Role of Artificial Intelligence in Traffic Optimization

Suyash Batra
Symbiosis Institute of Computer Studies & Research (SICSR), Symbiosis International (Deemed University), Pune, India
suyashbatra@gmail.com

***Abstract*—Traffic jam in cities is a burning issue, and especially in the booming cities like Pune, India. This study explores how Artificial Intelligence (AI) and Deep Reinforcement Learning (DRL) can also be used to optimise traffic signals control using the SUMO simulator and TraCI interface. An agent (Deep Q-Network, DQN) was written in Python based on PyTorch and trained to reduce the average vehicle waiting time at intersections. As the results show, the RL-based controller effectively minimizes waiting times relative to the static signal control, which results in a 16% improvement (i.e. a relative 84% reduction to the delay in the baseline) decrease in the waiting time under testing conditions. The results highlight the promise of AI-based adaptive control in improving the efficiency of traffic, decreasing congestion, and providing the foundation to intelligent transportation systems in urban India.**

Keywords—Artificial Intelligence, Deep Reinforcement Learning, DQN, Traffic Signal Optimization, SUMO, TraCI, Intelligent Transportation Systems, Urban Traffic Management.

I. Introduction

Traffic congestion is one of the long-term effects of high urbanisation and motorisation, which is reflected in long commutes, increased fuel usage, poor air quality, a rise in road-carriage accidents, and low productivity. The concept of Artificial Intelligence (AI) has become the key component of intelligent transport systems (ITS), replacing or supplementing the rule-based control with data-driven prediction and optimisation. In either short-term traffic prediction, adaptive signal control, dynamic routing, demand-responsive transit, or vision-based incident detection, modern machine-learning (ML), reinforcement-learning (RL), and computer-vision (CV) systems can discover complex spatiotemporal regularities and optimize control policies in the face of uncertainty. More importantly, the tools are not confined to simulations anymore; real field applications have proven to be beneficial to an extent. E.g. the SURTRAC adaptive signal system in Pittsburgh has resulted in over 25% reduction in average travel time, 31% reduction in frequency, 40% reduction in waiting time, and a 21% reduction in emissions. Similarly, the project green light of Google, that uses aggregated navigation information to derive timing plans, has been reported to reduce up to 30 per cent fewer stops and a 10 per cent reduction in emissions.

The magnitude of the problem can be illustrated by Pune (India). It is one of the slowest cities in the world with a travel time of 33 minutes and 22 seconds of 10 km, and vehicle ownership is growing by 250 000-300 000 vehicles annually, pushing the fleet over 3.5 million and mainly consisting of two-wheelers and cars. Despite a slight improvement in average speeds of 16 corridors (20.422.5 km h -1), the network is still weak at the time of peaks and post-event. The safety consequences are dire: in 2024, two-wheelers were present in 320 and 1,404 crashes respectively, with the former figure involving more than two-thirds of the total fatalities, making the interdependence between congestion and safety and the imperative of flow-enhancing but protective interventions obvious.

Examples from India and abroad underscore pertinent lessons. The AI-operated Adaptive Traffic Control System (ATCS) in Bengaluru recorded 2033 percent travel time reductions, when adaptive timings were implemented at a large number of junctions. When AI-enabled coordination was not limited to corridors, international pilots in China and elsewhere achieved a reduction of over 15 % in delays. Pune in particular is adopting facilitating solutions like adaptive-signal pilots, AI-enabled violation detection, and increased CCTV coverage, thus providing a basis upon which optimisation can be realised. The role of AI in traffic management in this paper therefore seeks to examine AI in traffic management in two aspects (i) to generalise on the state of the art in AI enabled urban traffic control and its effects demonstrated and (ii) to operationalise the same approaches to the congestion and safety profile in Pune and hence inspire AI interventions and an assessment framework that suits the mixed-traffic environment in India.

II. Literature Review

AI has been identified as a powerful solution in addressing the multidimensional problems of traffic congestion in cities. As the literature indicates, a wide range of solutions is represented: deep reinforcement learning and predictive analytics to federated models and real-time smart systems, all of which are designed to optimise traffic flow, im-prove safety, and enable sustainable mobility.

