This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

Fuzzy Inference Enabled Deep Reinforcement Learning-Based Traffic Light Control for Intelligent Transportation System

Neetesh Kumar, *Member, IEEE*, Syed Shameerur Rahman, and Navin Dhakad

***Abstract*— Intelligent Transportation System (ITS) has been emerged an important component and widely adopted for the smart city as it overcomes the limitations of the traditional transportation system. Existing fixed traffic light control systems split the traffic light signal into fixed duration and run in an inefficient way, therefore, it suffers from many weaknesses such as long waiting time, waste of fuel and increase in carbon emission. To tackle these issues and increase efficiency of the traffic light control system, in this work, a Dynamic and Intelligent Traffic Light Control System (DITLCS) is proposed which takes real-time traffic information as the input and dynamically adjusts the traffic light duration. Further, the proposed DITLCS runs in three modes namely Fair Mode (FM), Priority Mode (PM) and Emergency Mode (EM) where all the vehicles are considered with equal priority, vehicles of different categories are given different level of priority and emergency vehicles are given at most priority respectively. Furthermore, a deep reinforcement learning model is also proposed to switch the traffic lights in different phases (Red, Green and Yellow), and fuzzy inference system selects one mode among three modes i.e., FM, PM and EM according to the traffic information. We have evaluated DITLCS via realistic simulation on Gwalior city map of India using an open-source simulator i.e., Simulation of Urban MObility (SUMO). The simulation results prove the efficiency of DITLCS in comparison to other state of the art algorithms on various performance parameters.**

***Index Terms*— Deep learning, intelligent transportation system (ITS), fuzzy logic, dynamic traffic light control.**

I. Introduction

RAPID increase in number of vehicles, despite of limited transportation physical infrastructure, is a critical issue in development of smart cities. High density of the vehicles raises many environmental (air, noise pollution), health (stress, diseases), collision and economical (cost of fuel) issues, which cause time delay and fuel wastage because of chaotic road traffic situations. Traffic light control system plays an important role in vehicles movement and handling traffic congestion, accidents etc. in the cities. However, inefficient vehicle traffic management system causes numerous issues such as long waiting time, wastage of fuel, high carbon footprints and vehicular accidents [1], [2]. Current traffic light control system deploys a fixed duration traffic light adjustment without taking care of real-time traffic or considering the traffic scenario to a limited extends [1]. Further, fixed duration traffic light control system splits the traffic signal duration to equal time in every phase without considering real time traffic data. Some other traffic light control algorithms take the traffic information as the input from sensors i.e., inductive loop detectors to capture the traffic data. The input data is then processed to determine the phase duration of the green light signal. However, such traffic light control systems will be paralyzed when the inflow rate of the traffic is relatively large at the certain point of times. Thus, many times, traffic-policemen have to directly manage the traffic light intersections by hand-waving signals.

Thus, it is very prominent to deploy an intelligent and dynamic traffic light control system to manage vehicular dynamics efficiently, so that traffic may flow smoothly despite of high density in traffic and limited infrastructure. Therefore, utilization of Intelligent Transportation System (ITS) has elevated the expectations of society and the individual traveller based on promise of the technologies involved. Major objectives of ITS are to evaluate, develop, analyse and integrate new sensor, information and communication technologies, and algorithms to achieve traffic efficiency. Thus, it can improve environmental quality, save energy, conserve time, enhance safety, comfort for the drivers and pedestrians or other traffic groups. Few research efforts have been made on dynamic traffic light control systems using reinforcement learning technique, which defines the states by the waiting queue length [15], [16]. Reinforcement learning technique is aimed to make an agent learn the optimal action policy through an interaction with the environment in order to maximize the reward such as minimization of average waiting time in traffic management scenarios. However, real traffic conditions cannot be accurately captured by the queue length [17]. Additionally, the number of state increases which indicates an exponential growth in the complexity of the traditional reinforcement learning system. With the recent advancements in machine learning techniques, deep neural networks are the major focused areas to deal with when there is large number of states and to solve such vehicular traffic problems [18]–[20]. Further, with technological advancement, more traffic states

Manuscript received July 9, 2019; revised December 2, 2019 and January 28, 2020; accepted March 23, 2020. This work was supported by the Ministry of Science and Technology (DST), ICPS Division, New Delhi, Government of India through the Project IoT based automated, Real Time and Effective Traffic Signal Scheduling for Smart City under Grant T-678. The Associate Editor for this article was S. Arhin. *(Corresponding author: Neetesh Kumar.)*

Neetesh Kumar and Syed Shameerur Rahman are with the Department of Information Technology, Atal Bihari Vajpayee Indian Institute of Information Technology and Management Gwalior, Gwalior 474015, India (e-mail: nkiiitmg@iiitm.ac.in).

Navin Dhakad is with the Indian Institute of Information Technology Pune, Pune 411048, India.

Digital Object Identifier 10.1109/TITS.2020.2984033

can be captured with newer sensors, cameras which can lead to more accurate traffic light management [5], [26], [27]. However, there are certain limitations of the existing systems are itemized as follow:

- For existing traffic light control systems, traffic signals splatted only into fixed intervals and green/red lights duration can only be a manifold of the same interval.
- Existing studies were focused on only finding the accurate traffic duration but not considered vehicle priority [1]. As obvious, many priority and emergency vehicles such as fire-brigade, ambulance etc., need to be reached their destinations as soon as possible.
- Existing traffic light controllers are unable to understand the traffic behaviour and operate in different modes i.e., emergency/priority/fair mode.
- Traditional ITS solutions are not cost effective in terms of IT and IoT infrastructure (Rasberry pi, Cloud etc.) for developing vehicular networks. Such type of systems are commercially expensive for the developing countries.

Traditional traffic light control systems are lacking in meeting the current requirements of the smart city, therefore, in order to overcome the above mentioned limitations of the existing systems, in this work, an efficient, Dynamic and Intelligent Traffic Light Control System (DITLCS) is proposed. The significant contributions of the proposed DITLCS are listed as follow:

- In DITLCS, time duration for each phase of traffic light signal is calculated based on extracted information (traffic load, vehicles heterogeneity) from vehicular network.
- Considering heterogeneous traffic load, an algorithm is proposed utilizing fuzzy inference system to choose one out of three modes: 1) Fair Mode (FM):-all vehicles have same weightage 2) Priority Mode (PM):- vehicle weights are differentiated based on priority 3) Emergency Mode (EM):-emergency vehicle routes are prioritized first.
- An agent is trained utilizing deep reinforcement learning on traffic data which mimics the behaviour of an experienced human agent. This agent can run on light weighted computing device (Raspberry Pi) to predict traffic light signal phase for the each mode of the DITLCS.
- To verify the effectiveness of DITLCS, a realistic simulation is developed utilizing an open source simulator tool i.e., Simulation of Urban Mobility (SUMO) [16] on Gwalior city, India. Results of DITLCS are compared with several state of the art algorithms on various testing parameters which proved the effectiveness of the model.

