International Journal of Computing and Digital Systems
ISSN (2210-142X)
Int. J. Com. Dig. Sys. 15, No.1 (Mar-24)

http://dx.doi.org/10.12785/ijcds/150171

Dynamic Traffic Management Using Reinforcement Learning

Aryaan Shaikh[1], Babasaheb Bhalekar[2] and Pravin Futane[3]

[1,2,3]*Department of Information Technology, Vishwakarma Institute of Information Technology, Pune, India*

Received on 25 Apr. 2023, Revised on 29 Jan. 2024, Accepted on 7 Feb. 2024, Published on 1 Mar. 2024

Abstract: Traffic congestion has become a major problem in this rapidly growing world. Everyone operating a vehicle, as well as the traffic police in charge of managing the traffic, finds it difficult to become stuck in heavy traffic. For this a set, predetermined timing for traffic flow for each direction at the junction is utilized by traditional traffic light controllers. However, the concept of a fixed time traffic signal controller does not work well in places with uneven traffic. A dynamic traffic control system is therefore required, which regulates the traffic signals in accordance with the volume of traffic. This paper proposes a model that uses reinforcement learning (RL) along with deep neural networks (DNN) to manage discretions (signal status) for an environment with the help of Simulation of Urban MObility (SUMO). A simulation of real-world environment consisting a network of Four-way crossroad junction that contains 4 arriving lanes and 4 exiting lanes is used to train the agent. The main objective of this research study is to construct a model that can independently determine the best course of action and aims to provide better traffic management that will decrease the average waiting time, cause lower congestion, and provide a smooth flow of traffic.

Keywords: Deep Neural Networks, Deep Q-learning, Dynamic Traffic Management, Reinforcement Learning, Simulation of Urban MObility, Traffic Patterns

1. INTRODUCTION

Road traffic is currently the most common issue faced by cities all over the world. As the population is growing, the need for transportation is also rapidly increasing. Widespread everyday usage of vehicles put a lot of pressure on a city's transportation infrastructure, resulting in long traffic jams. Large red-light delays also play a significant part in worsening the vehicle congestion. This delay problem is usually caused by using traditional pre-timed traffic regulation systems.

Metropolitan cities in India particularly cities like Pune and Bengaluru, which ranked second and sixth respectively amongst 390 cities [1] in the TomTom traffic index, estimated for having the most average travelling time. Major reason for having such high average travelling time can be due to low number of flyovers or an inefficient traffic control management system. Current traffic management systems have a fixed pre-determined timer for traffic flow for each lane, which is not sensitive to actual traffic. Therefore, it could be useful to have a dynamic traffic control management system that prioritizes traffic lanes based on the existing state of congestion with the usage of technology that have learnt to accurately assess traffic density. This can help cities to become more efficient, grow economically, and people's daily lives would be made easier.

Nowadays, traffic control system uses traffic lights with pre-set timers that run on a pre-set schedule [2] and show green lights to each approaching vehicle for the same amount of time throughout each cycle, regardless of the flow of traffic. This might work best in locations with even traffic density, but in areas with uneven traffic volume, the sequence is less useful. Investigations now underway suggest manually constructed restrictions based on actual traffic data [3]. Whatever the case, these rules remain pre-defined and cannot be effectively balanced with intermittent traffic. Therefore, using an intelligent traffic signal control system can be a way to effectively manage traffic congestion. Hence, the primary goal of this research is to develop a model capable of autonomously determining optimal courses of action for enhancing traffic management by reducing average waiting times, minimizing congestion, and ensuring a seamless flow of traffic.

In recent years, many nations have focussed on creating advanced traffic surveillance systems in order to tackle the issues brought on by traffic congestion. Such systems' key objective is to examine the current traffic condition by gathering information on vehicle density, vehicle type, speed, and vehicle count. Systems for monitoring traffic can make use of radar sensors, video cameras, loop detectors and ultrasonic detectors. Video cameras are being utilised to compile and evaluate traffic data as a result of recent

E-mail: aryaan.21910912@viit.ac.in, babasaheb.21910926@viit.ac.in, pravin.futane@viit.ac.in

and ongoing breakthroughs in the disciplines of image processing, computer vision and deep learning. Regulating highway road traffic is currently a challenging undertaking due to the clearly expanding population.

There are many other methods like use of sensors, Machine Learning (ML), Internet of Things (IoT), Artificial Intelligence (AI), Neural Network (NN), Deep Learning and Computer Vision that can be used for managing traffic dynamically. Techniques like ML, AI, Deep Learning or computer vision depends on identifying and detecting vehicles for traffic flow. But these techniques become inefficient as they become unable to detect traffics effectively due to large number of uneven dimensions vehicles available on the road. Moreover, use of sensors and IoT can be very costly. Therefore, RL model is chosen over other methods.

The primary rule of any classic traffic control system is to alternate between green and red lights for a set period of time. When the traffic network is small, such traffic control strategies perform brilliantly. The dynamic traffic management system, in contrast, responds in a way that reduces traffic in metropolitan networks serving metropolitan areas by taking into account ongoing traffic conditions and RL computations to get better over time. The proposed system will examine vehicles at a four-way intersection for incoming traffic density and take the best possible action to reduce it. The proposed estimator utilised learns with time, therefore it's possible that the system's foundational periods won't produce flawless results for the traffic that has been identified. The aim to develop a framework for intelligently managing and directing traffic using RL, and to achieve smooth vehicle traffic is to reduce environmental problems including increased air pollution, fuel waste, and accident risk.

RL is a method of learning the best course of action performed by an agent through a process of trial and error that includes observing the environment, selecting an action in accordance with the present conditions. For every correct action the agent will be receiving gains/rewards from the environment. The most ideal strategy will be chosen from the one that maximises the anticipated long-term benefit. People have been using the RL process to gradually adjust traffic lights in accordance with steady traffic [4]. Due to two major challenges, traditional RL is challenging to implement. Firstly, how to describe condition and how to show the relationship between condition and choice. Hence, a RL algorithm is required that naturally extracts from raw real-time traffic data all features (machine-created highlights) useful for adaptable traffic signal management as well as learns the best traffic signal control arrangement.

2. LITERATURE SURVEY

Over the past 3 decades, study on AI based traffic management systems, particularly using RL have been carried out. In order to demonstrate a traffic management system that successfully improved efficiency of a junction, Sadayoshi et al. suggested a diffused RL approach via Genetic Algorithm [5] . But it was unable to be put into use because of the constraints of computational capacity. Initial proposal for an actual dynamic traffic manager was suggested by Rob Pringle et al. which utilized a Q estimator that is trained for regulating the signal system automatically through a Cerebellar Model Articulation Controller (CMAC) [6].