Reinforcement learning is a key focus of adaptive traffic signal control. The combined approach proposed by the authors of [10] as a deep reinforcement learning algorithm achieved an improvement in vehicle delay by up to 86 percent, compared to fixed-time and longest-queue-first approaches. The authors of [13] tried different DRL algorithms in a com- complementary approach including DQN, SAC, A2C, and PPO. They underlined the role of reward functions design and proposed a custom simulation environment that can be used together with Gym.

Based on this, the authors of [11] showed that a new state-space formulation resulted in 32 and 37 percent reductions in delay of semi-actuated and fixed-time controls respectively. These results confirm the useful-ness of DRL in the ability to model temporal dynamics and reconfigure to real-time traffic changes.

Solving the problem of scalability and data confidentiality, the authors of [6] proposed a federated PPO architecture, which empowers distributed training at intersections without undermining the data security. This solution helped 47.69 percent faster convergence of these models and 27.34 percent shorter wait times of vehicles. On the same note, [4] and [3] authors discussed coordinated control strategies with MARL and discussed simulation platforms like SUMO and VISSIM which are vital in practical use.

Congestion prediction is made more efficient by using predictive analytics, which builds AI models. The authors of [12] combined RL with LSTM, ARIMA, and GNNs for traffic fore-casting in Belgrade. Their hybrid system took on average 33 percent less time per vehicle and emitted fewer, 16 percent less, emissions, highlighting the impact of AI in supporting sustainability.

Also, the models used in [7] were CNN and RNN-LSTM to identify and forecast vehicle volumes, with the authors reporting that on the CARLA simulation platform, 50% increase in flow rate and a 70 percent decrease in vehicle delay were achieved.

AI has also been useful in traffic routing optimization. The researchers in [8] suggested the ML-based systems that dynamically adapt routes according to the real-time conditions. They had significant travel time, fuel, and emissions reductions in their simulations.

In addition, the authors of [16] highlighted the potential of AI to man-age road networks when under pressure, particularly, with the emergence of autonomous vehicles. They also established some of the barriers to deployment such as infrastructure costs and ethical concerns.

The role of AI goes to traffic law enforcement and compliance with rules. The authors of [17] introduced a complete smart system that can manage weather-sensitive signal ad-adjustments, computer vision violation detection, and automatic imposition of fines. A holistic urban mobility solution was also provided, giving route recommendations that would reduce congestion.

Simultaneously, the authors of [14] paid attention to Electronic Traffic Law Enforcement (ETLE), strengthening the role of AI in stimulating digital legal infrastructures and better justice delivery in traffic control.

Lastly, the authors of [15] imagined the world in which the manual traffic policing is substituted by intelligent surveil-lance, vehicle detection, and inter-signal communication with the help of AI. Their camera-based system was more economically effective than RFID-based one.

III. METHODOLOGY

In this paper, Python 3.11 was used as the main programming language, due to its high efficiency, better readability and compatibility with scientific libraries. The version chosen provided improved performance and strong error-handling features, thus, being able to run simulation scripts and artificial intelligence models. Eclipse SUMO (Simulation of Urban Mobility), an open-source, microscopic traffic simulator was used to simulate traffic and was found fitting in the study of complex urban transport networks. SUMO provided the base scenario to simulate real world traffic environment that includes vehicles, intersection, and road signs. TraCI API (Traffic Control Interface) was used to allow real time communication of the simulation with the external algorithms. The SUMO-Python interface helped to exchange data in both directions allowing dynamic control of the traffic lights and change the route as well as obtain the live traffic conditions to analyze them.

