**Student Name:** Diwash Poudel

**Student Id:** 220075907

**Module Code & Name**: CET351 – Research

**Programme:** Computer System Engineering

**Assignment Title:**

CET351 Assignment 2 Research Paper

**College:** ISMT, Kathmandu, Nepal

**Word Count:** 3300

**CET351- Research**

**Traffic Light Scheduling Framework for SDN-Enabled Smart Transportation System Using Deep Reinforcement Learning**

**Diwash Poudel**

**Abstract**

This paper suggests a traffic-light scheduling system for congested city regions that makes use of deep reinforcement learning and a software-defined control interface. The architecture involves a software-defined control system for generating traffic light control signals and monitoring traffic conditions. In order to intelligently operate traffic lights, a Deep Reinforcement Learning (DRL) model is used, taking into account inputs from the real-time traffic environment such as vehicle count, speed, and density. Congestion detection and prevention signals are generated by a threshold policy on the control server. Together with the signals for congestion prevention, the DRL agent creates efficient traffic light control signals. The suggested framework is tested using SUMO simulator simulations on an Indian city's OpenStreetMap. Comparative results show increases in performance parameters like throughput, speed, queue length, and average waiting time. These results show how the suggested method can improve traffic flow and relieve congestion.

**Introduction**

In order to reduce congestion in congested urban areas, this paper suggests a traffic-light scheduling system that blends deep reinforcement learning and software-defined control. The framework attempts to improve throughput and speed in metropolitan areas by intelligently operating traffic lights using a Deep Reinforcement Learning (DRL) model and real-time traffic data. To create traffic light control signals and keep track of traffic conditions, a software-defined control system is used. The DRL agent provides efficient traffic light control signals by identifying congestion using a threshold strategy and collaborating with congestion prevention signals. The proposed system shows gains in performance parameters including average waiting time, throughput, queue length, and speed through simulations on an Indian city's OpenStreetMap. This approach leverages the advantages of software-defined networking and deep reinforcement learning to enhance traffic management in smart transportation systems, addressing the limitations of traditional traffic light control systems.



**Fig.** The map that was used for the study of sumo

Multi-Phase Adaptive (MPA) is a traffic lighting system that uses advanced algorithms to optimize intersection traffic flow (Vikash Gayah, 2023). It adjusts signal timing based on real-time conditions, considering factors like traffic volume and pedestrian demand (Farhad Pooran, 2001). MPA relies on accurate data from sensors and cameras, but may face challenges with sudden fluctuations and sensor quality. Regular maintenance and adjustments are needed to ensure optimal performance and address limitations (Najmeh Samadiani 1, 2019).

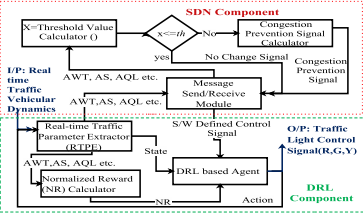
Fixed-Time Traffic Light Control (FTLC) operates on fixed timing patterns, allocating signal phases based on predetermined schedules (Mansoure Kafash, 2013). It lacks adaptability to real-time traffic conditions and cannot optimize traffic flow. FTLC's inflexibility may lead to inefficient operations, longer wait times, and increased congestion (AKÇELIK, 2011). Upgrading to advanced systems like adaptive or intelligent traffic lights can overcome these limitations by dynamically adjusting signal timings based on traffic demands (Li, et al., 2007).

In this study, three traffic signal control algorithms—SDDRL (Self-Drive Deep Reinforcement Learning), NFM (Novel Fuzzy Model), and DQN (Deep Q-Network)—are evaluated. While NFM uses fuzzy logic to find the best timings based on linguistic norms, SDDRL uses AI agents to dynamically change signal timings based on current traffic conditions. DQN combines Q-learning and deep neural networks to estimate optimal action-values for traffic light control. The performance of these algorithms will be assessed using travel time, congestion level, average reward, Q-value convergence, and exploration rate. The objective is to enhance traffic efficiency, reduce congestion, and minimize waiting times at intersections through these intelligent control algorithms.