In the ordinary RL based method the traffic light durations are divided into same size time intervals and are increased only by multiples of these fixed time intervals. This is not an efficient method. So Ilhan Tunc suggested a Fuzzy logic and deep Q learning based Traffic management system. In the method proposed, the phase sequence is determined by using the Deep Q-Learning algorithm, and the green light duration is determined according to the traffic intersection state. Fuzzy logic controller divides the Lane into cells. There are two input values for Fuzzy Logic Control. One is GP of road in green phase and RP of road in red phase [7]. The output is the time duration for the next Lane.

To get better results from the RL agent Salah Bouktif suggested to make use of double deep Q-Network (DDQN) along with prioritized experience replay (PER) for the agent architecture [8]. The idea is to declare both state and reward in a consistent and straightforward manner. This makes system more reliable to be setup in real scenario. The limitation of this system is the duration of the signal phase. The duration of the green phase does indeed impact the performance of the traffic signals.

Fang Fei et al. employed a Multi-Agent Reinforcement Learning (MARL) approach for modeling Traffic Signal Control (TSC) as a Markov game [9]. It highlighted the coordination among independent traffic light controllers at intersections for global traffic efficiency. Acknowledging deployment challenges, the study aims to address them and emphasizes transitioning from simulation (using SUMO) to real-world application. Qian Sun's also proposed a similar multi-Agent collaborative hierarchical RL framework for urban traffic called NavTL [10], employing a two-level hierarchy with dynamic graph neural networks and deep Q-networks. It innovates hierarchical RL, improves feudal RL, and integrates graph neural networks for effective vehicle-signal cooperation and signal-signal coordination.

Min Chee et al. adapted a RL model to simulate a fuzzy based neural system (FBNS) for integrating an arrangement to attain a flexible control for managing traffic in a broad region [11]. While Ella et al. suggested RL for parameterization evaluation for a FBNS traffic management for an individual junction [12]. The above techniques have their basis on RL. However, the main objective is to tune the FBNS system's characteristics. It is important to stress the viability and benefits of using a framework temporal variation RL based algorithm [13] for controlling traffic signals in order to make the model more responsive to the real-world traffic. The fundamental disadvantage of such a technique is that it only allows traffic to flow in one direction, not both, at four-way crossroads.

The key property for any adaptive traffic management system is its capability to maximize the dynamic signalling strategy [14] in accordance with its goal. This makes it compulsory for any junction approaches to be monitored in a manner that is comparable to the actual traffic signal. Adaptive traffic controlling systems are considered different as they take into account the real-world scenario of traffic on any junction or crossroad and predicts accordingly the optimal ways to smoothly and quickly clear the traffic. This enables junctions to subsist variations of traffic and improve the junction's working capability. Liyan Shi et al. employed one such adaptive traffic control system using intersection clustering and reinforcement learning for reducing learning time and addressing data observation limitations [15]. The methodology models traffic pressure, introducing lane pressure to minimize intersection pressure during signal control. The paper emphasizes the importance of traffic signal phase design with a continuous action control problem using a Gaussian distribution for optimal strategy. Kranti Shingate et al. also proposed Adaptive Traffic Signal Control with Dedicated Short-Range Communication and RL for real-time optimization [16]. The paper highlighted vehicle detection, machine-crafted features, and model-less RL for responsiveness. Multi-agent systems and Markov Decision Process models show potential to improve traffic signal control efficiency, representing ongoing efforts to address congestion intelligently.

In order to design a better traffic light layout, Zhang et al. designed a RL based model for managing the current traffic scenario more efficiently. Dedicated short-range communication is used to identify vehicles in partially visible conditions [17]. The agent performs a one-sided manoeuvre in favour of identified vehicles if there is a significant difference between undetected and detected vehicles at an intersection. So, this technology performs better for automobiles equipped with remote communication than it does for undetected vehicles. The experience repeats and aims network [18] utilised during the agent's training phase are a crucial element that significantly enhances algorithm stability. Henglong et al. [19] presented an approach of calculating and tallying the amount of traffic congestion. An evaluation was carried out utilising a network of neural networks and the Time Spatial Imagery (TSI) technique. Even though this technology measures vehicles accurately, but it is unable to control changing traffic lanes.

There are a lot of methods which were introduced that used vehicle data extracted from GPS (Global Position System) [20] for predicting, extracting and detecting vehicle states for traffic management. These methods were based on algorithms like Genetic Algorithm, Support Vector Machine, fuzzy logic [21]. However, results indicated that traffic management can be improved using partial recognition.

For achieving an effective traffic flow management, the multiple agent strategy for traffic controlling was used by Arel et al. that combined RL with a multi-agent system [22]. In this, an outbound agent and a central agent are both employed as sorts of agents. LQF (Linear Queue First) is the technique used by the outbound agents to schedule traffic lights, and the centralized agent trains a value function i.e., Q-learning that is influenced by local and neighbouring agent's traffic conditions. The Q Learning multi-agent method outperforms the LQF task scheduling at low arrival rates.

Anum Mushtaq et al. proposed a two-phase Deep Reinforcement Learning (DRL) approach for traffic management. Using Deep Q-Learning, the system optimizes traffic flow by dynamically adjusting signals based on real-time information [23]. The first phase focuses on intelligent traffic light optimization, while the second phase involves DRL-guided vehicle rerouting to reduce congestion. The paper explores credit assignment in reinforcement learning, defining states as vehicle speed and position, actions as traffic light operations, and cumulative waiting time as the reward.

The research study conducted by Stern, R. E. et al. uses the combined-autonomy ring road [24] scenario as an example to compare the relative effectiveness of explicit and learned regulators [25]. Model-based techniques to RL, for instance, Dynamic Programming, manually designed controllers, optimal control is supported by model-free RL methods in the computational aspects for framework. The complexity of these techniques, however, varies greatly, with some requiring prohibitive computational costs.

The proposed study by Mishra, K. D et al. by using detection and tracking of vehicles, monitoring, and counting was built using real-time video data. The system [26] was built using the OpenCV computer vision library and the YOLOv3 object detection method based on CNN. Also, traffic lights were controlled using this information based on the volume of vehicles. Jiasong et al. used DL techniques to demonstrate a framework for assessing volume of traffic [27]. Data from footages were obtained using Unmanned Aerial Vehicle. They proposed a Deep Vehicle Counting Framework for counting different vehicles according to their vehicle types. The only shortcomings for these systems were that it involves a lot of computation and processing time.

3. METHODOLOGY

The research involves reinforcement learning over supervised or unsupervised machine learning as it is difficult to collect a relevant and quality dataset. Moreover, an agent will perform much better than a pre labelled or unlabeled dataset in real world traffic scenarios as it will learn from its mistakes. The model proposed is based on reinforcement learning and does not require any pre labelled dataset to be trained on, instead it learns by experience. Deep Q-Learning also uses past experiences and results saved in the memory that can be used in taking further actions. These tools and techniques are further discussed in detail below.

