used to implement a real-world scenario in Shiraz City to localize the system and consider all the challenges of a real-world scenario.
- A distributed Multi-Agent Reinforcement Learning (MARL) algorithm has been used at each intersection for traffic signal control.
- The cutting-edge advantage actor-critic (A2C) algorithm has been applied where deep neural networks (DNN) are used for policy and value approximations.
- The currently running traffic control system of Shiraz City, which has been implemented using SCATS, has been compared with the proposed method.

The remainder of this paper is structured as follows: Section 2 examines related works. Background and formulations are described in Section 3. In Section 4, the proposed method is introduced. Section 5 describes numerical experiments and evaluation results and finally, in Section 6, conclusion of the research is outlined.

2. Related Work

Researchers have always been interested in traffic signal control systems. Studies on this subject are divided into various traffic light control methods.

A. G. Sims et al. (1980) [24] introduced a system called SCATS, which is an urban traffic control system. The system consists of several small computers in the control center in Sydney. Specifically, it is an intelligent transportation system that manages real-time signal timing in traffic lights and uses traffic light sensors to detect vehicles in each lane. In this system, induction loops are used to detect the presence of vehicles. Information about the passage of vehicles is collected at intersections and transmitted to the traffic control center. The center analyzes this information then the appropriate green time, which is the system output, is reached. The SCATS system offers many benefits by reducing travel time, reducing accidents, saving fuel, and reducing air pollution. Moreover, implementing this control method, in addition to the high cost of purchase and installation, requires a human operator to control the system remotely, so it can be disrupted due to a lack of proper maintenance.

Hosur et al. (2019) [25] proposed a framework using IoT technologies that evaluate the traffic density via IR sensors to achieve dynamic timings for the traffic light. In their proposed system, they considered some threshold distance when the sensor detects any vehicle within this distance using IoT technologies. When other roads are empty of vehicles, it switches to a green light. The IoT can help to access components from far places, and their proposed system is beneficial for non-peak hours and saves power during non-peak hours. The disadvantage of their work is that they do not consider peak hours because most vehicles will only be present during these rush hours, which is an essential factor for traffic system control.

Liang et al. (2019) [26] changed the traffic light signal durations according to the discrete values of the actions. They collected the data from sensors and divided the whole intersection into small grids [21]. The information received from these sensors is difficult to process to find the duration of green and red lights. Such algorithms have low performance in peak traffic conditions, and the main reason is to ignore the impact of the current phase time on future traffic [27]. Value-based methods are more suitable for solving problems with discrete states than with continuous states such as traffic flows [21].

Lillicrap et al. (2015) [28] extended the idea of Q-learning to the continuous action domain. Although their proposed algorithm was able to discover policies whose performance was competitive with predefined scheduling algorithms due to the dynamics of the environment, it required a complete state sequence to update the policy. In fact, policy-based methods can work with continuous states, but the convergence of the training process is complicated, which is an important factor for continuous traffic flow [21]. In addition, this method has a high bias and variance [22].

Aslani et al. (2017) [29] proposed actor-critic adaptive traffic signal controllers to optimize traffic signal controllers in the traffic network of Tehran city for 24 h. They also developed different actor-critic algorithms based on different function approximations and compared them with six different scenarios. They showed that actor-critic in ATSC with centralized agents performed better than Q-learning. This work focused on discrete action RL but did not realize continuous actions.

Chu et al. (2019) [22] proposed a decentralized and fully scalable MARL algorithm for the deep RL agent called the advanced actor-critic (A2C) in the ATSC. They also proposed two methods to improve learning by enhancing observability and reducing learning difficulty for each local agent: the fingerprint of neighboring agents and spatial discount factor. They compared their multi-agent A2C algorithm with the independent A2C and IQL in both the synthetic traffic scenario and the real-world scenario of Monaco. The results of their work showed the optimality of the proposed algorithm compared to other decentralized MARL algorithms.

Wang et al. (2021) [21] also proposed an A2C algorithm, but they applied a region-aware cooperative strategy based on a graph attention network to overcome the problem of partial observability of each local agent.

Hongwei Ge et al. (2021) [30] proposed a MARL algorithm for traffic signal control. They also proposed transfer and encoder paradigms to enhance the agent's learning ability. Specifically, they improved the cooperation strategies of the algorithm. Their focus in this work was to increase the capability of agents to learn. They showed the robustness of their algorithm, but it limits its implementation in real scenarios. According to the research presented, Cases [24] attempted to control the traffic signals with SCATS which is an adaptive traffic signal control, but it cannot do it in dynamic conditions. Moreover, Case [25] proposed a framework using IoT. Cases [26,28,29] provided traffic light control based on reinforcing learning methods. Cases [21,22,30] proposed multi-agent Reinforcement Learning for traffic signal control. It should be mentioned that most previous research focused mainly on single intersections and simulated data, while it is quite rational to use real-world data from IoT sensors for multiple intersections. Moreover, there is no research available in the literature for Shiraz City that addresses traffic signal control using Machine Learning algorithms with real-world data.

In this paper, we used the Advantage Actor-Critic (A2C) algorithm combined with IoT approaches for the global control of each local agent. We also used observations and fingerprints inspired by neighboring agents in the state. Therefore, each local agent has more additional information about the distribution of regional traffic and cooperative strategy. In addition, this algorithm is implemented in a real-world scenario with six intersections of Shiraz City to consider all the challenges of a real-world scenario.

3. Background

3.1. Machine Learning (ML)

Machine Learning is a subset of artificial intelligence with applications in all fields that can learn from data and make predictions or decisions. There are three types of Machine Learning: supervised, unsupervised, and Reinforcement Learning. Supervised learning makes decisions based on labels of data during training. Unsupervised learning makes decisions based on pattern-finding without prior knowledge of labels, and Reinforcement Learning works based on the reward or penalty criteria it receives during training [31].

3.2. Deep Learning (DL)

Deep Learning or Deep Neural Network is a special Machine Learning scheme that allows a computational model to receive raw data, such as images, text, etc., as input and automatically discover the data representation for a variety of tasks. This computational model can efficiently extract information from data with a large number of states since it has numerous processing layers. Deep Learning works by using a back-propagation algorithm to determine how the computational model should change its internal parameters to obtain data at each layer that is used from the previous layer. These neural networks are initialized with a set of parameters θ, and map an input vector to an output vector through a number of hidden layers. Connections between neural network layer units (neurons) are known as weights (model parameters). Deep neural networks (DNN) are a type of neural network that have multiple hidden layers [32].

Deep Learning can be integrated with Reinforcement Learning, called Deep Reinforcement Learning, which is currently accepted as an advanced learning framework in control systems. While Reinforcement Learning can solve difficult control problems, Deep Learning also helps approximate nonlinear functions from complex data sets. Many Deep Reinforcement Learning methods have recently been used in various ITS applications. On the other hand, there is tremendous interest in control mechanisms like Reinforcement Learning in ITS, such as traffic control systems. In the following subsection, RL is defined in detail [31].

3.3. Reinforcement Learning (RL)

As mentioned, Reinforcement Learning is one of the types of Machine Learning that can be combined with Deep Learning so that it can be used in problems with a large number of states, such as the problem of traffic control, which is explained in detail below.

RL is a framework of MDP, a general mathematical framework of sequential decision-making algorithms. MDP consists of five members in one tuple:

- A series of states $\mathcal{S}$.
- A series of actions $\mathcal{U}$.
- Transition function $T(s_{t+1} \mid s_t,\ a_t)$ maps a state-action pair for each time t to the next state s_{t+1} distribution.
- Reward function that gives a reward when transitioning to the next state s_{t+1} for selecting action a_t from state s_t.
- Discount factor γ between 0 and 1 for future rewards [31].