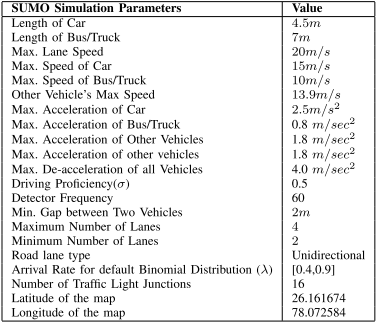
The three algorithm are:

1. Self-Drive Deep Reinforcement Learning (SDRRL)
2. Novel Fuzzy Model (NFM)
3. Deep Q Network (DQN)
4. SDRRL (**Algorithm 1**)

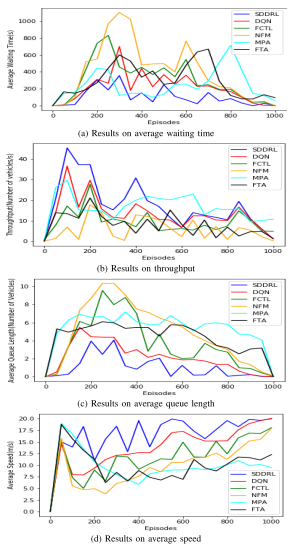
DRL and SDN are the two main parts of SDDRL. Real-time vehicular traffic dynamics are first inputted into the component by sensors placed at each intersection. In order to calculate the normalized reward, the NR calculator module receives essential parameters from the RTPE module, such as Average Waiting Time (AWT), Average Speed (AS), Average Queue Length (AQL), and Throughput (TP). Message send/receive module of the SDN component, which exchanges the message between SDN and DRL components, is simultaneously sent these parameters. Current traffic state is determined by the RTPE module using traffic parameters and provided to the DRL-based agent. The Threshold Calculator module of the SDN component is activated at the same time to calculate the threshold (X). If X is below the predetermined threshold, no change signal is generated because there is no congestion on the route; otherwise, the Congestion Prevention Signal Generator module is called to generate the congestion prevention signal. Use of this signal modifies or updates the phase duration set by the relevant traffic light controller. Following the completion of this phase, the DRL agent generates the output, the traffic light control signal. (Kumar, et al., 2021).



**Fig.** Logical flow of SDDRL



The vehicular network simulation utilized the SUMO tool, integrated with Python through the Traffic Control Interface (TRACI). Traffic was generated on OpenStreetMap using the randTrips.py script, controlling arrival rates based on rush hour observations. The yellow signal duration was set at three seconds for easier signal switching. The deep network simulation considered parameters leaky ReLU, discount factor, and learning rate. The software-defined control involved five control nodes connected to a server, considering factors like waiting time, queue length, speed, and a threshold value for congestion evaluation. Setting the threshold congestion value to 0.5 improved traffic flow and performance metrics, with a suggested value of 0.6 adjustable based on specific application requirements.



**Fig.** Relative consequences of SDDRL with other best in class calculations

The exploratory discoveries and similar investigation of SDDRL were examined utilizing four significant execution measurements. SDDRL's behavior was examined under a variety of vehicle arrival rates. Fig. ( The outcomes for AWT, TP, AQL, and AS are depicted in a), b), c), and d, respectively. The performance metric and the episodes are represented by the x-axis. The figures show different appearance rates: 1, 2, 2.5, 3, and 3.5 times.

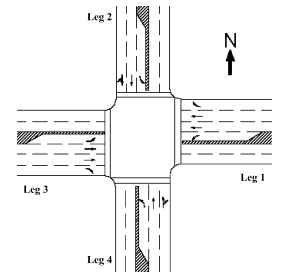
From the outcomes, it is seen that AWT and AQL are negligible for extremely low appearance rates, as would be considered normal. However, the primary objective of this work is to ensure that SDDRL maintains its performance even for higher arrival rates (2.5x, 3x, and 3.5x) or congested routes.. On the other hand, AS exhibits a reciprocal behavior, which is desired. Moreover, TP is fundamentally kept up with for all situations. However, it has been observed that TP decreases with increasing episodes. This is because the simulation is divided into episodes, and as vehicles complete their journeys, the number of vehicles decreases. Therefore, TP decreases with an increasing number of episodes.

The results indicates that SDDRL remains effective in handling peak traffic with minimal performance variations. The DRL component is trained with a comprehensive reward system, while the SDN control server monitors congestion and responds dynamically.

