Multi-Agent Deep Reinforcement Learning-Empowered Channel Allocation in Vehicular Networks

Anitha Saravana Kumar, *Student Member, IEEE*, Lian Zhao, *Senior Member, IEEE*, and Xavier Fernando, *Senior Member, IEEE*

***Abstract*—Channel allocation has a direct and profound impact on the performance of vehicle-to-everything (V2X) networks. Considering the dynamic nature of vehicular environments, it is appealing to devise a blended strategy to perform effective resource sharing. In this paper, we exploit deep learning techniques predict vehicles' mobility patterns. Then we propose an architecture consisting of centralized decision making and distributed channel allocation to maximize the spectrum efficiency of all vehicles involved. To achieve this, we leverage two deep reinforcement learning techniques, namely deep Q-network (DQN) and advantage actor-critic (A2C) techniques. In addition, given the time varying nature of the user mobility, we further incorporate the long short-term memory (LSTM) into DQN and A2C techniques. The combined system tracks user mobility, varying demands and channel conditions and adapt resource allocation dynamically. We verify the performance of the proposed methods through extensive simulations and prove the effectiveness of the proposed LSTM-DQN and LSTM-A2C algorithms using real data obtained from California state transportation department.**

***Index Terms*—Advantage actor critic (A2C), channel allocation, deep Q-learning network (DQN), long short-term memory (LSTM), multi-agent deep reinforcement learning (MADRL), spectrum efficiency, mobility, vehicular networks.**

I. Introduction

VEHICLE to everything (V2X) type vehicular communication is envisioned to culminate in an efficient intelligent transportation systems paradigm. V2X communication is also essential for upcoming autonomous vehicles. The 5G automotive association (5GAA), a consortium formed by telecommunications, technology, and automotive industries, is working on developing end-to-end solutions for cellular V2X technologies [1]. IEEE 802.11p is the dominant standard used for vehicular networking. However, the wireless access in vehicular environments has introduced changes to the conventional IEEE 802.11p to handle periodic and event-driven messages of the vehicular networks. V2X scenario incorporates vehicles to road side unit (RSU) connections known as vehicle to infrastructure (V2I) communication and vehicle to vehicle (V2V) communications [2]. It is crucial that the V2X systems not only need to be intelligent, self-learning, and adaptive but also ultra reliable with low latency.

Vehicular communication requires high bandwidth connections from servers for infotainment applications like video streaming. Whereas, safety related information shared among vehicles requires ultra reliable, low latency communications. Hence, dynamic channel allocation is more challenging to meet different QoS requirements in real-time with limited spectrum. Moreover, the shortage of ubiquitous road-side infrastructure results in coverage issues. Hence, V2X demands effective utilization of RSU resources. Also, RSU is required to support high-priority vehicles by providing pervasive coverage and guarantee better QoS for V2X communications [3].

A. Related Work

Efficient resource allocation for vehicular networks is of interest to many. Significant challenges are, seamless data transfer in a highly dynamic situation that includes low-speed to high-speed vehicles in a shared environment, and considerable variation in data services to support delay-sensitive vehicular communication. Various devices are employed in vehicular networks with different hardware parameters that demand competent interfaces [5]. Also, connected vehicles suffer from an extensive range of impairments which are not limited to shadowing, jamming, multiuser interference, path loss, frequency selective channels, and loss of connectivity with RSU due to low earliest deadline first (EDF) limitation. The EDF is initially proposed for wireless networks that demand delay sensitive quality of service (QoS) in [6]. The vehicle's speed information has a significant impact to the spectrum sharing in handling massive data allocation to channels and quality of experience (QoE) of connected vehicles [7], [8]. A changing network environment causes inaccurate channel state information, which affects the accuracy of resource sharing [9], and frequent handoff of channel allocation [10]. V2X mainly requires efficient channel allocation, uninterrupted data transfer, and smart handoff techniques for highly mobile user equipment. The typical wireless resource allocation approach has long formulated the design objective and constraints as an optimization problem. In this section, we discuss resource allocation using conventional, reinforcement learning, and deep-reinforcement techniques.

Manuscript received October 2, 2021; accepted December 2, 2021. Date of publication December 13, 2021; date of current version February 14, 2022. This work was supported in part by the Natural Sciences and Engineering Research Council of Canada (NSERC) under Grant RGPIN-2020-04678. The review of this article was coordinated by Prof. Bin Lin. *(Corresponding author: Lian Zhao.)*

The authors are with the Department of Electrical, Computer, Biomedical Engineering, Ryerson University, Toronto, ON M5B 2K3, Canada (e-mail: anitha.saravanakumar@ryerson.ca; l5zhao@ryerson.ca; fernando@ryerson.ca).

Digital Object Identifier 10.1109/TVT.2021.3134272

1) *Resource Allocation based on Conventional Techniques:* Until now, most resource allocation techniques utilize instantaneous knowledge of the system hence, long- and short-term variations not appropriately incorporated. For example, a V2V resource allocation method based on cellular V2X technology is proposed in [11] to improve a vehicular network's reliability and delay. The authors have optimized resource allocation by choosing the best receiver for V2V correlation identification and appropriate channel assignment to reduce overall latency. In [12], the authors' objectives are to maximize the system throughput in multi-channel cognitive vehicular networks. This is a non-linear integer programming NP-hard problem. Since, heterogenous vehicles demand different resource allocations, a semi-markov decision process (SMDP) based resource allocation for vehicular cloud computing is presented in [13]. A quick guarded message is essential in vehicular networks to transmit alarming messages. Hence, joint power control and resource allocation for safety-related message communication is proposed in [14]. In [15], an energy-efficient resource scheduler for networked fog centers is proposed for better resource management. A delay-optimal virtualized radio resource scheduling scheme is proposed through stochastic learning based on a software-defined heterogeneous vehicular network framework in [17].
2) *Resource Allocation based on Reinforcement Learning Techniques:* Many intelligent algorithms have been proposed in the open literature to address the problem [18], [19]. One promising artificial intelligent technique is reinforcement learning (RL), in which the agent interacts with the environment and selects an action. Discerning action, the agent reaches a new state and obtains a reward if it attains the goal and get punished otherwise [20]. Recent growth in RL has witnessed its success in numerous fields such as robotics, medical applications, and digital games, in addition to vehicular networks. An adaptive cloud resource allocation is proposed in [21]. It is based on SMDP and RL algorithms in a vehicular cloud system to guarantee QoS and QoE. In [22], RL uses a proximal policy optimization algorithm to learn the changing vehicular networks and perform resource allocation for the local vehicular fog computing environment. RL-based user scheduling and resource allocation are demonstrated with an objective to minimize the age of computing results [23]. In [24], a continuous-time markov decision process problem is formulated for offloading mobile video applications to improve the performance of V2V communications, and it is resolved using the RL algorithm.
3) *Resource Allocation based on Deep Learning Techniques:* RL is extensively used for resource allocation techniques. However, RL faces some difficulties in dealing with ample state space since it is challenging to traverse every state and obtain a value function or model for every station action pair directly and explicitly. Hence, Deep learning sheds light on solving complex optimization problems (at least partially). Deep learning allows multi-layer computation models that learn data representations with multiple levels of abstraction [25], [26]. Each layer computes a linear combination of outputs from the previous layer and then introduces nonlinearity through an activation function to improve its expressive power. Deep learning has seen a recent surge in a wide variety of research areas due to its exceptional performance in many tasks. In [27], the author proposed a deep reinforcement learning (DRL) technique to tackle a complex decision making the problem for collaborative computing approach in vehicular networks. In [28], [29], authors applied deep learning models for wireless resource allocation in vehicular networks to enhance resource allocation problems. Also, the authors noted that policy gradient-based algorithms could learn stochastic policies and tend to be more effective in high dimensional or continuous action space than value-based algorithms.
Furthermore, in [30], the authors proposed a joint optimization DRL-based double deep Q network algorithm, considering mobile edge computing platform to reduce the cost of energy consumption, the latency of computation, and communication. Another deep Q-learning (DQN) model is proposed in [31]. The authors suggested a multi-time scale framework and joint optimal resource allocation for communication, caching, and computation strategy. The author considered vehicle's mobility, the limited storage capacities, the computational resources at the RSUs, and complex service deadline constraint in vehicular networks. In [32], a deep reinforcement learning-based dynamic resource management (DDRM) is used to solve the markov decision process (MDP) problem.
Additionally, a DQN based decentralized resource allocation mechanism for V2V communications is proposed in [33] to tackle the latency constraints in V2V communications. In this study, the DQN model is trained and tested using the data generated from the interactions of an environment simulator and the agents. In [34], the author proposed inter-slice resource management. The author has incorporated long short-term memory LSTM into the advantage actor-critic (A2C) algorithm to achieve better system utility with different moving users.

