International Journal of Intelligent Transportation Systems Research

SUMO-Based Intelligent Traffic Lights Control System:

A Case Study of Congested Areas in Colombo

--Manuscript Draft--

Manuscript Number:	IJIT-D-25-00334
Full Title:	SUMO-Based Intelligent Traffic Lights Control System: A Case Study of Congested Areas in Colombo
Article Type:	Research Paper
Funding Information:	
Abstract:	This research addresses the escalating issue of traffic congestion within Colombo Metropolitan Area, Sri Lanka, where rapid urbanization and increasing transportation demand have exacerbated the strain on existing road transit infrastructure. Even with multiple traffic management strategies like fixed-time signal control and manual efforts, congestion continues, resulting in financial losses, longer travel durations, and higher fuel consumption. A new method for traffic monitoring and management has been proposed that leverages Reinforcement Learning (RL) technique to regulate traffic signals in real-time dynamically. Our RL-based designed system aims to improving the traffic movement, reduce waiting periods, and boost the overall efficiency of city transport network. Primarily, simulation method is utilized thereby leveraging the open-source traffic simulation software SUMO (Simulation of Urban MObility) for illustrating traffic situations and assessing effectiveness of the proposed system. Findings suggest that applying reinforcement learning to traffic signal control can reduce congestion, improve road safety, and offer a sustainable solution to urban transportation challenges. Additionally, this study contributes to the growing area of intelligent transportation systems and establishes a foundation framework for more flexible and responsive traffic management approaches in fast-developing urban regions.
Corresponding Author:	Sameera Abar, Ph.D. KCGI: Kyoto Joho Daigakuin Daigaku Kyoto, JAPAN
Corresponding Author Secondary Information:	
Corresponding Author's Institution:	KCGI: Kyoto Joho Daigakuin Daigaku
Corresponding Author's Secondary Institution:	
First Author:	N.W.V.C. Kanishka Chathuranga, Master's degree
First Author Secondary Information:	
Order of Authors:	N.W.V.C. Kanishka Chathuranga, Master's degree
	Sameera Abar, Ph.D.
	Yoichi Terashita, Doctorate
Order of Authors Secondary Information:	
Author Comments:	Kindly assess the suitability of our paper for publication in your journal.

Noname manuscript No.
(will be inserted by the editor)

SUMO-Based Intelligent Traffic Lights Control System:

A Case Study of Congested Areas in Colombo

N.W.V.C. Kanishka Chathuranga · Sameera Abar · Yoichi Terashita

Received: date / Accepted: date

Abstract This research addresses the escalating issue of traffic congestion within Colombo Metropolitan Area, Sri Lanka, where rapid urbanization and increasing transportation demand have exacerbated the strain on existing road transit infrastructure. Even with multiple traffic management strategies like fixed-time signal control and manual efforts, congestion continues, resulting in financial losses, longer travel durations, and higher fuel consumption. A new method for traffic monitoring and management has been proposed that leverages Reinforcement Learning (RL) technique to regulate traffic signals in real-time dynamically. Our RL-based designed system aims to improving the traffic movement, reduce waiting periods, and boost the overall efficiency of city transport network. Primarily, simulation method is utilized thereby leveraging the open-source traffic simulation software SUMO (Simulation of Urban MObility) for illustrating traffic situations and assessing effectiveness of the proposed system. Findings suggest that applying reinforcement learning to traffic signal control can reduce congestion, improve road safety, and offer a sustainable solution to urban transportation challenges. Additionally, this study contributes to the growing area of intelligent transportation systems and establishes a foundation framework for more flexible and responsive traffic management approaches in fast-developing urban regions.

Keywords Traffic Signal · SUMO · Traffic Simulation · Traffic Flow Prediction · Predictive Analysis · Reinforcement Learning

N.W.V.C. Kanishka Chathuranga
Department of Data Science, Kyoto College of Graduate Studies for Informatics, Japan
Tel.: +8170-3793-5659
E-mail: chamodkanishka77@outlook.com

Sameera Abar
Department of Web System Development, Kyoto College of Graduate Studies for Informatics, Japan E-mail: s_abar@kcg.ac.jp

Yoichi Terashita
Department of Data Science, Kyoto College of Graduate Studies for Informatics, Japan

1 Introduction

Traffic congestion continues to be a major issue in many cities worldwide, particularly in developing nations. It leads to significant delays, excessive fuel consumption, and financial losses. Poorly designed road systems often result in small key areas within numerous developing countries that frequently become congestion hotspots. Insufficient implementation of traffic management in and around these critical areas is likely to lead to prolonged traffic jams [1]. Sri Lanka has an extensive roadways network of about 119,000 kilometers, with a high road density of 1.7 kilometers per square kilometer as compared to regional peers. This network includes National, Provincial, Pradeshiya Saba (regional government), and Local Authority roads and those built under the ambit of mass transit development projects. In Sri Lanka, around twelve thousands (12,380) kilometers of National roads - comprising class “A” (trunk), class “B” (main), and Expressways - are managed by the central government through RDA (Road Development Authority) [2].

In recent decades, transportation demand has significantly increased especially in the capital Colombo Metropolitan Area. Colombo is the economic center of Sri Lanka, with government and private offices, factories, hospitals, hotels, schools, and the sea harbor driving high daily travel demand. This central location also attracts many ad-hoc travelers, as it serves as the main highways and rail transport hub. Consequently, most travelers from western regions to other parts of the country must routinely pass through Colombo [3].

As traffic demand has increased, roads congestion has also escalated, resulting in several adverse effects. This implies losing monies, spending extra time traveling, and having to bear relatively higher costs to operate vehicles in terms of fuel expenses. Even though various measures have been implemented to reduce traffic congestion in Colombo, majority have not yielded any significant positive results. This indicates that the issue is escalating rapidly and needs to be tackled continuously to prevent adverse effects. Primary cause of traffic bottlenecks in cities is that the roads are overcrowded with unprecedented number of vehicles exceeding their permissible capacity. Colombo, being the central hub that offers numerous job opportunities and serves as the commercial capital of Sri Lanka, consistently experiences heavy traffic, particularly during peak hours.

Typically, city centers in metropolis experience heavy traffic during rush hours, particularly at intersections. Significant delays occur during school drop-off times in the mornings (from 7:15 to 7:45 a.m.), proceeded by commuter traffic later on (office start timings i.e. 8:00 to 9:00 a.m). Whereas in the afternoons and evenings, congestion is normally noted from 1:30 to 2:30 p.m. and again from 5:00 to 8:00 p.m. respectively [4]. Ineffective intersection control strategies are a basic cause of traffic realted issues. Managing traffic at intersections during peak hours, when the highest volumes occur, is increasingly challenging and daunting task. Road users encounter difficulties at crossings or intersections because of extended queue lengths and longer waiting times when automatic traffic signals are in operation. To alleviate this issue, sometimes traffic officers are also deployed to manage traffic at intersections during busy periods. Underlying cause was thought to be that the signal cycle timings could not be adjusted appropriately to improve the situation for increased traffic flow rates [5]. During peak hours in Sri Lanka, dominant city center intersections are handled by traffic police wardens (with grade-separated intersections used in severe traffic) who mainly focus on clearing queues from major roads, giving less attention to minor roads. Thus, it's important to evaluate the effectiveness of manual traffic control practices versus traffic signals.