Rest of the paper is organized as follows. The literature review is presented in Section II. The model and problem statement are introduced in Section III. The background on reinforcement learning is introduced in Section IV. The technical details of the DITLCS is presented in Section V. The Simulation study of the model is given in Section VI. Finally, the paper is concluded in Section VII.

II. Literature Review

Due to immense growth in vehicles and limited transport infrastructure, ITS has been widely focused as a prominent area of research, thus, traffic optimization has been studied in recent decades. Author [7], [8] utilized fuzzy model to develop the traffic light controller with the aim of minimizing Average Waiting Time (AWT) and Queue Length (QL) but they missed heterogeneity. Further, Mir and Hassan [9] developed a traffic light controller utilizing neural network and fuzzy controller to minimize AWT and QL for car vehicles. To tackle the emergency vehicles, Younes and Boukerche [12] developed a heuristic based algorithm and verified via simulation in SUMO, however, they ignored the other priority vehicles. In order to account the priority vehicles, Khan *et al.* [11] developed a EVP-STC protocol utilizing micro-controller, ZigBee, GPS and other IoT devices, and varied its performance effectiveness using PTV Vissim. However, authors considered line opening time but missed several important parameters in vehicular dynamics.

Abadi *et al.* [2] parented a strategy to predict the traffic flow on road transportation vehicular networks but with limited traffic data. They utilized autoregressive model and Monte Carlo simulation technique, however, it fails to consider vehicle priority and make traffic light control decisions. Quek *et al.* [3] proposed a novel fuzzy neural approach in order to analyse and predict road traffic utilizing pseudo outer-product fuzzy neural network using the truth-value-restriction. This work accounts the vehicle heterogeneity to analyse the traffic behaviour but not designed to flow traffic. Zhang *et al.* [4] suggested a force-driven traffic simulation for the future connected autonomous vehicle-enabled smart transportation system, however, they missed the requirement traffic control by understanding the vehicular dynamics. Kachroo and Sastry [10] developed a theoretical mathematical model to analyse the density based travel time for real time vehicular dynamics. Authors show that it is important for intelligent transportation system applications where travel time is an important factor. Bekiaris-Liberis *et al.* [6] developed a methodology for highway traffic state estimation with mixed connected and conventional vehicles. Authors validated the performance of the developed estimation schemes through simulations using a well-known second-order traffic flow model. Belletti *et al.* [5] utilized multi task deep reinforcement learning for expert level control of ramp metering they proved neural network based deep reinforcement is effective to the problems whose parameters are known. The domain experts [1] have used and suggested the application of Convolutional Neural Network (CNN) with reinforcement leaning referred deep reinforcement leaning to control the dynamic traffic light but authors missed the heterogeneity among the vehicles.

III. Model and Problem Statement

This work considers heterogeneous vehicular traffic network of the developing counties with the aim of developing a dynamic and intelligent traffic light signal scheduling system. A pictorial view of the model is shown in Fig. 1. The proposed model works in two steps; first, the traffic light control system gathers the vehicle traffic information via a vehicular network [14] with the help of cameras, Radio-Frequency Identification (RFID) sensors etc., that is represented by the red dotted lines in the Fig.1. The traffic light controller (IoT device

TABLE I
LITERATURE SUMMARY

Reference	Objective	Techniques	Simulation Tools	Priority	Multi-mode
Lianget al. [1]	To minimize waiting time	Deep reinforcement learning	SUMO	No	No
Abadi et al. [2]	Traffic flow prediction	Autoregressive model, Monte Carlo	VISUM	No	No
Quek et al.[3]	Road traffic analysis and prediction	Fuzzy neural network	SNNS	Yes	No
Zhang et al. [4]	To create a model for future CAV-enabled ITS	Smoothed Particle Hydrodynamics (SPH)	SPH	No	No
Belletti et al. [5]	To conduct non-parametric test	Deep reinforcement learning	Mathematical analysis	No	No
Liberis et al. [6]	To estimate traffic state for mixed traffic	Kalman filter to estimate total density	Microscopic platform	yes	No
Kafash et al. [7]	To minimize queue length and waiting time	Fuzzy rules	Matlab	No	No
Azimirad et al. [8]	To minimize average waiting time	Fuzzy model	Mtalab	No	No
Mir et al. [9]	To minimize queue length and waiting time	Neural network and fuzzy rules	Matlab	No	No
Kachroo et al. [10]	To manage density base travel time dynamics	Hyperbolic partial differential equations	Mathematical analysis	No	No
Khan et al.[11]	To delope platform and protocol	Microcontroller and heuristics	PTV Vissim	Yes	No
Younes et al. [12]	To minimize delay and enhance throughput	Heuristic	SUMO	Yes	No
This work	To improve AWT, Tp, AQL, AS, CO_2	Fuzzy and Deep reinforcement learning	Python, Tracki, SUMO	Yes	Yes

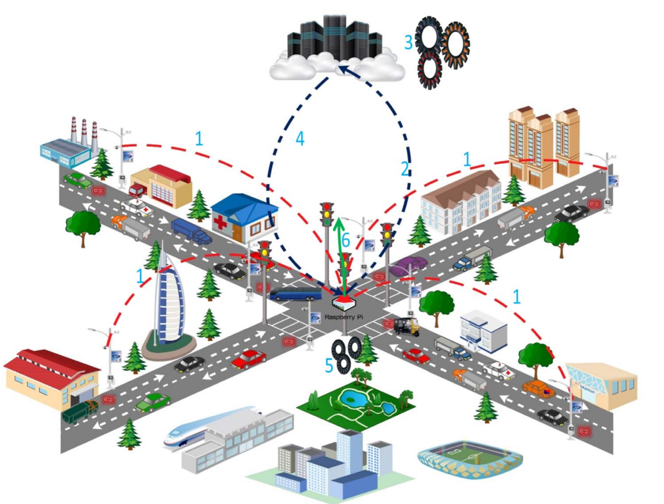


Fig. 1. The proposed DITLCS architecture.

i.e., Raspberry Pi) then processes the input numerical data to obtain the vehicle traffic's state and other parameters such as queue length, waiting time etc., which also has been assumed in many previous studies [1], [15], [16]. Based on the input traffic data, step 1 of DITLCS acts, that is a fuzzy inference system finds the optimal mode of the operation among the three defined modes namely FM, PM and EM. Next, step 2 finds traffic signal phase and duration. In DITLCS, traffic light controllers at the intersections are used to manage the vehicle flow. A traffic light has three signals, namely Red(R), Yellow(Y) and Green(G). One traffic light cannot control all the vehicles when there are vehicles flowing from multiple directions. Thus, multiple traffic lights need to work in synchronization at each intersection to control the traffic flow from multiple directions. At such traffic light junctions, the traffic lights keep changing in such a way that there is non-conflicting flow of traffic in multiple direction. The time duration of staying at one particular signal (R, Y, G) is called one phase. The total number of phases is decided by the number of possible signals at the junction. All the phases keep on changing cyclically at the junction. It is called a cycle which keeps on repeating and one of the objectives is to dynamically set the signal phase duration. Once the traffic signal is red at an intersection the vehicles are not allowed to pass through. On the contrary, if the signal is green the vehicle can have free passage. The objective is to minimize the average waiting time of the vehicles considering heterogeneity among the vehicles such as ambulance and fire-brigade and to optimize the traffic flow intelligently at an intersection by dynamically choosing a mode (FM/PM/EM) of the operations and duration of each phase. Each modes of operation has its own objectives 1) FM is to minimize the average waiting time of all vehicles 2) PM to minimize the average waiting time of vehicles in the decreasing order of priority 3) EM is to minimize the average waiting time of the emergency vehicles by allowing them to cross the intersection first.