Multi-agent deep reinforcement learning (MADRL) focuses on multiple agents that learn the environment cooperatively and competitively to produce an action. Here, an autonomous group of agents yield a shared environment to earn rewards and act independently to attain a common goal [35]. Considering the large number of benefits, we explore MADRL to determine the V2X channel allocation based on vehicles' mobility rate in the work [36]. This paper is a significant extension of our earlier work based on the vehicle's priorities and the service mobility factor (SMF). In our work we consider channel allocation in vehicular networks based on their SMF and its priority. SMF is calculated based on their mobility rate and geographic position of vehicles. We use two DRL techniques, DQN and A2C, to allocate the channel at RSU. Our network consists of communication between 5G macro-base stations (MBS) to RSUs and RSUs to vehicles [37]. Also, we apply Mode 4 specified in 3GPP 5G system for V2X communication and tune RSUs to dynamic channel allocation of vehicles.

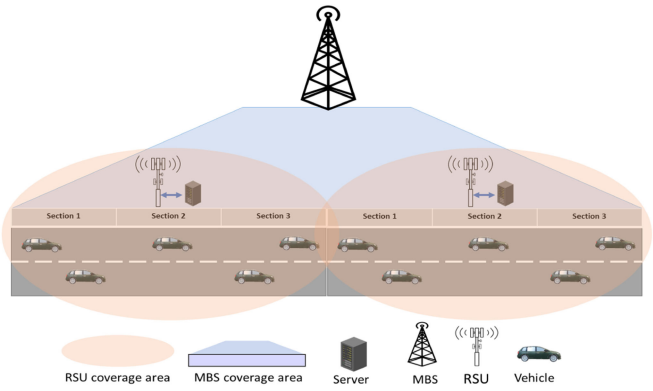

Fig. 1. System model for considered vehicular networks in a freeway.

B. Our Contributions

In this work, we consider channel allocation among vehicles with various mobility rates as a multiagent problem and assume vehicles as agents. We have considered licensed primary users as high priority users, while unlicensed secondary users are categorized as medium priority users and low priority users.

- We propose a mobility-aware priority-based channel allocation using DRL techniques where, the channels are allocated to vehicles based on their SMF and its priority. SMF is calculated based on the mobility rate and geographic position of the vehicles.
- We use LSTM to capture the temporal variation regularity of service requests due to user mobility and append that with DRL techniques. Also, we combine LSTM observations with the powerful learning and decision-making capabilities of DQN and A2C. The research capitalizes on LSTM's knowledge to optimize its bandwidth allocation policy based on a comprehensive understanding of the dynamic environment. We calculate the reward based on the user's SMF, transmission cost, and the used bandwidth. We also compare the LSTM-DQN and LSTM-A2C under extensive settings with conventional SMDP.

The rest of the paper is organized as follows; in Section II, we discuss our System model. In Section III, we describe our projected SMDP model. In Section IV, we present proposed deep RL based mobility aware channel allocation. Section V presents simulation results, and Section VI concludes the paper.

II. System Model

A. Network Structure

We consider a cognitive-enabled vehicular communication network with N_R RSUs along the road as shown in Fig. 1. All N_R RSUs and their users under an MBS can communicate among themselves. In our work, high priority users (H-users) have priority to communicate in the licensed spectrum. Ambulance, fire trucks, on board units, and police vehicles are H-users. Medium priority users (M-users) and low priority users (L-users) are unlicensed users. Examples for M-users are school buses, trucks, and heavy-duty vehicles. L-users are cars and vans. In our model, H-users, M-users, and L-users share the licensed channels. H-users have the priority to access the spectrum. At the same time, M-users occupy bandwidth in cooperation with H-users. L-users can opportunistically utilize the spectrum bands when those channels are unoccupied by both H-users and M-users. Once a user intends to utilize the approaching RSU, it communicates its mobility rate (m_r) and geographic position to the intended RSU. Thus, the considered RSU based on the given information predicts the section speed and allocates the channel.

TABLE I
System Parameters

User Priority	H	M	L
User Arrival Rate	λ_h	λ_m	λ_l
Average Service Time	$1/\mu_h$	$1/\mu_m$	$1/\mu_l$
Number of users	n_h	n_m	n_l
Services with c channels	n_{hc}	n_{mc}	n_{lc}
Service Request Arrival event	A_h	A_m	A_l
Service Request Completion event	F_{hc}	F_{mc}	F_{lc}
Transfer Vector	T^m, T^l	T^l, T^l_{cmin}	-
Rewards	R_h	R_m	R_l

B. Channel Model

In our work, the H, M, and L users share K number of channels from one RSU. The users arrive with poisson distribution, with the mean rate of λ_h for H-users, λ_m for M-users, and λ_l for L-users. At initialization, the number of channels that can be allocated to each vehicle is c, where $c \in (1, \ldots, C)$, $C \leq K$ is the maximum number of the channels allocated to one service. Each RSU is covering D_R meters as coverage diameter. As the vehicle enters the RSU coverage area, its service rate of the requests is calculated using mobility rate m_r calculated using $\mu = m_r/D_R$. RSU accepts the request based on the following initial criteria. First, RSU agrees with the request based on the availability of channels. Second, it considers vehicles within the RSU range for channel allocation and calculates the vehicle's service deadline established on its mobility rate. Based on channel availability, a user can get a high transmission rate if many channels are assigned for the same service, reducing the cost of occupying the channel as its gains can be completed in a shorter period. The residence time for a vehicle connected with a single RSU is an exponential distribution with a mean time of $1/\mu_D$ [4]. The average service time for one allocated channel use is $1/\mu_h$, $1/\mu_m$, and $1/\mu_l$ for H, M, and L-users respectively. When all channels are occupied and an H-user arrives, we clear an existing L-user and allocate that channel to the H-user. Similarly, if the channel is busy and an M-user enters, then the L-user channel is vacated and handed the channel to the M-user. When an M-user leaves the coverage area of an RSU, then the channel allocation is transferred to the MBS. The same approach is maintained for L and H users as well. System parameters of these three classes are given in Table I. Generally, an SMDP is divided into five parts 1) State Space, 2) Action Space, 3) Transition Probabilities, 4) Reward Model, and 5) Decision Epochs [4]. This section discusses all the parts of SMDP.

C. Mobility Scenario

Section speed is the speed in km/hrs of a vehicle passing a given location on a highway. Section speeds and travel times of motor vehicles may vary because of different physical factors (curvature, sight distance, frequency of intersections, and

roadside development), various traffic factors, and different environmental factors. In this part, we analyze the section speed distribution of a vehicle. A study of the relation between spot speeds and travel times reveals how to consider the speed during channel allocation at RSU. To analyze the speed distributions, we should consider the mean travel time of the vehicle. We also assume the speed of a vehicle fluctuates fairly smoothly with the maximum speed not more than 2 or 3 times the minimum.

D. Problem Formulation

In this paper, we aim to maximize the spectrum efficiency. Channel allocation is done for N vehicles, hence mathematically we can denote $1 \dots N$ sharing the aggregated bandwidth B and having fluctuating demands $m = (m_1 \dots m_N)$. We aim to maximize the long term reward expectation $\mathbb{E}\{R(b,m)\}$, where the notion $\mathbb{E}(\cdot)$ denotes to take the expectation of the argument, given as,

$$\underset{b}{argmax}\,\mathbb{E}[R(B,m)] = \underset{b}{argmax}\,\mathbb{E}\{\alpha \cdot SE(b,m)\} \tag{1}$$

$$\begin{aligned} &s.t.: \boldsymbol{b} = (b_1, \dots, b_N) \\ &b_1 + \cdots + b_N = B; \\ &m = (m_1, \dots, m_N) \\ &m_i \sim mobility\, model, \forall i \in [1, \dots, N]. \end{aligned}$$

The critical challenge to our problem statement is to provide maximum spectrum efficiency to the volatile demand of vehicles without having known a priori due to the mobility model. Hence, LSTM -DQN, and LSTM- A2C are exactly matching solutions to solve the problem.