A. *Proposed Framework*

The two main components of the system are the agent and the environment as shown in Fig. 1. The entity that acts on the environment is known as agent and is responsible for taking actions to obtain optimum results. The distribution of vehicles at a crossroads/junction is the known as the environment. Traffic Controller Interface (TraCI) package in SUMO is used for providing the environment. SUMO environment generates an unspecified number of cars from an irregular source distribution, which in turn provides the agent with a contribution as a state. Time provided for each green signal will be at least of 5s in SUMO. This least time is provided to avoid accidents and provide a least possible time for vehicles to flow at a junction as sudden traffic density change can cause system to fluctuate green signal instantly causing confusion.

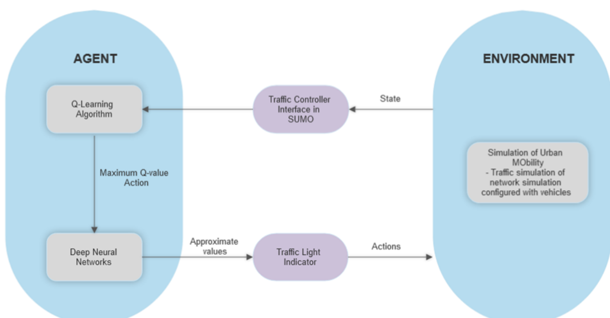

Figure 1. Architecture of proposed system

The Q-learning algorithm uses this state and selects the most appropriate action with the greatest Q-value. Traffic signals act accordingly to change the environment. DNN and Adam optimizer are used to provide approximations of values to improve results. In the start the agent will not be efficient enough, but after many trials and errors it will be able to control the flow of traffic with much more effectiveness. The key objective of this system is to provide a quicker and more effective way for vehicles to move across traffic signals based on the conditions of traffic to avoid long waiting time of vehicles.

B. *Reinforcement learning*

In real world, humans try to reach their goals by performing certain steps. Every step either takes us towards or away from our goal. Similarly, in reinforcement learning each action performed has some rewards or penalty associated with it. In this process, humans search for different ways in order to achieve their goal. There can be many ways to reach the goal with varying results based on the actions performed. There can be many outputs as there are many solutions to a problem.

An agent in RL can be considered parallel to a new born baby in the real world. If a new born is rewarded for doing a task, it will affect his behaviour in a positive way. Similarly, Rewards can help improve the frequency and quality of the behaviour of an agent. This will help agent maximise performance for better results.

Contrary to conventional ML algorithms, RL is the field of AI in which machine learns on its own by experiencing different actions in an environment. It is just about choosing the right course of actions in according to earn optimum rewards. RL is considered more superior to many methods for traffic flow management as it offers more efficient and cost-effective solutions. Moreover, it is not necessary to define every environmental variable as the learning is based on system's success.

C. *Simulation of Urban MObility (SUMO)*

Characterizing the states, actions, rewards, learning mechanism and environment is a very crucial part in the design process for this system that uses a RL algorithm. In SUMO, the real-world environment is expressed by a Four-way crossroad junction that contains 4 arriving lanes and 4 exiting lanes as shown in Fig. 2.

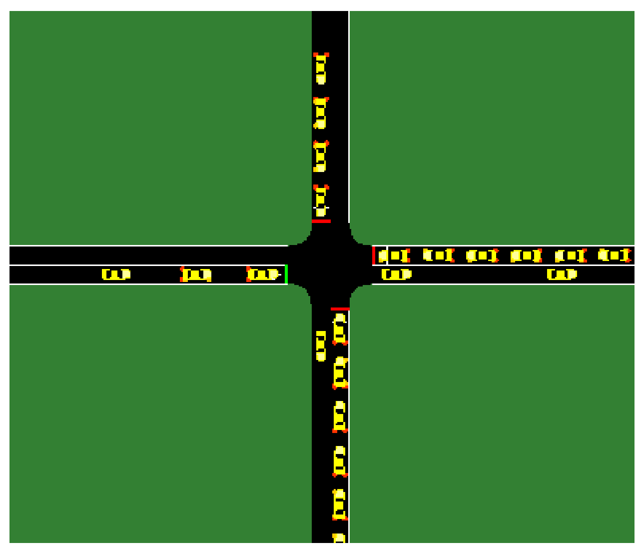

Figure 2. SUMO environment

Some of the Parameters in Sumo are:

1. Queue Length: The queue length is the length of a line of vehicles waiting at a specific point, such as a traffic signal or an intersection. It indicates the number of vehicles that are delayed and waiting for their turn to proceed.

2. Vehicle Speed: The vehicle speed represents the current speed of individual vehicles moving within the simulation. Each vehicle has allocated speed randomly by the configuration file generated using randomTrips.py file.

3. Waiting Time: Waiting time is the time a vehicle spends waiting at intersections or traffic signals before being al-

lowed to proceed. It is related to queue length and traffic signal timings.

In SUMO, there are four traffic lights for every junction and each of them is denoted by a coloured strip situated on the right sideway of the lane as shown in Fig. 3 for every vehicle coming towards the junction. This coloured strip indicates the current status of the traffic signal for that lane. For instance, it will display red light for lanes that are still closed and green for the lanes that are free to go. Five seconds in the real world is equivalent to 1 millisecond in the SUMO Simulation.

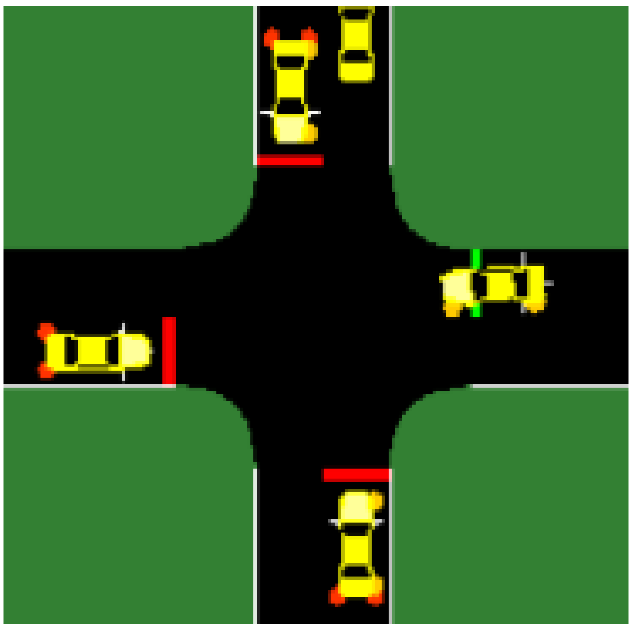

Figure 3. Traffic Lights at a junction in SUMO

D. Deep Q-Learning

Q in Q-learning symbolizes quality learning. Quality learning refers to effectiveness of a particular action at boosting the upcoming rewards. Q-learning algorithm is simple yet incredibly effective because the agent is enabled to accurately choose exactly which actions to carry out. The agent learns by Deep Q-Learning (DQL) strategy. DQL combines DNN with q-learning. In this strategy, a NN is used to simulate Q-value function. Policy is not necessary as the agent learns actions that go against the current policy. Q-learning focuses on discovering and learning a policy that gains maximised rewards. For every input i.e., the state, a Q-value is associated with every possible action that can be performed.