IV. BACKGROUND ON REINFORCEMENT LEARNING

Reinforcement learning is one of the types of machine learning algorithms which is different from other machine learning approaches such as supervised learning and unsupervised learning [15]. There is an agent in reinforcement learning which interacts with the environment to get rewards from a set of actions. The ultimate goal of the agent is to take an action according to a given scenario and to maximize the numerical rewards in the long run. In reinforcement learning, an agent is an action executor which changes the state of the environment from one to another. Reinforcement learning model is well defined by a three-tuples (S, A, R) which has the following meanings.

1. S: Possible set of state space. s is a specific state (s $\in$ S);
2. A: Possible set of action space. a is an action (a $\in$ A);
3. R: Reward is something an agent receives after action execution

A policy is a set of sequential actions performed by an agent. The ultimate goal in reinforcement learning model is to interact with the environment and learn an optimal policy to maximize the cumulative rewards.

An agent obtains reward or punishment at every time step according to the action chosen. The Agent tries to increase the cumulative reward at long run. We take two observations during each time step. First observation is taken before executing the action and the second observation is taken after executing the action. Let r1 and r2 represents the reward observed before and after executing the action. Finally the total reward R is calculated as follows.

$$R = r1 - r2 \tag{1}$$

Agent Goal: The objective of the agent is to maximize the cumulative reward in the long run, the agent needs to find an action policy π that maximizes the following cumulative future reward, the same reward is calculated using the following Eq. 2 [1].

$$Q_\pi(s,a) = \mathbf{E}[R_t + \alpha R_{t+1} + \alpha^2 R_{t+2} + \ldots | S_t = s, A_t = a, \pi] \tag{2}$$

In the equation, α is the discount factor, which is usually in [0, 1]. It means that the nearest rewards are worthier than the rewards in the further future.

$$R_t = \sum_{i=1}^{n} P_i * WT_i \tag{3}$$

where, P_i is the priority weight of i^{th} vehicle defined as per the mode of operation, WT_i is the waiting time of i^{th} vehicle, n is total number of vehicle in the intersection.

The assumption is that the cumulative reward is equal to the sum of the immediate rewards. If the estimated future optimal reward is obtained, the cumulative reward now can be calculated. The Eq. 2 can be solved by many approaches but the dynamic programming approach is the best in terms of time complexity [1], but for that the number of states should be finite to make the computing complexity easier and faster. When the number of states becomes large, there is a need to approximate the Q-value [1]. In case of vehicular traffic network the number of states are infinite, hence, there is a need to approximate the Q-value.

V. The Proposed Fuzzy Inference Enabled Deep Reinforcement Learning Model

As discussed, reinforcement learning model is well defined by a three-tuples (S, A, R). This section defines all the three parameters in terms of vehicular network scenario.

States: The state of the traffic at an intersection is defined by two parameters namely position and speed of the vehicle in the vehicular network. The whole traffic junction is partitioned into square grids all of same size with length c. The primary condition to set up a grid should be such that, no two vehicles will be on the same grid. The value of c is modifiable according to the needs. In every grid, the state space is defined by two-tuples (position, speed) of the vehicle present inside the grid area. As shown in Fig. 2, the position is a binary value where, zero (0) means there is no vehicle present inside the grid, and one (1) indicates the presence of vehicle. The speed dimension is a float value which indicates the current speed of the vehicle in meter/second.

Action Space: An agent is responsible for choosing an appropriate action for the smooth flow of the vehicles at the junction based on the current traffic scenario. In this system, the agent scans the state of the traffic and chooses one of two actions: 0 - don't change the traffic signal phase and 1 - turn on green lights for the next traffic light in sequence.

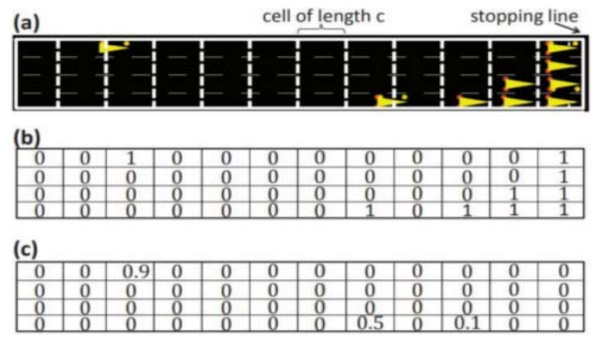


Fig. 2. Demonstration of conversion of traffic state into input matrix.

Algorithm 1 DITLCS

Input: qL, WT (from real-time vehicle traffic network)
Output: Dynamic traffic light signal control

```
phase = 0;
while True do
    mode = findMode(qL,WT);
    if mode == fair then
        W_T = initialize to 1;
    end if
    if mode == priority then
        W_T = initialize based on vehicle priority;
    end if
    if mode == emergency then
        W_T = initialize based on vehicle priority;
        while emergency vehicle is present do
            makeSignalGreen;
        end while
    end if
    agent = DQN(P, V);
    if agent == 1 then
        ChangeTrafficSignalPhase;
        phase = (phase + 1) % 4;
    end if
    if agent == 0 then
        duration = findDuration(W_T, phase);
        setGreenSignal(duration) //a function to set traffic
        signal to green;
    end if
end while
```