III. Channel Allocation Based on Semi Markov Decision Process

1) *State Space:* The state-space is a time-domain method that presents a suitable and compact way to model and analyze systems with multiple inputs and outputs. In our proposed state space model, we have five units. The first three units relate the channel allocation. Channels allocated to H-users are denoted by $\boldsymbol{n}_h = \{n_{h1}, n_{h2}, \dots, n_{hC}\}^T$, where, n_{hC} represents number of H-user services allocated with C channels. Similarly, channels allocated to M and L users are denoted by $\boldsymbol{n}_m = \{n_{m1}, n_{m2}, \dots, n_{mC}\}^T$ and $\boldsymbol{n}_l = \{n_{l1}, n_{l2}, \dots, n_{lC}\}^T$ respectively, with $\sum_{c=1}^{C} c(n_{hc} + n_{mc} + n_{lc}) \leq K$ is the given condition. The fourth part is the SMF denoted by η, given as,

$$\eta = \frac{d_v}{m_r}, \tag{2}$$

where, d_v is the distance of the vehicle to the intended RSU and m_r is the mobility rate of the vehicle. η^* is the threshold value considered and it varies depending on the priorities of vehicles. In our model, RSU continously receives the user details about their mobility rate, geographic position and determines the SMF. If, $\eta \leq \eta^*$, it shows the vehicle will move out of the coverage of current RSU sooner. The fifth part is event, given as $\mathbf{e} \in \{A_h, A_m, A_l, F_{hc}, F_{mc}, F_{lc}\}$. A_h, A_m, and A_l are the arrival events of H, M, and L users' service requests respectively. F_{hc}, F_{mc}, and F_{lc} are completion events of H, M, and L user services using c channels respectively. The system space is given as,

$$S = \{s | s = n_h, n_m, n_l, \eta, \mathbf{e}\}. \tag{3}$$

2) *Action Space:* The action space is the set of possible actions and evaluations that we can consider after observing the information. The system should choose an action based on a new request. When an L-user request occurs and satisfies the condition $\eta \leq \eta^*$, c channels are allocated to that user. It is denoted as,

$$a(n_l, n_m, n_h, A_l) = \begin{cases} (l, c), & \eta \leq \eta^*, \\ 0, & \text{otherwise.} \end{cases} \tag{4}$$

Likewise, when an M-user request arrives, the system assign c channels, if $\eta \leq \eta^*$ condition is satisfied. It is denoted as,

$$a(n_l, n_m, n_h, A_m) = \begin{cases} (m, c, \mathbf{T}^l), & \eta \leq \eta^*, \\ (m, c_{min}, \mathbf{T}^l_{cmin}), & \eta > \eta^*, \\ 0, & \text{otherwise,} \end{cases} \tag{5}$$

where, $\mathbf{T}^l = \{T^l_1, T^l_2, .., T^l_{cmin}, .., T^l_c\}^T$ is the transfer vector for L-user. When $\eta \leq \eta^*$, T^l_c is the number of L-user services allocated with c channels that are transferred to MBS to accommodate M-user services in the RSU range. Otherwise, T^l_{cmin} is the least minimum L-user services allocated with c_{min} channels, also $T^l_{cmin} < T^l_c$.
Similarly, when H-user request arrives, the system will accept the request with c channels and is given as,

$$a(n_l, n_m, n_h, A_m) = \begin{cases} (h, C, \mathbf{T}^m, \mathbf{T}^l), & \eta \leq \eta^*, \\ (h, c, \mathbf{T}^m, \mathbf{T}^l), & \eta > \eta^*, \\ 0, & \text{otherwise,} \end{cases} \tag{6}$$

where, $\mathbf{T}^m = \{T^m_1, T^m_2, \dots, T^m_r\}^T$ is the transfer vector for M-user. T^m_r is the number of M-user services allocated with r channels that are transferred to MBS to accommodate H-user service request. When $\eta \leq \eta^*$, maximum number of channels C is allocated to a single H-user service to attain high tranmission rate and also satisfies $C < K$. When $\eta > \eta^*$, the system assign c channels. As all continuing services are completed, we denote the idle action space as $a(n_l, n_m, n_h, A) = -1$, and $F \in (F_{hc}, F_{mc}, F_{lc})$. The summarized action space is given as,

$$A = \begin{cases} (a(n_l, n_m, n_h, A_l), a(n_l, n_m, n_h, A_m), \\ a(n_l, n_m, n_h, A_h), a(n_l, n_m, n_h, F)). \end{cases} \tag{7}$$

3) *Transition Probabilities:* Note, we use the Continuous Time Markov Decision Process. Hence, the time interval between any two state-action pairs (s, a) is an exponentially distributed variable. Assuming time interval $\tau(s, a)$ as the expected value, the $\delta(s, a)$ is the mean occurrence rate of this event. Note, $\delta(s, a)$ is the reciprocal of $\tau(s, a)$. We can calculate transition probability $p_r(s'_i | s, a)$ using

mean rate of events. The mean rate for events is given as follows:

$$\delta(s,a) = \begin{cases} \delta_0(s,a) + (c\mu_l + \mu_d), & e = A_l, a = (l,c), \\ \delta_0(s,a) - \sum_{l=1}^{C} \mathbf{T}^l(l\mu_l + \mu_d) + (c\mu_m + \mu_d), & \\ \quad e = A_m, a = (m,c,\mathbf{T}^l), & \\ \delta_0(s,a) - \sum_{l=1}^{c_{min}} \mathbf{T}^l(l\mu_l + \mu_d) + (c_{min}\mu_m + \mu_d), & \\ \quad e = A_m, a = (m,c_{min},\mathbf{T}^l_{cmin}), & \\ \delta_0(s,a) - \sum_{l=1}^{C} \mathbf{T}^l(l\mu_l + \mu_d) + (C\mu_m + \mu_d) & \\ \sum_{m=1}^{C} \mathbf{T}^m(m\mu_l + \mu_d) + (c\mu_h + \mu_d), & \\ \quad e = A_h, a = (h,C,\mathbf{T}^l,\mathbf{T}^m) & \\ \delta_0(s,a) - \sum_{l=1}^{C} \mathbf{T}^l(l\mu_l + \mu_d) + (c\mu_m + \mu_d) & \\ \sum_{m=1}^{C} \mathbf{T}^m(m\mu_l + \mu_d) + (c\mu_h + \mu_d), & \\ \quad e = A_h, a = (h,c,\mathbf{T}^l,\mathbf{T}^m) & \\ \delta_0(s,a), \qquad otherwise, & \end{cases} \tag{8}$$

where, $\delta_0(s,a)$ can be expanded as below:

$$\delta_0(s,a) = \lambda_h + \lambda_m + \lambda_l + \sum_{c=1}^{C}[n_{lc}(\mu_l + \mu_d) + n_{mc}2(\mu_m + \mu_d) + n_{hc}2(\mu_h + \mu_d)]. \tag{9}$$

For instance, M-users transition probability is given as,

$$p_r(s_i'|s,a) = \begin{cases} \frac{\lambda_l}{\delta(s,a)'}, & s_1' = (n_1 - \mathbf{T}^l, n_m + \mathbf{I}^c, A_l), \\ \frac{\lambda_m}{\delta(s,a)'}, & s_2' = (n_1 - \mathbf{T}^l, n_m + \mathbf{I}^c, A_m), \\ \frac{\lambda_h}{\delta(s,a)'}, & s_3' = (n_1 - \mathbf{T}^l, n_m - \mathbf{T}^m, n_h + \mathbf{I}^c, A_h), \\ \frac{(n_{lc} - \mathbf{T}^m)(c\mu_l + \mu_d)}{\delta(s,a)}, & s_4' = (n_1 - \mathbf{T}^l - \mathbf{I}^n, n_m + \mathbf{I}^c, F_{lc}), \\ \frac{(n_{mc}+1)(c\mu_m + \mu_d)}{\delta(s,a)}, & s_5' = (n_1 - \mathbf{T}^l, n_m, F_{mc}), \\ \frac{(n_{mc}+1)(c_{min}\mu_m + \mu_d)}{\delta(s,a)}, & s_6' = (n_1 - \mathbf{T}^l_{cmin}, n_m, F_{mc}), \\ \frac{(n_{lc} + \mathbf{T}^l)(l\mu_l + \mu_d)}{\delta(s,a)}, & s_7' = (n_1 - \mathbf{T}^l - \mathbf{I}^c, n_m + \mathbf{I}^m, F_{lc}), \\ \frac{(n_{mc})(c\mu_m + \mu_d)}{\delta(s,a)}, & s_8' = (n_1 - \mathbf{T}^l, n_m + \mathbf{I}^m - \mathbf{I}^c, F_{mc}), \\ \frac{(n_{mc})(c_{min}\mu_m + \mu_d)}{\delta(s,a)}, & \\ \qquad s_9' = (n_1 - \mathbf{T}^l_{cmin}, n_m + \mathbf{I}^m - \mathbf{I}^c, F_{mc}), & \end{cases} \tag{10}$$

where, $\mathbf{I}^c, \mathbf{I}^m$, and $\mathbf{I}^n$ are vectors with C elements, in which all the elements are zeroes except the c^{th}, m^{th}, and the n^{th} element being 1, respectively.

