Intelligent Traffic Signal Management using DRL for a Real-time Road Network in ITS

ANANYA PAUL*, Indian Institute of Engineering Science and Technology, Shibpur, India
KRISHNENDU BERA†, Indian Institute of Engineering Science and Technology, Shibpur, India
DEVTANU MISRA†, Indian Institute of Engineering Science and Technology, Shibpur, India
SATTWIK BARUA†, Indian Institute of Engineering Science and Technology, Shibpur, India
SAURABH SINGH†, Indian Institute of Engineering Science and Technology, Shibpur, India
NISHU NISHANT KUMAR†, Indian Institute of Engineering Science and Technology, Shibpur, India
SULATA MITRA, Indian Institute of Engineering Science and Technology, Shibpur, India

The acceleration of urbanisation and the development of the pace of industrialisation help to grow the population of metropolitan areas, thus increasing the density of traffic flow. The sole way of managing traffic congestion is to mitigate it through optimising traffic signals at the intersections of a vast road network. The synchronization amongst the traffic signals at intersections is strongly needed in order to alleviate congestion and to allow vehicles to travel smoothly along intersections. Reinforcement Learning (RL) techniques in Intelligent transportation system (ITS) are not feasible for the management of traffic signals of large road networks due to enormous information of the state-action pairs. To overcome this problem, the emerging technology of Deep Learning allows RL to form Deep Reinforcement Learning (DRL) to measure up previously unwavering decision-making issues, for handling high-dimensional states and action spaces. DRL agents perform tasks through perception, monitoring the environment through action and learning as well as analysing the results of actions. In the present work, a single DRL agent is trained using the Policy Gradient algorithm in four different categories of Deep Neural Networks (DNN) to control the traffic signals dynamically. In case of a static road network, the functional implementation and efficacy of the Policy Gradient algorithm cannot be analysed accurately due to the less intricate details of static network. Hence, two different dynamic real time road networks have been considered here. Moreover, the real-time spatio-temporal information congregated from the dynamic real time map is provided as an input, so that the traffic signal duration can be adjusted adaptively in order to manage the traffic flow appropriately. The viability of the simulation experiment is investigated using three separate simulation metrics against the baseline, which is fixed signal duration frameworks and indeed the suggested method outperforms the baseline.

CCS Concepts: • **Computing methodologies** → **Reinforcement learning**; *Agent / discrete models.*

Additional Key Words and Phrases: Adaptive traffic signal Management, ITS, DRL, Policy gradient, Real time road network

1 INTRODUCTION

The vehicle industry has significantly developed due to flourishing technical progress and developments. Traffic uncertainty and unpredictability have surpassed the ability of traffic signal systems to work effectively on previously determined time schedules. As the traffic volume escalates, it also efficiently contributes to traffic congestion and road accidents. Highly dense traffic and slower speeds of vehicles characterise traffic congestion, which is a condition that exceeds the capacity of the lane in terms of traffic volume. It not only drains scarce public services, but also renders mobility resources unable to make the full collective usage that is incapable of allocating transport capital equitably. A number of traffic congestion problems occur, including unnecessary delays in travel, increased fuel wastage due to frequent frequencies of braking and intermediate gear [1]. However, there are common features and complex processes

*Corresponding Author
†Authors contributed equally to this research.

of managing traffic congestion in different metropolitan areas. To effectively monitor traffic flow, coordination between traffic signals at intersections must be improved; otherwise, congestion will continue to propagate across time to many other neighbouring intersections [2]. Interdependence among traffic signals at various intersections seems to be so significant for a metropolitan area that the variations in traffic signals can have repercussions on each other. Current traffic signal control methods also rely heavily on oversimplified knowledge and regulatory approaches, whereas recently there are enormous data, improved processing resources and sophisticated methods for driving smart transport. With an increase in population and modern innovations in transport systems, transport has developed into smart networks known as ITS [3]. Machine Learning (ML), on the other hand, seeks to control systems with minimal human intervention. Integration of ML and ITS provides adequate solutions for optimising traffic signal difficulties.

RL is a framework of ML, that is interactive with the environment and establishes the best policy for sequential decision-making in the various fields of sciences through trial and error method [4]. An RL agent verifies the status of the environment and therefore, carries out an operation. Any action performed earns a reward or penalty and this reward is determined by the environmental impact of the action. The goal of the agent is to learn the best selection technique to maximize the discounted cumulative reward by means of repeated ambient interactions. Generally, for a large scale road network, as state-action pairs extend exponentially, the difficulty of using RL in traffic signal management increases in an exponential manner. Deep Learning has been highly appreciated and combined efficiently with RL approaches to yield DRL [5] in order to solve this dilemma. DRL has been an effective solution for sequential decision-making control problems in recent years, and has shown an incredible success in complex, dynamic and high dimensional environments.