Remainder of the paper is structured as stated. Next section deals with the account of related works in similar direction to proposed scheme. Section 3 & 4 detail the underlying methodology, test environment and implementation considerations whereas results, discussions and implications of the research are provided in Section 5. Finally paper concludes thereby highlighting future work paths.

2 Literature Review

2.1 Relevant Research Publications

A SUMO Based Simulation Framework for Intelligent Traffic Management System [6]: Study presents an Intelligent Traffic Management System (ITMS) incorporating a Deep-Neuro-Fuzzy model for optimizing traffic flow and alleviating congestion in metropolitan areas. Utilizing Dijkstra's algorithm, subject system selects optimal routes based on road segment weights derived from the model. To assess the model's effectiveness, authors combined it with the open-source traffic simulation tool SUMO, which allows for the simulation of various traffic-related situations, including route optimization and traffic signal control. Additionally, a custom GUI was developed for controlling simulation parameters and visualizing traffic flow feedback. Representative findings indicate that the model can be successfully utilized in practical scenarios, confirming its capacity to replicate realistic traffic situations and improve routing by comparing it with various pre-existing algorithms.

Simulation of modern Traffic Lights Control Systems using the open source Traffic Simulation SUMO [7]: "OIS" (Optical Information Systems) project developed and tested innovative traffic control mechanisms using advanced sensors beyond traditional inductive loops. An agent-based traffic light algorithm was introduced, optimizing junction flow by using jam length as input. Because there were no practical testing opportunities available, the efficacy of algorithm was confirmed using the open-source traffic simulation software SUMO, which showed enhancements in traffic flow at busy intersections.

Traffic Signal Control System Based on Intelligent Transportation System and Reinforcement Learning [8]: This work proposes a Traffic Signal Control System (TSCS) harnessing an Intelligent Transportation System (ITS) architecture and a Reinforcement Learning (RL) algorithm to minimize vehicle queues at engaged intersections. Here the system includes components for queue detection using computer vision and an RL-based signal program. Development stages involved system architecture design, prototype creation, and rigorous testing, including module-specific and real-time integration tests. Simulations in a medium-sized city showed significant improvements, reducing vehicle queues by 29 present, waiting time by 50 present, and lost time by 50 present compared to fixed traffic signal phases.

Traffic Adaptive Control Framework for Real Time Large-Scale Emergency Evacuation [9]: Authors present a traffic adaptive control framework for real-time large-scale emergency evacuations, aimed at efficiently managing evacuation traffic flow. Designed framework combines a prescriptive reference model, which specifies desired traffic states in a rolling-horizon manner, with an adaptive control system that utilizes real-time traffic data from sensing systems. The adaptive control system devises and implements traffic management strategies to guide traffic towards desired conditions, ensuring a smooth and efficient evacuation during both natural and man-made emergencies.

Traffic Sub-Area Division Expert System for Urban Traffic Control [10]: This paper discusses a method for effectively controlling traffic networks by partitioning them into sub-areas for better coordination and reduced complexity. It introduces an automated expert system based on an integrated correlation index to balance multiple demands in traffic command and control. System's framework includes a knowledge representation method and a fuzzy logic-based inference engine. A case study in Zhangjigang City, China, demonstrates that the system successfully addresses multiple traffic demands, and its results align closely with those of transportation experts. A notable drawback in such approaches could potentially be the complete reliance on domain specific knowledge that needs continuous maintenance and modification in ever-evolving traffic situations.

Deep Reinforcement Q-Learning for Intelligent Traffic Signal Control with Partial Detection [11]: Hereby a Deep Q-Network (DQN)-based reinforcement learning model is putforward for optimizing traffic signal control at isolated intersections in partially observable environments. This model is made to work well with low detection rates using data from connected automobiles, in contrast to majority of current systems that assume complete vehicle detection. For these kinds of situations, authors suggest a novel state representation, incentive system, and network architecture. According to simulations, the prototype outperforms conventional actuated controllers in both full and partial detection scenarios. Additionally, this scheme determines the detection rate thresholds necessary for satisfactory and ideal traffic control performance.

Universal Attention-Based Reinforcement Learning Model for Traffic Signal Control [12]: AttendLight, an end-to-end reinforcement learning (RL) algorithm is introduced for adaptive traffic signal control thereby employing a universal model that can manage junctions with different numbers of roads, lanes, signal phases, and traffic patterns, in contrast to earlier approaches that call for independent training for each and every intersection. Two respective attention mechanisms are deployed one for handling various signal phase combinations and other for managing changeable road-lane inputs allow it to achieve this flexibility. Across a range of intersection kinds and deployment settings, AttendLight routinely outperforms conventional and other RL-based techniques in tests conducted on both synthetic and real-world datasets.

Traffic Signal Control with State-Optimizing Deep Reinforcement Learning and Fuzzy Logic [13]: An improved traffic light control method is created using a Deep Q-Network (DQN) enhanced with fuzzy logic to reduce vehicle waiting time and improve traffic efficiency. Turning traffic may experience lengthy delays due to the bias of traditional DQN techniques against straight-moving cars. Authors changed the reward system to take waiting periods in each lane into consideration in order to remedy this. Comparing the suggested approach to traditional DQN with fuzzy logic, simulations utilizing SUMO software revealed an average overall waiting time reduction of 18.46%. To further speed up emergency response times, an ambulance prioritization system was also implemented. Overall the strategy performed better than current methods in other cases that are studied in this context.

Calibration of SUMO Microscopic Simulator for Sri Lankan Traffic Conditions [14]: This research introduces an automated calibration framework for the SUMO microscopic traffic simulation software to finetune parameters related to car-following and lane-changing behavior under heterogeneous traffic conditions. Paradigm was used to calibrate four lane-changing and one car-following parameters in an urban corridor in Colombo, Sri Lanka, using the Simultaneous Perturbation Stochastic Approximation (SPSA) algorithm. Efficiency of the framework was demonstrated by the results, which closely matched observed traffic speeds. The method's automated nature makes it simple to scale to bigger networks, which makes it appropriate for correctly simulating traffic conditions in Sri Lanka.

Traffic - Light Control with Reinforcement Learning [15]: This paper presents a real-time traffic light control method using deep Q-learning to reduce urban congestion. The model makes decisions on dynamic phase changes using a reward function that is dependent on throughput, travel time, delays, and queue lengths. Online learning from real-time traffic and offline learning from data with a set timetable are combined in training. System incorporates a ”memory palace” to rectify training data imbalance and a ”phase gate” to streamline learning across various signal phases. Subject approach considerably outperformed conventional fixed signal schemes when tested on an intersection in Hangzhou, China, cutting down on vehicle waiting times, line lengths, and overall travel times.

2.2 Conventional Techniques for Traffic Management

In addition to aforementioned works, some of previously proposed approaches by some researchers are incorporated for further discussion on the matter under investigation.

Signal Coordination Approach: Here, traffic signal coordination involves linking multiple traffic signals to enhance the flow of specific directional movements. It sets the timing of green lights along a series of signals to ensure smooth traffic flow, especially on busy arterial roads with frequent signals. Signal coordination has drawbacks, including negative community impacts, increased traffic, high maintenance costs, and the need for qualified staff. Despite reducing stops and delays, factors like traffic growth, complex layouts, and incidents can hinder free-flow travel [16].