Thus in a fully observable MDP, the agent at any time t observes the state of the environment $s_t \in S$, and performs an action $u_t \in \mathcal{U}$ based on a policy $\mu(u|s)$. Then, enters the next state using the transition function $s_{t+1} \sim p(\cdot|s_t,\ u_t)$ and receives an immediate reward $r_t = r(s_t,\ u_t,\ s_{t+1})$. The total future reward under policy π is defined as follows:

$$R_t^{\pi} = \sum_{\tau=t}^{\infty} \gamma^{\tau-t} r_{\tau}, \tag{1}$$

where $0 \leq \gamma \leq 1$ is a discount factor which is a trade-off between future and immediate rewards. The total expected reward is also shown as its Q-function $Q^{\pi}(s, u) = \mathrm{E}[R_t^{\pi} \mid s_t = s, u_t = u]$. The optimal Q-function $Q^* = max_{\pi}\ Q^{\pi}$, which leads to an optimal greedy policy $\pi^*\ (u \mid s)$: $u \in \arg max_{u'}\ Q^*\ (s, u')$, is derived from solving the Bellman equation $\mathbf{7}Q^* = Q^*$ according to the dynamic programming (DP) operator $\mathbf{7}$ [33]:

$$\mathbf{7}\ Q(s,\ u) = r(s,\ u) + \gamma \sum_{s' \epsilon S} p(s'|s,\ u)\ max_{u' \epsilon \mathcal{U}}\ Q(s',\ u'), \tag{2}$$

where $r(s, u) = E_s^{'}\ r(s, u, s')$ is the expected reward. In practice, r and p are unknown to the agent, so RL performs DP based on the sampled experience $(s_t,\ u_t,\ s'_t,\ r_t)$ instead of the above equation.

RL is a good choice for learning ATSC based on real-world traffic data [34]. It tries to learn optimal control based on interacting with the environment. For complicated

traffic conditions, the methods of this approach can solve traffic congestion problems more effectively [35]. Three methods of Reinforcement Learning are value-based (critic-only), policy-based (actor-only), and actor-critic.

In value-based methods, we train a neural network to learn a value-function. Then, we choose the action with the highest value. Although this method has low variance in estimating the returns, it requires an optimization method in each state to find the optimal actions in different states [29].

Policy-based methods directly optimize a policy without using a value function [36]. This method has the problem of high variance and slow learning.

Finally, the Actor-Critic (AC) method aims to take advantage of all the good stuff from both value-based and policy-based while reducing the bias and variance of policy-based methods [37]. Actor-critic algorithms are divided into two categories: actor-network, and critic-network. The actor uses the policy function to predict the probability distribution of all available actions. In fact, it is used to select actions. The critic also uses the value function to evaluate the performance of the action chosen by the actor so that, in the next state, the actor can choose better actions [21]. Actor-critic is one of the most complete and robust types of RL methods, which has the superiority of fast learning and the potential to perform precisely in unseen traffic situations [29].

3.4. Advantage of Actor-Critic

A2C is the synchronous version of the asynchronous advantage actor-critic (A3C) algorithm [38], and both of them update the policy gradient using the critic network. This network calculates the optimal state-value V_s based on the current state. The actor network uses this V_s to regularly update the parameter θ of the policy function and then select the next action a_{t+1}. The input of the actor-network is the local agent's observations, and the output is the action. Specifically, there are two types of value function approximations; state-value $V_\pi^*(s)$ and action value $Q_\pi^*(s,\ a)$ which can be described as follows:

$$\begin{aligned} V_\pi^*(s) &\approx Q(s, \mathcal{w}) \\ Q_\pi^*(s,\ a) &\approx Q(s, a, \mathcal{w}) \end{aligned} \tag{3}$$

The following describes the distinction between action-value and state-value:

$$Adv_t = Q(s,\ a) - v(s) \tag{4}$$

$v(s)$ is state-value function at time step t, and $Q(s,\ a)$ is the action-value function corresponding to an action at the current state. After the advantage between action-value and state-value is obtained, the critic network is updated using the Mean Square Error (MES) according to the following equation:

$$\mathcal{L}(\mathcal{w}) = \frac{1}{2|B|}\sum_{t\epsilon B}(R_t + \gamma V(s') - V(s)^2 \tag{5}$$

Meanwhile, the loss function of actor-network needs to be updated.

$$\mathcal{L}(\theta) = -\frac{1}{|B|}\sum_{t\epsilon B}\log \pi_\theta(s_t,\ a_t) Adv_t \tag{6}$$

3.5. Multi-Agent Reinforcement Learning

Many real-world issues involve controlling multiple intersections simultaneously. Therefore, the issue of cooperation between intersections in urban traffic networks is very important because the action of each intersection affects the traffic volume of the others. Therefore, cooperation makes the vehicles cross the intersections more smoothly and faster. The issue of cooperation between traffic signals has recently been addressed using the multi-agent Reinforcement Learning (MARL) technique [39]. MARL can discover the

optimal policy, which in turn enables the vehicles to leave the multi-intersection in the shortest time [40]. In ATSC, multiple agents cooperate to optimize global network traffic targets. Consider a multi-agent network $G(\mathcal{V}, \mathcal{E})$, where each agent $i \epsilon \mathcal{V}$ performs a discrete action that communicates to a neighbor agent via the edge $ij \epsilon \mathcal{E}$, and shares the global reward $r(s, u)$. Then, the joint action space for this network is $\mathcal{U} = \mathrm{x}_{i\epsilon\mathcal{V}}\mathcal{U}_i$.

A Reinforcement Learning agent can be added to a network and distributedly compute actor-critic on top of each node. Each local agent can work independently to perform the optimal action and cooperate with other agents. Through network connections, neighbors' information can be shared in MARL. Information that is more than one hop distant can also be propagated throughout the network by exchanging messages through the hop connections to obtain an approximate global optimization [1]. Moreover, by increasing the observability and lowering the learning difficulty of each local agent, the fingerprint method has been used to stabilize the learning process. In this way, we incorporate the observations and fingerprints of the neighboring agents in the state so that each local agent is better informed about the regional traffic distribution and the cooperation strategy of the neighboring agent. In fact, in the fingerprint method, the recent real-time policy of neighbors is given to each local agent rather than the long-term behavior of neighbors. This is based on two facts in ATSC: (1) Traffic state in short windows changes slowly; therefore, the current step policy is quite similar to the last step. (2) According to the given current state and policy, the dynamics of the traffic state are Markovian.

4. The Proposed Method

Shiraz is one of Iran's most crowded metropolitan cities, with 63 junctions. One of the cons of the city is that there are still fixed-time traffic light systems that are implemented with the SCATS system and work according to a predefined schedule that causes increased travel time, fuel consumption, and air pollution, in addition to which, heavy traffic behind red lights causes psychological damage to the driver. Therefore, in order to intelligently control the intersections and examine this issue more closely, six real-world intersections were selected to be examined in this research.

The area between Imam Ali Bridge and Deh Bozorgi Bridge, according to the data received from the Transportation and Traffic Organization of Shiraz Municipality, consists of four Bridges, including six intersections, which have heavy traffic congestion due to their proximity to offices, parks, historical gardens, and tourist attractions. Thus, it requires efficient traffic control measures such as traffic signal controls. Therefore, this area of Shiraz has been chosen as the study area due to the fact that the traffic output of one intersection affects the traffic volume of another intersection.

In order to implement real-world intersections, an agent-based traffic simulator was used. Therefore, in this paper, the Simulation of Urban MObility (SUMO) was used for agent-based traffic simulation, which is an open-source, highly portable, microscopic, and ongoing multi-modal traffic simulation package that is built to handle massive networks.