4) *Reward Model:* There are two parts in the reward function. The first part is the instant reward $i(s,a)$ given from the user after an action is selected and the second part is the system cost $g(s,a)$. The reward function can be formulated as,

$$r(s,a) = i(s,a) - g(s,a). \tag{11}$$

With an increase in H-user requests, both M-users and L-users are transferred to the MBS. In that case $i(s,a)$ can be given as,

$$i(s,a) = \begin{cases} 0, & a(n_l,n_m,n_h,F) = -1, \\ -R_l, & a(n_l,n_m,n_h,A_l) = 0, \\ R_l - R_c \times c, & a(n_l,n_m,n_h,A_l) \\ \quad = (l,c), \eta \le \eta^*, & \\ R_m - R_c \times c - B_t, a(n_l,n_m,n_h,A_m) & \\ \quad = (m,c,\mathbf{T}^l),\ \eta \le \eta^*, & \\ R_m - R_c \times c_{min} - B_{mint}, & \\ a(n_l,n_m,n_h,A_m) = (m,c_{min},\mathbf{T}^l_{cmin}), & \\ \quad \eta > \eta^*, & \\ R_h - R_C \times C - O_{tmax}, a(n_l,n_m,n_h,A_h) & \\ \quad = (h,C,\mathbf{T}^m,\mathbf{T}^l),\ \eta \le \eta^*, & \\ R_h - R_c \times c - O_t, a(n_l,n_m,n_h,A_h) & \\ \quad = (h,c,\mathbf{T}^m,\mathbf{T}^l),\ \eta > \eta^*, & \end{cases} \tag{12}$$

where, R_l, R_m, and R_h denote reward from L, M, and H users respectively. R_c denotes the transmission cost of lodging one channel. $B_t = \{\sum_{c=1}^{C} T_c E_t + cT_c U_t\}$, $B_{mint} = \{\sum_{c=1}^{C} T_{cmin} E_t + cT_{cmin} U_t\}$, $O_{tmax} = \{\sum_{c=1}^{C} T_c E_t U_t (1+C) + \sum_{r=1}^{C} rT_r E_t U_t\}$, and $O_t = \{\sum_{c=1}^{C} T_c E_t U_t (1+c) + \sum_{r=1}^{C} rT_r E_t U_t\}$ are the overall cost involved in transferring affected L and M user services respectively. At the condition, $\eta \le \eta^*$ for H-users, we allocate maximum channels for single service which results in shorter duration of serive completion. Thus, less cost is incurred for occupying the channels. The system cost $g(s,a)$ can be defined as,

$$g(s,a) = \tau(s,a)o(s,a), \tag{13}$$

where, $\tau(s,a)$ is the time interval and $o(s,a)$ is the cost rate of the system.

5) *Decision Epochs and Goal Achievement:* Expected long term reward always focuses on agent's ability to maximize the long-term reward [16]. It is defined from the agent's goal. In our model, it is given as:

$$h^{\pi}_{s_0} = \lim_{Y \to \infty} \frac{E^{\pi}_{s_0} \sum_{m=0}^{Y} [i(s_m,a_m) - \tau_m o(s_m,a_m)]}{E^{\pi}_{s_0} \sum_{m=0}^{Y} \tau_m}, \tag{14}$$

where, s_0 is the initial state and τ_m is the time difference between any two states of a decision epoch. Since, our local reward for any action depends on initial cost and user satisfaction, we can find the optimal policy to obtain the average reward. The optimal policy that generates a long-term reward is given as,

$$\pi^* \in \underset{a}{argmax}\ h^{\pi}. \tag{15}$$

IV. Multi-Agent Deep Reinforcement Empowered Mobility Aware Channel Allocation

A. Basics of DQN, A2C and LSTM

1) Deep Q-Network (DQN): DRL has attracted much attention in recent years due to its capability to provide a good approximation of the objective value (referred to as Q-value) while dealing with extensive state and action spaces. In deep Q-learning, a DNN parameterized by θ called deep Q-network (DQN), represents the action-value function. In contrast to Q-learning methods that perform well for small-size models

but perform poorly for large-scale models, DRL combines a deep neural network with Q-learning, referred to as DQN, for overcoming this issue. Using DQN, the deep neural network maps from the (partially) observed state to an action, instead of storing a lookup table of Q-values. Furthermore, large-scale models can be represented well by the deep neural network so that the algorithm has the ability to preserve good performance for very large-scale models. The user fully observes network state-action space with soft policies and stores the transition tuple $(s_t, a_t, R_{t+1}, s_{t+1})$ in a replay memory at each time step. By applying Q-learning to the given setting, DQN updates the Q-value at time t and it is given as,

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha[(R_{t+1} + \gamma \max_{a'} Q(s_{t+1}, a') - Q(s_t, a_t)], \tag{16}$$

where, $\gamma \in [0, 1]$ is the discount factor, α is the learning rate set to $0 \leq \alpha \leq 1$, generally is set close to zero. The choice of a_t in state s_t follows some soft policies, usually the ε-greedy, meaning that the action with maximal estimated value is chosen with probability $1 - \varepsilon$ while a random action is selected with probability ε. Over many episodes of the markovian process, the replay memory accumulates experiences. A mini-batch of experience D are uniformly sampled from the memory at each step which contributes in updating θ, hence the name experience replay. The algorithm mainly aims at minimizing the time difference error between the learned value and the current estimate value. To minimize the sum-squared error:

$$\sum_{\mathcal{D}} [R_{t+1} + \gamma \max_{a'} Q(s_{t+1}, a_t), a'; \theta^-) - Q(s_t, a_t; \theta)]^2, \tag{17}$$

where, θ^- is the parameter set of a target Q-network, which is duplicated from the training Q-network parameters set θ periodically and fixed for a couple of updates. Experience replay improves sample efficiency through repeatedly sampling stored experiences and breaks correlation in successive updates, thus also stabilizing learning [31].

2) Advantage Actor-Critic (A2C): In the field of RL, A2C algorithm combines two types of RL algorithms (policy based and value based) together. Value based algorithms learn to select actions based on the predicted value of the input state or action. The A2C model is synchronous; it provides better consistency among agents, suitable for disaggregated deployments. At each time step t, the agent receives a state $s_t \in S$ and selects an action a_t from the set of possible actions $\mathcal{A}$ according to its policy $\pi(a_t|s_t)$. The agent reaches the next state s_{t+1} after interacting with the environment. The total reward at time-step t is $R_t = \sum_{k=0}^{\infty} \gamma^k r_{t+k}$, where γ is the discount factor takes the value between 0 and 1. The goal of the RL agent is to maximize the expected return from each state s_t, which can be estimated by the action-value function $Q^\pi(s_t, a_t)$ and the state-value function $V^\pi(s)$. The state-action value $Q^\pi(s_t, a_t)$ estimates the expected return for selecting action a in state s at time t and it is given as $Q^\pi(s_t, a_t) = \mathbb{E}[r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \cdots | s_t, a_t]$ and it is simplified as $Q^\pi(s_t, a_t) = \mathbb{E}[R_t|s_t, a_t]$. The state value function $V^\pi(s)$ is given as $\mathbb{E}[R_t|s_{t=s}]$ which gives the average expected return from state s.

The advantage actor critic is given as $\mathcal{A}^\pi(s_t, a_t) = Q^\pi(s_t, a_t) - V^\pi(s_t)$ which represents the advantage of performing action a_t at state s_t. Usually the temporal difference (TD) error can be used to estimate $\mathcal{A}^\pi(s_t, a_t)$ and its given as $\mathcal{A}^\pi(s_t, a_t) = \mathbb{E}[R_t|s_t, a_t] - V^\pi(s_t)$, that is expressed as $[r_t + V^\pi(s_{t+1|s_t,a_t} - V^\pi(s_t))] = \delta(s_t)$. The gradient of the actor is $\nabla_\theta \mathcal{J}(\theta)$ and loss function of critic is given as $\mathcal{L} = \delta(s_t)^2$.

3) Long Short-Term Memory Networks (LSTM): LSTM networks are a special category of RNNs that are suitable for learning long-term dependencies [38]. The key part that enhances LSTMs' capability to model long-term dependencies is a component called the memory block. In LSTM, the memory block is a recurrently connected subnet that contains functional modules called the memory cell and gates. The memory cell is in charge of remembering the temporal state of the neural network and the gates formed by multiplicative units are responsible for controlling the pattern of information flow. According to the corresponding practical functionalities, these gates are classified as the input gate, the output gate and the forget gate. The input gate controls how much new information flows into the memory cell, while the forget gate governs how much information of the memory cell still remains in the current memory cell through recurrent connection, and the output gate determines how much information is used to compute the output activation of the memory block and further flows into the rest of the neural network. Through the cooperation between the memory cell and the gates, LSTM is endowed with a powerful ability to predict time series with long-term dependences.