Traffic Queue Length Approach: proposed system hardware comprises of multiple sensors, which are simple and well-known for estimating queue lengths. However, a single sensing device is insufficient; therefore, an array of sensors is necessary, arranged to maximize traffic data output. Sensor topology is vital for accurate queue length estimation. Sensor’s reading numbers can vary by direction and should be sized based on peak-hour traffic. Practical placement and spacing of sensors depend on vehicle types, lengths, and traffic conditions [17]. This approach is not much cost-effective, and it is difficult to install sensors everywhere on roadways.

2.3 Inception of Innovative Approaches

2.3.1 Reinforcement Learning

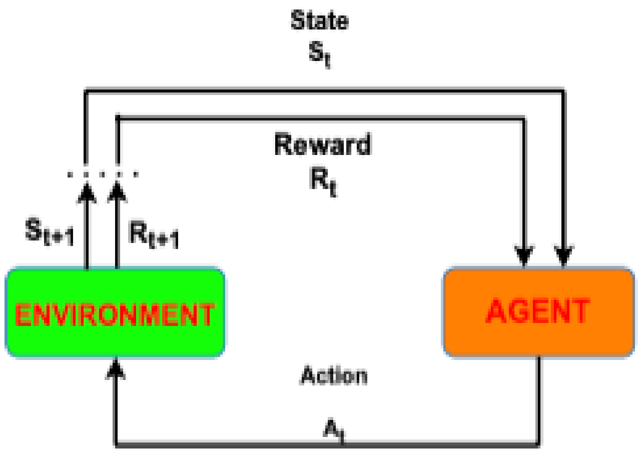

Fig. 1: Reinforcement Learning Procedure [19]

This study introduces Reinforcement Learning (RL: a system based on rewards and penalties feedback to amend the behavior of software agents) to overcome the limitations of fixed-time signal control systems. RL is a branch of machine learning that enables intelligent agents to adjust traffic signal timings based on real-time traffic conditions, optimizing a specified reward [18]. Prime goal of the RL approach is to minimize delays, reduce congestion, shorten travel durations, enhance safety, and improve traffic flow. This method could enhance the coordination of traffic signal controllers, which in turn may reduce traffic congestion in urban areas. By implementing reinforcement learning, traffic light systems can become more intelligent and able to adapt to ever-changing runtime conditions. These algorithms rely on data to use negative reinforcement to deter congestion while employing positive reinforcement to promote smooth traffic flow. As a result, urban transportation networks can evolve to be more intelligent and flexible [19]. The dynamic characteristics of this innovative traffic management strategy are illustrated in Figure 1, which outlines how the various state encounters interacts with surroundings or environment and the involvement of problem-solving and decision-making process.

2.3.2 Simulation of Urban MObility (SUMO)

In 2001, German Aerospace Center (DLR) initiated the creation of SUMO, an innovative open-source traffic simulation software package that’s designed to replicate a traffic road network within an imitated city’s infrastructure. Since the simulation is multi-modal, it encompasses not only the movements of cars within urban layout but also the public transportation systems along street network, including various train networks [20]. SUMO features a robust simulation engine suitable for modeling everything from individual intersections to entire metropolitan areas, along with a “remote control” interface (TraCI) that allows for real-time adjustments to the simulation. This simulator represents a comprehensive set of applications designed to assist in the preparation and execution of traffic simulations rather than being just a traffic simulation itself. In order to mimic traffic using ”SUMO”, it is imperative to showcase road networks and traffic demands in a specific format, which means that both need to be either imported or created from various sources [3].

A person’s identity is influenced by their departure time and the path they select, which encompasses various sub-routes that dictate a particular mode of transportation. For instance, in a theoretical scenario, an individual might drive their vehicle to the nearest public transport station and then continue their journey using various forms of conveyance. While a person may also choose to walk, this action is not simulated explicitly; instead, it is accounted for by calculating the time it would take for them to arrive at their destination [20].

Traffic flow is modeled microscopically, implying each vehicle is represented individually with specific locations and speeds. At every one-second time lapse

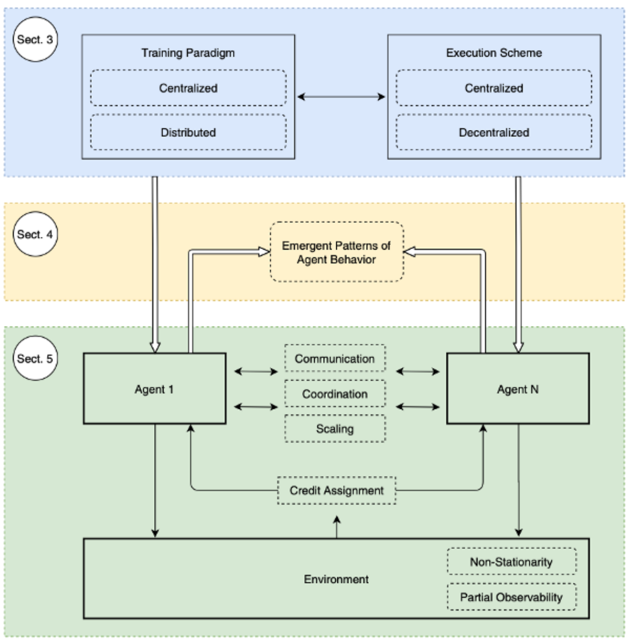

Fig. 2: Multi-Agent Reinforcement Learning (MARL) Network [24]

or step, these values are updated based on vehicle in front and diverse roadway attributes or characteristics. This simulation operates in a time-discrete and space-continuous manner, as detailed in the findings yielded by SUMO; essentially this functions as a purely microscopic traffic simulation. Each vehicle is explicitly defined, identified by its name, time of departure, and route within the given network, as outlined in the research based on SUMO. Additional details about each vehicle, such as lane utilization, speed, and position, can also be specified if desired on need-cum basis. Furthermore, vehicles can be categorized by type to reflect their physical characteristics and the parameters of the movement mode employed.

Traffic signals are necessary requisite for traffic management and improving flow. Each simulated intersection can include traffic lights, and some German junctions allow right turns on red. An update to the right-of-way rules regarding this is being introduced. SUMO also provides effective traffic simulation capabilities that enable vehicles to operate without collisions while supporting a range of vehicle types. It includes multi-lane roads for lane changes and applies right-of-way rules at junctions, managing different priority scenarios. SUMO generates XML (eXtensible Markup Language) outputs that track network states over time and supports data input from multiple XML files for flexible and convenient management. Detectors can also produce GNU Plot (GNU: a recursive acronym that stands for ”GNU’s Not Unix) or CSV (Comma Separated Values) sort of outputs [20].