In the present work, efficiency of the Policy Gradient algorithm is tested in two real time dynamic road networks with multiple intersections in which traffic signal is connected to each intersection. The following are suggested contributions from the method proposed:

- An adaptive traffic signal management system based on DRL is proposed to monitor multi-intersection traffic signals as simultaneous control of all the traffic signals of the network permits more efficient traffic handling.
- The traffic flow is a sequential spatio-temporal data stream. The DRL agent uses this data stream of the traffic environment to coordinate the traffic signals on the intersections. In contrast, this understanding is indiscriminate in the sense in which the agent controls a single traffic signal.
- To precisely describe the temporal information of the traffic environment, a stack of five frames is used to define state representation. It encourages the agent to obtain more environmental knowledge, leading to a better choice of actions.
- The agent is trained using Policy Gradient algorithm in several Deep Neural Network (DNN) models such as Fully Connected Neural Network (FCNN), Convolutional Neural Network (CNN), Long Short Term Memory (LSTM), Gated Recurrent Unit (GRU) to estimate the best action selection policy to ensure effective traffic signal management.
- The productivity of Policy Gradient algorithm is already analyzed in a static network in [6]. However, dynamic real time networks help to portray actual scenarios with its associated complexities which is not possible to mention in static network. So, in this work, the entire simulation experiment is done in two different real time dynamic road networks, which is downloaded from OpenStreetMap (OSM).

This paper is divided into four more parts. In section 2, research gaps in former researches are being discussed. Section 3 presents the present work followed by section 4 which describes the simulation results and discussion. Finally, section 5 concludes the research work.

2 RELATED WORK

Recent progress in RL and DNN configurations has shown promising results to solve large dimensional dynamic control problems. Influenced by these accomplishments, two types of RL are built in [7]- the deep policy gradients and the value-based policy, which can better predict traffic flow in an intersection. Agents receive a glimpse of the current status of an integrated camera and generate control signals in any state. The value-base policy agent first calculates the value of all control signal, but the agent selects the signal by direct observation. It then chooses the maximum value for the optimum control action. These approaches have produced promising results and have shown they can find more stable policies compared with the anticipated work in the optimisation of traffic signals.

In order to learn the optimal policy for traffic signal management in congested scenarios, a policy gradient approach is suggested in [8], based on periodic circumstances and on time baseline. It is applied to a highly dense roundabout and evaluated on real data-set which shows that the policy reduces overall waiting time and emissions significantly while avoiding traffic congestion. It has been also shown that the suggested time base policy gradient algorithm is better than the classical baseline. In this case, only the roundabout network was considered, where it is not clear whether the algorithm will be functioning well in a real time dynamic road network also.

A number of algorithms for DRL is proposed in [9] in order to design signal control. A DNN is set up to learn the Q function from the inputs that have been sampled. On the basis of the DNN, it is necessary to model the control activities and the change of the system states in the appropriate signal timing policies. Although there have been a few approaches towards approximation of the maximum discounted future reward, this method shows that DNNs provide a more powerful and convenient tool in order to achieve the goal. After setting up a number of policies, the explore-exploit dilemma will be faced and for this, possibly better policies may be opted in terms of exploiting the current working policy.

In [6], a single DRL agent uses the policy gradient algorithm to handle a traffic signal of several intersections. The agent is trained, in particular, on spatio-temporal environmental data to act in a variety of deep-neural networks. Three different simulation metrics are analysed for the simulation experiment. Policy gradient method is executed in various deep neural network models, namely, in order to bring about improved control of traffic signals. FCNN and CNN are used for approximating the action selection policy. But the simulation experiment has been executed in a static network here.

A promising and scalable multi-agent approach to DRL using the Deep Q Network (DQN) algorithm is introduced in [10]. It examines traffic signal control policies with the help of new rewarding features and it is suggested to combine the popular Deep Q learning algorithm with a coordination algorithm in order to manage traffic signals. It is demonstrated that this approach reduces travel times in relation to previous research on reinforcement methods. It works for multi-agent signal control earlier, but the DQN algorithm can oscillate, a problem which was also found in previous works on profound reinforcement learning.

3 PRESENT WORK

3.1 RL:

In recent years, RL techniques for controlling the traffic signals are being explored by researchers in order to mitigate traffic congestion. The purpose of RL is to learn an optimal policy starting from the initial state. An agent in RL interacts with the environment by trial and error method to learn the best policy for selecting an action for maximising the discounted cumulative reward in the longer term. The RL environment can be defined as a type of Markov Decision Process (MDP) which contains the tuple $< \boldsymbol{S}, \boldsymbol{A}, \boldsymbol{P}, \boldsymbol{R}, \boldsymbol{\gamma} >$ where $\boldsymbol{S}$ is the state space, $\boldsymbol{A}$ is a set of actions, $\boldsymbol{P}$ is the state transition probability function, $\boldsymbol{R}$ is known as the reward function and $\gamma \in (0, 1)$ is a discount factor.

