Project Phase-2 Report On

AI-assisted Smart Signal Traffic Management System

Submitted in partial fulfillment of the requirements for the award of the degree of

Bachelor of Technology

in

Computer Science and Engineering

By

Athulram K R (U2003051)
Alona Mary Sebastian (U2003028)
Colin Michael (U2003064)
Diya Anna Sunil (U2003218)

Under the guidance of

Dr Preetha K G

Department of Computer Science and Engineering
Rajagiri School of Engineering & Technology (Autonomous)
(Parent University: APJ Abdul Kalam Technological University)
Rajagiri Valley, Kakkanad, Kochi, 682039
April 2024

CERTIFICATE

This is to certify that the project report entitled ***"AI-assisted Smart Signal Traffic Management System"*** *is a bonafide record of the work done by* ***Athulram K R (U2003051)***,***Alona Mary Sebastian (U2003028)***, ***Colin Michael (U2003064)*** ,***Diya Anna Sunil (U2003218)*** *submitted to the Rajagiri School of Engineering & Technology (RSET) (Autonomous) in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology (B. Tech.) in Computer Science and Engineering during the academic year 2023-2024.*

Dr.Preetha K G
Project Guide
Head of the Department
Professor
Dept. of CSE
RSET

Dr.Sminu Izudheen
Project Coordinator
Professor
Dept. of CSE
RSET

ACKNOWLEDGMENT

We wish to express our sincere gratitude towards **Dr Sreejith P S**, Principal of RSET, and **Dr Preetha K G**, Head of the Department of Computer Science Department for providing us with the opportunity to undertake our project, "AI-assisted Smart Signal Traffic Management System".

We are highly indebted to our project coordinators, **Dr Sminu Izudheen**, Professor, Department of Computer Science and Engineering for her valuable support.

It is indeed our pleasure and a moment of satisfaction for us to express our sincere gratitude to our project guide **Dr Preetha K G** for her patience and all the priceless advice and wisdom she has shared with us.

Last but not the least, We would like to express our sincere gratitude towards all other teachers and friends for their continuous support and constructive ideas.

Athulram K R

Alona Mary Sebastian

Colin Michael

Diya Anna Sunil

Abstract

In response to the challenges of urban traffic management, this project aims to transform traditional traffic signal control systems with the integration of artificial intelligence (AI) and data analytics. By combining AI-assisted traffic light simulation and data fusion techniques, the goal is to alleviate the adverse effects of traffic congestion on both economic productivity and environmental well-being. Through this approach, the system predicts traffic flow dynamics and optimizes signal timings in real-time, adapting to fluctuations in traffic patterns, weather conditions, and other relevant factors.

At the core of this endeavour lies a sophisticated framework comprising a distinct module, the Reinforcement Learning (RL) module. The RL module harnesses Q-learning algorithms, enabling the system to learn and refine decision-making processes within the traffic environment, ensuring efficient traffic signal management. The culmination of this effort is an innovative approach that minimizes travel time and fuel consumption while enhancing overall traffic efficiency, ultimately contributing to the creation of more sustainable and livable urban environments.

In addition to optimizing traffic signal control, this project aims to foster collaboration and knowledge sharing among stakeholders involved in urban traffic management, by promoting interdisciplinary cooperation and community engagement, seeking to develop inclusive and context-sensitive solutions that address the unique challenges faced by diverse communities. Through partnerships with local authorities, transportation agencies, and community organizations, an endeavour to co-create strategies prioritising safety, accessibility, and sustainability in urban transportation systems. By fostering a culture of innovation and collaboration, this project aspires to build resilient and equitable cities where all residents can enjoy safe, efficient, and environmentally friendly mobility options.

Contents

List of Abbreviations

PIR Passive Infrared

ALSI Agent Learning for a Single Intersection

CLFE Cognition,Learning,Feedback and Evolution

ALMI Agent Learning for Multiple Intersections

ATLC Adaptive Traffic Light Control

DDPG Deep Determinsitic Policy Gradient

RACS Regional Cooperative Strategy

A2C Advantage Actor Critic

GAT Graph Attention Network

SUMO Simulation of Urban Mobility

IQL Independent Q Learning

IA2C Independent Advantage Actor Critic

MDRL Multi Agent Reinforcement Learning

MSA Multi Agent System

MDP Markov's Decision Process

3DQN Double Dueling Deep Q Network

MARL4TS Multi Agent Reinforcement Learning for Traffic Signal Control

CMDQN Cooperative Multiagent Deep Q Learning Network

DMN Decentralised Multiagent Deep Q network

DDPG Deep Deterministic Policy Gradient

MADDPG Multi Agent DDPG

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

As urban populations and the number of vehicles continue to rise, traffic congestion has emerged as a pressing concern. Beyond merely causing delays and frustration for drivers, traffic jams contribute to heightened fuel consumption and air pollution levels.While its impact may appear widespread, mega cities bear the brunt of traffic congestion, exacerbated by its relentless growth. This underscores the importance of real-time calculation of road traffic density for optimizing signal control and implementing efficient traffic management strategies.