Rewards: The importance of the reward in reinforcement learning model is to provide a feedback based on the previously executed action so that forthcoming actions can be taken in an efficient way. Higher reward indicates that the executed action is much significant. In each iteration, the offered rewards to an agent can be positive or negative. As discussed the agent tries to minimize the average waiting time, delay and queue length and to maximize average speed for vehicles at the junction, therefore, two observations are taken at each iteration one before the action is executed, and the other after the action is executed. Let's say that r1 be the reward before execution of the action and r2 be the reward after the action execution then, the final i.e., $R(= r1 - r2)$ reward can be calculated as follows:

$$r1 = \sum_{i=1}^{n}\sum_{j=1}^{x} W_{ij} * (NWT_{ij} - NS_{ij}) + \sum_{k=1}^{nL} W_k(ND_k + NqL_k) \tag{4}$$

$$r2 = \sum_{i=1}^{n}\sum_{j=1}^{y} W_{ij} * (NWT_{ij} - NS_{ij}) + \sum_{k=1}^{nL} W_k(ND_k + NqL_k) \tag{5}$$

$$X = \frac{Curr_{value}X - Min_{value}X}{Max_{value}X - Min_{value}X} \tag{6}$$

where, n and nL are the number of traffic intersection and lane in the vehicular network respectively, and x and y are the number of vehicle in the intersection before and after the execution of an action respectively. W_{ij} is the weight of j^{th} vehicle in i^{th} intersection, and NWT_{ij} and NS_{ij} are the normalized waiting time and speed of j_{th} vehicle in i_{th} intersection respectively. W_k, ND_k and NqL_k are the lane weight (sum of vehicles on the lane), normalized delay [13] and queue length [13] of the lane respectively. In Eq. 6, X is the normalizing parameter (NWT_{ij}, NS_{ij} ND_k and NqL_k), and $Curr_{value}X$, $Max_{value}X$ and $Min_{value}X$ are the current, maximum and minimum value of the parameter X respectively. The maximum and minimum values for each parameter are the maximum and minimum values achieved till the current iteration for the respective parameter. The positive value of R indicates that the chosen action is the best suited for the state space of traffic.

A. Deep Reinforcement Learning Model

In case of vehicular traffic network, the number of state space are large. Thus, we make use of a deep learning technique, Convolutional Neural Network (CNN) [16] to approximate Q value [1]. Hence, the proposed solution for the vehicular network architecture is the combination of CNN to decide whether to change the traffic light phase to the next phase and the fuzzy logic to decide the mode of operation (FM, PM, EM). Since, in case of vehicle traffic network, we have apparently infinite states to work with, storing the Q-value for each and every state is not a good idea as we might run out of memory sometimes, and the processing cost is also high. Thus, we need an approximation model for estimating the Q-value for the given vehicle traffic network state. For finding the optimal approximate Q-values, recursive relationship has been followed, known as Bellman Optimality Equation as shown in Eq. 7 [1].

$$Q^*(s,a) = \mathbf{E}[R_t + \alpha Q^*(S_{t+1}, a^{'})|S_t = s, A_t = a], \quad forall\ s \in S, \quad a \in A, \tag{7}$$

The cumulative reward received by the agent is the sum of all immediate reward received from executing an action and the optimal future reward thereafter.

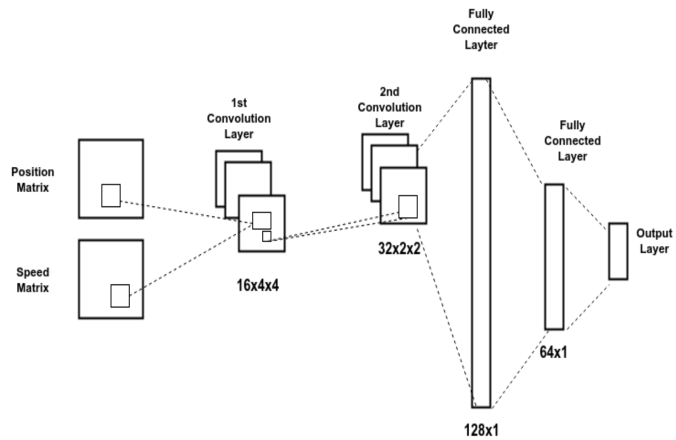


Fig. 3. The proposed convolutional neural network architecture.

B. Convolutional Neural Network

The architecture of the proposed CNN is shown in Fig. 3. The first layer of convolution network has 16 filters of 4×4 with stride 2, and it applies activation function as Rectified Linear Unit (ReLU) [1]. The equation of ReLU activation is given below.

$$f(x) = \begin{cases} x & x > 0 \\ \beta x & x \leqslant 0 \end{cases} \tag{8}$$

The second convolution layer has 32 filters of size 2x2 with stride 1 that also applies activation function as ReLU. Third and fourth layers are fully connected convolution layers of size 128 and 64 respectively. Final layer is linear layer which outputs Q values corresponding to every possible action that agent takes. Now, we will discuss about the training phase of our convolutional neural network. Firstly, we initialize our convolutional neural network(CNN) with random weights. At the beginning of every time step, agent observes the current state of the vehicle traffic in terms of the matrix (position, speed) which forms the state space S_t and it is fed as input to the neural network and performs an action A_t, where $A_t \in [0, 1]$ that has highest future cumulative reward. After performing the action, agent receives reward R_t which can be either positive or negative and next state S_{t+1} as the result of chosen action. Agent then stores this experience (S_t, A_t, R_t, S_{t+1}) in memory. As the memory is of limited size, maximum memory size is defined at the starting phase of the training. Once, the threshold limit of the memory is reached, the oldest data is removed. Deep Neural Network (DNN) is trained by extracting training examples of type (S_t, A_t) from the memory. After collecting the training data, agent learns features θ by training the DNN to minimize the following Mean Squared Error (MSE).

$$MSE(\Theta) = \frac{1}{M}\sum_{t=1}^{m}[R_t + \alpha max\, Q(S_(t+1), a, \Theta) - Q(S_t, A_t, \Theta]^2 \tag{9}$$

where, m is the size of the input data set. Since, m is large, the computational cost to calculate MSE(Θ) is large, hence,

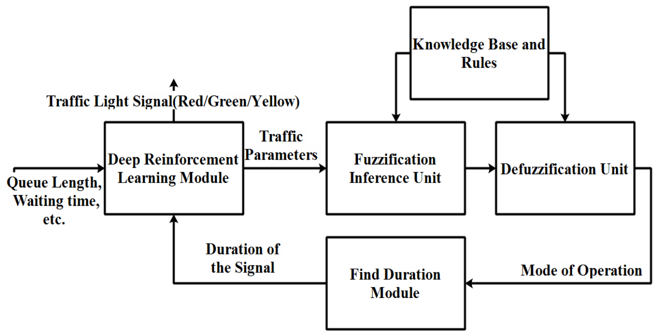


Fig. 4. Logical flow in DITLCS components.

we use stochastic gradient descent algorithm RMSProp with mini batch of size of 32.

C. Fuzzy Inference System

As discussed, the proposed DITLCS vehicular traffic system operates in three modes FM, PM and EM. The selection of the best suited operating mode for the particular traffic scenario is done by the fuzzy logic as shown in Fig. 4.