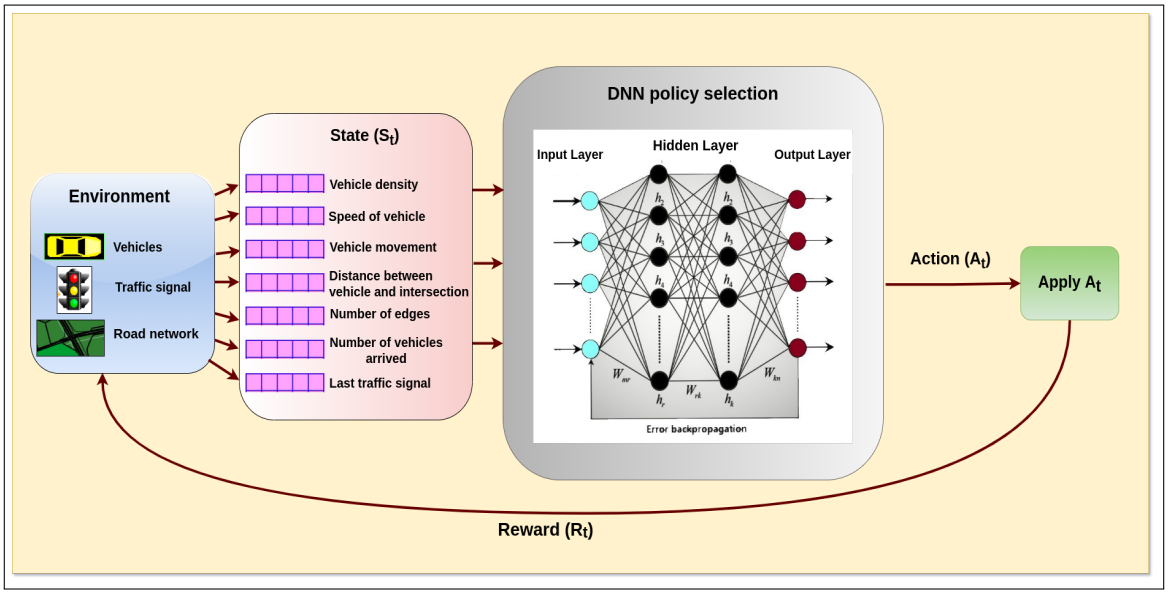


Fig. 1. Reinforcement Learning Framework

In the present work, two simulators Simulation of Urban MObility (SUMO) and Traffic Control Interface (TraCI) have been considered in order to create the environment and the state of the environment is not only dependent on traffic signals of the road network but also the vehicles present in it. At each discrete time steps t, the current state $s_t \in \boldsymbol{S}$ is received by the agent (Fig. 1). It then selects an action $a_t \in \boldsymbol{A}$ to perform and it is acted upon the environment. The agent obtains a reward R_t for its action and the environment gets moved to a new state $s_{t+1} \in \boldsymbol{S}$. If the agent's action leads to a favourable environmental reward, then the chance of carrying out such an action will be increased otherwise it will decrease. However, RL algorithms face challenges due to complex and enormous state-action pairs of vast road networks.

3.2 DRL:

In contrast to RL, DRL seems to be more powerful and stabilised, particularly for high-dimensional state-action problems. The behaviour of an agent is defined by a policy π ($\pi : \boldsymbol{S}$ X $\boldsymbol{A} \rightarrow \boldsymbol{R}$) in RL. In DRL, weights and biases of the neural network, which can also be called as a set of parameters θ, are described to parameterise the policy π, which is

defined as π_θ. A probability distribution over actions is returned by π_θ. The task of an agent is to follow a policy $\pi_\theta(a_t|s_t), a_t \in \boldsymbol{A}, s_t \in \boldsymbol{S}$. The agent is trained for numerous episodes (K), each of which has multiple iterations (T). The cumulative sum of discounted reward at t^{th} iteration of an episode (G_t) is defined as follows (Equ. 1):

$$G_t = \sum_{i=t}^{T} \boldsymbol{\gamma}^{i-t} R_i \qquad [11] \tag{1}$$

3.3 Policy Gradient:

Function approximations based on Neural Network are sufficient to make RL effective at high dimensional state spaces (i.e. a policy used to map input traffic to a signal traffic control measurement). It could either be accomplished using action-value methods or by directly learning a parameterised policy using policy gradient method by measuring the value of actions. The policy gradient algorithm is intended by a gradient-ascent approach to optimise the parameterised policy function $\pi_\theta(a_t|s_t)$ to maximise the expected value of G i.e. ($J(\theta)$).

As stated by the policy gradient theorem (Equ. 2), the derivative of the expected reward ($\nabla J(\theta)$) is the expectation of the product of the discounted reward (G_t) and gradient of the log of the policy $\pi_\theta(a_t|s_t)$.

$$\nabla J(\theta) = E_{(\pi,\theta)}[G_t \nabla log \pi_\theta(a_t|s_t)] \tag{2}$$

The parameters of the neural network are modified using the following update rule (Equ. 3) for each episode k, according to gradient ascent, until $J(\theta)$ is maximised.

$$\theta_{k+1} = \theta_k + \alpha \nabla J(\theta_k) \qquad [11] \tag{3}$$

3.4 The method framework:

The agent is trained using policy gradient algorithm which is outlined in Fig. 2. Let the number of intersections in a road network is N. Initially, the parameters (θ) of the neural network are set to random values using normal distribution and the replay memory (*EPISODE*), which serves as a buffer, is set to null. The current state of the environment at time step t is s_t which is sent to the DNN model. The DNN generates a probability distribution set (P_t) over actions for s_t. The P_t consists of the probability distribution of action at intersection n (p_t^n), $\forall n \in N$ (Equ. 4).