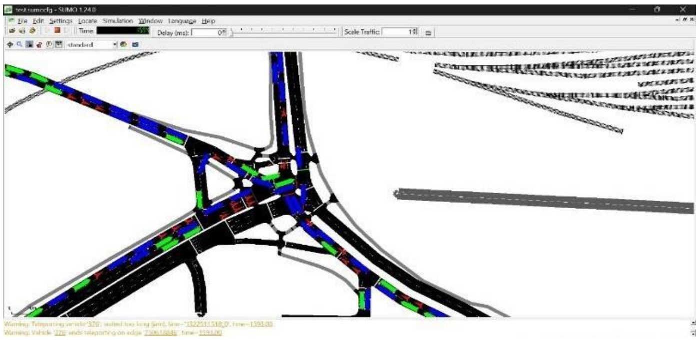

Fig.1: Sumo GUI Simulation

In Fig.1 shows an instance of an ongoing simulation. There are multiple different types of vehicles used in the simulation done for the static runs, training and testing of the model.

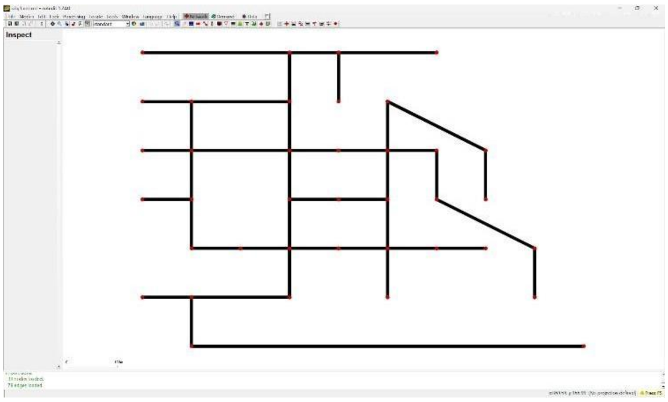

Fig.2: Training Road Network

The network in Fig.2 represents a generic road network on which the RL model was trained on. The network was taken for an open source directory. The experiments were done using machine-learning with the help of the PyTorch, which was chosen due to its dynamic computation graphs and ability to use a graphics card to accelerate the training and testing of deep reinforcement learning models. These models were designed to maximise traffic flow by adaptation of control measure through simulation.

Other computational work was handled through NumPy which provided with efficient operations on large sets of data and Matplotlib which was used to visualise the results of the simulations such as congestion rates, numbers of vehicles and comparisons of performance of various models.

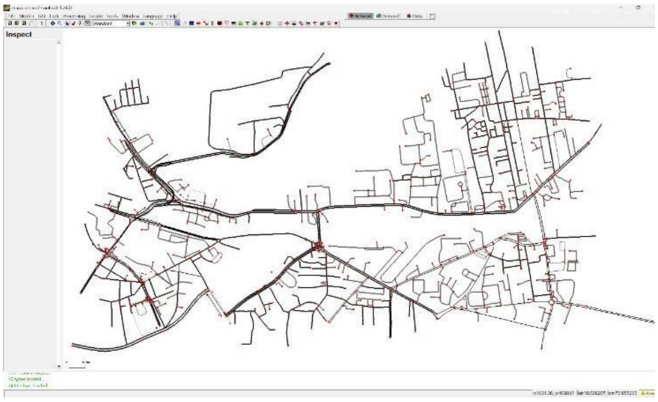

Fig.3: Pune City Layout (For Testing & Static)

The experiment took into account different topologies of the network to depict urban traffic pattern. These topologies defined the interconnection of the intersections and road segments affecting the development of routing strategies and traffic light control policies. The network in Fig. 3 is related to a high-traffic place called RTO Chowk. The openstreetmap.com site had the source of this road network. Both regular and non-regular topology were represented according to various city systems and in Fig. 3 these two topological systems are depicted. All simulation experiments were launched and controlled by a central root configuration file, that specified the simulation parameters, network inputs, and traffic demand patterns. This arrangement provided recurrence and uniformity in numerous simulation iterations. These tools and models created a comprehensive approach to model, analyze, and optimize traffic systems with the help of artificial intelligence.