3 Design Methodology

3.1 Q-Learning Method

Reinforcement learning is increasingly popular and have got traction in the recent years within the context of diverse application scopes. Chiefly, it’s applied in areas such as game theory and operations research, allowing computers to make decisions without historical or labeled data [21]. Q-learning is a type of model-free reinforcement learning. It can also be seen as a technique for Asynchronous Dynamic Programming (ADP). This approach enables agents to learn how to act optimally in Markovian environments by experiencing the outcomes of their actions (either penalities and rewards) without having to create representations of those environments [22]. Markov Decision Process (MDP) is a framework for sequential decision-making. Markovian environ or condition is inherently probabilistic, so the outcomes of actions are random. Policies dictate how

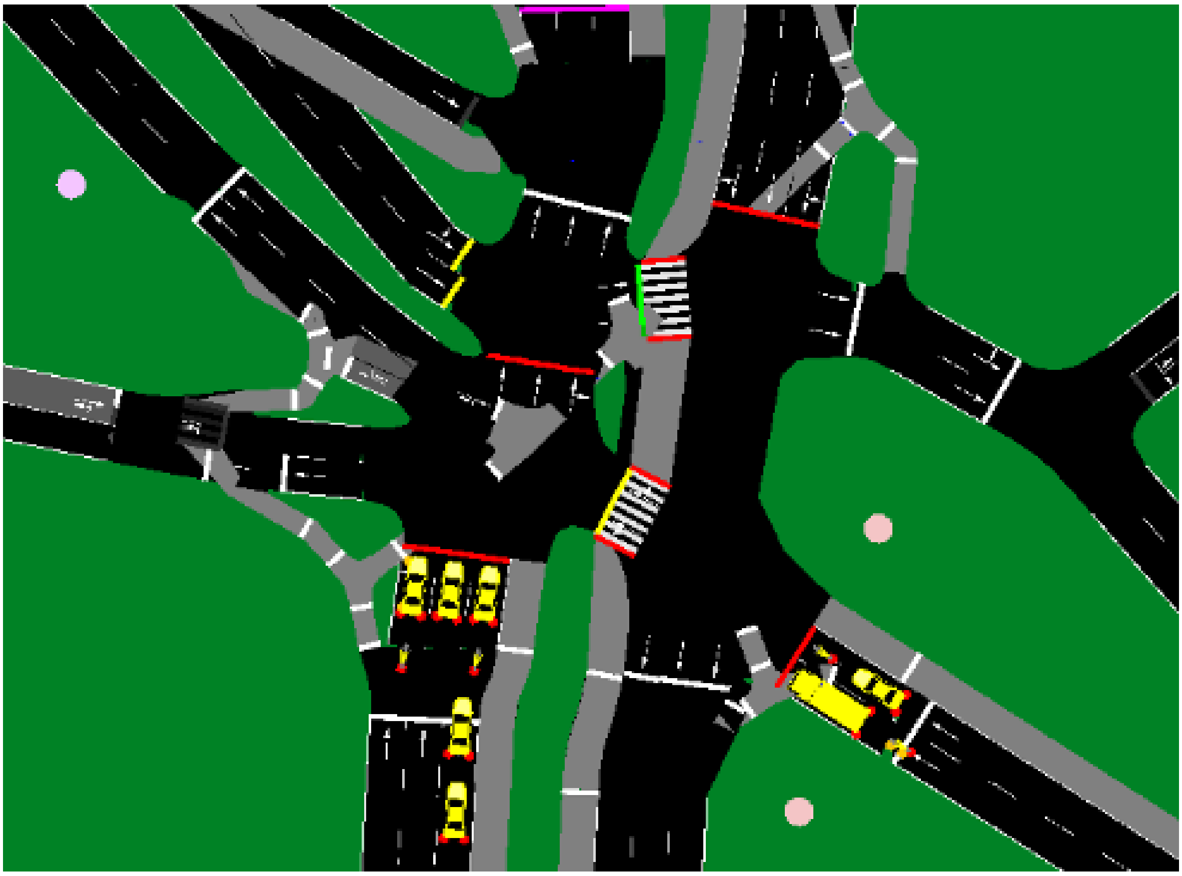

Fig. 3: Borella Junction view in SUMO

actions are chosen in a state, and MDPs are used to create reinforcement learning algorithms [21].

3.2 Deep Q Network Method

Term 'Deep' in 'Deep Q-Networks' (DQN) highlights the use of Deep Convolutional Neural Networks (CNNs) that are inspired by the way us human's visual cortex interprets images and are crafted to analyze and retrieve features from visual information [8]. Under this setting, CNNs work on unprocessed image data to recognize objects and pinpoint their locations. Following this, somewhat improved finer data is sent to an agent, resulting in a clearer representation of the environment to improve decision-making and problem-solving abilities [9].

Google's Deep Mind developed the Deep Q Network (DQN) by combining Q Learning and Deep Learning paradigms. This algorithm learned to play 49 Atari games, often surpassing human achieved high scores [8]. We can gain insight into the strength of this algorithm. Our study ultimately aims to evaluate the performance of this algorithm towards managing traffic signal timing for large area maps using reinforcement learning.

3.3 Multi-Agent Reinforcement Learning (MARL)

A multi-agent system is composed of several distributed entities known as agents that independently make decisions and interact with one another in a common environment [23]. Each agent strives to accomplish a particular goal, requiring a variety of skills to execute intelligent actions. Depending on the specific task, complex interactions between agents may arise, leading them to either work together or compete against one another to outperform their rivals [9].

MARL has been utilized across various fields, such as distributed control, telecommunications, and economics [22]. It is typically represented as either a multiagent mediated Markov Decision Process (MDP) or a team Markov game in which the agents commonly share a reward function. A more complex yet demanding cooperative testing scenario involves distinct reward functions exclusive to each agent, while the unified ob-

Fig. 4: Borella Area view in Netedit

jective is to optimize the average long-term return for the deliberate actions of all agents combined [24].

3.3.1 Decentralized MARL Network

Every agent independently decides its course of actions based upon the local information observed/perceived and communication protocols received by them from their neighboring agents through the network. This optimizes the globally averaged return across the whole network [11]. Research indicates that a decentralized method in Multi-Agent Reinforcement Learning (MARL) can be effectively used for traffic monitoring and management chores. In this approach, each agent depicts a traffic light and operates independently, attempting to optimize its performance without cooperating with the other traffic lights. Besides this, agents are responsible for actualizing their own rules and policies for the traffic operation functions.

3.3.2 Centralized MARL Network

Centralized MARL (Multi-Agent Reinforcement Learning) methods train multiple agents collaboratively using shared information, addressing non-stationarity within multi-agent environments. These methods feature Shared Information, which uses collective data; Value Decomposition which divides global value functions into agent-specific contributions; and Actor-Critical which integrates policy learning with value function optimization [12].

Table 1 summarizes the salient feature and contrast oriented factors among simulation-based approach and various learning-based methods, highlighting their environment coordination, algorithms, models, communication levels, action types, robustness, and performance ratings. While both Q-Learning and DQN operate as single-agent models with no environment coordination, MARL Centralized coordinates the entire system, facilitating high communication and collaboration among agents, whereas MARL Decentralized limits coordination to individual agents with lower communication overhead. Algorithms differ as well; Q-Learning uses the traditional Bellman Equation with a Q-table model, whereas DQN and MARL methods leverage neural networks for handling complex environments. In terms of robustness, MARL Centralized demonstrates superior resilience by incorporating all agents collectively, while Q-Learning and DQN's robustness is agent-dependent, and MARL Decentralized provides independent robustness but with less system-wide cohesion. Cumulative performance rating (CPR) further reflects these trends, with MARL Centralized achieving the highest scores, indicating better overall effectiveness, followed by moderate scores for DQN and MARL Decentralized, and lowest for Q-Learning, illustrating trade-offs between simplicity, coordination, and performance.