1. NFM (**Algorithm 2)**

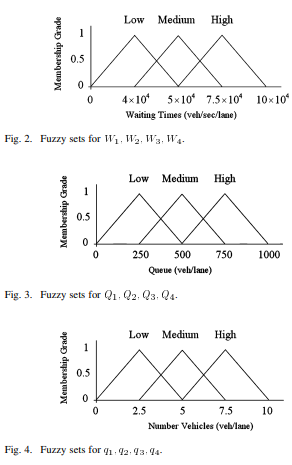
A Novel Fuzzy Model The development of models of physical processes using fuzzy logic is a novel concept. Fluffy models are less remotely intricate; they are very suitable for non-linear processes and easy to comprehend. Models with fewer rules are more beneficial. A crucial aspect of a metropolitan traffic organization is the cycle chosen in this study, which reduces the amount of time that cars spend waiting at specific intersections.

The figure shows the two Stages Signalized Convergence form, which is used in this paper for referencing and presenting single crossing points. Stage 1 is represented by Legs 1 and 3 and Stage 2 by Legs 2 and 4. The length of the line is a significant factor in determining the traffic situation at an intersection.



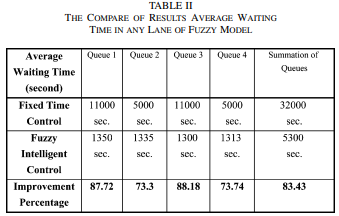
**Fig.** Signalized junction of two phase

The fixed-time control and fuzzy intelligent control models both considered the following control variables: Infers that the traffic signal is green in paths 2 and 4 and red in paths 1 and 3 for stage 1 convergence. The cars should halt in pathways 1 and 3 and can proceed in paths 2 and 4. A traffic signal that is red in lanes 2 and 4 and green in lanes 1 and 3 denotes a phase 2 intersection, in contrast. Consequently, the vehicles can go in paths 1 and 3 and they ought to stop in paths 2 and 4.

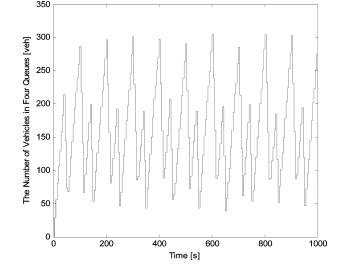


**Fig.** a fuzzy set graph

The principal phase of fluffy displaying includes choosing execution factors for a solitary crossing point traffic framework. Input and output variables were previously described. The output variables represent the state variables, which serve as input for the fuzzy model. Choosing the fuzzy sets or linguistic variables for the inputs and outputs is the second stage. The fuzzy sets in this instance are divided into three categories: "Low (L)," "Medium (M)," and "High (H)." The following step involves creating fuzzily defined relationships or rules between the inputs and outputs using the data sequences from the urban transportation network that are currently accessible. These regulations are written as if-then sentences, with "Go" denoting the extension of the green phase and "STOP" denoting its end. Fuzzy numbers describe the characteristics of parameters and. In the final stage, fuzzification algorithms are used to calculate the precise numerical crisp value of the output, which represents the number of vehicles joining the queue within a particular time interval for the supplied inputs. In fuzzy logic systems, the precise output value is frequently determined using the center of gravity fuzzification technique (Azimirad, et al., 2010).

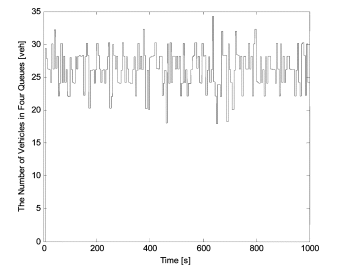
****

For a single intersection, a fuzzy model of the urban traffic network was developed. The length of the lines and the average wait time for automobiles in each lane depend on the status of the variables. On the other hand, a fuzzy signal controller was developed in order to display the percentage of improved traffic. The outcomes of the diversion and the degree of advancement demonstrate that the Cushioned Canny of Controller, which differs from Fixed-time Control, reduces the frequent holding up of time vehicles in any way of crossing point. The organizing and study of different fluffy demonstrating and control measures in non-disengaged convergences and complicated crossing spots in urban rush hour.