The purpose of the experimental pipeline is to contrast between a traditional fixed-time (static) signal plan with a reinforcement-learning (RL) controller that has been acquired in simulation. Experiments were carried out in Python 3.11, Eclipse SUMO to simulate microscopic traffic, SUMO-TraCI to control and monitor the run-time, and PyTorch to implement the RL agent. NumPy and Matplotlib were used to carry out auxiliary numerical and plotting tasks. The RL training and evaluation logic are implemented in training.

A. Data preparation and network generation

To model the RTO Chowk intersection, the simulation network was developed based on OpenStreetMap (OSM) data, which had to be extracted with Overpass -Turbo. NetEdit was used to translate the raw .osm file to a SUMO-compatible network (.net.xml) and thus provide an accurate mapping of road segments and intersection layouts. The synthesis of traffic demands was performed using randomTrips.py utility of the SUMO that created trips on the network with controlled parameters including interval and duration. Duarouter was used to compute routing to generate the.rou.xml files to ensure realistic vehicle paths. Training (train.sumocfg) and static baseline testing (static.sumocfg) were kept in separate configuration files, so that comparisons can be made systematically. This process is aligned with well-known workflows of creating reproducible traffic simulation environments by using OSM and SUMO tools [18].

Example:

python randomTrips.py -n pune.net.xml -o pune.trips.xml -p 5 -e 3600
where: -p is period of vehicle spawn
-n is the network file name
-o is the output trip file name
-e is the time end value

duarouter -n pune.net.xml -t pune.trips.xml -o pune.rou.xml
where: -t is the name of the trip file
-o is the name output route file

Configurations: Separate .*sumocfg* files were prepared for training (*train.sumocfg*) static (*static.sumocfg*) and testing (*test.sumocfg*). Specifically, the RTO Chowk intersection .net.xml and .rou.xml were used for static baseline and RL evaluation.

B. Static (baseline) configuration

A fixed-time signal control scheme was used in this work as a point of reference to be evaluated. This setup was defined in the static.sumocfg file and run over the RTO Chowk intersection network. This setup followed a fixed cycle time and sequence of phases which were constant through the entire simulation and did not respond to changes in traffic demand. This kind of rigidity often causes unnecessary traffic management, since even in the event of low or lumpy traffic one might still be waiting.

The tripinfo outputs produced by the SUMO in this example of a static run were used as benchmark measures such as waiting times, trip times, and teleport events. Setting up this fixed baseline provided a fixed reference point with which to compare it with the adaptive reinforcement-learning controller, a technique commonly used in traffic signal control research to demonstrate relative merits of dynamic systems [10].

C. Reinforcement-learning controller

The adaptive controller that is outlined in the following section was developed using a Deep Q-Network (DQN), which was chosen due to its suitability in sequential decision-making with uncertainty. The agent, implemented in PyTorch and connected to SUMO through the TraCI API, used lane-level vehicle counts as inputs and selected one of four discrete states of the traffic signal as actions. To strike a balance between exploration and exploitation, an epsilon-greedy exploration policy was implemented.

Experience replay also increased training stability by accumulating historical interactions and allowing arbitrary sampling, which decreased the number of correlations between consecutive states. The regular renewal of the target network also helped maintain stability of Q-value estimation. The reward function involved punishing long waiting time and queue length and rewarding improved traffic flow hence the learning objective was directly related to congestion reduction.

As a result, the reinforcement-learning controller was able to develop dynamic switching strategies that adjusted to the changing demand at RTO Chowk. Unlike the fixed-time benchmark which was strict in its cycles, the DQN agent continuously updated its policy according to feedback, resulting in a more efficient intersection control. The reinforcement-based method outperformed comparator-based approaches in previous research, thus supporting the viability of AI-supported adaptive control in complex urban traffic conditions [10].