In this section, different features of the traffic simulation are explained. Moreover, six real-world intersections of Shiraz City, along with the Reinforcement Learning control method that was applied for multi-agent Reinforcement Learning, are described. The goal is to design challenging and real-world traffic environments.

4.1. Environment, Agents and Traffic Demands

The environment is the traffic network, consisting of streets, vehicles, intersections, and agents interacting with each other. Agents are also considered in this article as signalized intersections or traffic signals.

The study area of Shiraz is shown in Figure 1, and includes four bridges, two of which have north- and south-signalized intersections, as shown in Figure 2.

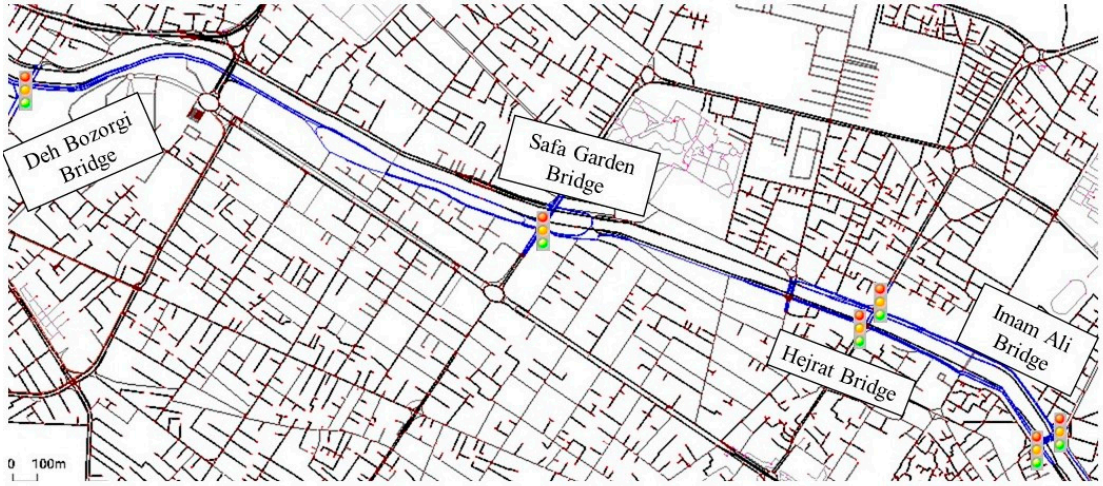

Figure 1. Shiraz traffic network.

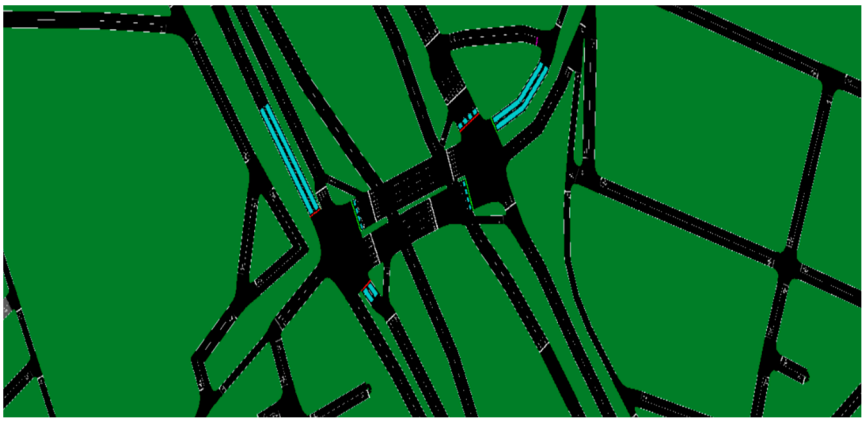

Figure 2. North and south intersections.

The four Bridges are as follows:

- Imam Ali Bridge: It has two intersections, north and south, both of them have two phases, and the duration of the green phase is 27 s in the existing static configuration. At the northern intersection, the first phase is N–S and S–N, which means that vehicles are allowed to pass from north to south and south to north. The second phase of the northern intersection is W–E. At the southern intersection, the first phase is N–S, the second phase is E–W and W–E.
- Hejrat Bridge: It also has two intersections, and the duration of the green phase is 42 s for both of them. At the northern intersection, the first phase is N–S and S–E, and the second phase is W–E. At the southern intersection, the first phase is N–S, and the second phase is E–W.
- Safa Garden Bridge: It has one intersection with two phases; the first phase is N–S, which, due to the congestion on this side, has a longer duration for the green phase of 70 s. The second phase is W–E, which lasts 24 s.
- Deh Bozorgi Bridge: It has a three-phase intersection; the first phase is W–E for 30 s, the second phase is east to west for 29 s, and the third phase is N-S which lasts for 22 s.

We should mention that the yellow phase duration is considered 3 s for all of the intersections in existing static configuration. In addition, the phases are shown in Figure 3.

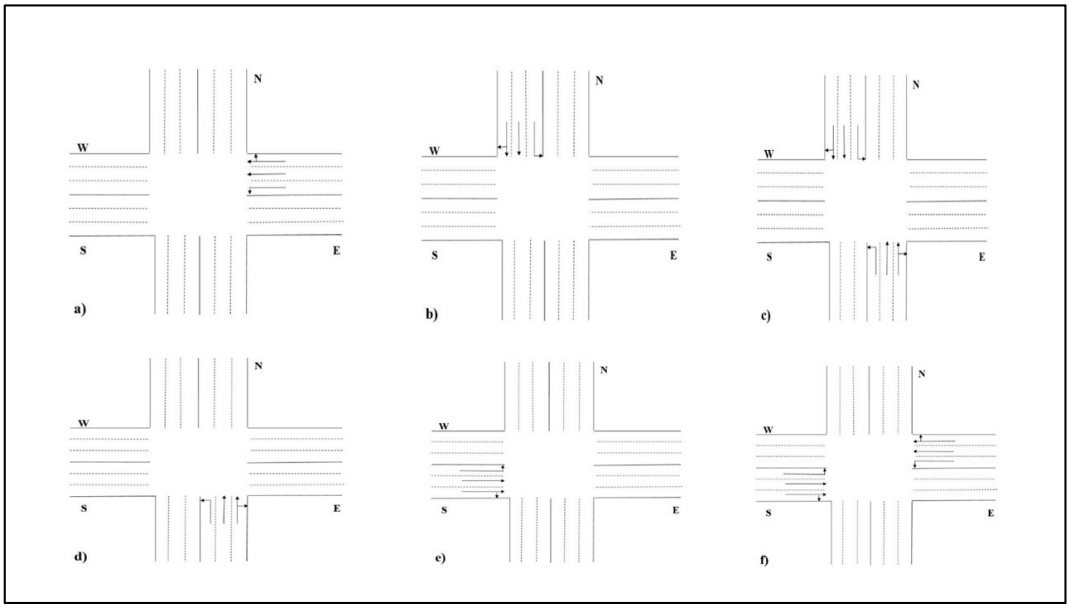

Figure 3. Phases implemented in Shiraz City: (**a**) east to west, (**b**) north to south, (**c**) north to south, (**d**) south to north, (**e**) west to east. (**f**) west to east.

To take into account all of the challenges of the real world, four groups of time-varying traffic flows were simulated. For all four groups, traffic flows that entered the intersections from arterial streets were also considered.