**Fig.** The number of vehicles in the four queues at the intersection without a controller.

The numbers above show how many cars are parked in four lines at an uncontrolled intersection. The graph graphic depicts 1000s of time simulations, and every 50 seconds the intersection's total number of vehicles and queues was displayed. According to the illustration, each queue represents a lane or traffic direction at the intersection. The number of vehicles in each queue represents the current volume of traffic or level of congestion in that lane. Without a controller, traffic is uncontrolled and moving cars are not coordinated to move simultaneously. The graph shows the increase in each queue's number of vehicles over time.

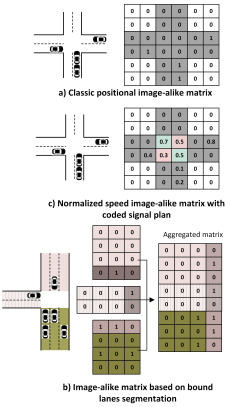


**Fig.** The number of vehicles in each of the four intersection queues using a fuzzy controller.

The graph displays the number of vehicles in four queues, each representing a separate traffic lane, at a crossroads without a controller. It sheds light on how traffic is distributed, how flows, and how a traffic signal controller affects things. Changes in vehicle numbers indicate traffic events and help evaluate the controller's effectiveness in managing traffic and achieving balance. These insights support informed decision-making to optimize traffic flow and reduce congestion.

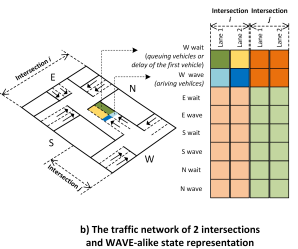
1. DQN (**Algorithm 3**)

The described working principle is focused on state representation in coordinated control across many junctions, notably in the context of the Deep Q-Network (DQN) for traffic control. Most methods in this sector demand familiarity with the current traffic signal setup. Placing this data in the center of the binary matrix representation, which resembles an image, is one method to include it.

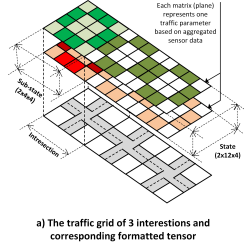


In the context of deep learning, the most popular traffic state configurations for a particular intersection are represented by a binary matrix, in which each element stands for a road network segment or cell. With binary features indicating the color of each traffic light, the traffic light configuration is represented as an additional layer in the state space in this representation. However, adding the traffic light configuration as an additional layer substantially expands the state space, posing memory and computation difficulties during learning. To address this, alternative approaches are used.

In these approaches, A Hawk-dimensioned two-dimensional tensor is constructed from traffic data gathered from various sensors. The tensor has multiple channels (C) representing different traffic parameters such as the number of halting vehicles and mean speed. The height (H) and width (W) of each channel correspond to the segments or cells in the road network.



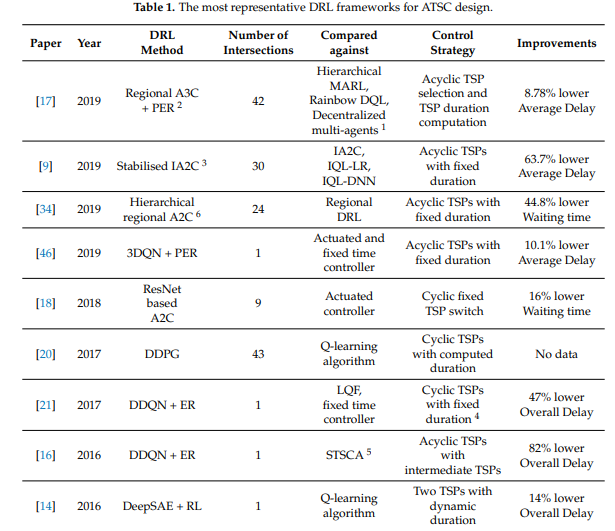
The state tensor represents the overall state of the traffic network, with each intersection represented by sub-states. Each sub-state corresponds to one intersection bound and contains segments representing traffic lanes for each direction. The aggregated traffic data is assigned to the respective segments based on their channel affiliation.



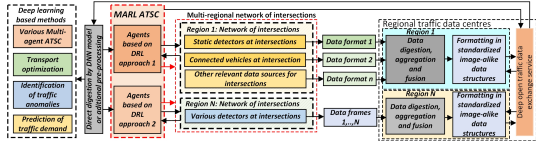
**Fig (a and b).** most frequently used traffic state configurations for the extensive intersection network.