In Deep Q-learning all prior results and experiences are saved by the memory that can be used in taking further actions. The action to be performed next is usually decided by highest value in Q-table of the network. In the start, using DQL will not be able to find the optimal Q-value effectively. But, as the number of results and experiences increase, DQL will more effectively judge the closest state-action pair closest to the best Q-value.

In Q-learning agent learn a Q-value. Q-value can be represented as Q(st , at) which is derived from the Bellman Equation. Observed state is denoted by st and action denoted by at. Discrete time index is represented by t and the outcome of this expected collective future reward is given by:

$$Q(s_t, a_t) = r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \gamma^3 r_{t+3} + \ldots (1)$$

$$Q(s_t, a_t) = Q(s_t, a_t) + \alpha(r_{t+1} + \gamma \max Q(s_{t+1}, a_t) - Q(s_t, a_t)) \ldots (2)$$

In the above equation, (st+1 , at) denotes Q-value of the upcoming stage and rt+1 denotes the rewards obtained after completing an action. Reward at each step is represented by rt in Eq. (1). The value of future rewards is determined by the discount factor represented by γ. The value of γ varies from 1 to 0. When γ is 1 the agent wants to seek long term rewards, while if γ is 0 it only considers immediate benefits. After every step the Q function is updated by updating the Q-value as shown in Eq. (2). Q-value for action-state pair is enhanced by errors caused, which is levelled by learning rate denoted by α. If learning rate is factored 1, it will only be taking into account the most recent knowledge, while if it factors 0, it will prevent the agent to learn anything. Learning rate, α determines how much newly acquired data supersedes previously known data. DNN approximates Q function when the state and action space is complex. In this case, original value is used rather than the updated Q value.

E. Training Agent

The training phase of how an agent is trained to control the flow of traffic is represented in Fig. 4.1.

Model Parameters:

- $\gamma = 0.99$
- $\epsilon = 0.0$
- max_memory_size = 10^6
- $\epsilon\_dec = 5 \times 10^{-4}$
- $\epsilon\_end = 0.05$
- Learning Rate = 0.1
- Input Dimension = 4
- Batch Size = 1024
- Output Size = 4

The model agent is implemented using the PyTorch machine learning framework. The steps for this process are:

1) Initializing replay memory capacity (Taken 10^6 standard).
2) Initializing the neural network with random weights.
3) Taking the same neural network and making it the target network.
4) For each SUMO episode:
    a) Initializing the initial state.
    b) For every time step:
        i) Based on the ϵ value, choose an action from exploitation or exploration.
        ii) Carry out the chosen action in SUMO.
        iii) Examine the next state and the reward.
        iv) Save experience in replay memory (State, action, Next_state, reward, done).
        v) Take a random sample of the replay memory.
        vi) Pre-process states from the batch.
        vii) Pass a group of previously processed states to the neural network (policy network).
        viii) Calculate the difference in Q-values between the output and the target.
        ix) The next state needs a pass to the target network.
        x) Gradient descent modifies the policy network's weights to reduce loss.
5) Weights in the target network are modified to match those in the policy network after x time steps.
6) Call Traci to stop the SUMO episode.
7) Store the training result plot (Average waiting time vs episode) in the Result directory.

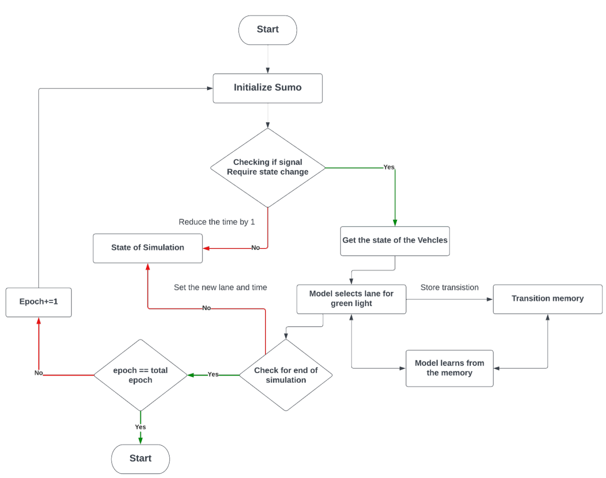

Figure 4. Agent Training process flow

F. Testing Agent

Testing phase is done to evaluate the model performance on generalize condition. The testing phase is shown in Fig 4.2.

1) Initializing the SUMO environment.
2) Loading the trained agent.
3) For each SUMO simulation step: a.
    a) Scanning the state of the SUMO environment at each step.
    b) The agent chooses the action pair.
    c) Carrying out the chosen action in SUMO.
4) Calling the 'Traci.close()' method of the Traci API.

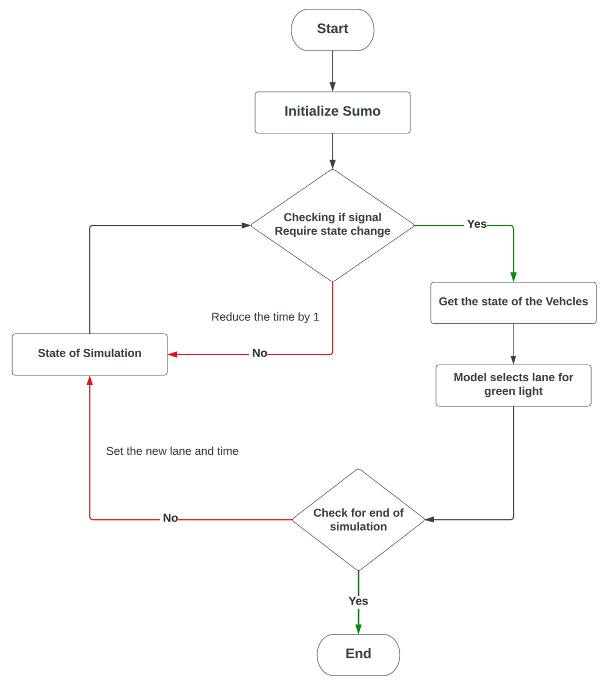

Figure 5. Agent Testing process flow

4. RESULTS

If some lane has heavy traffic or have a significantly high number of vehicles then that specific lane will be prioritized by the agent to avoid traffic congestions. For such cases where there is uneven density of vehicles, the system will not allot equal time (green signal) to all the terminals of the traffic signal. The agent used in the system will automatically prioritise more time to the busy lane as compared to other less traffic lanes for smooth flow of traffic and to avoid jams. Only the terminal with heavy traffic will be given priority and will signal green light. All other terminals will remain red until the vehicle count at the previous terminal becomes less. When the volume of traffic becomes less, the green light will shift to other lanes with the greatest number of vehicles.