- The first group had a traffic flow that covered the entire route. This stream entered the Imam Ali Bridge and exited the Deh Bozorgi Bridge through Hejrat and Safa Garden Bridges.
- The second group of flows also covered the entire route, but unlike the first group, it entered from the Deh Bozorgi Bridge and passed through the Safa Garden and Hejrat Bridge, and exited the Imam Ali Bridge.
- The third flow group entered from the north of the Deh Bozorgi Bridge and exited from its south, and vice versa. The same flow was defined for Safa Garden Bridge.
- Finally, the fourth group included the flows from the north of the northern Hejrat intersection to the south of the southern intersection of Hejrat Bridge and vice versa, from the south to the north of the southern intersection to the northern intersection of the Hejrat Bridge. The same flow was defined for Imam Ali Bridge.

We defined these flow rates according to information from the Transportation and Traffic Organization of Shiraz Municipality at different times of the day.

4.2. IoT Agents as MA2C for Traffic Signal Control

In this paper, we employed RL, a data-driven method for adaptive traffic signal control in intricate urban traffic networks, and because our simulation scenario included four bridges with six interconnected intersections of Shiraz City, we could not use the centralized RL. Therefore, the multi-agent RL (MARL) was used, which can overcome the scalability issue. This means that more intersections could be controlled. Distributed MARL was installed in the traffic light system in such a way that an RL agent was located at each intersection to manage the local traffic lights for vehicles in all directions. Surveillance cameras, which were IoT sensors, were also placed on each side to capture the queue lengths of the vehicles. Additionally, the agent gathered local traffic data that was tracked by cameras and recorded the information in a local IoT database. As an IoT technique, agents also gathered information from neighbors by exchanging information across network connections. Neighbor data were also kept in the same database, as seen in Figure 4. Based on the data in the database, the actor-critic algorithm, which is an RL type, selects

the optimal control action from a list of predefined actions, and the IoT actuator, such as a traffic light, applies the selected action to the environment. Moreover, in order to better communicate and coordinate between intersections, the combination of MARL and A2C (MA2C) was used in this paper, which makes each intersection not only aware of its own policy but also aware of the policy of other intersections. Therefore, the traffic impact of neighboring intersections can be controlled at the desired intersection. Finally, in order to stabilize the learning procedure by improving the observability of each local agent, the fingerprint method of neighboring agents was incorporated. This means that observations and fingerprints of neighboring agents were included in the local agent state. In the fingerprint method, we incorporated the most recent neighborhood policies π_{t-1}, $\mathcal{N}_i = [\pi_{t-1,\,j}]_{j\in\mathcal{N}_i}$ in the DNN inputs, where $\mathcal{N}_i$ is the neighborhood of agent i. The local policy is calculated as:

$$\pi_{t,\,i} = \pi_{\theta_i^-}(\cdot\,|s_{t,\,\mathcal{V}_i},\ \pi_{t-1},\ \mathcal{N}_i) \tag{7}$$

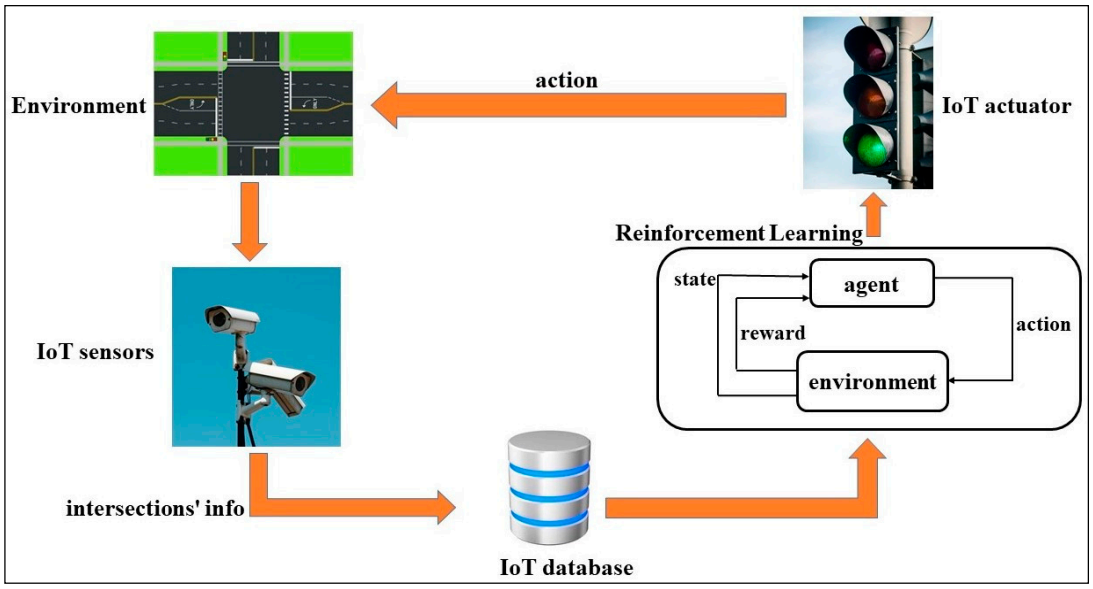

Figure 4. IoT and MARL approach for ATSC.

Therefore, each local intersection contains more information on neighbors' policies in addition to the regional traffic distribution and cooperation strategy.

The following is a definition of state, action, and reward based on [22].

❖ **State Definition**

We define each state as follows:

$$s_{t,\,i} = \{\text{wait}_t[l],\ \text{wave}_t[l]\}_{ji\in\varepsilon,\ l\in L_{ji}}, \tag{8}$$

where l is the incoming lanes of the intersection i. wait[s] is the cumulative delay of the first vehicle, and the wave[veh] measures the total number of vehicles entering the intersection lanes. Both wait and wave are measured as shown in Figure 2 by using induction-loop detectors (ILD) marked in blue. *LaneAreaDetector* in SUMO is also used to obtain this information.

❖ **Action Definition**

In this paper, we define the action for an intersection of all possible phase combinations of traffic lights. This definition allows the agent to control the traffic signal more flexibly. In other words, a set of all existing static phases of Shiraz is defined for each intersection so that the RL agent chooses one of them that lasts for Δt at each step; in which Δt is the interaction period between each agent and the traffic environment.

❖ **Reward Definition**

The reward should be measurable and evaluable, directly dependent on the state and indicating to the agent whether the chosen action is good or bad. In this paper, the queue length at each incoming lane and the average waiting time of drivers are considered as a reward and measured at time $t + \Delta t$.

$$r_{t,\,i} = \sum_{ji \in \varepsilon,\; l \in L_{ji}} (queue_{t+\Delta t}[l] + a \cdot wait_{t+\Delta t}[l]), \tag{9}$$

where a[veh/s] is a tradeoff factor. This reward definition emphasizes traffic congestion and trip delay.

4.2.1. DNN Structure

The traffic flows are complicated spatial–temporal data, so MDP may become non-stationary if the agent only knows the current state. One direct strategy is to include all historical states as A2C input. However, this dramatically increases the dimension of the state and may reduce A2C's attention to recent traffic conditions. Fortunately, long–short term memory (LSTM) is a potential DNN layer that keeps hidden states to memorize brief history. Therefore, we used LSTM as the last hidden layer to extract representations from various types of states. We defined the states for each input line as the input of neural networks, which include the number of input vehicles at the intersection and their waiting time within 50 m of the intersection. In addition to the wave and wait as state, we also included a neighbor policies node as input. Then we processed these values with a fully connected layer. Finally, we concatenated the output in an array and then normalized them. In order to select the best action from the available actions, we used the Softmax as an activation function, and for the critic we also used the linear relation as an activation function to return the reward [22]. Figure 5 shows the DNN architecture in actor-network and critic-network.