D. Training protocol and CLI reproducibility

The training procedure was designed to subject the reinforcement-learning agent to a wide range of traffic environment, thus improving its generalization properties. The training sessions were carried out with Python 3.11 and used Simulation of Urban MObility (SUMO) platform as a simulator, and PyTorch as a deep-learning framework. A central training configuration file (train.sumocfg) was used to configure each session, thus leading to reproducibility across experimental runs. Domain randomisation was done by varying traffic-demand conditions, thus counteracting the tendency to overfit a single pattern of traffic, and selecting the model based on the smallest mean vehicle waiting time in epochs. This organised training path complies with modern research practices in adaptive traffic -signal studies, where robustness and reproducibility are the key elements in eliminating the simulation deployment gap [6].

Example training run:

python train.py --train -m pune_model -e 2000 -s 3600 --train-cfgs train.sumocfg --nogui

where: -m is model name

- e is number of epochs to run

- s is number of steps per epoch

-cfgs is name of the config file to run

E. Testing protocol

Testing phase was carried out to compare the trained model of reinforcement-learning with the fixed-time baseline in the same conditions. To be impartial, the agent was tested using the test.sumocfg file using the RTO Chowk network with a fixed set of simulation parameters and a fixed set of traffic demand profile. The results reported on performance measures including the mean waiting time, trip time and teleport events, thus, allowed a direct comparison with the fixed control. Unlike the baseline where performance remained constant, the reinforcement learning model was adaptable and adjusted its responses to changing real time traffic flows. The approach is reflective of the standard practice in the field of evaluation because the conventional forms require equity and repeatability to prove the claim of performance [12].

Example testing run:

python train.py -m pune_model -e 10 -s 3600 --test-cfg test.sumocfg --nogui

F. Reward function and evaluation formulas

We trained the learning agent by optimizing a direct traffic efficiency indicator that was used as a reward function. The function took the total wait at the controlled interchange as its input. The reward at time t was defined as the reduction in wait at the previous step compared to the current one with a tiny penalty for long queues that discourages it.

$$R_t = (W_{t-1}^{junction} - W_t^{junction}) - 0.01 \cdot Wt$$

where:

- R_t = *reward at time step t.*
- W_{t-1} = *total waiting time on the junction's lanes at the previous decision step.*
- W_t = *total waiting time on the junction's lanes at the current step*

A positive reward is that the action taken reduced total congestion, while a negative reward is that traffic has become more congested.

To analyze, we consider all network vehicles' wait times $\{w_1, w_2, \ldots, w_n\}$ and calculate traditional traffic performance indices, e.g., average, median, higher percentiles (75 and 90), and maximum wait. Also, we monitor the percentage of vehicles that experience teleport events in order to capture system robustness. To provide a clean contrast between static control and learned RL policy, we report relative gain in performance as an improvement percentage:

$$Improvement(\%) = \frac{W_{static} - W_{RL}}{W_{static}} * 100$$

where:

- W_{static}= *mean waiting time under static (non-RL) control.*
- W_{RL} = *mean waiting time under reinforcement learning*

This construction ensures that results focus not just on average efficiency but also fairness among vehicles by considering median and percentile quantities, and the improvement metric provides a direct quantitative correspondence between optimized and baseline control approaches. This method offers direct inter-comparability between static control and reinforcement learning approaches at the intersection of RTO Chowk.

IV. RESULTS

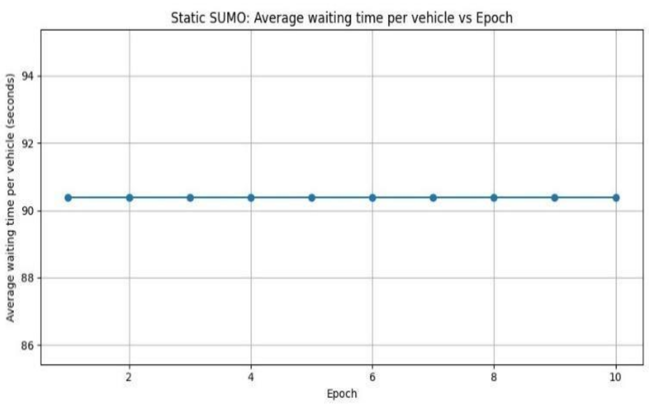

Fig.4: Static Setup

Fig. 4 shows the performance of fixed-time static controller in 10 simulation runs in the RTO Chowk intersection. The findings reveal that the mean waiting time per vehicle is equal at all epochs, a value of about 90.5 seconds. This invariance is an expression of the rigidity of static control which engages pre-determined signal cycles without considering demand changes. Although predictable, this strategy is not flexible enough to adjust to changing traffic patterns, leading to long delays and poor use of the road capacity [3].