The major difference between existing approach [12] and proposed approach is the usage of DQL. In proposed method if the state-action pair complexity is too high, it uses approximation to judge the optimal Q-values using all prior results and experiences that are saved in the memory. The action to be performed next is usually decided by highest value in Q-table of the network. This increases the overall accuracy of the system.

Five cases have been shown to evaluate the proposed system with different traffic conditions. In these cases, the system is given different situations based on varying traffic and lanes. This shows how the system will react on such situations for obtaining the best possible solutions to manage the traffic.

A. *Case 1: More traffic on one of the Lane*

When the traffic on south lane is more with respect to the other lanes. The model allocates green signal to south lane having a greater number of vehicles. The fig 6 depicts the phase before signal change and the fig 7 shows the phase when green signal is given to south lane.

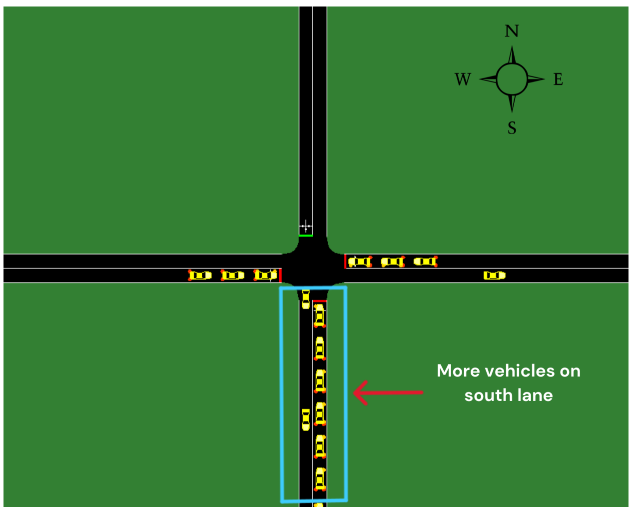

Figure 6. Depicts the phase before signal change.

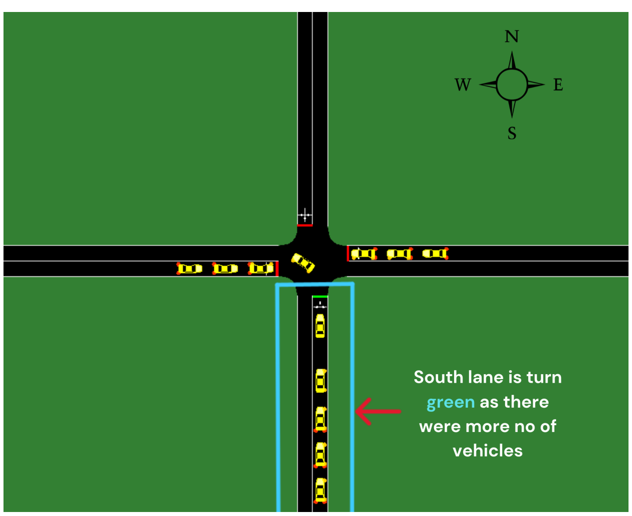

Figure 7. Shows the phase when green signal is given to south lane.

B. *Case 2: When there is incoming vehicle on the lane*

Green signal on West Lane persists as the model detects the incoming vehicle (1) on west lane and thus increases the green signal time for vehicle (1) to pass as shown in fig 8, 9 and 10.

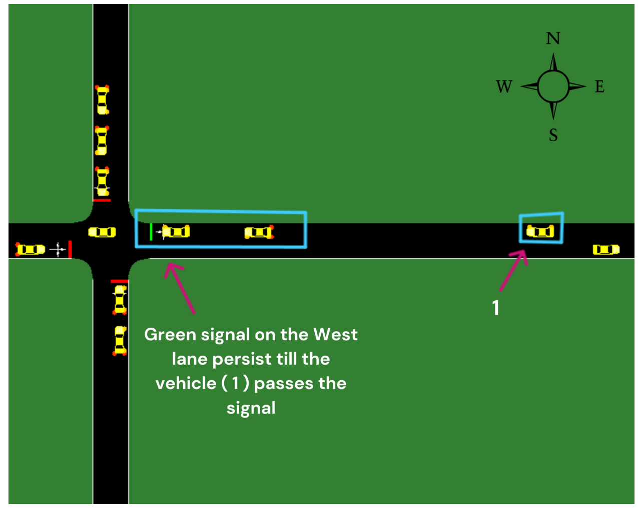

Figure 8. Showing vehicle (1) approaching the signal.

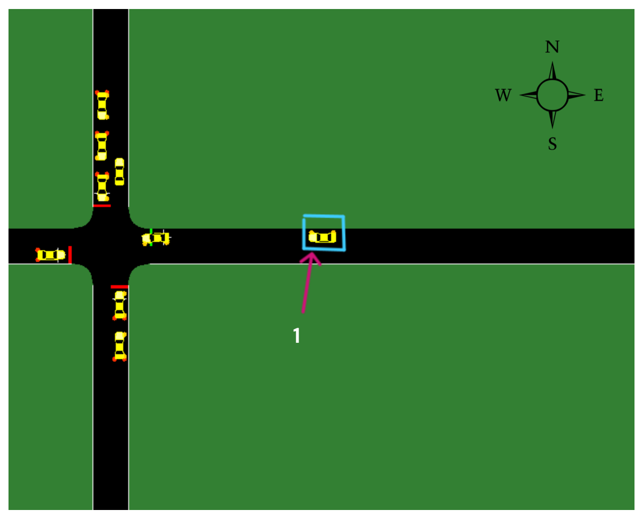

Figure 9. Model detects vehicle (1) on the lane.