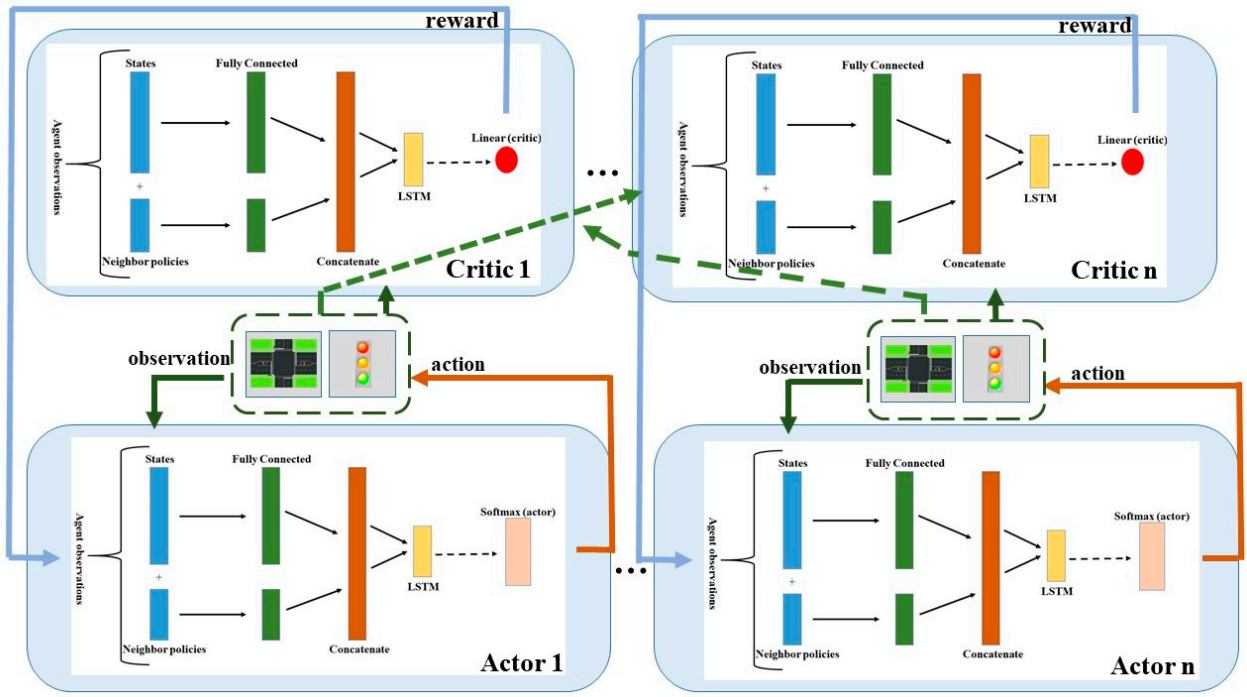

Figure 5. The DNN architecture and training.

DNN Training

In our method, agents learn their policy since each agent is distributed within the six intersections. Therefore, as shown in Figure 5, each agent has its actor-network and

critic-network. We train the actor and the critic of the DNN separately. The input of each actor-network is the local agent's observations or state (Equation (8)). Based on the model that has been trained so far, it performs an action which has predefined phases that are mentioned in sub-Section 4.1. These phases have been obtained from the Shiraz transportation organization and municipality. Based on the selected action, each agent obtains a reward from the environment (Equation (9)), and based on it, the critic-network examines whether the selected action is appropriate for the existing conditions or not, and then the weights are updated. During the training process, the input of each critic-network includes not only the global states of all intersections but also the global actions of all agents. After that, each agent receives a reward, then we enter the new state, and these steps are repeated again. The model is trained at any given time based on the data given to it. Moreover, due to intersection cooperation, neighboring information can be shared via network connections of IoT devices. By exchanging messages through the hop connections, information that is more than one hop away can also be propagated throughout the network to achieve an approximate global optimization. Additionally, by increasing the observability and lowering the learning difficulty of each local agent, the fingerprint method has been utilized to stabilize the learning process. In this way, we include the latest policies or the last actions of the neighboring agents in the state so that each local agent is better informed about the regional traffic distribution and the cooperation technique of the neighboring agent.

4.2.2. Normalization

Normalization is a crucial factor in DNN training. A greedy policy is applied for each wave and wait state to gather statistics relevant to a certain traffic environment and use them to produce an accurate normalization. To avoid the gradient explosion, all normalized states are clipped to [0, 2]. Similarly, to stabilize the mini-batch updating, we normalized the reward and clipped it to $[-2, 2]$ Also, the wave and wait normalization factors are 5 veh and 100 s, respectively [22].

5. Numerical Experiments and Evaluation Results

As mentioned in Section 4, we executed our traffic signal control method in the SUMO [23] simulation, which can model microscopic traffic conditions. We also used Traffic Control Interface (TraCI) (https://sumo.dlr.de/docs/TraCI.html (accessed on 1 January 2022)) as an API, which gives online access from Python to traffic simulation to retrieve simulated objects' values and manipulate their behavior. We train the MA2C algorithm over 1M steps, which is around 1400 episodes. Then we evaluate the obtained model over 10 episodes. Additionally, 10 distinct seeds are used to create various training and evaluation episodes. For MDP, we set $\gamma = 0.99$. Four time-varying traffic flow groups are designed as unit flows of 325 veh/hr. This traffic flow strongly matches into the real world. The general configurations of the simulation are shown in Tables 1–3.

Table 1. The parameter values of the proposed model.

Model_Config	Values
Gamma	0.99
Batch_size	40
Reward_norm	1.0
Reward_clip	2.0

Table 2. The train configuration.

Train_Config	Values
Total_step	1×10^6
Test_interval	2×10^4
Log_interval	1×10^4

Table 3. The environment configuration.

Env_Config	Values
Clip_wave	2.0
Clip_wait	2.0
Agent	MA2C
Episode_length_sec	1400
Norm_wave	5.0
Norm_wait	100.0
Flow_rate	325
Yellow_interval_sec	3

We evaluated MA2C-based ATSC in two traffic environments: two synthetic traffic grids for evaluating the results with synthetic data and a real-world six-intersection traffic network extracted from Shiraz City for evaluation with real data.

5.1. Synthetic Traffic Grid

As illustrated in Figure 6, two synthetic intersections are formed by three lanes with a speed limit of 8.89 m/s, and vehicles with 5 m length are defined for it. The action space for two intersections contains four possible phases: E–W straight phase, E–W left-turn phase, and N–S straight phase, N–S left-turn phase. Moreover, each vehicle's route is generated randomly during run-time.

Figure 6. A traffic grid with two synthetic intersections.

5.1.1. Training Results

Figure 7 plots the training curve.