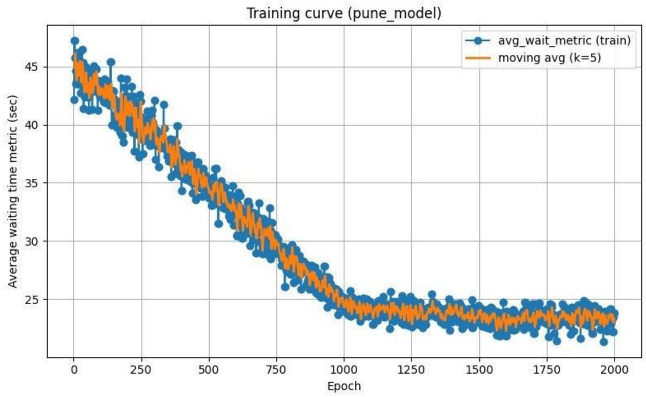

Fig.5: Training curve

Fig. 5 demonstrates training performance of the reinforcement learning agent with 2000 epochs using the mean vehicle waiting time as the measure of the agent performance. The curve shows a slow decrease in values over 45 seconds at the first epochs till about 23 seconds at the end of the training, and the moving average line will support the general downward tendency. This shows that the agent gradually learnt better phase-switching strategies, and in so doing, traffic flow efficiency increased. It reaches a plateau at about 1200 epochs, indicating that it converged to an optimised policy where further training only produced slight levels of improvement. These results affirm the ability of the reinforcement learning to dynamically hone the decisions on the traffic control with time [10].

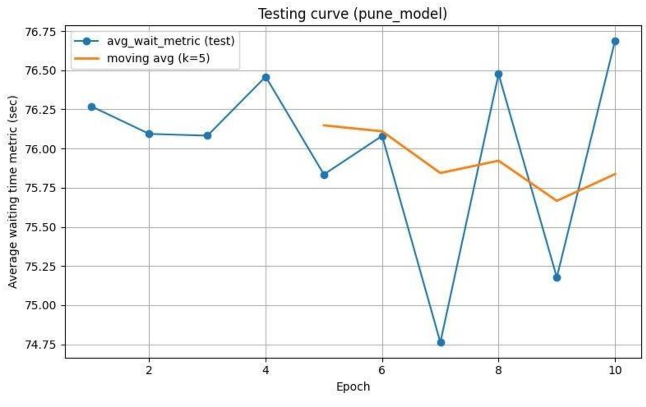

Fig.6: Testing curve

Fig. 6 shows the performance of the trained reinforcement-learning controller during ten runs in a simulation. The mean waiting time per vehicle is variable in a small range, that is, at approximately 74.8 -76.7s/veh, and its peaks and troughs are stochastic variations of traffic demand. The moving average stays near 76s during the test and, thus, regularizes temporal variations and emphasizes the issues of stability of the learned policy. Such findings indicate that, despite being placed in invisible circumstances at the time of evaluation, the RL agent repeatedly employs good phase-switching mechanisms, a sign of robustness as well as flexibility out of the training context.

The RL controller in Fig. 6 shows a significant improvement when compared to the static baseline in Fig. 4 where an average wait time of approximately 90.5s was observed during all the runs. A waiting time of approximately 76s reduces the average waiting time by an estimated 16 per cent as compared to the fixed-time setup. The given improvement is especially important in the light of dynamic, unpredictable character of traffic demand, in which static controllers are rigid. The flexibility of the RL agent to changing flows highlights the importance of data-based control techniques, which provide quantifiable benefits to operational returns compared to conventional fixed operational cycles and supports the possibility of reinforcement learning in practice in real-world traffic control [11].