An organized and thorough representation of the traffic network is provided by the state representation setup in Figure a, which makes it easier to implement multi-agent DQL control frameworks. In the overall state tensor, each network intersection is represented by a sub-state. Two segments inside this sub-state stand in for the lanes of traffic in either direction at a certain intersection bound. These segments include channel affiliation and traffic metrics in their aggregated traffic data. Figure provides an overview of the complete configuration of this state representation, highlighting the included sub-states. This representation enables a detailed understanding of the traffic network and supports the implementation of effective control strategies using multi-agent DQL frameworks.

Overall, these working principles focus on representing the state of the traffic network in a structured manner, incorporating relevant information such as traffic light configurations and traffic parameters (Martin Gregurić 1, 2020).



The decentralized multi-agent technique is based on distributed constraint optimization, multi-step returns, and off-policy A3C, as seen in the Table. It is kept stable by a spatial discount factor and fingerprinting. Additionally, the STSCA, a shallow ANN-based traffic signal control agent, is contrasted with it. The Wolpertinger Entertainer Pundit MLP engineering and DDPG calculation are also used by Every Specialist.



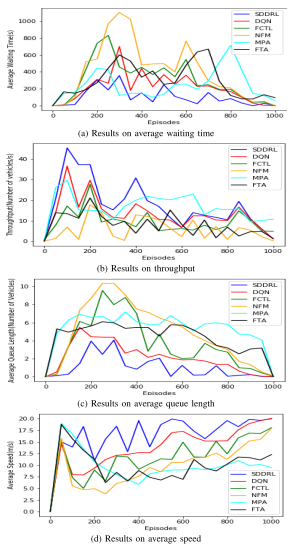
**Fig.** The intersection network utilized the Deep Open Traffic Data concept.

The use of Deep Reinforcement Learning (DRL) algorithms in Advanced Traffic Signal Control (ATSC) is supported by the data in this section. In comparison to conventional algorithms and RL-based techniques, recent implementations of DRL in ATSC have demonstrated encouraging outcomes. The effectiveness of traffic light control is increased through the use of DRL, machine learning, multi-agent systems, and intricate Deep Neural Network (DNN) models. Future large-scale ATSC is thought to benefit from these techniques. The paper examines improvements in state, action, and reward modeling, emphasizing scaling, data accessibility, learning convergence, and DNN model complexity as important research areas. In order to implement sophisticated traffic signal control systems in real-world settings, practical options are shown.

Standardized image-like state representation is crucial for interoperability and scalability in ATSC using DRL. Centralized data acquisition and preprocessing, along with Deep MARL and transfer learning, can enhance multi-intersection signal control. Multi-objective evolutionary algorithms optimize complex DRL frameworks, while DOTD integrates traffic, weather, and road condition data for advanced ATSC using DQN and DRL algorithms.

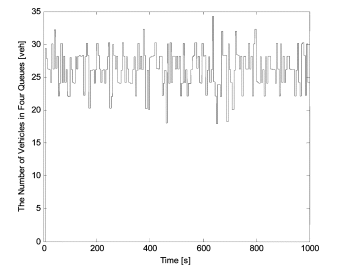
**Comparison of different algorithm**

The comparison between SDDRL (Self-Dueling Deep Reinforcement Learning), NFM (Novel -Fuzzy Model), and the precise traffic scenario, dataset, and implementation details all influence the DQN (Deep Q-Network)'s traffic signal control accuracy.



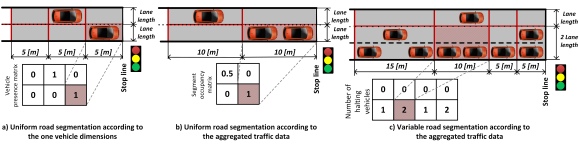
**Fig.** Graphical comparison of SDDRL's results with those of other cutting-edge algorithms

Figures (a)-(d) look at the exhibition measurements, SDDRL outperforms by 66.62 percent, 49.70 percent, 30.3%, 54.77 percent, and 28.34 percent, respectively. For TP, SDDRL works on by 66.60%, 47.06%, 26.88%, 54.41%, and 24.76% over similar models. The performance of AQL is enhanced by 69.80%, 59.25%, 61.73%, 63.38%, and 30.89% when compared to the aforementioned models. In terms of AS, SDDRL performs better than NFM, FCTL, MPA, FTA, and DQN by respective margins of 40.90%, 29.55%, 42.83%, 43.67%, and 16.62%. When compared to NFM and DQN cutting-edge models, SDDRL shows a considerable improvement in AWT, TP, and AQL because it takes into account constant traffic boundaries and generates strong prize-based traffic signal management signals.