$$P_t = \{p_t^n, \forall n \in N\} \tag{4}$$

If the value of p_t^n is greater than 0.5 then the value of the action at n (a_t^n) is set as 1 otherwise 0. An action set (A_t) is formed which consists of $a_t^n, \forall n \in N$ (Equ. 5).

$$A_t = \{a_t^n, \forall n \in N\} \tag{5}$$

Subsequently, A_t is applied in the environment and reward R_t, next state s_{t+1} is received. Then, the tuple $< s_t, P_t, A_t, R_t, s_{t+1} >$ is stored in the buffer *EPISODE*. If one episode is completed i.e. all the T iterations of one episode are over, then for each tuple in *EPISODE*, the following steps are executed. Since, a single agent controls multiple intersections, the agent at first calculates the average binary cross entropy loss ($Loss_t$) $\forall N$ (Equ. 6).

$$Loss_t = \frac{-1}{N} \sum_{\forall n \in N} (p_t^n \; log_2 a_t^n + (1 - p_t^n) \; log_2 (1 - a_t^n)) \tag{6}$$

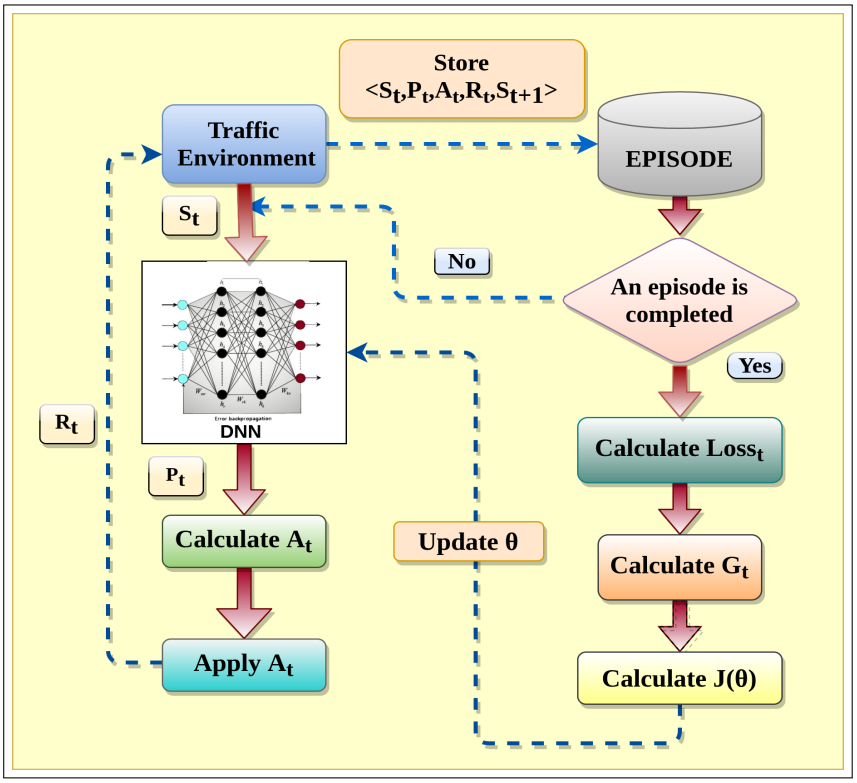


Fig. 2. Steps of Policy Gradient

Then, it calculates G_t (Equ. 1) and $J(\theta)$ ((Equ. 7)) in turn.

$$J(\theta) = Loss_t * G_t \tag{7}$$

The value of θ is updated in DNN using optimizer and Equ. 2, 3. All these steps are repeated until K episodes are completed.

3.5 Problem Statement:

A method for reduction of traffic congestion with DRL is proposed that leads towards determination of an ideal approach for estimating the duration of the traffic signal, as per the state of the environment. In order to attenuate traffic congestion, the issue of adjusting traffic signals is overlaid. Moreover, a good correlation between the signals enables vehicles to move through intersections quickly. Thus, with the help of the policy gradient algorithm, a single DRL agent is used to monitor the traffic signals of all intersections of a dynamic road network. To optimise the traffic signal, the agent is trained in FCNN, CNN, LSTM and GRU. The agent collects all the necessary information about traffic flow from SUMO and TraCI and then, it analyses the information and uses the aforementioned DNNs in order to determine an action depending on the current state.

3.6 Setting up the learning environment:

The principal constituents of the DRL environment are identified, e.g. state, action and reward. The DRL agent efficiently decides the traffic flow characteristics by learning the right action selections procedure at the current state.