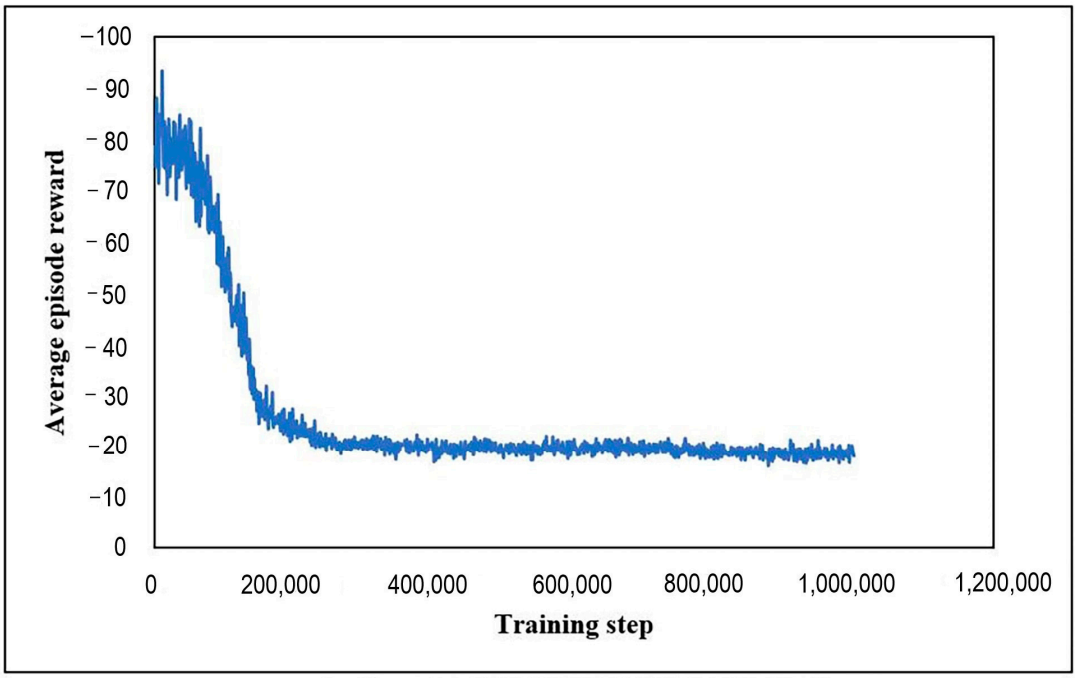

Figure 7. MA2C training curve for synthetic traffic grid.

As Reinforcement Learning learns from accumulated experience and eventually reaches the local optimum, the learning curve initially rises and then converges. Therefore, this algorithm has achieved a good result.

5.2. Evaluation Results

The network's average queue length for each simulation step is shown in Figure 8.

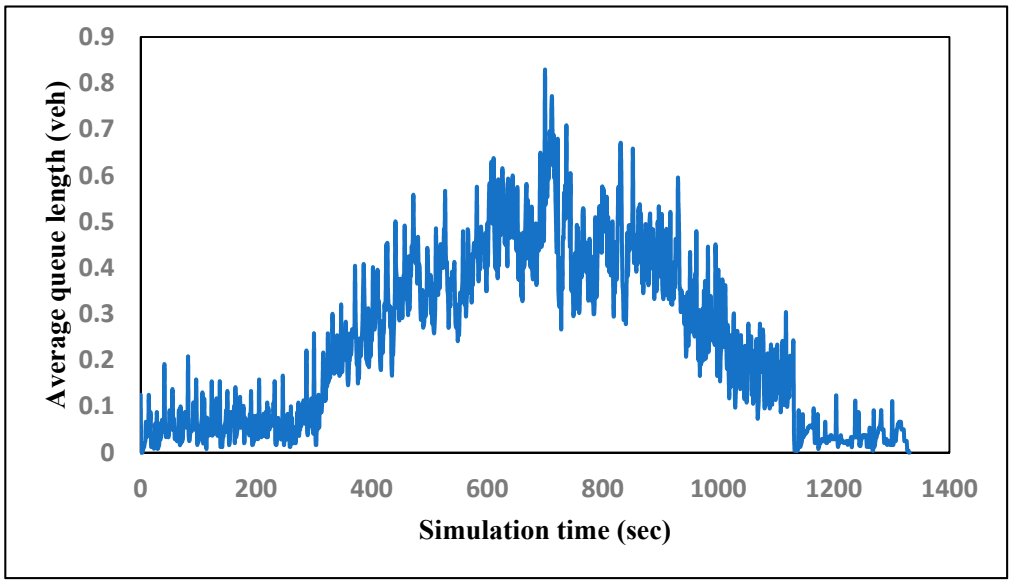

Figure 8. Average queue length in synthetic traffic grid.

This figure represents the total number of vehicles at all junctions at red lights. First, because the number of vehicles is small, the queue length is also short, then as the number

of vehicles in the middle of the simulation time increases, the queue length also increases and again decreases with the decreasing number of vehicles.

5.3. Shiraz Traffic Network with Real Data

We exported the map of Imam Ali Bridge to Deh Bozorgi Bridge in Shiraz City from Open Street Map (OSM) as shown in Figure 9. The map was converted into SUMO-compatible topology by the netconvert tool as shown in Figure 1.

Figure 9. A map of Imam Ali Bridge to Deh Bozorgi Bridge downloaded from OSM.

After conversion, we applied the algorithm we used to each traffic light in Figure 1. Along with the real-world map, we also used real-world traffic data, which are the implemented phases in Shiraz that are obtained from the transportation and municipality traffic organization of Shiraz City. In totally, there were six signalized intersections: five were two-phase, and the last one had three phases. In addition, as stated in Section 4, four time-varying traffic flow groups were designed as unit flows of 325 veh/hr to simulate the peak-hour traffic and to take into account real-world challenges and evaluate the robustness and optimality of the algorithm.

5.3.1. Training Results

Figure 10 plots the training curves of the MA2C algorithm. It converges to reasonable policy and has a stable convergence.

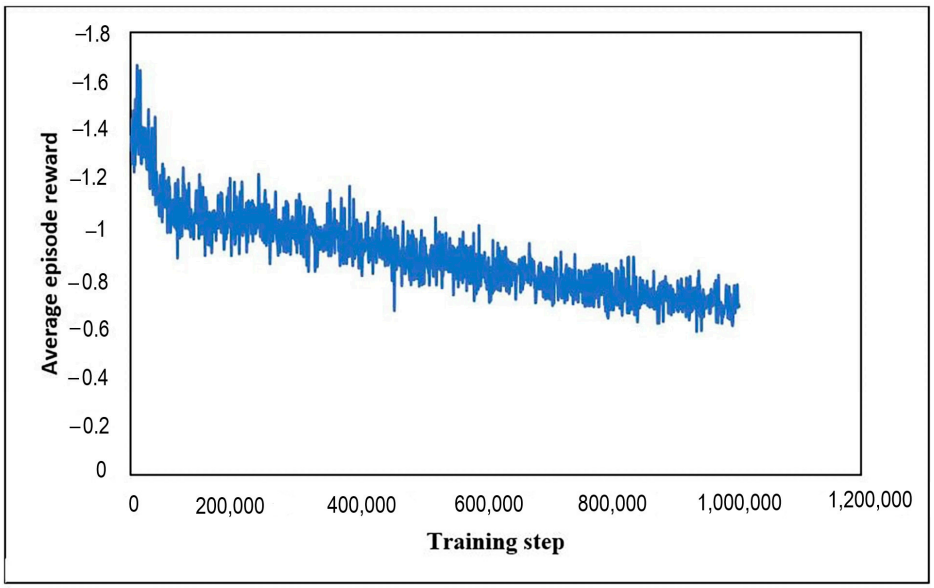

Figure 10. MA2C training curve for Shiraz traffic network.

5.3.2. Evaluation Results

Figures 11 and 12 represent the average queue length of vehicles and their waiting time (average intersection delay) over the simulation time, respectively, in which our proposed system was compared with the traditional traffic control system of Shiraz City using the fixed-time traffic signal control system.