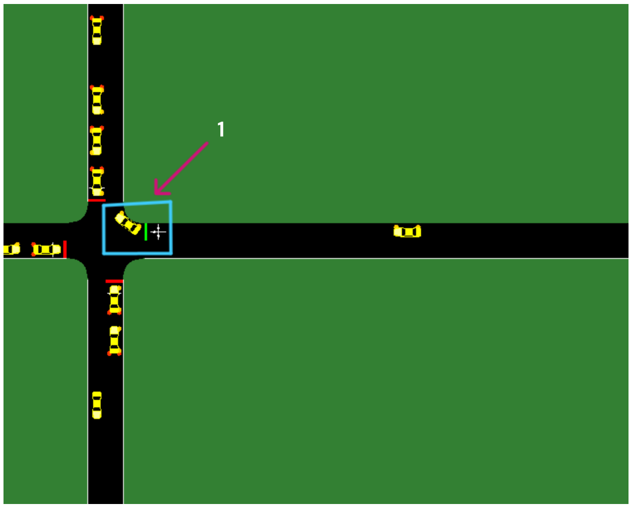

Figure 10. Model has not changed the signal.

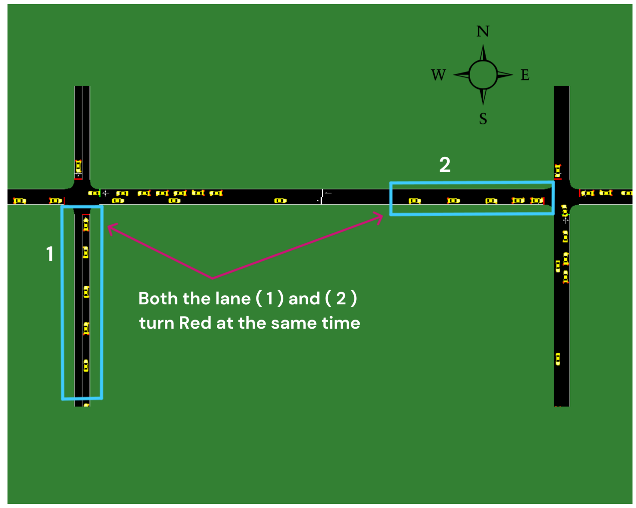

Figure 12. Both turned red at same time.

C. *Case 3: Synchronizations between two junctions*

Let suppose there are two signals side by side then a Multi agent strategy is used by the system to synchronize the traffic signals and provide a much better pathway for vehicles to travel between traffic signals. Both the junctions are synchronized with each other to maximize traffic flow. Both lane (1) and (2) turned green at same time as shown in fig 7.1 and 7.2. This synchronization enables the vehicles released from one junction to move from another junction without again waiting at the next one. This ultimately decreases the overall average waiting time.

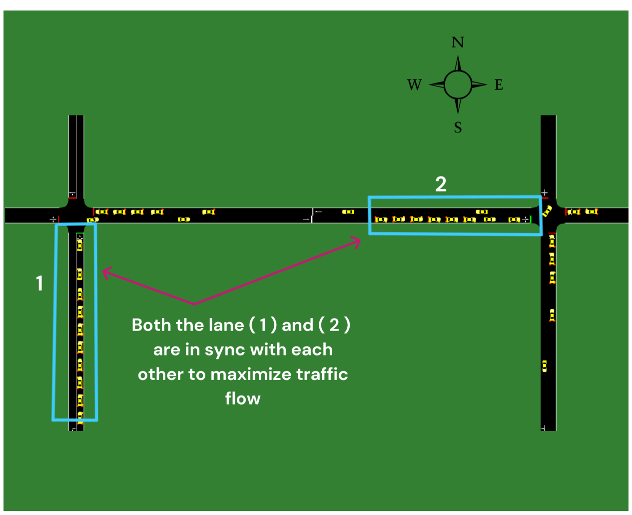

Figure 11. Both lane 1 and lane 2 turned green.

D. *Case 4: Both Lane having same amount of traffic*

In this case the results are similar to that of a classical pre-timed signal controller. If the model detects equal amount of traffic on both lanes the it will allocate same amount of green signal time to each after one another as shown in fig 13 and 14. The average waiting time will either be equal to or less than that of a pre timed signal controller

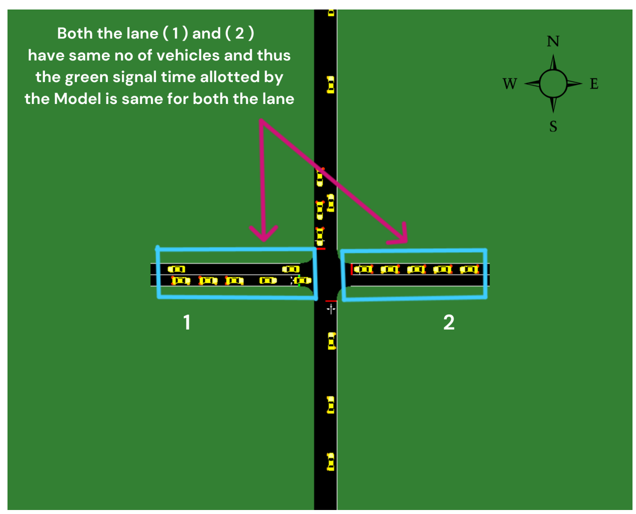

Figure 13. Same number of vehicles on both lanes.

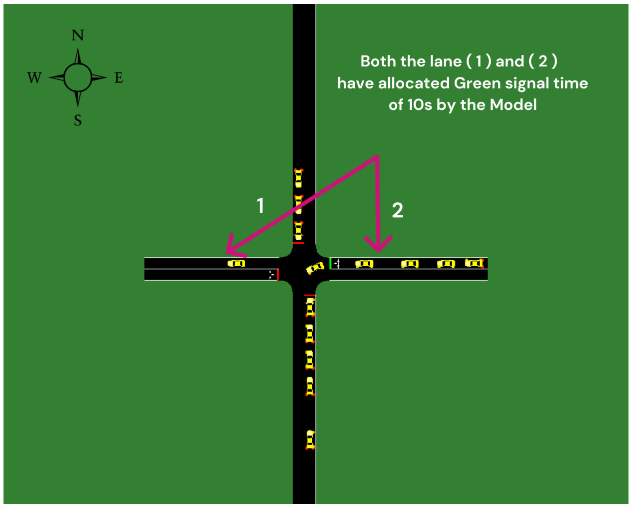

Figure 14. Same time is allocated to both lanes.

E. Case 5: When there is Incoming vehicle from another traffic Signal

Green signal on North Lane will be persisting as the model detects the incoming vehicle (1) on North Lane. Moreover, as the traffic signals are synchronized the traffic signal will thus increases the green signal time for vehicle (1) to pass as shown in fig 15, 16 and 17.