B. DRL Based Channel Allocation

In our proposed DRL structure, we propose both LSTM-DQN and LSTM-A2C models. LSTM captures the temporal variation regularity of service requests due to user mobility and further applies the powerful learning and decision-making capability of the A2C mechanism to optimize its channel allocation policy based on the comprehensive understanding of the dynamic environment. To capture the temporal correlation of service requests, we define the state $s_t = (o_{t-T}, o_{t-T+1}, \ldots, o_{t-1})$ as a series of observation vectors, where each observation vector $\boldsymbol{O_t}$ is the number of arrived vehicles in each section within the t^{th} scheduling period. Then the action $a_t = b_1, \ldots, b_N$ is defined as the channel allocation to each vehicle. We design the reward function as,

$$r_t = r_h(SE) I_{SMF}(s_t, a_t) + r_m(SE) I_{SMF}(s_t, a_t) + r_l(SE) I_{SMF}(s_t, a_t), \tag{18}$$

where, $I_{SMF}(s_t, a_t) = [0, 1]$ is an indicator function based on the service mobility factor, whether the bandwidth allocation is provided by RSU. $I_{SMF}(s_t, a_t) = 1$ indicates the channel allocation, if $I_{SMF}(s_t, a_t) = 0$, otherwise.

C. LSTM-DQN Based Channel Allocation

In our proposed LSTM-DQN based model as in Fig. 2, $s_t = (o_{t-T}, o_{t-T+1}, \ldots, o_{t-1})$ is the state-observation from LSTM network. The agent then takes an action $a_t \in \mathcal{A} = 1, \ldots, K$ and receives a reward r_t. The objective of the agent is to take actions that maximize the total rewards. The future reward is calculated

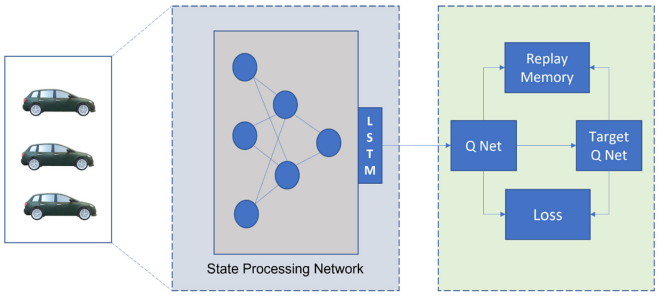

Fig. 2. LSTM-DQN architecture.

Algorithm 1: The LSTM-DQN.

Initialize Parameters:action-value function θ, target action-value fuction $\theta^* = \theta$, replay memory $\boldsymbol{\mathcal{D}_t}$ to capacity N

for i=1 to T do
  Append $s_t = (o_{t-T}, o_{t-T+1}, \ldots, o_{t-1})$
  State processing network output s_t
  Given s_t, select a random action a_t with probability ε
  Otherwise select $a_t = \max_{a'} Q(s_t, a; \theta)$
  Obtain r_t from equation (18) and new state s_{t+1}
  Store (s_t, a_t, r_t, s_{t+1}) into; $\boldsymbol{\mathcal{D}_t}$
  Sample (s_i, a_i, r_i, s_{i+1}) from; $\boldsymbol{\mathcal{D}_t}$
  set target $y = r_i + \max_a Q^*(s_{i+1}, a)$
  update the parameters of θ to make (s'_i, a_i) close to
  $[Q^*(s'_t, a'_t); \boldsymbol{\theta^-} - Q^*(s_t, a_t; \boldsymbol{\theta})]^2$ with network
  parameters θ
  Every C steps rest $\theta^* = \theta$
end for

with $R_t = \sum_{t=0}^{\infty} \gamma^t r_t$, where γ discount factor remains between 0 and 1. To achieve the objective, LSTM-DQN considers the action corresponding to the maximum action-value function as follows,

$$Q^*(s_t, a_t) = \max_{\pi} \mathbb{E}[r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \cdots | s_t, a_t, \pi]. \quad (19)$$

$Q^*(s_t, a_t)$ is the expected value of the future discounted rewards under the policy $\pi = P(a_t|s_t)$. The agent stores the past experiences like $e_t = \langle s_t, a_t, s'_t, R(s_t, a_t)\rangle$ at episode t into a dataset $\boldsymbol{\mathcal{D}_t}$ and selects mini-batch items from the dataset to the Q-value of the neural network $Q^*(s_t, a_t)$. LSTM-DQN approximates the Q-value function with vector $\boldsymbol{\theta}$, hence $Q^*(s_t, a_t)$ approximates to $Q^*(s_t, a_t; \boldsymbol{\theta})$. In order to make $Q^*(s_t, a_t; \boldsymbol{\theta})$ close to the target value $Q^*(s'_t, a'_t; \boldsymbol{\theta^-})$ the loss function can be given as,

$$\mathcal{L}_{dq}(\theta) = \mathbb{E}(s_t, a_t, r, s'_t) \sim \mathbb{U}[Q^*(s'_t, a'_t; \boldsymbol{\theta^-}) - Q^*(s_t, a_t; \boldsymbol{\theta})]^2, \quad (20)$$

where, $\mathcal{L}_{dq}(\theta)$ is the loss function and $\boldsymbol{\theta^-}$ is the parameters of the target network. To minimize the parameter $\boldsymbol{\theta}$ of $\mathcal{L}_{dq}(\theta)$, gradient based approach is used,

$$\boldsymbol{\theta^{i+1}} \leftarrow \boldsymbol{\theta^i} - \alpha \nabla \mathcal{L}_{dq}(\theta^i) = \boldsymbol{\theta^i} - \alpha[Q^*(s'_t, a'_t); \boldsymbol{\theta^-} - Q^*(s_t, a_t; \boldsymbol{\theta})]. \quad (21)$$

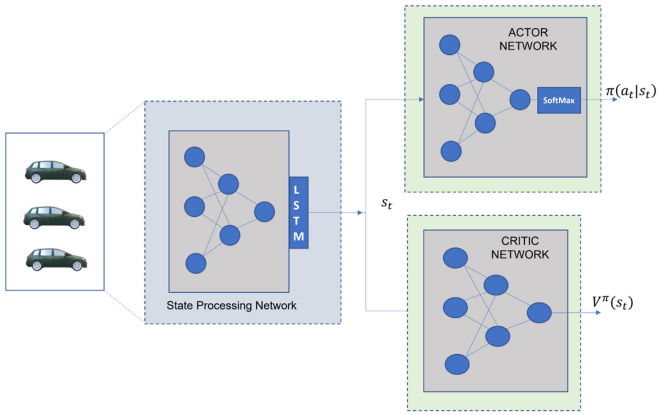

Fig. 3. LSTM-A2C architecture.

D. LSTM-A2C Based Channel Allocation

1) Actor Network: From Fig. 3, the actor network is responsible for generating actions based on current states. First, the actor neural network observes s_t as input and extracts action-related features. Second, the neural network maps the output into the probability of different actions $\pi(a_t|s_t)$ using softmax function.

2) Critic Network: The critic network is responsible for estimating state values. First, the neural network takes the state processing s_t output as its input to extract value-related features. Second, the neural network obtains the state value $V^\pi(s_t)$. We can estimate the state processing network and the critic network with parameter θ_{ct} and the action network with parameter θ_{ac}. Also, we include entropy to loss function of the actor-network to inspire the exploration [34].

$$\mathcal{L}_{ac} = -[\delta_t(s_t, \theta_{ct}) log\pi(a_t|s_t; \theta_{ac}) + \zeta \mathbb{S}\pi(a_t|s_t; \theta_{ac})], \quad (22)$$

where, ζ is the weight of the action entropy $\mathbb{S}$. The parameter update of the actor network can be expressed as,

$$\theta_{ac} \leftarrow \theta_{ac} + \frac{\partial log\pi(a_t|s_t; \theta_{ac})}{\partial \theta_{ac}} \delta_t(s_t, \theta_{ct}) + \zeta \frac{\partial \mathbb{S} log\pi(a_t|s_t; \theta_{ac})}{\partial \theta_{ac}}. \quad (23)$$

The loss function of the critic network is given as,

$$\mathcal{L}_{ct} = (r_t + \gamma V^\pi(s_{t+1}; \theta_{ct}) - V^\pi(s_t; \theta_{ac}))^2. \quad (24)$$

parameter update of the critic network is given as,

$$\theta_{ct} \leftarrow \theta_{ct} + \delta_t(s_t, \theta_{ct}) \frac{\partial V^\pi(s_t; \theta_{ct})}{\partial \theta_{ct}}. \quad (25)$$

The total loss function for the A2C is given as,

$$Total\ Loss = Actor\ loss + critic\ loss * critic\ weight - entropy\ loss * entropy\ weight. \quad (26)$$

V. Performance Evaluation

A. Dataset Collection from PeMS

Data collection for this research is downloaded from Caltrans Performance Measurement System (PeMS) [39]. In this work we consider, the speed of vehicles between two detector stations