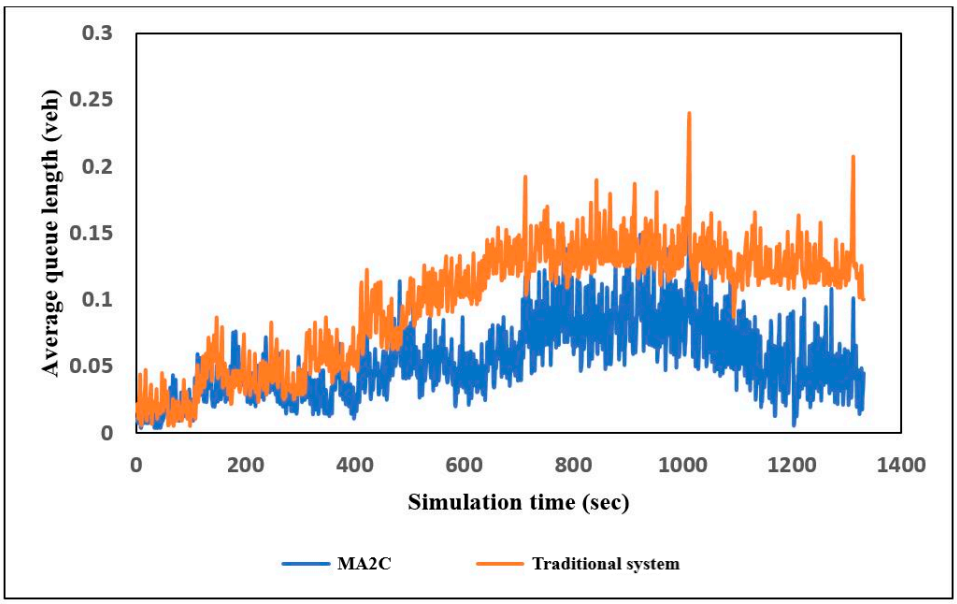

Figure 11. Average queue length in Shiraz traffic network.

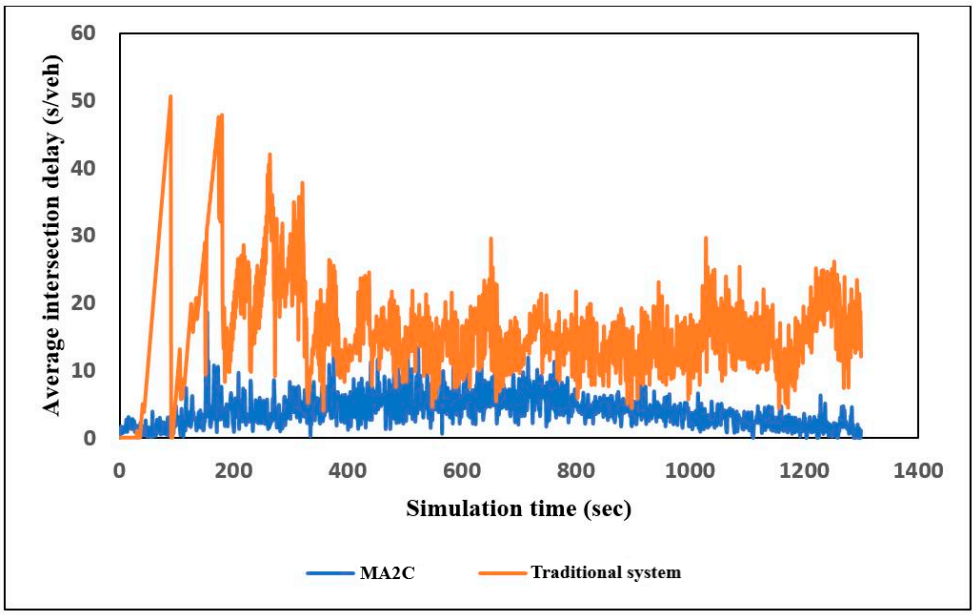

Figure 12. Average intersection delay in Shiraz traffic network.

As shown in Figure 11, the MA2C algorithm performed better than Shiraz City's traditional traffic control system, which uses fixed-time scheduling and does not consider environmental traffic conditions. The main reason for the efficiency of this system is that by taking into account the traffic volume on each side of the intersection using IoT technologies, it switches the traffic light into green for that side and into red if the traffic on the opposite road is less congested, or free of vehicles. Furthermore, the robustness of our system appears during peak hours because it works according to traffic conditions. Therefore, as

seen in the figure, the MA2C algorithm can manage the queue length of vehicles more effectively.

As the results show in Figure 12, our system could reduce and then maintain intersection delays by coordinating and distributing traffic homogeneously among neighboring intersections, specifically when the local traffic flow is maximized greedily in the middle of the simulation time. Therefore, the MA2C algorithm can significantly reduce vehicle waiting time compared to the fixed-time traffic signal control system of Shiraz City.

6. Conclusions

In this paper, we proposed a method using the MARL algorithm to reduce the traffic at six signalized junctions of Shiraz City by changing their phases in real-time. Also, we utilized real-world traffic data received from the transportation and municipality traffic organization of Shiraz City. The proposed method was then applied to two scenarios: (1) two fictitious intersections and (2) a real-world map of Shiraz City received from OSM. In fact, we compared our proposed algorithm with the traditional system of Shiraz that uses fixed-time scheduling for traffic signal control. In order to solve the challenges of cooperation between multiple intersections, the fingerprint method was used to improve their observability. Results showed that using the MARL approach with data from IoT sensors would decrease the average queue length and waiting time at the intersection compared to the fixed-time scheduling implemented in Shiraz. Moreover, the importance of this method was more pronounced with the higher number of vehicles during peak hours. For future work, more intersections should be considered for deployment of the traffic networks to implement the proposed system in the real-world Shiraz City. Moreover, considering the impact of pedestrians on the traffic signal control system would more efficiently improve traffic management.

Author Contributions: Project administration, S.D.; Writing & editing, S.D.; Data curation, M.Z.; Supervision, R.J.; review, A.F. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: My manuscript has no associated data.

Conflicts of Interest: All Authors declare that they have no conflict of interest.

References

1. Liu, Y.; Liu, L.; Chen, W.P. Intelligent traffic light control using distributed multi-agent Q learning. In Proceedings of the IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC, Yokohama, Japan, 16–19 October 2017. [CrossRef]
2. Lin, Y.; Jia, H.; Yang, Y.; Tian, G.; Tao, F.; Ling, L. An improved artificial bee colony for facility location allocation problem of end-of-life vehicles recovery network. *J. Clean. Prod.* **2018**, *205*, 134–144. [CrossRef]
3. Zhang, C.; Tian, G.; Fathollahi-Fard, A.M.; Wang, W.; Wu, P.; Li, Z. Interval-valued intuitionistic uncertain linguistic cloud petri net and its application to risk assessment for subway fire accident. *IEEE Trans. Autom. Sci. Eng.* **2020**, *19*, 163–177. [CrossRef]
4. U.S. Department of Transportation, Smart City Challenge: Lessons for Building Cities of the Future. Available online: https://ops.fhwa.dot.gov/publications/fhwahop08024/index.htm#toc (accessed on 3 February 2021).
5. Gao, J.; Shen, Y.; Liu, J.; Ito, M.; Shiratori, N. Adaptive traffic signal control: Deep reinforcement learning algorithm with experience replay and target network. *arXiv* **2017**, arXiv:1705.02755.
6. Hunt, P.B.; Robertson, D.I.; Bretherton, R.D.; Royle, M.C. The SCOOT on-line traffic signal optimisation technique. *Traffic Eng. Control.* **1982**, *23*, 1982.
7. Luk, J.Y.K. Two traffic-responsive area traffic control methods: SCAT and SCOOT. *Traffic Eng. Control.* **1984**, *25*, 14.
8. Kao, Y.-C.; Wu, C.-W. A self-organizing map-based adaptive traffic light control system with reinforcement learning. In Proceedings of the 2018 52nd Asilomar Conference on Signals, Systems, and Computers. Pacific Grove, CA, USA, 28–31 October 2018; pp. 2060–2064.
9. Gartner, N.H. Demand-Responsive Decentralized Urban Traffic Control. Part I: Single-Intersection Policies. 1982. Available online: https://trid.trb.org/view/1410964 (accessed on 3 February 2021).
10. Sen, S.; Head, K.L. Controlled optimization of phases at an intersection. *Transp. Sci.* **1997**, *31*, 5–17. [CrossRef]
11. Henry, J.-J.; Farges, J.L.; Tuffal, J. The PRODYN real time traffic algorithm. In *Control in Transportation Systems*; Elsevier: Amsterdam, The Netherlands, 1984; pp. 305–310.