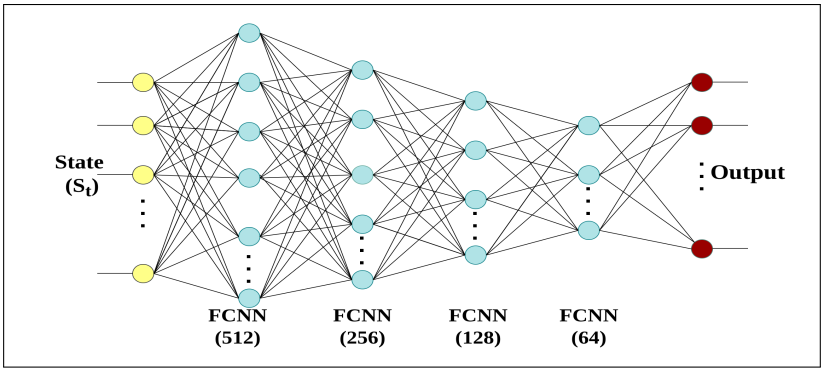
- **State:** The learning skill of an agent has a close connection with a precise and achievable state classification. For managing traffic signals, multiple state representations are used. Based on the current traffic scenario at and near the intersection, the agent determines how to monitor these traffic signals. The state description of the present work comprises vehicle density, the distance between the vehicle's current location and the intersection, direction of vehicle movement, speed of vehicles, a lane's number of edges, number of vehicles arrived in the network at the last time step and the final traffic signal of a lane.
  In the state representation the temporal information is used as well. The agent uses a five-framed stack to receive sufficient temporal details about the environment to accommodate the state of the last five time steps.
- **Action:** The action that has been taken by the agent in the specified state results in a new state and has a remarkable impact on learning. As the traffic signals from several intersections are regulated by one single agent, the gained space-time comprehension of the entire network is one of the most important feature of the agent. This aims to improve the strategy by integrating each and every traffic signals present in the network. Here, the agent decides which lane will receive the green signal in the next time step. Primarily it defines the duration of the signals of the lanes of the road network to minimize congestion.
- **Reward:** As $R_t \in \mathbb{R}$, the reward R_t is definitely a scalar attribute. This reward is to be assessed, i.e. to penalise or reward the agent for its subsequent actions. The objective of the agent in the current work is to bring down the average waiting time of the vehicles. The waiting time is determined by the summation of time the vehicles stop. Positive reward means that the actions done in the current phase reduce the waiting time, where, by the term "negative reward", it is meant the waiting time in the current state is improved from the preceding. The agent decides in the future to adjust its behaviour for certain states of the system, depending upon these rewards.

3.7 Network architecture

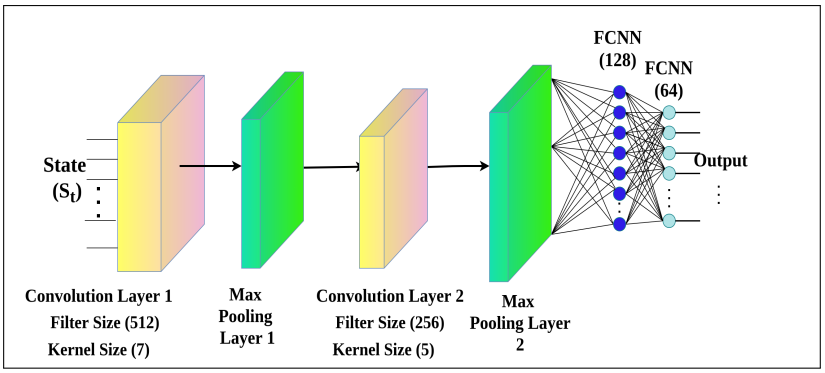
Four different DNN models are used in this work (Fig. 3). The size of the state of the environment determines the number of the input nodes for all the networks. It is known that a Fully Connected layer operates on flattened inputs where each input is connected to all of the nodes present in the network. Here, in case of FCNN, there are four flattened hidden layers with 512, 256, 128, 64 number of nodes respectively. The CNN uses a set of 1-Dimensional convolution layers and 1-Dimensional max pooling layers twice in this work. Two convolution layers are used with filters and kernel size of (512, 7) and (256, 5) respectively in each layer. The output of the last max pooling layer is then propagated to two fully connected layers with 128 and 64 nodes respectively.

Recurrent Neural Network (RNN) models are also used here in addition to FCNN and CNN models. As because the traffic signal is a sequence based decision making process, RNN is best suited for this sort of problem. Yet short-term memory fails with RNN. LSTM and GRU are developed as a mitigation for this limited memory. Both LSTM and GRU have physical properties called gates that can accommodate the broad information sequence [12].

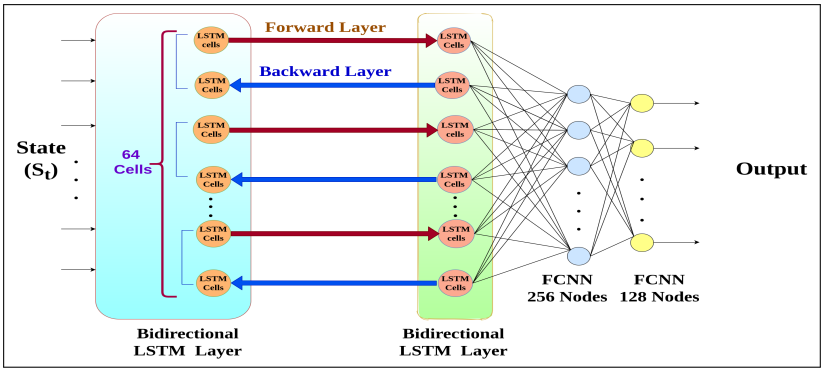
In the case of LSTM, two bidirectional LSTM layers with 64 LSTM cells in each layer are taken into consideration. The output of the LSTM layer is passed to two flattened layers with 256 and 128 nodes. On the other hand, The structure of the GRU network is the same as that of LSTM. Only the GRU layer is used instead of the LSTM layer.