The term fuzzy refers to things which are not really clear or are vague. In the real world, many times, a lot of situations are encountered when it is hard to determine whether the state is true or false. Thus, fuzzy logic provides a very valuable flexibility for reasoning. In this way, the inaccuracies and uncertainties can be considered for any situation. In Boolean system truth value, 1.0 represents absolute truth value and 0.0 represents absolute false value. However, in fuzzy system, there is no logic for absolute truth and absolute false value. But in fuzzy logic, there is intermediate value too present which is partially true and partially false. The Architecture of our fuzzy inference system is shown in Fig. 4. All the fuzzy system has the following subsystems

- **Rule Base:** It contains the set of rules in form of IF-THEN conditions. In this work, we have considered three parameters for our rule formation; waiting time of vehicle, queue length, and time of operation. More information of the rule formation based on the above parameters will be discussed in the upcoming section/subsection.
- **Fuzzification:** As mentioned, it is the technique to convert the inputs i.e., crisp numbers into fuzzy sets. Crisp inputs are those which we get from the vehicular network such as waiting time, queue length, and time of operation. Let A is a given set. The membership function can be used to define a set A is given by

$$\mu A(x) = \begin{cases} 1 & x \in A \\ 0 & x \notin A \end{cases} \tag{10}$$

- **Inference Engine:** Based on the input it fires which rule in our rule base is to be executed.
- **DeFuzzification:** It is used to convert the fuzzy sets obtained by inference engine into a crisp value. In our case, we use λ cut method [28] to get the crisp output. λ is a threshold value and all the membership value greater than λ will be taken into consideration.

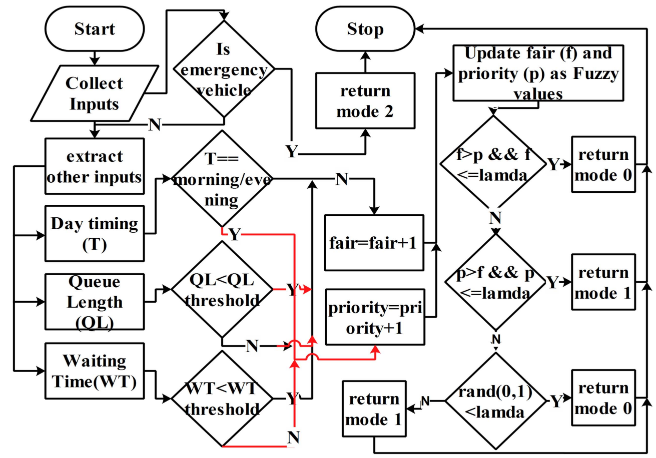


Fig. 5. Logical flow of operations in $findMode$ fuzzy inference system.

D. DITLCS Logical Flow

Fig.1 shows the logical flow among the DITLCS components in vehicular traffic control system. Algorithm 1 explains the complete flow. The High definition camera (HD camera)/sensors placed at the intersections to capture the vehicle data. The input information is then fed into a fuzzy inference system as shown in Fig. 4, fuzzy module returns a decisive mode of execution in which DITLCS must run. The mode of execution can be one any one out of three 1) FM 2) PM 3) EM. Fig. 5 pictorially represents a logical flow of fuzzification and de-fuzzification process of the DITLCS, where 0, 1 and 2 denote FM, PM and EM respectively. Further, the technical details about the execution modes operation is discussed in algorithm 2. If more than one fuzzy set satisfies the threshold condition, one among all the qualified is chosen as mode of execution. Once, the mode of execution is chosen, the algorithm 3 initiates its execution. The dynamicity of the DITLCS is governed by the algorithm 3. In order to make effective dynamic traffic light signal scheduling decisions, deep reinforcement learning agent is trained which tries to maximize the cumulative reward according to the action chosen. The deep reinforcement learning agent takes position and velocity of each vehicle in the vehicular system and then converts them into the matrix form [1].

$$P = \begin{vmatrix} P_0 \\ P_1 \\ P_2 \\ P_3 \end{vmatrix} \quad V = \begin{vmatrix} V_0 \\ V_1 \\ V_2 \\ V_3 \end{vmatrix}$$

where, P is the position matrix, and V is the velocity matrix. These inputs are consumed by the agent during its run and it returns its binary output in the form of 0 or 1. Zero (0) value implies that there is no need to change the traffic light phase i.e., if the current phase of the traffic light in south-north direction is green and when the agent returns 0, the traffic signal is left unchanged. On the other hand, if the agent returns 1, immediately the traffic light phase is switched to green to red or vice-versa. Once, the deep reinforcement learning agent returns 0, the next task is to find the duration

Algorithm 2 Find Mode

```
Input: qL, WT, T
Output: Returns mode of operation
Function findMode(qL, wT):
    mode = [fair, priority, emergency]
    time = T, queueThreshold = 10, lamda = [0.4,0.9],
    rule = 3, waitingThreshold = 20000 (milli seconds);
    if emergency vehicle is present then
        return mode[2]
    end if
    // Rules for fuzzification
    if (time == morning or time == evening then
        priority = priority + 1
    else
        fair = fair + 1
    end if
    if qL < queueThreshold then
        fair = fair + 1
    else
        priority = priority + 1
        fair = fair + 1
    end if
    if WT < waitingThreshold then
        fair = fair + 1
    else
        priority = priority + 1
    end if
    f = fair/rule, p = priority/rule // Defuzzification
    if fair > priority and fair <= lamda then
        return mode[0]
    end if
    if priority > fair and priority <= lamda then
        return mode[1]
    end if
    if random.uniform(0,1) < 0.5 then
        return mode[0]
    else
        return mode[1]
    end if
end
```

of the green signal. To do this, algorithm 3 (one of the module of DITLCS) is designed. In the proposed DITLCS, different vehicles are assigned to a different/same weights (depends on the modes of execution) that is based on type and priority of vehicles. The edge which is having the highest weight will have the relatively larger duration for the green signal.

$$W_N = \sum_{i=1}^{x} W_i, \quad W_S = \sum_{i=1}^{y} W_i, \quad W_E = \sum_{i=1}^{z} W_i, \quad W_W = \sum_{i=1}^{k} W_i, \tag{11}$$

where, W_N is the weight in north direction, W_S is the weight in south direction, W_E is the weight in east direction and W_W is the weight in west direction. x, y, z, k are the number of

Algorithm 3 Find Duration

```
Input: W_T, phase
Output: Returns duration of signal
Function findDuration(W_T, phase):
    N = 0, S = 0, E = 0, W = 0, T = 60;
    for v: vehicles in north direction do
        N = N + W_T[v] ;
    end for
    for v: vehicles in south direction do
        S = S + W_T[v] ;
    end for
    for v : vehicles in east direction do
        E = E + W_T[v] ;
    end for
    for v : vehicles in west direction do
        W = W + W_T[v] ;
    end for
    if phase == 0 then
        return S/(N+S+W+E) * T;
    end if
    if phase == 1 then
        return E/(N+S+W+E) * T;
    end if
    if phase == 2 then
        return N/(N+S+W+E) * T;
    end if
    if phase == 3 then
        return W/(N+S+W+E) * T;
    end if
end
```

vehicles in north, south, east and west direction respectively.

$$\begin{aligned} TotalWeight &= W_N + W_S + W_E + W_W, \\ &= \sum_{i=1}^{x} W_i + W_S = \sum_{i=1}^{y} W_i \\ &\quad + \sum_{i=1}^{z} W_i + W_W = \sum_{i=1}^{k} W_i \end{aligned} \tag{12}$$

VI. Simulation Study

The simulation is prepared on the real map of the Gwalior city in India utilizing a well-known open source simulator Simulation of Urban MObility (SUMO) [1]. The performance behaviour of the proposed DITLCS is analyzed on several performance metrics such as Average waiting time (AWT), Throughput (TP), Average Queue Length (AQL), Average Speed (AS) and Carbon emission (CO_2). Further, the DITLCS is compared with fuzzy based traffic light controllers i.e., FCTL [7], NFM [8], FIRNTL [9] and priority based traffic light controllers i.e., EVP-STC [11], ETLSA [12]. The simulated results evidenced about the effectiveness of the model.