12. Genders, W.; Razavi, S. Asynchronous n-step Q-learning adaptive traffic signal control. *J. Intell. Transp. Syst.* **2019**, *23*, 319–331. [CrossRef]
13. Fehon, K.; Peters, J. Adaptive Traffic Signals, Comparison and Case Studies. 2010. Available online: https://www.semanticscholar.org/paper/Adaptive-Traffic-Signals-%2C-Comparison-and-Case-Fehon-Peters/3a0da73ec54249b3366158663c8b4c834e6646c1 (accessed on 3 February 2010).
14. Dubey, A.; Lakhani, M.; Dave, S.; Patoliya, J.J. Internet of Things based adaptive traffic management system as a part of Intelligent Transportation System (ITS). In Proceedings of the 2017 International Conference on Soft Computing and its Engineering Applications (icSoftComp), Changa, India, 1–2 December 2017; pp. 1–6.
15. Bingham, E. Reinforcement learning in neurofuzzy traffic signal control. *Eur. J. Oper. Res.* **2001**, *131*, 232–241. [CrossRef]
16. Darmoul, S.; Elkosantini, S.; Louati, A.; Said, L.B. Multi-agent immune networks to control interrupted flow at signalized intersections. *Transp. Res. Part C Emerg. Technol.* **2017**, *82*, 290–313. [CrossRef]
17. Srinivasan, D.; Choy, M.C.; Cheu, R.L. Neural networks for real-time traffic signal control. *IEEE Trans. Intell. Transp. Syst.* **2006**, *7*, 261–272. [CrossRef]
18. Sánchez-Medina, J.J.; Galán-Moreno, M.J.; Rubio-Royo, E. Traffic signal optimization in 'La Almozara' district in Saragossa under congestion conditions, using genetic algorithms, traffic microsimulation, and cluster computing. *IEEE Trans. Intell. Transp. Syst.* **2009**, *11*, 132–141. [CrossRef]
19. Bazzan, A.L.C. Opportunities for multiagent systems and multiagent reinforcement learning in traffic control. *Auton. Agents Multi-Agent Syst.* **2009**, *18*, 342–375. [CrossRef]
20. Qu, Z.; Pan, Z.; Chen, Y.; Wang, X.; Li, H. A distributed control method for urban networks using multi-agent reinforcement learning based on regional mixed strategy Nash-equilibrium. *IEEE Access* **2020**, *8*, 19750–19766. [CrossRef]
21. Wang, M.; Wu, L.; Li, J.; He, L. Traffic Signal Control With Reinforcement Learning Based on Region-Aware Cooperative Strategy. *IEEE Trans. Intell. Transp. Syst.* **2021**, *23*, 3774–3785. [CrossRef]
22. Chu, T.; Wang, J.; Codecà, L.; Li, Z. Multi-agent deep reinforcement learning for large-scale traffic signal control. *IEEE Trans. Intell. Transp. Syst.* **2019**, *21*, 1086–1095. [CrossRef]
23. Krajzewicz, D.; Erdmann, J.; Behrisch, M.; Bieker, L. Recent development and applications of SUMO-Simulation of Urban MObility. *Int. J. Adv. Syst. Meas.* **2012**, *5*, 128–138.
24. Sims, A.G.; Dobinson, K.W. The Sydney coordinated adaptive traffic (SCAT) system philosophy and benefits. *IEEE Trans. Veh. Technol.* **1980**, *29*, 130–137. [CrossRef]
25. Hosur, J.; Rashmi, R.; Dakshayini, M. Smart Traffic light control in the junction using Raspberry PI. In Proceedings of the 2019 3rd International Conference on Computing Methodologies and Communication (ICCMC), Erode, India, 27–29 March 2019; pp. 153–156.
26. Liang, X.; Du, X.; Wang, G.; Han, Z. A deep reinforcement learning network for traffic light cycle control. *IEEE Trans. Veh. Technol.* **2019**, *68*, 1243–1253. [CrossRef]
27. Wang, T.; Cao, J.; Hussain, A. Adaptive Traffic Signal Control for large-scale scenario with Cooperative Group-based Multi-agent reinforcement learning. *Transp. Res. Part C Emerg. Technol.* **2021**, *125*, 103046. [CrossRef]
28. Lillicrap, T.P.; Hunt, J.J.; Pritzel, A.; Heess, N.; Erez, T.; Tassa, Y.; Silver, D.; Wierstra, D. Continuous control with deep reinforcement learning. *arXiv* **2015**, arXiv:1509.02971.
29. Aslani, M.; Mesgari, M.S.; Wiering, M. Adaptive traffic signal control with actor-critic methods in a real-world traffic network with different traffic disruption events. *Transp. Res. Part C Emerg. Technol.* **2017**, *85*, 732–752. [CrossRef]
30. Ge, H.; Gao, D.; Sun, L.; Hou, Y.; Yu, C.; Wang, Y.; Tan, G. Multi-agent transfer reinforcement learning with multi-view encoder for adaptive traffic signal control. *IEEE Trans. Intell. Transp. Syst.* **2021**, *23*, 12572–12587. [CrossRef]
31. Haydari, A.; Yilmaz, Y. Deep reinforcement learning for intelligent transportation systems: A survey. *IEEE Trans. Intell. Transp. Syst.* **2020**, *23*, 11–32. [CrossRef]
32. Garg, D.; Chli, M.; Vogiatzis, G. Deep reinforcement learning for autonomous traffic light control. In Proceedings of the 2018 3rd IEEE International Conference on Intelligent Transportation Engineering (ICITE), Singapore, 3–5 September 2018; pp. 214–218.
33. Bellman, R. A Markovian decision process. *J. Math. Mech.* **1957**, *6*, 679–684. [CrossRef]
34. Sutton, R.S.; Barto, A.G. *Reinforcement Learning: An Introduction*; MIT Press: Cambridge, UK, 2018.
35. Wu, T.; Zhou, P.; Liu, K.; Yuan, Y.; Wang, X.; Huang, H.; Wu, D.O. Multi-agent deep reinforcement learning for urban traffic light control in vehicular networks. *IEEE Trans. Veh. Technol.* **2020**, *69*, 8243–8256. [CrossRef]
36. Williams, R.J. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Mach. Learn.* **1992**, *8*, 229–256. [CrossRef]
37. Konda, V.R.; Tsitsiklis, J.N. Actor-critic algorithms. *NIPS* **2000**, *12*, 7.
38. Mnih, V.; Badia, A.P.; Mirza, M.; Graves, A.; Lillicrap, T.; Harley, T.; Silver, D.; Kavukcuoglu, K. Asynchronous Methods for Deep Reinforcement Learning Volodymyr. *Int. Conf. Mach. Learn.* **2013**, *48*, 1928–1937.

39. Wei, H.; Xu, N.; Zhang, H.; Zheng, G.; Zang, X.; Chen, C.; Zhang, W.; Zhu, Y.; Xu, K.; Li, Z. Colight: Learning network-level cooperation for traffic signal control. In Proceedings of the CIKM'19: The 28th ACM International Conference on Information and Knowledge Management, Beijing, China, 3–7 November 2019; pp. 1913–1922.
40. Song, J.; Jin, Z.; Zhu, W. Implementing traffic signal optimal control by multiagent reinforcement learning. In Proceedings of the 2011 International Conference on Computer Science and Network Technology, Harbin, China, 24–26 December 2011; Volume 4, pp. 2578–2582.