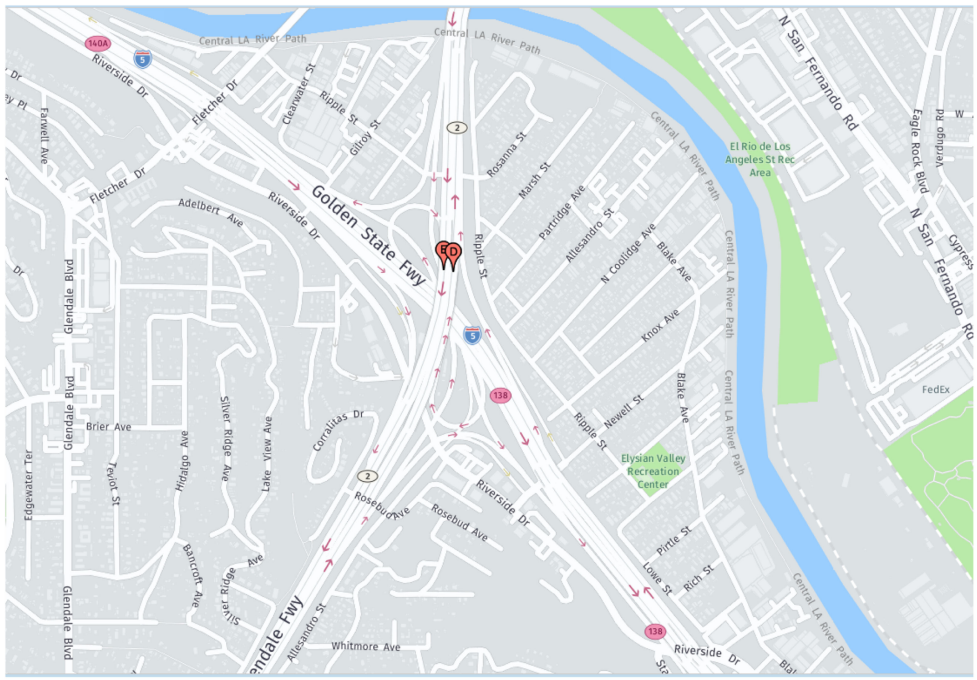

Fig. 4. The traffic flow between two station in the Glendale free way.

Algorithm 2: The LSTM-A2C.

Initialize Parameters: θ_{ac}, θ_{ct}
Buffer length T and time step $t = T + 1$;
for i=1 to T do
Choose an random action $a_i \in A$; perform a_i
Agents observe $\boldsymbol{O_i}$ at end of i^{th} scheduling period
append $\boldsymbol{O_i}$ to the buffer end.
end for
for each iteration do
Append $s_t = (o_{t-T}, o_{t-T+1}, \ldots, o_{t-1})$
State processing network output s_t
Calculate $\pi(a_t|s_t)$, $V^{\pi}(s_t)$ using s_t
Each step compute r_t using (18)
Append o_t to form $s_{t+1} = (o_{t-T+1}, \ldots, o_t)$,
s_{t+1} Output of state processing network is s_{t+1}
Calculate $V^{\pi}(s_{t+1})$
Calculate $\delta_t(s_t) = (r_t + \gamma V^{\pi}(s_{t+1}; \theta_{ct}) - V^{\pi}(s_t))$
Update $\theta_{ac}, \theta_{ct}, t \leftarrow t + 1$
Calculate the total loss using (26)
end for

767484 and 767497 of Glendale free way as shown in Fig. 4. We consider a duration of 2021-04-01 00:00:00 to 2021-07-31 23:59:00 for the data set. We update the dataset with a frequency of every 30 seconds. The whole sample points in the dataset we used include 12,000 samples, among which 80% is used for training, and the remaining 20% is used for testing. As shown in Fig. 4, we obtain the statistical data recorded at several stations near district 7, Los Angeles County. The red dots represent the positions of data detection, which we marked as A, B, C, D, etc., and the red arrows represent the direction of traffic flow.

TABLE II
HYPERPARAMETER FOR LSTM-DQN AND LSTM-A2C

Hyperparameter	Value	Algorithm
Units per layer	[100,100]	All
Activation Function	Sigmoid	All
Batch Size	64	All
Loss Function	MSE	DQN
Reward decay (γ)	0.9	All
ϵ-greedy value	0.99	DQN
Experience Replay Memory Size	100,000	DQN
Packet size	512bytes	All
Exploration steps	1,000	All
Speed	10-140 km/hr	All
Value loss factor	1	A2C
Entropy loss factor	0.1	A2C

B. Experimental Setup

In terms of hardware, all experiments are carried out in Colab [40] with GPU Tesla k80, CPU @ 2.20 GHz, 13 GB of RAM, and hard disk of 108 GB. All the models are built using Keras API. The proposed LSTM-DQN and LSTM-A2C architecture are implemented on the Tensorflow platform (v1.14.0) [41]. Hypermaraters for LSTM are as follows, the learning rate is 0.0001, and the batch size is 64. The ReLu is used in the activation function. Adam optimizer is used. The hyperparameters of DQN and A2C are shown in Table II.

C. Simulation Results

Fig. 5 depicts lane speed prediction over time in minutes. The LSTM updates the input in its memory continuously, which enables long-term learning. There is a fluctuation in the pattern during training the dataset. When the learning minutes increase,

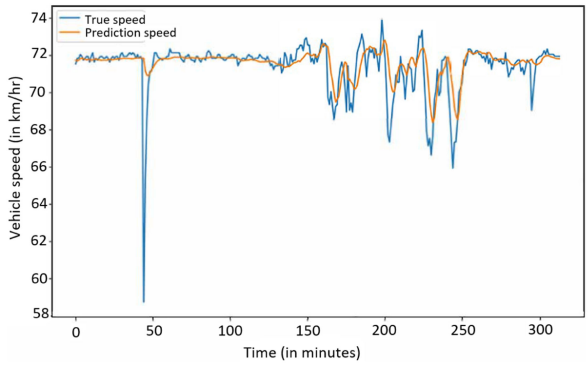

Fig. 5. Mobility prediction of vehicles in highway using LSTM.

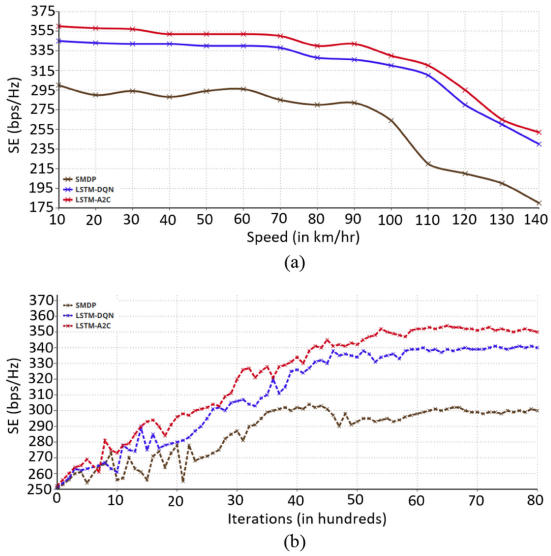

Fig. 6. Spectrum efficiency comparison for LSTM-A2C, LSTM-DQN, and SMDP.

the previous dataset helps LSTM achieve greater prediction accuracy. Therefore, this proves that LSTM is capable of learning and remembering long-term dependencies.

Fig. 6 depicts the SE with various vehicular speeds and an increasing number of iterations. Fig. 6(a) shows the SE with varying user speeds in a single iteration. The SE is high at low speed and decreases with an increased speed due to the elevated number of hand-offs to MBS and consecutive RSUs. In Fig. 6(b), the proposed LSTM-A2C achieves the highest SE among the three methods, indicating that LSTM-A2C can best capture the temporal variations of service requests and adjust the bandwidth allocation flexibly to improve SE. LSTM-DQN also shows some improvement in SE because it conservatively allocates bandwidth to get stable SE. However, this is inferior to LSTM-A2C. In addition, LSTM-A2C also converges to the same final level as LSTM-DQN but exhibits a more stable convergence curve.

A reward is always part of the problem definition and should be based primarily on the agent's goals. Fig. 7 depicts rewards for various episodes. From Fig. 7, SMDP produces good rewards during initial episodes but it gradually decreases with increasing episodes. SMDP is a model-based algorithm that can perform well with synthesized transition probabilities while not

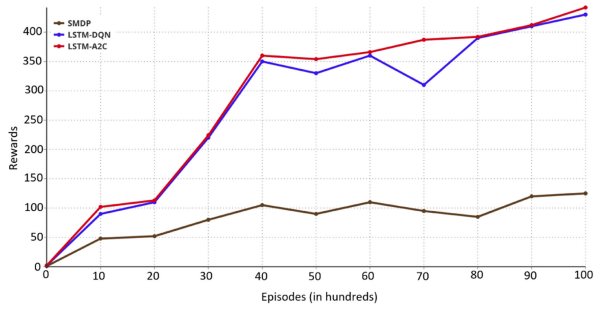

Fig. 7. Reward vs Episodes.