3.3.3 Methodology Comparison Chart

- **EC** = Environment Coordination
- **CPR** = Computational Power Requirement
- **NN** = Neural Network
- **DNN** = Deep Neural Network

Table 1: Comparison of Simulation and Learning-Based Methods

Feature	Q-Learning	DQN	MARL Centralized	MARL Decentralized
Env. Coordination	No	No	Full system	None
Algorithm	Bellman Equation	NN	NN	NN
Model	Q-table	DNN	DNN	DNN
Communication	None	None	High	Low
Action Type	Single Agent	Single Agent	All Agents	Individual
Robustness	Agent-Dependent	Agent-Dependent	All Agents	Independent
CPR	Low	Moderate	High	Moderate

4 Implementation and Test Environment

4.1 Tools and Software Application Used

4.1.1 SUMO GUI

This application is primarily designed to conduct simulations that identify traffic congestion bottlenecks. It

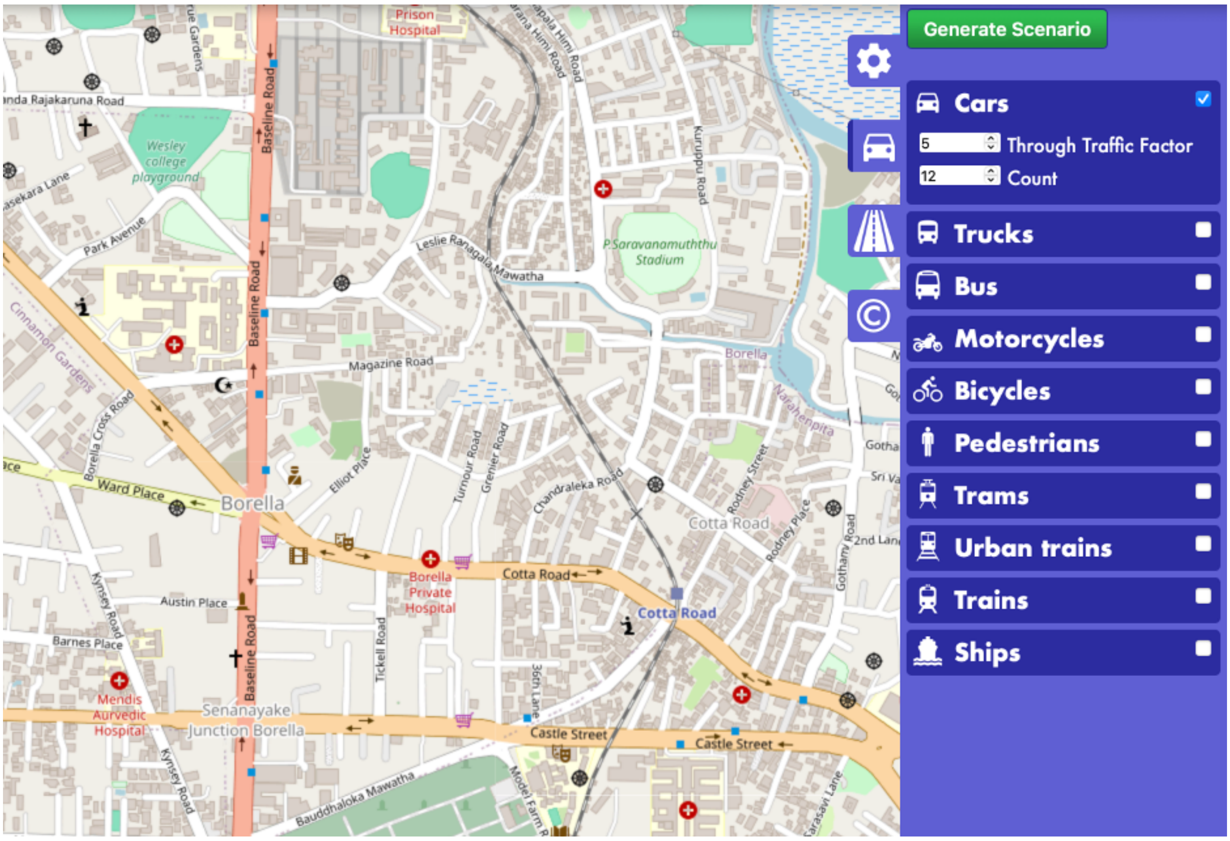

Fig. 5: OSM with Open Street Map

effectively replicates real-world conditions by incorporating a variety of vehicle categories—such as cars, buses, motorcycles, pedestrians, and trucks—that are pertinent to the Colombo region. Borella Junction area (a preview is shown in Figure 3) has been selected in Colombo for performing simulations due to its distinctive urban dynamics and intricate traffic patterns, which are likely to yield valuable insights for the proposed case study.

4.1.2 Netedit

Principal objective of this application is to enable the effective management and modification of maps in alignment with the specific parameters and preset criteria established by the user. It harnesses a varied suite of tools designed to enhance usability and operational efficiency, thus optimizing the mapping process in a well-formed fashion. Figure 4's snapshot illustrates the Borella junction area on the Netedit map, highlighting four main junctions with traffic lights. We intend to optimize the timing of these lights applying reinforcement learning technique for improving the traffic flow and limiting congestion based on real-time dynamic conditions.

4.1.3 Sumo's OSM Web Wizard

A collection of Python scripts - OSM Web Wizard is an essential tool embedded within SUMO package for developing realistic maps for simulation purposes. It renders features and functionality for generating a variety of vehicle types and integrates seamlessly with OpenStreetMap excerpt, ensuring accurate and precise mapping. This makes it particularly beneficial for research and educational applications in traffic dynamics and urban planning.

Figure 5 illustrates the OpenStreetMap (OSM) Web Wizard, a tool designed to facilitate the creation of realistic and user-friendly maps. This novel and versatile platform leverages collaborative mapping techniques and open-source data to enhance the accuracy and details of geographic representations. By simplifying the mapping process, expert OSM Web Wizard empowers users to contribute effectively to the growing body of geographic information, fostering greater accessibility and engagement within the field of cartography.

4.1.4 Prototype Actualization

```
Train Model
FOR each episode:
START SUMO simulation.
  WHILE simulation is running:
    FOR each traffic light:
      GET current state.
      CHOOSE an action using Choose Action.
      SET traffic light phase.
      STEP simulation forward.
      COMPUTE reward and get next state.
      STORE (state, action, reward, next_state,
      done) in memory.
      PERFORM Replay to train the model.
  CLOSE simulation after each episode.
Test Model
START SUMO simulation.
WHILE simulation is running:
  FOR each traffic light:
    GET current state.
    PREDICT best action using the trained model.
    SET traffic light phase based on the action.
    STEP simulation forward.
CLOSE simulation.
```

Fig. 6: Pseudocode for the training and testing of DQN model.

Algorithm 1: General RL Training Framework

Initialize parameters (Q-table, neural networks, or agents)
for *each episode* **do**
  Initialize environment state s
  while *not terminal* **do**
    Select action a based on current policy
    Execute action a, observe reward r and next state s'
    Update parameters using algorithm-specific update rule
    $s \leftarrow s'$

```
def train_traffic_light(tl, q):
    # TRAIN PHASE | Train a specific traffic light
    for episode in range(num_episodes):
        traci.start(["sumo", "-c", "test.sumocfg"])
        done = False
        while not done:
            state = preprocess_state(tl)
            action = choose_action(state, tl)
            if action in traffic_lights[tl]:
                traci.trafficlight.setPhase(tl, action)
            traci.simulationStep()
            reward = -1   # Placeholder | Define a real reward function
            next_state = preprocess_state(tl)
            done = traci.simulation.getMinExpectedNumber() <= 0
            memory.append((state, action, reward, next_state, done))
            replay()
        traci.close()
    q.put(f"Training completed for {tl}")
def train_dqn():
    # Training DQN model using multiprocessing
    processes = []
    queue = Queue()
    for tl in traffic_lights.keys():
        p = Process(target=train_traffic_light, args=(tl, queue))
        processes.append(p)
        p.start()
    for p in processes:
        p.join()
        print(queue.get())
def test_traffic_light(tl):
    # TEST PHASE | Testing specific traffic light
    traci.start(["sumo", "-c", "test.sumocfg"])
    done = False
    while not done:
        state = preprocess_state(tl)
        action = np.argmax(dqn_model.predict(state, verbose=0)[0])
        if action in traffic_lights[tl]:
            traci.trafficlight.setPhase(tl, action)
        traci.simulationStep()
        done = traci.simulation.getMinExpectedNumber() <= 0
    traci.close()
    print(f"Testing completed for {tl}")
def test_dqn():
    # Testing trained DQN model using multiprocessing
    processes = []
    for tl in traffic_lights.keys():
        p = Process(target=test_traffic_light, args=(tl,))
        processes.append(p)
        p.start()
    for p in processes:
        p.join()
```

Fig. 7: An excerpt of code on utilizing SUMO | training DQN model | evaluating trained system to confirm its effectiveness in managing traffic signals.