A. Evaluation Methodology and Parameters

DITLCS is composed of mainly two components: deep reinforcement learning model and Fuzzy inference module. The deep reinforcement learning agent is dedicated to maximize the reward function which is directly linked to account waiting time and other parameters. The cumulative reward is measured in every cycle of the deep reinforcement learning module. Fuzzy inference module is specifically used to pick a mode of execution out of three (PM, FM and EM) by observing the current status of the vehicular traffic scenarios. In order to meet our requirements, the proposed DITLCS algorithms were implemented utilizing Python and, the simulation was prepared in SUMO. The deep reinforcement learning component of DITLCS was built with the help of Keras which is built on the top of Tensorflow [1]. The simulation was conducted on a major Indian city Gwalior, by making use of the realistic map which is converted into roads and traffic with the help of SUMO conversion tools such as netconvert.

The length of the vehicle depends on the type of vehicle (bus and truck length $= 7\ meter$, car $= 4.5\ meter$, bike length $= 1.5\ meter$) vehicular network. Other SUMO configuration parameters are as: limit-turn-speed value $=$ $5.5\ m/sec$, max lane speed $= 20.0\ m/sec$, truck and bus max speed $= 10\ m/sec$, car max speed $= 15\ m/sec$, other vehicles max speed $= 13.9 m/sec$, acceleration of truck and bus $= 0.8\ m/sec^2$, acceleration of car $= 2.5\ m/sec$, acceleration of other vehicles $= 1.8\ m/sec^2$, de-acceleration for all vehicles $= 4.0\ m/sec^2$, min gap-between any two vehicles $= 2.0\ meter$, sigma(driving-proficiency) $= 0.5$, detector frequency $= 60$. The arrival of the vehicles in the network follows Poisson distribution process where the arrival rate of vehicles are sampled from real-time traffic data, the arrival rate λ is between [0.4, 0.9]. Further, SUMO [1] provides an interface to mimic the real-time traffic flow, and it is also friendly to make use of the Python API i.e., TRACI (a part of sumo module) which is dedicated to extract the vehicle information from the intersection and to control the traffic light duration dynamically. Moreover, with the help of Python API (TRACI), Python and SUMO integration is very effective; this allows control of the SUMO traffic flow via written algorithms in Python. The duration of the yellow signal is fixed at 3 seconds which is the transition time for the signal change to green to red. The proposed deep reinforcement learning model is trained in iterations. The reward of our model is calculated at the end of each iteration based on the change in queue length and other parameters of the previous and current traffic flow. The goal of our model is to maximize the overall reward by dynamically changing the traffic signal duration. Deep network is simulated with Replay memory size ($M = 2000$), minimum batch size ($B = 64$), target network rate ($\alpha = 0.001$), discount factor ($\gamma = 0.99$), learning rate ($\epsilon_r = 0.0001$) and leaky ReLU ($\beta = 0.01$). For the traffic light signal implementation, the traffic load at each incoming road's intersections is dynamically generated following poission distribution process, hence the traffic density varies dynamically according to the traffic conditions.

B. Experimental Results

This sub-section presents the details of the experimental results of the proposed DITLCS along the comparison to the other state of the art algorithms on several parameters.

1) Average Waiting Time: Average Waiting Time (AWT) can be obtained by measuring the time interval between a vehicle arrival time and the time at which the the parameters are measured as shown in Eq. 13.

$$AWT = 1/N * \sum_{i=1}^{N} CT - AT_i \tag{13}$$

where $N =$ number of vehicles in the network and CT is the current time and AT_i is the arrival time of the vehicle at the intersection.

2) Throughput: Throughput (T_p) is the number of vehicles passing through an intersection per cycle i.e., when all the traffic light at an intersection have turned green once. (T_p) across all the intersection is measured by Eq. 14.

$$T_p = \sum_{i=1}^{N} qL_i \tag{14}$$

where, N = number of intersection and qL_i is the queue length in the i^{th} intersection.

3) Average Queue Length: Average Queue Length (AQL) is the average number of vehicle waiting at the traffic intersection. The AQL across the intersections can be calculated by Eq. 15.

$$AQL = 1/N * \sum_{i=1}^{N} v_i \tag{15}$$

where N is the number of traffic intersection and v_i is the number of vehicle waiting at i^{th} intersection.

4) Average Speed: It is the average speed of all the vehicles passing through the intersections in the network (Eq. 16).

$$AS = 1/N * \sum_{i=1}^{N} s_i \tag{16}$$

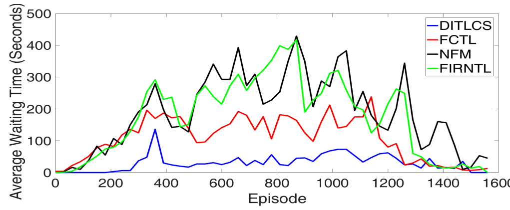
where, N is the number of vehicles in the network, and $s_i (= distance_i \div time_i)$ is the speed of the i^{th} vehicle.

5) Carbon Emissions: It is defined by the Eq. 17.

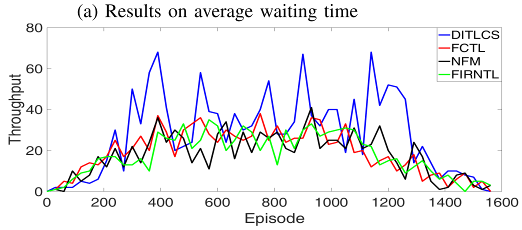
$$EmissionCO2 = Liters/Kilometers * MassCO2/Liters * KilometersTravelled \tag{17}$$

where Liters/Kilometer is the vehicle-technology-related parameter used to describe the fuel economy of a vehicle, mass CO2/Liter is the fuel-related parameter for estimating the amount of CO2 emission produced by each unit of fuel, and kilometers traveled is the traffic activity parameter for representing the travel distance of total vehicles.