**Fig**. The number of vehicles in each of the four intersection queues using a fuzzy controller.

This graph reveals how traffic is distributed among lanes, indicating higher or lower vehicle numbers and flow in different directions. It provides insights into traffic dynamics, the impact of the signal controller, and helps assess its effectiveness in managing flow and maintaining balance. Changes in vehicle numbers reflect traffic events and flow patterns, enabling informed decisions for traffic optimization and congestion reduction. NFM has been successfully applied in various domains, including traffic signal control, its performance in comparison to DRL-based approaches like SDDRL and DQN may vary depending on the specific problem and dataset.



**Fig.** Road segmentation in matrices with similar images

Figure (a) shows that the segment length corresponds to the average or minimum length of the cars when vehicle parameters are known. The same approach can be applied to other traffic data such as passenger count or vehicle direction. Figure (b) shows a cellular automata-style segmentation for aggregated data, while Figure (c) introduces a novel approach with variable segment sizes based on aggregated traffic data. Segments farther from the stop line are longer, allowing for better coverage. Occupancy is determined by the presence of at least one vehicle in a segment. DQN has proven to be far more effective than conventional traffic signal control techniques.. However, its performance can be sensitive to hyperparameters settings, network architecture, and exploration-exploitation trade-offs.

**Conclusions**

The paper introduces three algorithms: Self-Drive Deep Reinforcement Learning (SDRRL), Novel Fuzzy Model (NFM), and Deep Q Network (DQN) for traffic signal control. SDRRL utilizes AI agents to make real-time decisions on signal timings, while NFM employs fuzzy logic-based rules to determine optimal timings. DQN combines deep neural networks and Q-learning for estimating optimal action-values. Experimental evaluations of SDRRL demonstrate its effectiveness in maintaining performance under high arrival rates. NFM reduces average waiting time compared to fixed-time control, and DQN focuses on coordinated control across multiple intersections. The proposed framework and algorithms offer promising solutions for enhancing traffic management, leveraging software-defined networking and deep reinforcement learning to overcome limitations of traditional control systems.

# References

AKÇELIK, R., 2011. TIME – DEPENDENT EXPRESSIONS FOR DELAY, STOP RATE AND QUEUE LENGTH AT TRAFFIC SIGNALS.

Azimirad, E., Pariz, N. & Sistani, M. B. N., 2010. A Novel Fuzzy Model and Control of Single Intersection at Urban Traffic Network. 4(1), pp. 107 - 111.

Farhad Pooran, N. G., 2001. Implementation of the OPAC adaptive control strategy in a traffic signal network.

Kumar, N., Mittal, S., Garg, V. & Kumar, N., 2021. Deep Reinforcement Learning-Based Traffic Light Scheduling Framework for SDN-Enabled Smart Transportation System. 23(3), pp. 2411-2421.

Li, J., Pan, X. & Wang, X., 2007. State-space equations and the first-phase algorithm for signal control of single intersections. 12(2), pp. 1007-0214.

Mansoure Kafash, M. M. M. M. S. A. M., 2013. Designing fuzzy controller for traffic lights to reduce the length of queues in according to minimize extension of green light time and reduce waiting time.

Martin Gregurić 1, \*. V. 1. A. 2. a. M. 1., 2020. Application of Deep Reinforcement Learning in Traffic Signal Control: An Overview and Impact of Open Traffic Data.

Najmeh Samadiani 1, \*. H. 1. C. 1. L. 1.-H. C. 2. X. 1. a. H. 3., 2019. A Review on Automatic Facial Expression Recognition Systems Assisted by Multimodal Sensor Data.

Vikash Gayah, H. L., 2023. A Novel Max Pressure Algorithm Based on Traffic Delay.