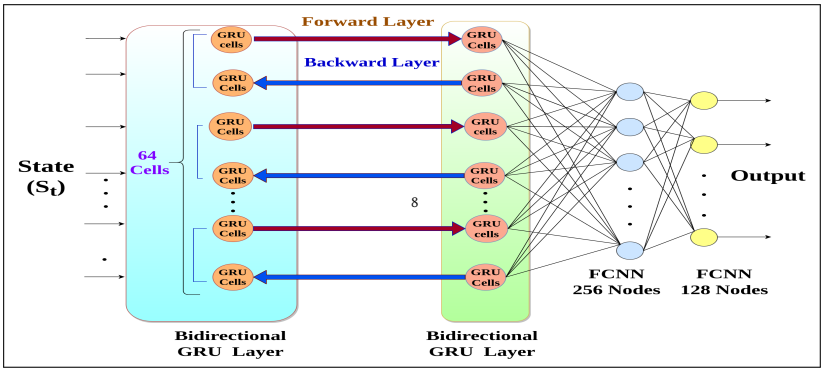
(a) FCNN



(b) CNN



(c) LSTM



(d) GRU

Fig. 3. Network architecture

For all the aforementioned networks, in the nodes of the hidden layer, Rectified Linear Units (ReLU) activation function is applied. Another problem that might be faced is data overfitting. A dropout layer has been added in order to prevent data overfitting after each fully connected layer. The number of output nodes is dependent on the number of signalised intersections in the network.

4 SIMULATION EXPERIMENTS

Flow simulator is used in order to evaluate the performance of the proposed system. OpenAI Gym and SUMO helped in implementing Flow. Design of road network, generation of traffic flow, controlling the traffic signal can be done by some flexible Application Program Interface (APIs), provided by SUMO. The MDP environment of RL is implemented using OpenAI Gym module. In the present work, two real-time road networks are considered, which are downloaded from OSM. With the help of OSM, real time maps can be obtained. The system specification required for conducting the simulation experiment is elaborated in Table 1.

Table 1. System Configuration

GPU Version	Nvidia Tesla T4
GPU Memory	16 GB
Python Version	3.7
TensorFlow Version	2.3

4.1 Environment hyper-parameters:

The hyper-parameters established in table 2 are used for the method proposed.

Table 2. Environment hyper-parameters

Hyper-parameters	Value
α	0.001
γ	0.99
K	100
T	400
Optimizer	Adam optimizer

4.2 Simulation environment:

Two road networks with numerous number of signalised intersections are considered in the present work. The number of vehicles considered in both of the road networks is 150.

- **Small scale network:** The number of traffic signals in this network is 17
- **Large scale network:** The number of traffic signals in this network is 35

4.3 Simulation metrics:

The performance of the proposed method is assessed on the basis of the following simulation metrics:

- **Fuel emission:** It measures fuel consumption of each vehicle and it is measured in ml/sec.

(a) Small scale network (b) Large scale network

Fig. 4. Simulation Networks

- **Waiting time:** It means the time the vehicles are waiting in a red signal. The measurement of this parameter is in sec.
- **CO_2 emission**: Quantifying CO_2 released by each vehicle is termed as CO_2 emission and it is calculated in mg/sec.

4.4 Simulation results:

For the two complex real-time road networks, two sets of simulation experiments are performed. Each experiment compares the efficiency of the proposed method against a baseline for FCNN, CNN, LSTM and GRU.

Fig. 5a- Fig. 5d show the plots of waiting time, CO_2 emission, fuel consumption of vehicles and penalty vs. iterations for small scale network. Fig. 5e- Fig. 5h show the plots of waiting time, CO_2 emission, fuel consumption of vehicles and penalty vs. iterations for large scale network.

In terms of average waiting time, average CO_2 emission, average fuel consumption and average penalty, the suggested model is compared in each network which is illustrated in Table 3 and the values for both networks is calculated for a single episode.

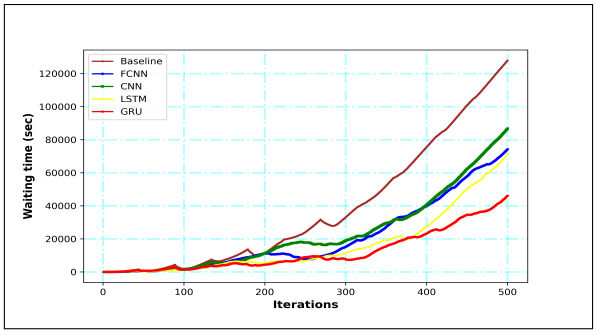
Table 3. Fuel emission, waiting time, CO_2 emission and penalty for an episode for small and large network