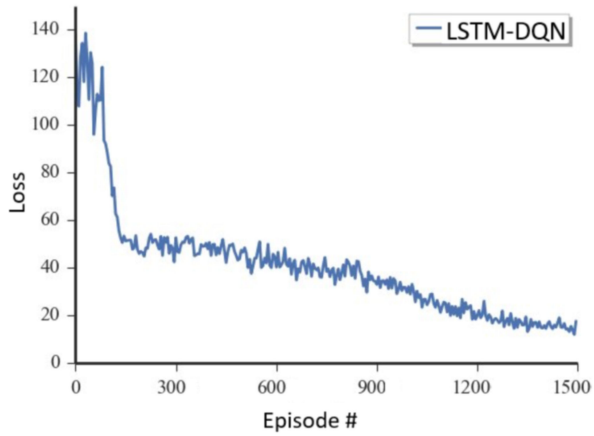

Fig. 8. Loss of LSTM-DQN.

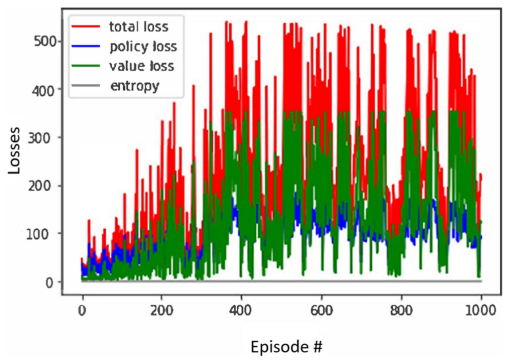

Fig. 9. Total loss of LSTM-A2C.

handling a massive dataset of real-time environment. Also, we can observe that all the RL algorithms have apparent performance improvements through learning and ultimately achieve higher system rewards than SMDP when the number of episodes increases. Although DQN shows increasing rewards among RL algorithms, its performance is not as good as the proposed A2C algorithm, which can achieve a reward of 350 after 4,000 iterations. In addition, the LSTM-A2C algorithm also exhibits superior performance to the LSTM-DQN algorithm in terms of both convergence rate and obtained utility.

The loss function is the TD function, the exact difference between the state at the actual bifurcation point and the state at the estimated future. Fig. 8, depicts the loss derived from (20). Loss decreases with an increase in the number of episodes, showing more accurate predictions of value for the current policy. Fig. 9, shows the total losses derived from (26). The

spikes in the losses are due to change in the environment. As the number of episodes increases LSTM-A2C, adjusts its initial value (start state) to choose the correct actions to reduce the loss.

VI. Conclusion

This paper investigates a multi-agent deep reinforcement learning approach for mobility-aware channel allocation for connected vehicles. We provide an intelligent channel allocation technique for vehicles that have different speeds at different sections. We incorporate the LSTM network into DQN and A2C algorithms to address the significant challenge of MADRL, like partial observability. Our proposed intelligent decision algorithms, LSTM-DQN and LSTM-A2C, accurately capture the demand variations plus user mobility and make appropriate resource allocation decisions in a dynamic network environment.

We have compared the proposed DRL techniques with conventional SMDP. Experimental results show that the proposed DRL techniques can guarantee higher spectrum efficiency and maintain better spectrum efficiency with large fluctuations in user requests, while the users have vastly varying vehicular speeds. We boldly claim that our proposed DRL techniques self-learn and self-adapt to the user mobility incurred variations. We anticipate that the proposed channel allocation techniques will be a valuable addition to the Internet of Vehicles. The extension of the proposed DRL channel allocation algorithm to a fully decentralized multi-agent reinforcement learning scenario is an appealing direction for future research.

References

[1] F. Tang, Y. Kawamoto, N. Kato, and J. Liu, "Future intelligent and secure vehicular network towards 6 G: Machine-learning approaches," *Proc. IEEE*, vol. 108, no. 2, pp. 292–307, Feb. 2020.

[2] J. Gao, M. Li, L. Zhao, and X. Shen, "Contention intensity based distributed coordination for V2V safety message broadcast," *IEEE Trans. Veh. Technol.*, vol. 67, no. 12, pp. 12288–12301, Dec. 2018.

[3] H. A. Shah and L. Zhao, "Multi-agent deep reinforcement learning based virtual resource allocation through network function virtualization in Internet of Things," *IEEE Internet Things J.*, vol. 8, no. 5, pp. 3410–3421, Mar. 2021.

[4] M. Li, L. Zhao, and H. Liang, "An SMDP-Based prioritized channel allocation scheme in cognitive enabled vehicular Ad Hoc networks," *IEEE Tran. Veh. Technol.*, vol. 66, no. 9, pp. 7925–7933, Sep. 2017.

[5] M. Noor-A-Rahim, Z. Liu, H. Lee, G. G. M. N. Ali, D. Pesch, and P. Xiao, "A survey on resource allocation in vehicular networks," *IEEE Trans. Intell. Transp. Syst.*, to be published, doi: 10.1109/TITS.2020.3019322.

[6] I. El Korbi and L. Azouz Saidane, "Performance evaluation of the earliest deadline first policy over Ad Hoc networks," *Int. J. Ad Hoc Ubiquitous Comput.*," vol. 10, no. 3, pp. 175–195, Aug. 2012.

[7] Z. Khan, P. Fan, and S. Fang, "On the connectivity of vehicular Ad Hoc network under various mobility scenarios," *IEEE Access*, vol. 5, pp. 22559–22565, Oct. 2017.

[8] W. Huang, L. Ding, D. Meng, J. Hwang, Y. Xu, and W. Zhang, "QoE-Based resource allocation for heterogeneous multi-radio communication in software-defined vehicle networks," *IEEE Access*, vol. 6, pp. 3387–3399, Jan. 2018.

[9] M. Li, J. Gao, L. Zhao, and X. Shen, "Deep reinforcement learning for collaborative edge computing in vehicular networks," *IEEE Trans. Cogn. Commun. Netw.*, vol. 6, no. 4, pp. 1122–1135, Dec. 2020.

[10] Y. Meng, Y. Dong, X. Liu, and Y. Zhao, "An interference-aware resource allocation scheme for connectivity improvement in vehicular networks," *IEEE Access*, vol. 6, pp. 51319–51328, Aug. 2018.

[11] Y. Hou, X. Wu, X. Tang, X. Qin, and M. Zhou, "Radio resource allocation and power control scheme in V2V communications network," *IEEE Access*, vol. 9, pp. 34529–34540, Feb. 2021.

[12] Z.-Q. Luo and S. Zhang, "Dynamic spectrum management: Complexity and duality," *IEEE J. Sel. Topics Signal Process.*, vol. 2, no. 1, pp. 57–73, Feb. 2008.

[13] M. Li, J. Gao, L. Zhao, and X. Shen, "Adaptive computing scheduling for edge-assisted autonomous driving," *IEEE Trans. Veh. Technol.*, vol. 70, no. 6, pp. 5318–5331, Jun. 2021.

[14] X. Li, L. Ma, R. Shankaran, Y. Xu, and M. A. Orgun, "Joint power control and resource allocation mode selection for safety-related V2X communication," *IEEE Trans. Veh. Technol.*, vol. 68, no. 8, pp. 7970–7986, Aug. 2019.

[15] M. Shojafar, N. Cordeschi, and E. Baccarelli, "Energy-efficient adaptive resource management for real-time vehicular cloud services," *IEEE Trans. Cloud Comput.*, vol. 7, no. 1, pp. 196–209, Jan–Mar. 2019.

[16] T. K. Das, A. Gosavi, S. Mahadevan, and N. Marchalleck, "Solving semi-Markov decision problems using average reward reinforcement learning," *Manage. Sci.*, vol. 45, no. 4, pp. 560–574, Apr. 1999.

[17] Q. Zheng, K. Zheng, H. Zhang, and V. C. M. Leung, "Delay-optimal virtualized radio resource scheduling in software-defined vehicular networks via stochastic learning," *IEEE Trans. Veh. Technol.*, vol. 65, no. 10, pp. 7857–7867, Oct. 2016.

[18] M. E. Morocho-Cayamcela, H. Lee, and W. Lim, "Machine learning for 5G/B5G mobile and wireless communications: Potential, limitations, and future directions," *IEEE Access*, vol. 7, pp. 137184–137206, Sep. 2019.

[19] C. Jiang, H. Zhang, Y. Ren, Z. Han, K. Chen, and L. Hanzo, "Machine learning paradigms for next-generation wireless networks," *IEEE Wireless Commun.*, vol. 24, no. 2, pp. 98–105, Apr. 2017.

[20] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA, USA: MIT Press, 2018.

[21] H. Liang, X. Zhang, J. Zhang, Q. Li, S. Zhou, and L. Zhao, "A novel adaptive resource allocation model based on SMDP and reinforcement learning algorithm in vehicular cloud system," *IEEE Trans. Veh. Technol.*, vol. 68, no. 10, pp. 10018–10029, Oct. 2019.