Algorithm 1 outlines a general Reinforcement Learning (RL) training framework. At the beginning of training process, core components such as Q-table, neural networks, or agent models are initialized depending on the specific RL method. For each episode, corresponding environment is reset to an initial state, and agent continuously selects actions according to its policy until a terminal state is reached. After executing each action, agent observes the resulting reward and next state and updates its internal parameters (e.g., value functions or policy networks) using an algorithm-specific update rule. This loop of interaction and learning enables the agent to gradually enhance its decision-making strategy over multiple episodes.

Figure 6 showcases a Pseudocode along with corresponding snippet of code Figure 7 on utilizing SUMO for traffic simulation and training the DQN model with experiences gathered. For each training episode, SUMO traffic simulation is initialized. While the simulation runs, agent iterates over each traffic light to: observe the current state, select an action via the policy, update the traffic light phase accordingly, advance the simulation by one step, calculate the reward and next state, and store the transition tuple (state, action, reward, next_state, done) into the experience replay buffer. Model is then trained through replay sampling. Upon episode completion, simulation gets terminated.

5 Results and Discussion

Our work evaluates the effectiveness of reinforcement learning algorithms for adaptive traffic signal control, using simulation results from SUMO to compare five approaches: baseline (Original Status), Q-Learning, Deep Q Network (DQN), Multi-Agent Reinforcement Learning (MARL) Centralized, and MARL Decentralized.

5.1 Overall System Performance

Table 2: Comparison of factors across different methods

Factor	Sim. (Orig.)	Q-learning	DQN	MARL Central.	MARL Decent.
Duration	266.77	47.64	46.78	43.89	41.32
Real-Time Factor	167.012	172.935	183.492	194.128	207.645
UPS	117244	119586	123438	127902	132154
UPS - Person	684	7360	7895	8412	8965

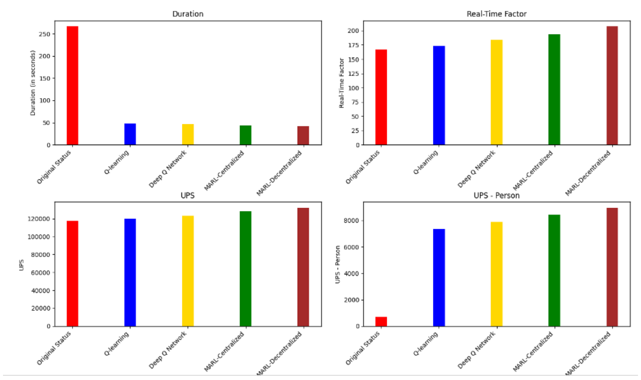

Fig. 8: Visual Graph of System Performance

Original Status model (Figure 8) had the longest simulation time at 266.77 seconds, while machine learning models significantly reduced this duration: Q-learning (47.64 seconds), Deep Q Network (46.78 seconds), MARL-Centralized (43.89 seconds), and MARL-Decentralized (41.32 seconds), highlighting the efficiency of reinforcement learning techniques. The Real-Time Factor (RTF) increased with these models, with the Original Status at 167.01 and MARL-Decentralized at 207.64, indicating higher computational demands despite shorter simulation times. In terms of Units Per Second (UPS), the Original Status recorded 117,24, whereas MARL-Decentralized achieved 132,15, demonstrating its ability to process more units. Similarly, the Original Status for UPS-Person had a recorded value of 684, whereas MARL-Decentralized reached 8965, demonstrating improved efficiency in terms of improved throughput. In conclusion, MARL-Decentralized outperforms the Original Status in performance metrics, confirming its position as the most effective solution for large-scale, real-time applications.

5.2 Vehicle Behavior Metrics

Table 3: Vehicle Performance

Factor	Sim. (Orig.)	Q-learning	DQN	MARL Central.	MARL Decent.
Inserted	5992	6148	6419	6792	7051
Running	2143	1497	1624	1756	1839
Waiting	2108	974	798	612	428
Teleports	8449	1943	1732	1484	1189
Emergency Braking	5	3	1	1	0

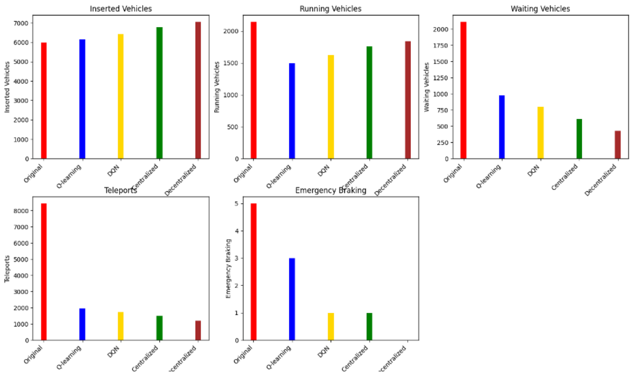

Fig. 9: Visual Graph of Vehicle Performance

As depicted in Figure 9, Number of inserted vehicles increased from the Original Status (5992) to MARL-Decentralized (7051), showing that reinforcement learning, especially MARL-Decentralized, enhanced vehicle management, and traffic flow. The count of running vehicles decreased significantly, dropping from Original Status (2143) to MARL-Decentralized (1839), indicating effective optimization in vehicle movement and reduced congestion. Waiting vehicles fell dramatically from 2108 to 428 in MARL-Decentralized, reflecting better traffic management and shorter wait times. The number of teleports, which signify vehicles circumventing the simulation, decreased from 8449 in the Original Status to 1189 in MARL-Decentralized, indicating better management. Instances of emergency braking fell from 5 to 0 in the MARL-Decentralized model, showing improved traffic flow and safety. Overall, these findings demonstrate that machine learning models, particularly MARL-Decentralized, significantly improve traffic flow, reduce congestion, and enhance system efficiency.