Following these implications when applied to practical traffic management, such an adaptive system has several advantages as compared to fixed-time signals. Reinforcement learning can be used to minimize the average waiting time, number of unnecessary stops and encourage smoother traffic flow by dynamically changing the duration of phases to achieve this desired outcome. This consequently results in fuel-saving, less emissions due to idling cars, and increased safety as there is no need to contend with long queues and spillbacks that habitually cause accidents. Simply put, whereas the concept of the use of a static control brings uniformity, reinforcement learning comes with flexibility, which makes it much more appropriate in the face of a complicated and dynamic traffic patterns of cities like Pune.

V. Conclusion and Future Work

This work describes the use of a Deep Q-network (DQN) based reinforcement-learning controller to achieve traffic-signal optimisation in the Simulation of Urban MObility (SUMO) framework. The reinforcement-learning (RL) controller was tested with performance being compared to a baseline comprising a fixed, static state at the RTO Chowk intersection, which is a location where congestion is intense in Pune. The experimental findings clearly indicate that the RL-based proposal performs better than the use of a fixed signal control that obtains large cutoffs in time vehicle waiting and has the capacity to respond to changes in the traffic flow. These advances highlight the opportunities of reinforcement learning to alleviate the inefficiency of fixed-time systems that cannot adapt to demand variations. The results are in agreement with the rest of the literature, according to which AI-controlled adaptive signal control has been demonstrated to positively influence traffic throughput, reduce idle periods, and assist in sustainable mobility targets [3].

A. *Future Work*

Even though it is emphasised in this research that reinforcement learning holds hope in traffic-signal optimisation, various opportunities are still available in future research directions. One, the experiments were limited by hardware limitations that limited the size of training and testing. Parameters were adjusted to less than real-world traffic volumes which restricted the ability of the model to be able to fully simulate large urban networks. Future research may use more efficient computing to facilitate longer training horizons, bigger state action space, and more detailed neural architectures. This would facilitate simulation of the parameters that are closer to the reality and reinforce validity of extrapolations outside of the controlled setting. Second, the current simulations involved various forms of vehicles in the conditions of strict discipline on a lane, which simplifies the driving behaviour in the real world. Real city traffic, especially in mixed-traffic situations, as in Indian cities, is described by uncertainty, high lane switching, and non-homogeneous interdependence. Such behavioural complexity should be integrated in future work by modelling stochastic heterogeneous traffic flow using heterogeneous driver models, which can capture irregular behaviour of human drivers. This would give a more realistic criterion to assess adaptive signal control policies. In addition to composition of traffic, a future study must expand analysis to form beyond single intersection to corridor and network level conditions. Besides that, inclusion of real-world traffic information collected by sensors, GPS paths, or camera feeds would minimize the divergence between simulation and deployment so that trained policies can be resistant to operational uncertainty[6].

VI. References

[1] Meess, H., Gerner, J., Hein, D., Schmidtner, S., Elger, G., & Bogenberger, K. (2024). First steps towards real-world traffic signal control optimisation by reinforcement learning. *Journal of Simulation, 18*(6), 957–972. https://doi.org/10.1080/17477778.2024.2364715

[2] Pillai, A. S. (2024). Traffic management: Implementing AI to optimize traffic flow and reduce congestion. *SSRN Electronic Journal*. https://doi.org/10.2139/ssrn.4916398

[3] Michailidis, P., Michailidis, I., Lazaridis, C. R., & Kosmatopoulos, E. (2025). Traffic signal control via reinforcement learning: A review on applications and innovations. *Infrastructures, 10*(114). https://doi.org/10.3390/infrastructures10050114

[4] Xiao, F., Lu, J., Li, L., Tu, W., & Li, C. (2025). Advances in reinforcement learning for traffic signal control: A review of recent progress. *Intelligent Transportation Infrastructure, 2025*(liaf009). https://doi.org/10.1093/iti/liaf009