Network	Small scale network					Large scale network				
Parameters	Baseline	FCNN	CNN	LSTM	GRU	Baseline	FCNN	CNN	LSTM	GRU
Fuel emission	331.12	283.90	315.55	299.45	281.45	1115.41	1045.43	1042.91	985.02	978.05
Waiting time	30323.38	13541.18	23242.19	16367.20	12248.48	109958.57	107775.52	97734.91	76946.81	74619.51
CO_2 emission	769432.88	659877.74	733344.55	695973.08	654225.09	2591901.02	2429270.30	2423600.89	2289264.52	2273034.11
Penalty	-34.62	-25.02	-31.52	-27.19	-24.92	-117.25	-111.25	-103.04	-92.75	-94.46

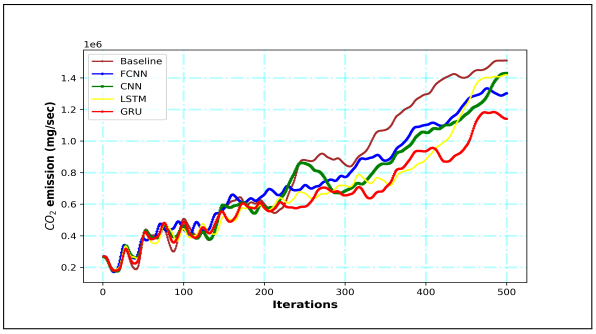
4.5 Discussion of Results:

From Fig 5a- Fig 5f, the proposed model can be seen to function better in the GRU network than any other network models.

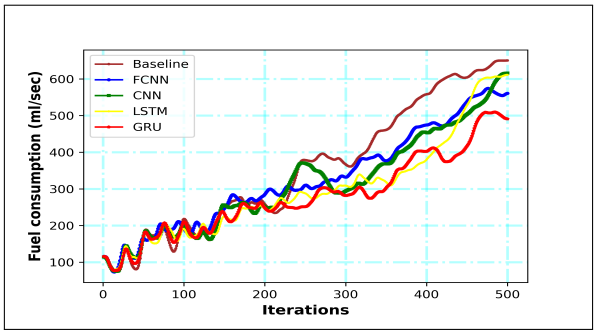
The values shown in Table 3 are comparable, suggesting that GRU outperforms all other networks. Using the data in the Table 3 it can be found that while comparing to baseline-