5.3 Persons Performance Efficiency

Table 4: Persons Performance

Factor	Sim. (Orig.)	Q-learning	DQN	MARL Central.	MARL Decent.
Inserted	713	748	779	804	826
Running	78	85	92	102	123
Jammed	37	21	14	9	5

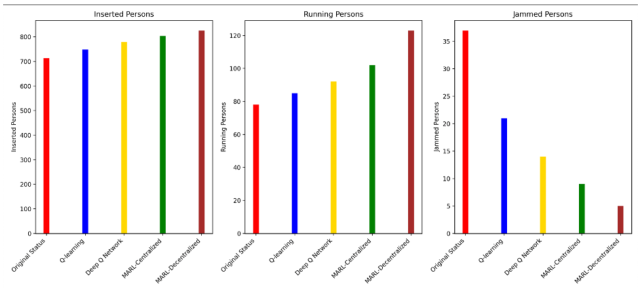

Fig. 10: Visual Graph of Person Performance

Analysis of results (Figure 10) highlights significant improvements in pedestrian management. Here, number of Inserted persons increased from 713 in Original Status to 826 in the MARL-Decentralized model, indicating enhanced management and insertion of individuals. In the same way, the number of running individuals increased from 78 to 123, indicating better management of pedestrian mobility and system responsiveness under learning-based controller and fewer delays. In contrast, Jammed persons dropped significantly from 37 to 5, demonstrating the model's effectiveness in alleviating congestion and ensuring smoother movement. In general, the results indicate that the MARL-Decentralized model significantly enhanced pedestrian movement, minimized congestion, and improved system efficiency, establishing it as the most successful model for managing pedestrian dynamics.

5.4 Traffic Statistics

Table 5: Traffic Statistics Comparison

Factor	Sim. (Orig.)	Q-learning	DQN	MARL Central.	MARL Decent.
Route Length	874.35	858.21	843.89	831.47	820.54
Speed	0.87	1.48	1.57	1.68	1.79
Duration	5219.81	1498.73	1439.12	1352.65	1271.98
Waiting Time	4773.45	1085.47	974.23	832.91	721.35
Time Loss	4946.44	1182.78	1093.63	973.51	862.17
Depart Delay	8607.49	389.25	223.67	348.71	256.93

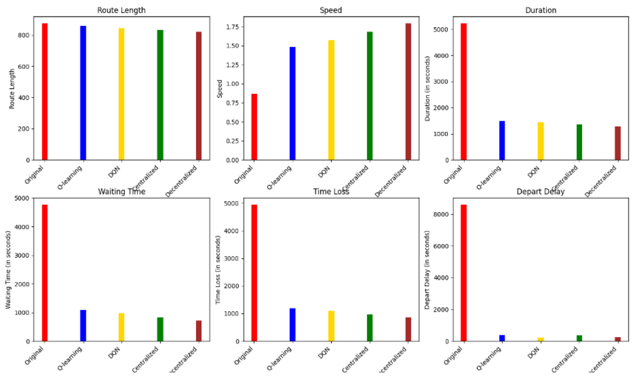

Fig. 11: Visual Graph of Traffic Statistic

As detailed in Table 5 & Figure 11, the performance insights uncovers that the Route Length reduced from 874.35 in the Original Status to 820.54 in the MARL-Decentralized model, signifying better path-finding by the machine learning models. Speed saw a notable increase, jumping from 0.87 to 1.79, which indicates improved traffic flow and vehicle efficiency. Duration was cut from 5219.81 to 1271.98, showcasing better route planning and reduced travel time. Waiting Time dropped from 4773.45 to 721.35, highlighting the models' effectiveness in minimizing idle time. Time Loss decreased from 4946.44 to 862.17, demonstrating success in reducing delays, while Depart Delay plummeted straight dramatically from 8607.49 to 256.93, emphasizing timely vehicle departures. Overall, the results demonstrate that the MARL-Decentralized model significantly enhances traffic system performance, reducing congestion and improving overall efficiency.

5.5 Pedestrian Route Statistics

Table 6: Pedestrians Statistics

Factor	Sim. (Orig.)	Q-learning	DQN	MARL Central.	MARL Decent.
Route Length	246.83	239.87	218.73	232.51	224.98
Duration	256.23	199.65	182.94	165.43	142.67
Time Loss	76.01	61.25	51.47	32.98	32.78

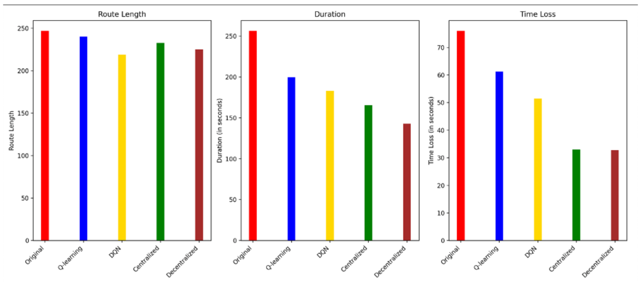

Fig. 12: Visual Graph of Pedestrian Statistics

Table 6 and associated bar chart in Figure 12 present pedestrian routing statistics. MARL-Decentralized model reduced duration from 256.23s to 142.67s, and time loss from 76.01s to 32.78s. Although the shortest route length was observed under DQN (218.73m), holistically best performance across all standard metrics was again achieved by the MARL-Decentralized approach, suggesting effective optimization of both vehicular and pedestrian mobility dynamics.

5.6 Discussion and Limitations

In this project, we designed a reinforcement learning model aimed at controlling traffic lights, which surpasses conventional fixed-time systems by adjusting to real-time traffic conditions. Implementing this model in Colombo could significantly advantage more than a million commuters by improving traffic flow and reducing delays, allowing them to utilize their time more effectively. This approach may also alleviate traffic congestion, resulting in lower fuel consumption and diminished emissions. By optimizing traffic light timings, air quality in the city can be adequately improved. Ultimately, this initiative has the potential to improve residents' quality of life while fostering a more sustainable urban setting.

At this point some potential frailties in relevance to this study need to be deliberated. Our research primarily relies on simulated traffic data generated by SUMO; although this offers a versatile simulation environment, data it generates might not accurately capture the intricacies of real traffic situations in Colombo, including driver behavior, road conditions, or unforeseen events. Absence of systematically gathered data in real-time creates obstacles for the effective application of these strategies in practical situations. This gap restricts our capacity to evaluate and modify strategies, which impacts the success of our interventions. Tackling this problem is crucial for effective implementation. Proposed work faced some hindrances due to the absence of sophisticated traffic monitoring and data collection gadget sand devices in Colombo. This constraint limited access to real-time and concise trails of traffic data, which could have enhanced the model's accuracy and practical usefulness. Implementing and training advanced models necessitates robust computational systems to effectively simulate maps and incorporate real-world data. Robust computing resources are essential for handling complicated algorithms and large datasets, ensuring the success of these applications. In addition to this, building the Q-learning model and incorporating it with SUMO requires a significant level of programming expertise. Insufficient experience in crafting and executing such intricate models may result in inefficiencies, errors, and less-than-ideal outcomes throughout the development and testing stages. Our effort emphasizes the necessity for continuous exploration in programming techniques and algorithms. Improving traffic light control with reinforcement learning requires exploring innovative reward functions, better state representation, and effective debugging, all of which necessitate focused R & D efforts.

6 Concluding Remarks & Outlook

This research conducted a comprehensive investigation of how reinforcement learning methods can be applied effectively to traffic light control. It highlighted the advantages of using these sophisticated algorithms over conventional fixed-time traffic signal systems, which frequently face challenges in adapting to real-time traffic conditions. Throughout this research, authors developed a deep understanding of various reinforcement learning approaches, such as Q-learning, policy gradients, and deep reinforcement learning. Unique characteristics and merits of each approach were carefully analyzed, highlighting how these methods are influenced by salient ideas such as exploration versus exploitation, reward systems, and state representation. These insights underscored the capacity of reinforcement learning to transform traffic fluidity management by fine-tuning signal timings, alleviating congestion, and ultimately enhancing urban mobility.