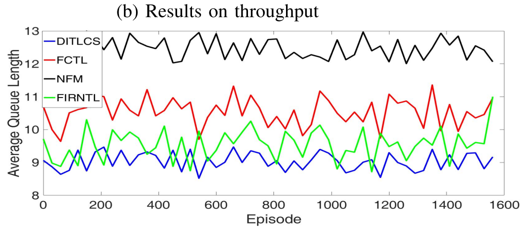
The Fig. 6 presents the comparative results of the DITLCS on various urban traffic simulation parameters. The X-axis indicates the number of simulation steps used by the model during the run. The Y-axis indicates the respective performance metric. Results in Figs. 6(a) to 6(e) are for the combined heterogeneous traffic, and Figs. 6(f) to 6(g) present the results for the priority vehicles. Fig. 6(a) shows that the proposed DITLCS is able to reduce the average waiting time significantly over the compared state of the art algorithms. Through the series of simulations, it is observed that DITLCS learns the environment effectively, and it adjusts traffic light phases as per the traffic load and heterogeneity



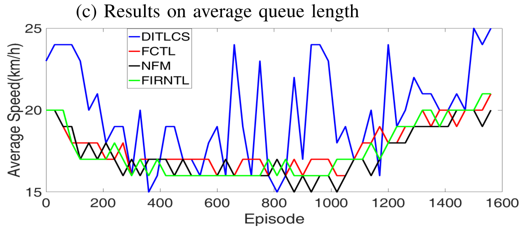
(a) Results on average waiting time



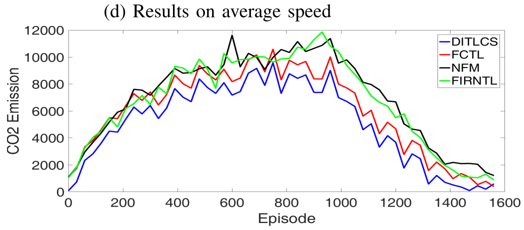
(b) Results on throughput



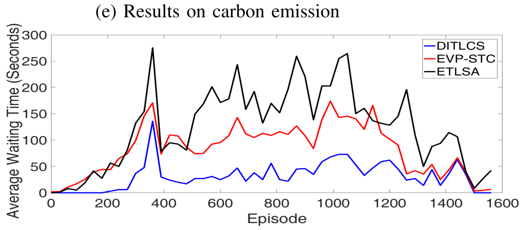
(c) Results on average queue length



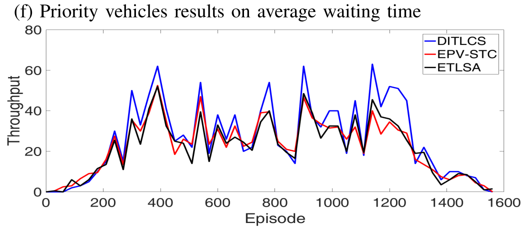
(d) Results on average speed



(e) Results on carbon emission



(f) Priority vehicles results on average waiting time



(g) Priority vehicles results on average waiting time

Fig. 6. Comparative results.

among the traffic. This results in reduced waiting time. The Fig. 6(b) and Fig. 6(c) present the comparative results of various models on throughput and queue length respectively. From the results, it is analysed that the DITLCS increases the throughput up to 68 and decreases the AQL down to 8.5 which is the noticeable improvement in the performance. The deep reinforcement learning model plays a significant role in improving the performance of DITLCS for throughput and AQL. It dynamically adjusted the traffic signal phase according to the load of vehicle traffic at each intersection, hence, more number of vehicles can be passed through the intersection which ultimately reduces the queue length at the intersection and increases the throughput. Additionally, this leads to increase in average vehicle speed. Therefore, in order to analyse the traffic smoothness, Fig 6(d) presents the average speed of the vehicles. From the results (Fig 6(d)), it is analysed that DITLCS candidly achieves highest speed among all the comparative state of the art algorithms. Fig. 6(e) shows the comparison of various models on the parameter carbon emission. The proposed DITLCS reduced 14.21%, 11.52% and 6.45% of CO_2 emission as compared to the existing NFM [6], FIRNTL [7] and FCTL[8] respectively. Through the set of experiment, it is analysed that the combination of fuzzy inference system and Deep reinforcement learning played a very significant role in minimizing the waiting time by accounting the heterogeneity of the vehicles. As obvious, carbon emission depends on the vehicle's delay in the congested traffic.

In order to verify the effectiveness of DITLCS on priority/emergency vehicles, a set of experiments are done in which DITLCS operates only in PM/EM, and results are collected for priority/emergency vehicles only and compared with two priority/emergency dedicated models i.e., EVP-STC[11] and ETLSA [12]. Results are obtained to analyse the performance in terms of average waiting time (Fig. 6 (f)) and throughput (Fig. 6 (g)) as the major concerns of this work. From the results, it is observed that DITLCS significantly reduces AWT for the priority vehicles as well; this also improves the throughput at the traffic light intersection in comparison to EVP-STC [11] and ETLSA [12]. With the set of experiments, it is analysed that the fuzzy inference system which is switching the system's execution modes accounting the heterogeneity of the vehicles, contributes towards minimization of the average waiting time of high priority vehicles such as ambulance and fire-brigade. Since, the experimental setup consists of heterogeneous vehicles including high priority vehicles and emergency vehicles. This way, DITLCS minimizes the average waiting time to a greater extend. In brief the experimental results as shown in Fig. 6 are evidenced for the effectiveness of the proposed DITLCS in terms of AWT, T_p, AQL, AS and CO_2. Hence, DITLCS can be a candidate solution for the dynamic traffic light control system in developing countries.

VII. Conclusion

This work explored the requirement of the heterogeneity aware, Dynamic and Intelligent Traffic Light Control System (DITLCS) on vehicular traffic network. Heterogeneity among the vehicles in terms of emergency and priority vehicles is taken into account. The proposed DITLCS acts in three phases namely Fair Mode (FM), Priority Mode (PM) and Emergency Mode (EM) where all the vehicles are considered with equal priority, vehicles of different categories are given different level of priorities and emergency vehicles are given at most priority. The DITLCS is the combination of deep reinforcement learning and fuzzy inference system. Deep learning reinforcement is dedicated to dynamically generate the phase of the traffic lights, and fuzzy inference system opts the mode of the execution based on the traffic behaviour, heterogeneity and dynamism in which deep reinforcement learning model acts. To verify the effectiveness of the DITLCS, a realistic simulation is developed utilizing an open source simulator tool i.e., Simulation of Urban MObility (SUMO) [1] on Gwalior city of India. Results of DITLCS are compared with several Fuzzy, neural network and priority based state of the art algorithms which proved the effectiveness and high-efficiency of the DITLCS.

Acknowledgment

Dr. Neetesh Kumar would like to put his sincere thanks to the all Co-PIs of the project, ABV-IIITM Gwalior, India, and DST, IDCPS Division, India, for kind of their support.