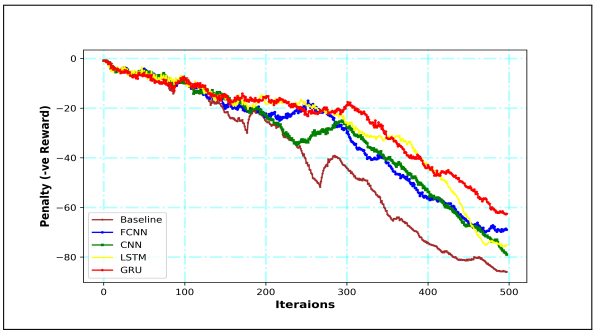
(a) Waiting time vs. Iterations



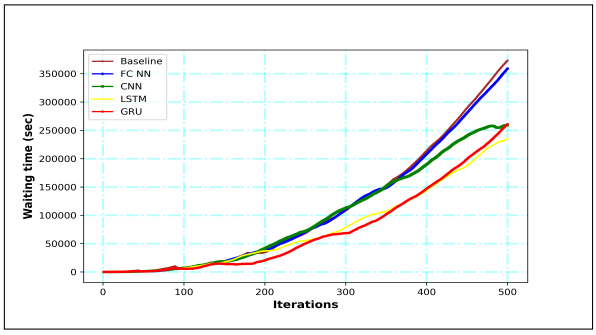
(b) CO_2 emission vs. Iterations



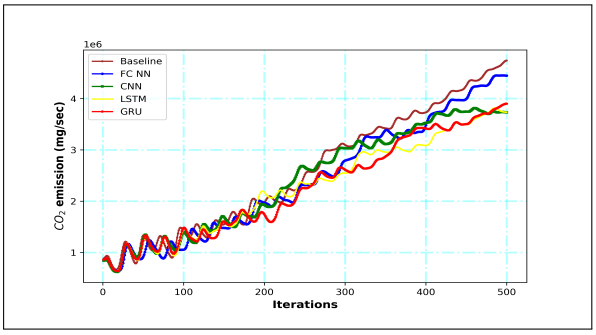
(c) Fuel consumption vs. Iterations



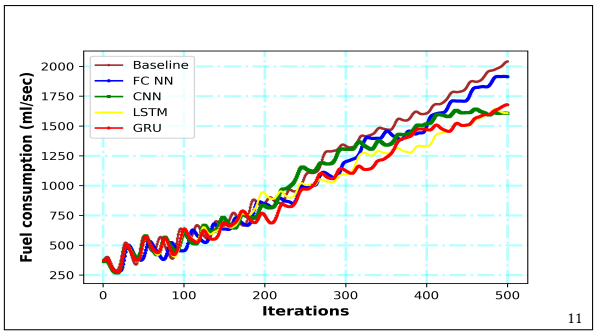
(d) Penalty vs. Iterations



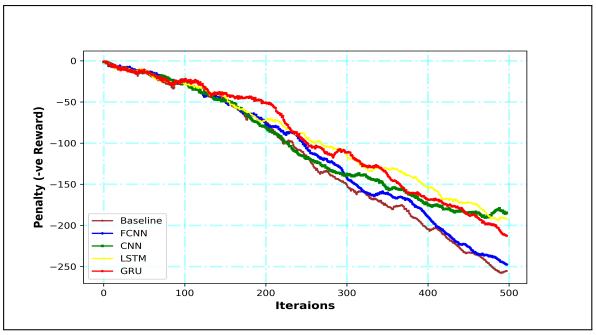
(e) Waiting time vs. Iterations



(f) CO_2 emission vs. Iterations



(g) Fuel consumption vs. Iterations



(h) Penalty vs. Iterations

Fig. 5. Simulation results: (a)-(d) for small scale network, (e)-(h) for large scale network

For small scale road network, in FCNN, CNN, LSTM and GRU, there are

- 55.34%, 23.25%, 46.02%, 59.60% reductions in average waiting time.
- 14.24%, 4.69%, 9.55%, 14.97% reductions in average CO_2 emission.
- 14.26%, 4.70%, 9.56%, 15.00% reductions in fuel emission.
- 27.72%, 8.95%, 21.46%, 28.01% reductions in penalty.

Moreover, for large scale road network, in FCNN, CNN, LSTM and GRU, there are

- 2.21%, 11.12%, 30.03%, 32.14% reductions in average waiting time.
- 6.63%, 6.50%, 11.12%, 12.24% reductions in average CO_2 emission.
- 6.28%, 6.50%, 11.60%, 12.24% reductions in fuel emission.
- 5.52%, 12.22%, 20.09%, 19.05% reductions in penalty.

5 CONCLUSION AND FUTURE WORK

This work provides a fully trainable DRL agent which is capable of operating numerous traffic signals in a dynamic real-time road network. The current work has encapsulated two state-of-the-art innovations, RL and Deep Learning, which is widely used as an optimisation technique on traffic signal management. By means of policy gradient algorithm, the agent can associate with information about the road traffic and this algorithm is assessed in four various DNNs, namely FCNN, CNN, LSTM, GRU. The waiting time, CO_2 emission and fuel consumption of vehicles have been reduced remarkably with the framework proposed. Furthermore, it is claimed to be a crucial factor in making it possible to deploy the policy gradient algorithm with the DRL agent in a real-life environment. Two dynamic real-time maps from OSM are considered here in order to check whether the algorithm works on both the small scale and large scale maps. In the conclusion, the policy gradient algorithm in the GRU network has been found to be the most effective in terms of performance rather than other DNNs.

The future debate may be driven by more complicated RL algorithms (e.g. A2C, PPO), which may be implemented in more advanced DNN models namely, Transformer, Linformer and Performer.

REFERENCES

[1] Sidina Boudaakat, Ahmed Rebbani, and Omar Bouattane. Smart traffic control system for decreasing traffic congestion. In *International Conference on Systems of Collaboration Big Data, Internet of Things & Security (SysCoBIoTS)*, pages 1–6. IEEE, 2019.

[2] Ananya Paul and Sulata Mitra. Dynamic traffic light control mechanism in VANET. In *International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, pages 436–440. IEEE, 2018.

[3] Ananya Paul and Sulata Mitra. Real-time routing for ITS enabled fog oriented VANET. In *17th India Council International Conference (INDICON)*, pages 1–7. IEEE, 2020.

[4] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

[5] Peter Henderson, Riashat Islam, Philip Bachman, Joelle Pineau, Doina Precup, and David Meger. Deep reinforcement learning that matters. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[6] Ananya Paul and Sulata Mitra. Deep reinforcement learning based traffic signal optimization for multiple intersections in ITS. In *International Conference on Advanced Networks and Telecommunications Systems (ANTS)*, pages 1–6. IEEE, 2020.

[7] Seyed Sajad Mousavi, Michael Schukat, and Enda Howley. Traffic light control using deep policy-gradient and value-function-based reinforcement learning. *Intelligent Transport Systems*, 11(7):417–423, 2017.

[8] Stefano Giovanni Rizzo, Giovanna Vantini, and Sanjay Chawla. Time critic policy gradient methods for traffic signal control in complex and congested scenarios. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1654–1664, 2019.

[9] Li Li, Yisheng Lv, and Fei-Yue Wang. Traffic signal timing via deep reinforcement learning. *IEEE/CAA Journal of Automatica Sinica*, 3(3):247–254, 2016.

[10] Elise Van der Pol and Frans A Oliehoek. Coordinated deep reinforcement learners for traffic light control. *Proceedings of Learning, Inference and Control of Multi-Agent Systems (NIPS)*, 2016.

[11] A. Haydari and Y. Yilmaz. Deep reinforcement learning for intelligent transportation systems: A survey. *arXiv preprint arXiv:2005.00935*, 2020.

[12] Ananya Paul and Sulata Mitra. Management of traffic signals using deep reinforcement learning in bidirectional recurrent neural network in ITS. In *International Conference on Intelligent Systems, Metaheuristics Swarm Intelligence (ISMSI)*. ACM, 2021.