[22] S. S. Lee and S. Lee, "Resource allocation for vehicular fog computing using reinforcement learning combined with heuristic information," *IEEE Internet Things J.*, vol. 7, no. 10, pp. 10450–10464, Oct. 2020.

[23] T. Şahin, R. Khalili, M. Boban, and A. Wolisz, "Reinforcement learning scheduler for vehicle-to-vehicle communications outside coverage," in *Proc. IEEE Veh. Netw. Conf.*, 2018, pp. 1–8.

[24] L. Hou, K. Zheng, P. Chatzimisios, and Y. Feng, "A continuous-time Markov decision process-based resource allocation scheme in vehicular cloud for mobile video services," *Comput. Commun.*, vol. 118, pp. 140–147, Mar. 2018.

[25] S. Hwang, H. Kim, H. Lee, and I. Lee, "Multi-agent deep reinforcement learning for distributed resource management in wirelessly powered communication networks," *IEEE Trans. Veh. Technol.*, vol. 69, no. 11, pp. 14055–14060, Nov. 2020.

[26] O. Naparstek and K. Cohen, "Deep multi-user reinforcement learning for distributed dynamic spectrum access," *IEEE Trans. Wireless Commun.*, vol. 18, no. 1, pp. 310–323, Jan. 2019.

[27] Y. S. Nasir and D. Guo, "Deep reinforcement learning for distributed dynamic power allocation in wireless networks," Aug. 2018. [Online]. Available: https://arxiv.org/abs/1808.00490

[28] L. Liang, H. Ye, G. Yu, and G. Y. Li, "Deep-learning-based wireless resource allocation with application to vehicular networks," *Proc. IEEE*, vol. 108, no. 2, pp. 341–356, Feb. 2020.

[29] R. Li *et al.*, "Deep reinforcement learning for resource management in network slicing," *IEEE Access*, vol. 6, pp. 74429–74441, Nov. 2018.

[30] L. T. Tan, R. Q. Hu, and L. Hanzo, "Twin-timescale artificial intelligence aided mobility-aware edge caching and computing in vehicular networks," *IEEE Trans. Veh. Technol.*, vol. 68, no. 4, pp. 3086–3099, Apr. 2019.

[31] H. Peng and X. Shen, "Multi-agent reinforcement learning based resource management in MEC- and UAV-assisted vehicular networks," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 1, pp. 131–141, Jan. 2021.

[32] Y. Chen, Z-Y. Liu, Y-C. Zhang, Y. Wu, X. Chen, and L. Zhao, "Deep reinforcement learning based dynamic resource management for mobile edge computing in industrial Internet of Things," *IEEE Trans. Ind. Informat.*, vol. 17, no. 7, pp. 4925–4934, Jul. 2021.

[33] H. Ye, G. Y. Li, and B. H. F. Juang, "Deep reinforcement learning based resource allocation for V2V communications," *IEEE Trans. Veh. Technol.*, vol. 68, no. 4, pp. 3163–3173, Apr. 2019.

[34] R. Li, C. Wang, Z. Zhao, R. Guo, and H. Zhang, "The LSTM-Based advantage Actor-Critic learning for resource management in network slicing with user mobility," *IEEE Commun. Lett.*, vol. 24, no. 9, pp. 2005–2009, Sep. 2020.

[35] T. T. Nguyen, N. D. Nguyen, and S. Nahavandi, "Deep reinforcement learning for multiagent systems: A review of challenges, solutions, and applications," *IEEE Trans. Cybern.*, vol. 50, no. 9, pp. 3826–3839, Sep. 2020.
[36] A. S. Kumar, L. Zhao, and X. Fernando, "Mobility aware channel allocation for 5G vehicular networks using multi-agent reinforcement learning," *Proc. IEEE Int. Conf. Commun.*, 2021, pp. 1–6.
[37] L. Liang, H. Ye, and G. Y. Li, "Spectrum sharing in vehicular networks based on multi-agent reinforcement learning," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 10, pp. 2282–2292, Oct. 2019.
[38] Y. Hua, Z. Zhao, R. Li, X. Chen, Z. Liu, and H. Zhang, "Deep learning with long short-term memory for time series prediction," *IEEE Commun. Mag.*, vol. 57, no. 6, pp. 114–119, Jun. 2019.
[39] California department of transportation - Caltrans, "Performance measurement system (PeMS)," 2021. [Online]. Available: https://pems.dot.ca.gov/
[40] Google colab, [Online]. Available: https://colab.research.google.com/
[41] Tensorflow, [Online]. Available: https://www.tensorflow.org/

Anitha Saravana Kumar (Student Member, IEEE) received the M.E. and Ph.D. degrees in electronics and communication engineering and information and communication from Anna University, Chennai, India, in 2004 and 2015, respectively. She is currently working toward the Ph.D. degree with the Department of Electrical, Computer, & Biomedical Engineering, Ryerson University, Toronto, ON, Canada. She has coauthored more than 30 journal papers and conference papers. Her current research interests include the autonomous vehicular network, resource management, and reinforcement learning. She was a Reviewer for the IEEE INTERNET OF THINGS JOURNAL and Encyclopedia of Computer Science and Technology, Taylor, and Francis publications. She was the TPC Member for the IEEE Globecom 2020 and Globecom 2021 conferences.

Lian Zhao (Senior Member, IEEE) received the Ph.D. degree from the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, Canada, in 2002. She joined the Department of Electrical and Computer Engineering, Ryerson University, Toronto, ON, Canada, in 2003, and was a Professor in 2014. Her research interests include the areas of wireless communications, resource management, mobile edge computing, caching and communications, and IoV networks.

She was the recipient of the Best Land Transportation Paper Award from IEEE Vehicular Technology Society in 2016, Top 15 Editor in 2015, for the IEEE TRANSACTION ON VEHICULAR TECHNOLOGY, Best Paper Award from the 2013 International Conference on *Wireless Communications and Signal Processing* (WCSP) and Best Student Paper Award (with her student) from Chinacom in 2011, the Canada Foundation for Innovation (CFI) New Opportunity Research Award in 2005, and Early Tenure and promotion to Associate Professor in 2006. She is an IEEE Communication Society (ComSoc) Distinguished Lecturer, has been the Editor of the IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY since 2013, the Editor of the IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS and the IEEE INTERNET OF THINGS JOURNALS the Co-Chair for Globecom 2020 and IEEE ICC 2018 Wireless Communication Symposium, workshop Co-Chair for IEEE/CIC ICCC 2015, local arrangement Co-Chair for IEEE VTC Fall 2017 and IEEE Infocom 2014, Co-Chair for the IEEE Global Communications Conference (GLOBECOM) 2013 Communication Theory Symposium.

She was a Panel Expert for the Discovery Grant Program and Evaluation Committee for the Research Tools and Instruments Grants Program of Natural Sciences Engineering Research Council of Canada (NSERC). She is a licensed Professional Engineer with the Province of Ontario and a Senior Member of the IEEE Communication Society and Vehicular Technology Society.

Xavier Fernando (Senior Member, IEEE) was born in Sri Lanka. He received the B.Sc. Eng. (First Class Hons.) degree in electrical and electronic engineering from the University of Peradeniya, Sri Lanka, in 1992, and the master's degree in telecommunications from the Asian Institute of Technology, Thailand, in 1994, and the Ph.D. degree in electrical and computer engineering specializing in wireless communications affiliation with TRLabs., from the University of Calgary, Calgary, AB, Canada, in 2001.

From 1994 to 1997, he was an R&D Engineer for AT&T, Thailand. He is currently a Professor with Ryerson University, Toronto, Toronto, ON, Canada, and the Director of Ryerson Communications Lab that has received over $3.2 Million research funding so far. From 2011 to 2012, he was with the Ryerson University Board of Governors. In 2008, he was a Visiting Scholar with the Institute of Advanced Telecommunications, U.K., and a MAPNET Visiting Fellow with Aston University, Birmingham, U.K., in 2014. He has authored or coauthored one monograph (translated to Mandarin) and coauthored two more books, 68 journal papers and 139 Conference Papers, five book chapters and holds three patents. His current research focuses on wireless communication and positioning.

Dr. Fernando is a licensed Professional Engineer with Ontario. He was an IEEE Communications Society Distinguished Lecturer and delivered more than 65 invited talks and keynote presentations worldwide. He has been in the organizing/steering/technical program committees of many conferences. He is an Associate Editor forthe IEEE INTERNET OF THINGS JOURNAL and *IEEE Consumer Technology Magazine*. He was a Program Evaluator for ABET, USA. His work has won 30 awards and prizes so far including, Professional Engineers Ontario Award in 2016, IEEE Microwave Theory and Techniques Society Prize in 2010, Sarnoff Symposium Prize in 2009, Opto-Canada Best Poster Prize in 2003, and CCECE Best Paper Prize in 2001. He was the recipient of the Ryerson University Service Excellence Award in 2012, and a finalist for Canadian Top Immigrant Award in 2012.