To overcome the limitations, we strive to dig further deep and analyze various algorithms to determine which one is most appropriate for the specific requirements and constraints of some particular scenarios. There is a need to take into account factors such as efficiency, accuracy, scalability, and ease of implementation to ensure the selected program more effectively meets the requirements of problem at hand.

Sensor devices can be installed at traffic lights locations for collecting real-time streaming video data to gather information, such as vehicle counts and other relevant data. Future work also focuses on preparing the models for retraining by carefully selecting the most effective parameters and methodologies. Ultimate milestone is to improve their performance to the highest achievable level, making certain they can produce excellent results in different situations. Moreover, advanced high-performance computing systems are to be incorporated for training the models, allowing for an in-depth comparison of their capabilities and efficiency.

Preparations to evaluate the model's effectiveness could also be done by simulating actual traffic scenarios in various locations throughout Colombo city. This strategy would likely enable us to evaluate how effectively the model responds to the intricacies and fluctuations of urban traffic patterns, offering important insights into its practical application accomplishment.

References

1. Dammulla.R, Mudunkotuwa.R.: Analysis on the Road Traffic Congestion in Colombo Metropolitan Area, CINEC Academic Journal, **vol. 5, no. 1**, p. 15-22,(2021) https://doi.org/10.4038/caj.v5i1.70
2. R. D. Authority: National Road Master Plan, Main Report, (2021) https://rda.gov.lk/images/publications/pdf/National_Road_Master_Plan_2021-2030/Natinal%20Road%20Master%20Plan%202021-2030%20Main%20Report.pdf
3. J. Weerawardana.: Greater Colombo Traffic Management,(2009) https://doi.org/10.13140/RG.2.2.21679.74406
4. A Vajeeran, GLDI De Silva.: Identification of Effective Intersection Control Strategies During Peak Hours, **48**,Pages 687-697,(2019) https://doi.org/10.1016/j.trpro.2020.08.069
5. T.W.K.I.M Dias, N.H.A.R.S Ekanayake.: Evaluating the Efficiency of a Traffic Signal Light over a Traffic Policeman. https://https:researchgate.net/publication/344674782_Evaluating_the_Efficiency_of_a_Traffic_Signal_Light_Over_a_Traffic_Policeman

6. B. Jang, M. Kim, G. Harerimana, J.W. Kim.: Q-Learning Algorithms: A Comprehensive Classification and Applications, **vol. 7**,(2019) `https://doi.org/10.1109/ACCESS.2019.2941229`
7. C. J. Watkins.: Technical Note Q-Learning,: Machine Learning, **Vols.8**, pages 279–292(1992) `https://doi.org/10.1007/BF00992698`
8. M. Sewak.: Deep Q Network (DQN), Double DQN and Dueling DQN, Deep Reinforcement Learning, pp 95–108,(2019) `https://doi.org/10.1007/978-981-13-8285-7_8`
9. D. Liao, Z. Zhang, T. Song, M. Liu.: An Efficient Centralized Multi-Agent Reinforcement Learner for Cooperative Tasks, **Vol 11**,(2023) `https://doi.org/10.1109/ACCESS.2023.3340867`
10. K. Zhang, Z. Yang, H. Liu, T. Zhang, T. Basar.: Fully Decentralized Multi-Agent Reinforcement Learning with Networked Agents,(2022) `https://proceedings.mlr.press/v80/zhang18n.html`
11. P. K. Sharmaa, R. Fernandeza, E. Zaroukiana, M. Dorothya, A. Basakb, D. E. Ashera.: Survey of Recent Multi-Agent Reinforcement Learning Algorithms Utilizing Centralized Training, **Vol 1**,(2021) `https://arxiv.org/abs/2107.14316`
12. T. Jayasinghe, T. Sivakumar, A. S. Kumarage.: Calibration of SUMO Microscopic Simulator for Sri Lankan Traffic Conditions, in Proceedings of the Eastern Asia Society for Transportation Studies, **Vol 13**, SRI LANKA,(2021) `https://www.researchgate.net/publication/357162895_Calibration_of_SUMO_Microscopic_Simulator_for_Sri_Lankan_Traffic_Conditions`
13. Romain Ducrocq, Nadir Farhi Cosys-Grettia.: Deep Reinforcement Q-Learning for Intelligent Traffic Signal Control with Partial Detection, France, **Volume 21**, pages 192–206(2023) `https://doi.org/10.1007/s13177-023-00346-4`
14. A. Oroojlooy, M. Nazari,D. Hajinezhad, J. Silva.:Universal Attention-Based Reinforcement Learning Model for Traffic Signal Control, NC,**Vol 1**(2020) `https://doi.org/10.48550/arXiv.2010.05772`
15. T. Meepokgit, S. Wisayataksin.: Traffic Signal Control with State-Optimizing Deep Reinforcement Learning and Fuzzy Logic,Appl. Sci. 2024, 14(17), 7908(2024) `https://doi.org/10.3390/app14177908`
16. E. P. Wannige.: Evaluating the Impacts of Coordinated Traffic Signal System , **p. 4**,(2018) `https://dl.lib.uom.lk/handle/123/13385`
17. I.T. K. Jayasekara, R. Sanjeewa, M. T. M.Perera.: An Automated Traffic Signal System Based on Traffic Queue Length, IRE Journals, **Vol 2: no 8**,10-11(2019) `https://www.irejournals.com/paper-details/1700925`
18. D. R. Kodagoda.: Deep Reinforcement Learning to Minimize Traffic Congestion with Emergency Facilitation, (2019). `https://doi.org/10.1109/ICter53630.2021.9774792`
19. S.M.M.R. Swapno, S.M.N Nobel.: Traffic Light Control Using Reinforcement Learning, (2024).`https://ieeexplore.ieee.org/document/10498933`
20. S. Piriyanth.: Sri Lankan traffic congestion, 55,(2023) `https://www.researchgate.net/publication/370604075_Sri_Lankan_traffic_congestion`
21. M. Behrisch, L. Bieker, J. Erdmann, D. Krajzewicz.: SUMO – Simulation of Urban MObility An Overview, (2011).`https://elib.dlr.de/71460/`
22. D. Krajzewicz, G. Hertkorn, P. Wagner.: SUMO (Simulation of Urban MObility) An open-source traffic simulation, p. 183-18(2002). `https://elib.dlr.de/6661/`
23. S. Konno, S. Abar, Y. Iwaya, T. Kinoshita.: Effectiveness of Autonomous Network Monitoring Based on Intelligent-Agent-Mediated Status Information. In: Okuno, H.G., Ali, M. (eds) New Trends in Applied Artificial Intelligence. IEA/AIE 2007. Lecture Notes in Computer Science, **Vol 4570** Springer, Berlin, Heidelberg. `https://doi.org/10.1007/978-3-540-73325-6_108`
24. K. Zhang , Z. Yang , H. Liu , T. Zhang ,T. Basar .:Finite-Sample Analysis for Decentralized Batch Multiagent Reinforcement Learning With Networked Agents, **Vol 66**, 5925 - 5940(2021).`https://doi.org/10.1109/TAC.2021.3049345`
25. S. Gronauer, K. Diepold.:Multi-agent deep reinforcement learning: a survey, **Vol 55**, 895–943(2022) `https://doi.org/10.1007/s10462-021-09996-w`