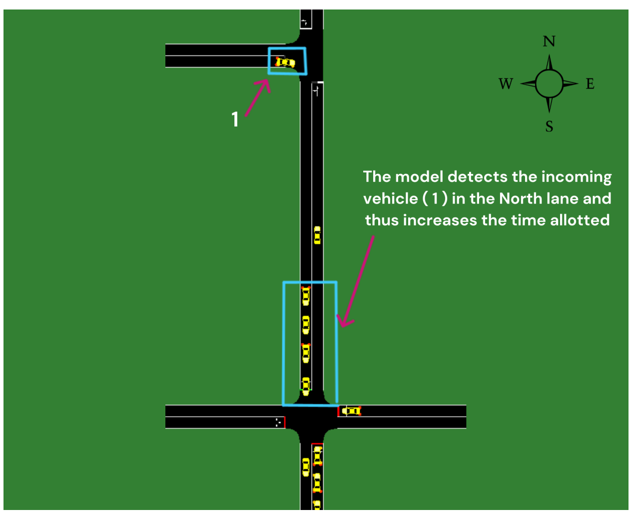

Figure 15. Model detects incoming vehicle on the lane.

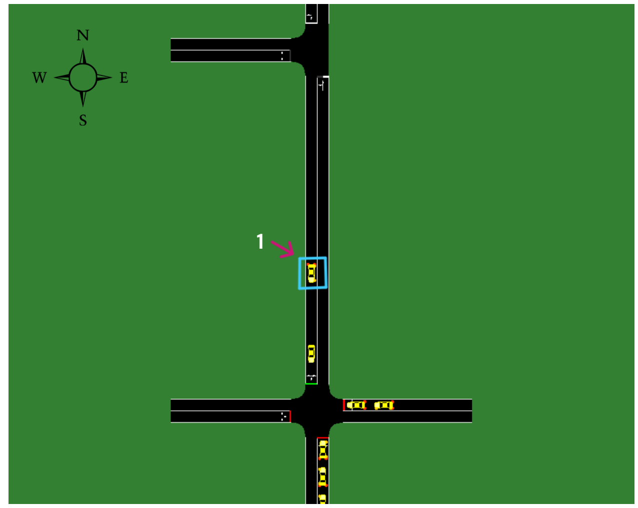

Figure 16. Vehicle (1) enters the lane.

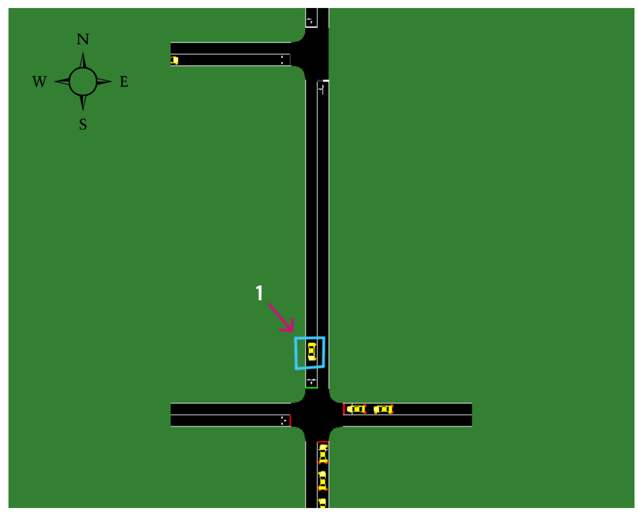

Figure 17. Agent holds the signal for vehicle (1) to pass.

Fig. 18 gives the visualization of model performance over time (epochs). At the start the average waiting time was relatively high but after every successive epoch the average waiting time has decreased radically. The model is basic neural network that trains itself using data generated in the replay memory avoiding the need of providing data beforehand. If the results are compared with classical traffic signal controllers, the proposed system show better performance for traffic signals with dense traffic

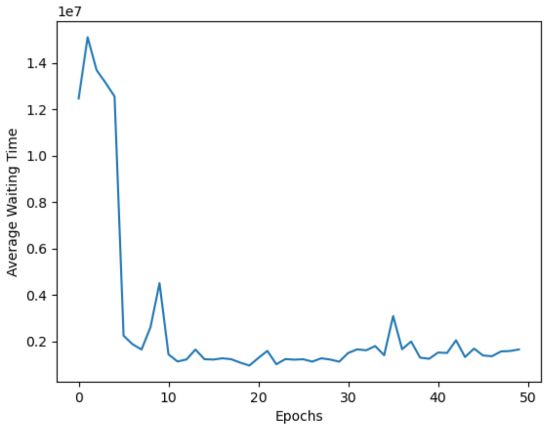

Figure 18. Model training plot

5. CONCLUSION

In this paper, a dynamic traffic management system is proposed that uses reinforcement learning, SUMO and deep neural networks to effectively control vehicle congestion across traffic signals in real time. The proposed model resulted in decreasing average waiting time when the number of epochs are increased. Although the proposed system can show similar performance as classical pre-timed signal controllers for medium or low traffic roadways, but it will always perform better for dense traffic regions. This system is trained for making appropriate decisions for signal timing, according to the positioning of vehicles across the signal junctions for reducing traffic.

Hence, this system can be very beneficial for busy city streets which lack infrastructure of roadways or have heavy traffic density and will reduce the need of manual interfaces or traffic officers at signal junctions. Conclusively, this system can help save considerable time, fuel and avoid public inconvenience.

6. DISCUSSION AND FUTURE SCOPE

The implication of this research is to reduce congestion, minimize waiting times, and enhance overall traffic management in urban areas, particularly in cities facing significant traffic challenges. This system only considers one lane at a time in SUMO and lacks when there are more lanes involved, then vehicles changing lanes can affect signal timings. The future scope for this system involves enhancing real-time adaptability through integrated data sources, multiple lanes, exploring multi-agent frameworks for holistic coordination, and developing interpretable models to instill trust. Research can be extended for prioritizing emergency vehicles like ambulance or fire brigade and improvising system for roadways without lanes are some future advancements to the system.

ACKNOWLEDGEMENT

The authors would like to deeply thank to Dr. Pravin Futane, Head of Information Technology Department from Vishwakarma Institute of Information Technology, Pune for his guidance, informative discussions, and plenty of suggestions for successfully carrying out this research study.

References

[1] Tomtom traffic index ranking 2022. [Online]. Available: https://www.tomtom.com/traffic-index/ranking/?country=IN

[2] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, and M. G. Bellemare, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, p. 529, 2015.

[3] S. B. Cools, C. Gershenson, and B. D'Hooghe, "Self-organizing traffic lights: A realistic simulation," *Advances in Applied Self-organizing Systems*, pp. 45–55, 2013.

[4] S. El-Tantawy, B. Abdulhai, and H. Abdelgawad, "Multiagent reinforcement learning for integrated network of adaptive traffic signal controllers (marlin-atsc): methodology and large-scale application on downtown toronto," *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 3, pp. 1140–1150, 2013.