[5] Bahamazava, K. (2025). AI-driven scenarios for urban mobility: Quantifying the role of ODE models and scenario planning in reducing traffic congestion. *Transport Economics and Management, 3*(1), 92–103. https://doi.org/10.1016/j.team.2025.02.002

[6] Li, M., Pan, X., Liu, C., & Li, Z. (2025). Federated deep reinforcement learning-based urban traffic signal optimal control. *Scientific Reports, 15,* 11724. https://doi.org/10.1038/s41598-025-91966-1

[7] Goenawan, C. R. (2025). ASTM: Autonomous smart traffic management system using artificial intelligence CNN and LSTM. *arXiv preprint arXiv:2410.10929*. https://arxiv.org/abs/2410.10929

[8] Dikshit, S., Atiq, A., Shahid, M., Dwivedi, V., & Thusu, A. (2023). The use of artificial intelligence to optimize the routing of vehicles and reduce traffic congestion in urban areas. *EAI Endorsed Transactions on Energy Web, 10*(1), e4. https://doi.org/10.4108/ew.4613

[9] Akinade, A. O., Adepoju, P. A., Ige, A. B., & Afolabi, A. I. (2024). Artificial intelligence in traffic management: A review of smart solutions and urban impact. *IRE Journals, 7*(12), 511–520.

[10] Gao, J., Shen, Y., Liu, J., Ito, M., & Shiratori, N. (2017). Adaptive traffic signal control: Deep reinforcement learning algorithm with experience replay and target network. *arXiv preprint arXiv:1705.02755*. https://arxiv.org/abs/1705.02755

[11] Muresan, M., Fu, L., & Pan, G. (2018). Adaptive traffic signal control with deep reinforcement learning – An exploratory investigation. *Proceedings of the Transportation Research Board 97th Annual Meeting*. https://doi.org/10.13140/RG.2.2.27674.80324

[12] Skoropad, V. N., Deđanski, S., Pantović, V., Injac, Z., Vujičić, S., Jovanović-Milenković, M., Jevtić, B., Lukić-Vujadinović, V., Vidojević, D., & Bodolo, I. (2025). Dynamic traffic flow optimization using reinforcement learning and predictive analytics: A sustainable approach to improving urban mobility in the city of Belgrade. *Sustainability, 17*(8), 3383. https://doi.org/10.3390/su17083383

[13] Boyko, N., & Mokryk, Y. (2024). Optimizing traffic at intersections with deep reinforcement learning. *Journal of Engineering, 2024,* Article 6509852. https://doi.org/10.1155/2024/6509852

[14] Hasibuan, E. S. (2023). AI used in electronic traffic law enforcement as law enforcement evidence in digital era. *Interdisciplinary Social Studies, 2*(11), 2612–2620. https://doi.org/10.1234/iss.2023.011

[15] Musmade, P., Rajas, S., & Khairnar, S. M. (2024). Artificial intelligence (AI) in traffic management: Toward a future without traffic police. *ADYPJIET, 1*(1), 17–23.

[16] Nasim, S. F., Qaiser, A., Abrar, N., & Kulsoom, U. (2023). Implementation of AI in traffic management: Need, current techniques and challenges. *Pakistan Journal of Scientific Research, 3*(1), 20–25.

[17] Sarjini, S., Sathya, M., & Sathya, M. (2025). AI-based traffic management system. *International Journal of Research Publication and Reviews, 6*(4), 9538–9544. http://www.ijrpr.com

[18] M. Raifer, Overpass Turbo, OpenStreetMap Wiki. [Online]. Available: https://overpass-turbo.eu/

[19] Navtegh, Traffic Light Management System using RL and SUMO — maps folder, GitHub repository. [Online]. Available: https://github.com/Navtegh/Traffic-Light-Management-system-using-RL-and-SUMO/tree/main/maps

[20] "SUMO Documentation," Simulation of Urban Mobility (SUMO), Eclipse SUMO User Documentation. [Online]. Available: https://sumo.dlr.de/docs/index.html