References

[1] X. Du and X. Liang, "Deep reinforcement learning for traffic light control in vehicular Networks," *IEEE Trans. Veh. Technol.*, early access, Mar. 2018, doi: 10.1109/TVT.2018.2890726.

[2] A. Abadi, T. Rajabioun, and P. A. Ioannou, "Traffic flow prediction for road transportation networks with limited traffic data," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 2, pp. 653–662, Apr. 2015.

[3] C. Quek, M. Pasquier, and B. L. B. Lim, "POP-TRAFFIC: A novel fuzzy neural approach to road traffic analysis and prediction," *IEEE Trans. Intell. Transp. Syst.*, vol. 7, no. 2, pp. 133–146, Jun. 2006.

[4] Y. Zhang, G. Zhang, R. Fierro, and Y. Yang, "Force-driven traffic simulation for a future connected autonomous vehicle-enabled smart transportation system," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 7, pp. 2221–2233, Jul. 2018.

[5] F. Belletti, D. Haziza, G. Gomes, and A. M. Bayen, "Expert level control of ramp metering based on multi-task deep reinforcement learning," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 4, pp. 1198–1207, Apr. 2018.

[6] N. Bekiaris-Liberis, C. Roncoli, and M. Papageorgiou, "Highway traffic state estimation with mixed connected and conventional vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 12, pp. 3484–3497, Dec. 2016.

[7] M. Kafash, M. B. Menhaj, M. J. M. Sharif, and A. Maleki, "Designing fuzzy controller for traffic lights to reduce the length of queues in according to minimize extension of green light time and reduce waiting time," in *Proc. 13th Iranian Conf. Fuzzy Syst. (IFSC)*, Aug. 2013, pp. 1–6.

[8] E. Azimirad, N. Pariz, and M. B. N. Sistani, "A novel fuzzy model and control of single intersection at urban traffic network," *IEEE Syst. J.*, vol. 4, no. 1, pp. 107–111, Mar. 2010.

[9] A. Mir and A. Hassan, "Fuzzy inference rule based neural traffic light controller," in *Proc. IEEE Int. Conf. Mechatronics Automat. (ICMA)*, Changchun, China, Aug. 2018, pp. 816–820.

[10] P. Kachroo and S. Sastry, "Travel time dynamics for intelligent transportation systems: Theory and applications," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 2, pp. 385–394, Feb. 2016.

[11] A. Khan, F. Ullah, Z. Kaleem, S. U. Rahman, H. Anwar, and Y.-Z. Cho, "EVP-STC: Emergency vehicle priority and self-organising traffic control at intersections using Internet-of-things platform," *IEEE Access*, vol. 6, pp. 68242–68254, 2018, doi: 10.1109/ACCESS.2018.2879644.

[12] M. B. Younes and A. Boukerche, "An efficient dynamic traffic light scheduling algorithm considering emergency vehicles for intelligent transportation systems," *Wireless Netw.*, vol. 24, pp. 2451–2463, Mar. 2018.

[13] Hua Wei, Guanjie Zheng, "Intellilight: A reinforcement learning approach for intelligent traffic light control," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, London, U.K., Aug. 2018, pp. 2496–2505.

[14] W. Farooq, M. A. Khan, and S. Rehman, "A cluster based multicast routing protocol for autonomous unmanned military vehicles (AUMVs) communication in VANET," in *Proc. Int. Conf. Comput., Electron. Electr. Eng. (ICE Cube)*, Apr. 2016, pp. 42–48.

[15] S. Sajad Mousavi, M. Schukat, and E. Howley, "Traffic light control using deep policy-gradient and value-function based reinforcement learning," 2017, *arXiv:1704.08883*. [Online]. Available: http://arxiv.org/abs/1704.08883

[16] W. Genders and S. Razavi, "Using a deep reinforcement learning agent for traffic signal control," 2016, *arXiv:1611.01142*. [Online]. Available: http://arxiv.org/abs/1611.01142

[17] N. Casas, "Deep deterministic policy gradient for urban traffic light control," 2017, *arXiv:1703.09035*. [Online]. Available: http://arxiv.org/abs/1703.09035

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

[19] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, vol. 1, no. 1. Cambridge, MA, USA: MIT Press, Mar. 1998.

[20] D. Silver *et al.*, "Mastering the game of go with deep neural networks and tree search," *Nature*, vol. 529, no. 7587, pp. 484–489, Jan. 2016.

[21] D. Silver *et al.*, "Mastering the game of go without human knowledge," *Nature*, vol. 550, no. 7676, pp. 354–359, Oct. 2017.

[22] M. Abdoos, N. Mozayani, and A. L. C. Bazzan, "Holonic multi-agent system for traffic signals control," *Eng. Appl. Artif. Intell.*, vol. 26, nos. 5–6, pp. 1575–1587, May 2013.

[23] H. Hartenstein and L. Laberteaux, "A tutorial survey on vehicular ad hoc networks," *IEEE Commun. Mag.*, vol. 46, no. 6, pp. 164–171, Jun. 2008.

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

[25] N. Kumar and D. P. Vidyarthi, "A green routing algorithm for IoT-enabled software defined wireless sensor network," *IEEE Sensors J.*, vol. 18, no. 22, pp. 9449–9460, Nov. 2018.

[26] J. Zhang, F.-Y. Wang, K. Wang, W.-H. Lin, X. Xu, and C. Chen, "Data-driven intelligent transportation systems: A survey," *IEEE Trans. Intell. Transp. Syst.*, vol. 12, no. 4, pp. 1624–1639, Dec. 2011.

[27] S. Bouchelaghem and M. Omar, "Reliable and secure distributed smart road pricing system for smart cities," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 5, pp. 1592–1603, May 2019.

[28] S. Chiu, "Adaptive traffic signal control using fuzzy logic," in *Proc. Intell. Vehicles Symp.*, Jun. 1993, pp. 98–107.

Neetesh Kumar (Member, IEEE) received the M.Tech. and Ph.D. degrees from Jawaharlal Nehru University, New Delhi. He is currently a Faculty Member with the Atal Bihari Vajpayee Indian Institute of Information Technology and Management (IIIT) Gwalior, India. He has published several research publications in worlds top tier publishers, such as IEEE journals and transactions, Elsevier, and springer journals. He is also acting as a Lead PI of several sponsored projects in the domain of Intelligent Transportation System (ITS). Broadly, his research interests include algorithm design, high performance computing, and ITS.

Syed Shameerur Rahman is currently pursuing the Integrated Master's degree with the Atal Bihari Vajpayee Indian Institute of Information Technology and Management Gwalior, Gwalior, India. His research interests include algorithm design, the IoT, and vehicular ad hoc networks.

Navin Dhakad is currently pursuing the bachelor's degree with the Indian Institute of Information Technology Pune, Pune, India. His research interests include algorithms, and intelligent transportation system (ITS).