[5] S. Mikami and Y. Kakazu, "Genetic reinforcement learning for cooperative traffic signal control," in *Proceedings of the First IEEE Conference on Evolutionary Computation*, 1994.

[6] B. Abdulhai, R. Pringle, and G. J. Karakoulas, "Reinforcement learning for true adaptive traffic signal control," *Journal of Transportation Engineering*, vol. 129, no. 3, pp. 278–285, 2003.

[7] I. Tunc and M. T. Soylemez, "Fuzzy logic and deep q learning based control for traffic lights," *Alexandria Engineering Journal*, vol. 67, pp. 343–359, 2023.

[8] S. Bouktif, A. Cheniki, A. Ouni, and H. El-Sayed, "Deep reinforcement learning for traffic signal control with consistent state and reward design approach," *Knowledge-Based Systems*, vol. 267, p. 110440, 2023.

[9] F. F. . S. N. (2023) Alleviating traffic congestion: Developing and evaluating novel multi-agent reinforcement learning traffic light coordination techniques. [Online]. Available: https://rosap.ntl.bts.gov/view/dot/68213

[10] S. Q. and et al., "Hierarchical reinforcement learning for dynamic autonomous vehicle navigation at intelligent intersections," in *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2023, pp. 4852–4861.

[11] M. C. Choy, D. Srinivasan, and R. L. Cheu, "Hybrid cooperative agents with online reinforcement learning for traffic control," in *2002 IEEE World Congress on Computational Intelligence. 2002 IEEE International Conference on Fuzzy Systems. FUZZ-IEEE'02. Proceedings (Cat. No. 02CH37291)*, vol. 2. IEEE, 2002.

[12] E. Bingham, "Reinforcement learning in neurofuzzy traffic signal control," *European Journal of Operational Research*, vol. 131, no. 2, pp. 232–241, 2001.

[13] R. Suryawanshi, O. Rachalwar, A. Hedaoo, A. Thakur, and T. Rane, "A review of traffic light control system with reinforcement learning," *International Journal of Scientific Development and Research*, vol. 8, no. 1, pp. 2455–2631, 2023.

[14] P. Jing, H. Huang, and L. Chen, "An adaptive traffic signal control in a connected vehicle environment: A systematic review," *Information*, vol. 8, no. 3, p. 101, 2017.

[15] S. L. . C. H., "A collaborative control scheme for smart vehicles based on multi-agent deep reinforcement learning," *IEEE Access*, 2023.

[16] J. K. . D. Y. Shingate K., "Adaptive traffic control system using reinforcement learning," *International journal of engineering research and technology*, vol. 9, 2020.

[17] R. Zhang, A. Ishikawa, W. Wang, B. Striner, and O. K. Tonguz, "Using reinforcement learning with partial vehicle detection for intelligent traffic signal control," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 1, pp. 404–415, 2020.

[18] J. Gao, Y. Shen, J. Liu, M. Ito, and N. Shiratori, "Adaptive traffic signal control: Deep reinforcement learning algorithm with experience replay and target network," *arXiv preprint arXiv:1705.02755*, 2017.

[19] H. Yang and et al., "A fast vehicle counting and traffic volume estimation method based on convolutional neural network," *IEEE Access*, vol. 9, pp. 150 522–150 531, 2021.

[20] J. Lu and L. Cao, "Congestion evaluation from traffic flow information based on fuzzy logic," in *Proceedings of the 2003 IEEE International Conference on Intelligent Transportation Systems*, vol. 1. IEEE, 2003.

[21] C. D. Fabritiis, R. Ragona, and G. Valenti, "Traffic estimation and prediction based on real-time floating car data," in *2008 11th international IEEE conference on intelligent transportation systems*. IEEE, 2008.

[22] I. Arel, C. Liu, T. Urbanik, and A. G. Kohls, "Reinforcement learning-based multi-agent system for network traffic signal control," *IET Intelligent Transport Systems*, vol. 4, no. 2, pp. 128–135, 2010.

[23] M. A. and et al., "Traffic flow management of autonomous vehicles using deep reinforcement learning and smart rerouting," *IEEE Access*, vol. 9, pp. 51 005–51 019, 2021.

[24] R. E. Stern and et al., "Dissipation of stop-and-go waves via control of autonomous vehicles: Field experiments," *Transportation Research Part C: Emerging Technologies*, vol. 89, pp. 205–221, 2018.

[25] A. Paul and S. Mitra, "Exploring reward efficacy in traffic management using deep reinforcement learning in intelligent transportation system," *ETRI Journal*, vol. 44, no. 2, pp. 194–207, 2022.

[26] M. K. Singh, K. D. Mishra, and S. Sahana, "An intelligent real-time traffic control based on vehicle density," *International Journal of Engineering Technology and Management Sciences*, vol. (5), 2021.

[27] J. Zhu and et al., "Urban traffic density estimation based on ultrahigh-resolution uav video and deep neural network," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 11, no. 12, pp. 4968–4981, 2018.

Aryaan M. Shaikh is currently pursuing a Master of Science (MS) degree in Computer Science at University of Florida, Gainesville (United States of America). He completed his B.Tech. degree from Vishwakarma Institute of Information Technology, Pune (India). He has more than 24 months of research experience. He has published 5 papers in reputed International Journals and Conferences until now. His research interests lie in the fields of Machine Learning, Neural Networks, Artificial Intelligence, Natural Language Processing, Blockchain and Robotics.

Babasaheb Bhalekar is studying in Vishwakarma Institute of Information Technology, Pune (India). He has completed his schooling and higher secondary education from a Kendriya Vidyalaya School. His research interests lie in Deep Learning, Machine Learning and Blockchain.

Dr. Pravin R. Futane is currently associated with Vishwakarma Institute of Information Technology, Pune (India) as a professor in Computer Engineering and Head of IT department from last 3 years. He is having total 24+ years of experience with 23 years of teaching experience including 15 years of administrative experience and 10 years of research experience parallelly. He was an Adjudicator to evaluate Ph.D. thesis for 4 different universities across India. He has so far acted as an External Examiner for Ph.D. thesis evaluation for 5 Research Scholar and a part of reviewing Ph.D. thesis for around 7 Research Scholars. He also has received a research grant or QIP funding for around 3 programs. He has published more than 70+ research papers in International Journals (37 papers) and Conferences (33 papers). His Google scholar paper citation is 200+, h-Index is 7 and i10 index is 5. His Scopus citation is 26 with h-Index 3 for 7 documents listed. He is a member of many professional bodies viz ISTE, IAMG and ISC. He has guided 22 PG Students and more than 75+ UG projects along with around 25+ internships from industry/research organization.