One of the significant challenges in India is the issue of traffic congestion. While many countries face traffic problems with automobiles, buses, trucks, motorcycles, scooters, and bicycles, India experiences additional challenges due to auto-rickshaws, two-wheelers, and heavy vehicles. These elements contribute significantly to congestion, leading to a surge in traffic, a higher incidence of accidents,Over time, there has been a rise in fatalities and prolonged commute durations. In India, in the event of an accident, individuals often take matters into their own hands, resulting in road blockages and confrontations. This impedes the passage of ambulances to the scene, sometimes for hours, exacerbating the situation.

The traffic controller plays a pivotal role in influencing traffic flow. Hence, there is a growing imperative to optimize traffic control measures to effectively manage the rising demand. There are three standard methods for traffic control that are being used currently:

1) Manual Controlling:

As implied by its name, manpower is necessary to oversee traffic control. Traffic police are deployed to designated areas to manage traffic, equipped with signboards, signal lights, and whistles for this purpose.

2) Conventional traffic lights with static timers:

Fixed timers dictate the operation of these lights. A preset numerical value is programmed into the timer, causing the lights to automatically switch between red and green based on this timing parameter.

3) Electronic Sensors:

An alternative advanced approach involves the installation of loop detectors or proximity sensors along the road. These sensors provide real-time data on traffic conditions, which is then used to regulate the operation of traffic signals.

1.2 Problem Definition

Design an AI-assisted traffic light simulation using data fusion which is the integration of artificial intelligence and data analytics in traffic management

1.3 Scope and Motivation

Conventional traffic control methods inefficient to solve current traffic situation. So a better solution is necessary to overcome the conventional methods. In a notable traffic scenario observed in recent years, a real-world experiment involved positioning several human-driven vehicles equidistantly along a ring road and instructing them to maintain a specific speed over a designated time frame. The experiment demonstrated that, even in the absence of external factors such as traffic lights or lane changes, solely human driving behavior resulted in the formation of stop-and-go waves and congestion. Using Autonomous Vehicles(AV) can overcome this limitation. Those kinds of vehicles collect real-time details and help to maintain the traffic, but those vehicles are not common nowadays so consider it as a futuristic technology.

So the dynamics of traffic flow are significantly influenced by human driving behavior, with traffic lights playing a crucial role in shaping overall traffic conditions. By optimizing these traffic signals, it becomes possible to mitigate waiting times and alleviate traffic congestion effectively.

1.4 Objectives

- Develop a smart method to resolve the traffic congestion in major metropolitan as well as big cities by predicting traffic to increase fuel efficiency and prevent driver frustration.
- It contribute towards minimizing the average travel time.
- This model solve traffic congestion by analyzing real-time traffic flow patterns and comparing them with existing traditional transportation methods.
- Minimize the average travel time in an area by predicting the traffic.

1.5 Challenges

The main challenge in automating traffic lights using AI is creating a system that can effectively adapt to diverse and unpredictable traffic conditions. Unforeseen events and varying traffic patterns pose difficulties in developing an AI system that consistently makes accurate decisions for optimal signal timings.

1.6 Assumptions

- **Data Availability:** Assumption of access to diverse and reliable real-time and historical traffic data sources crucial for comprehensive traffic flow pattern analysis.
- **Traffic Pattern Consistency:** Expectation of consistent traffic behaviors and flow patterns, considering variations within regular traffic dynamics for effective modeling.
- **Model Adaptability:** Expectation that developed AI models can adapt to changing real-time traffic conditions to analyze and optimize traffic flow patterns efficiently.

1.7 Societal Relevance

This holds immense societal significance as it promises to revolutionize the way we navigate our daily commutes. This not only has the potential to significantly reduce waiting times and alleviate congestion but also contributes to a more sustainable and efficient transportation network. Traffic lights can dynamically respond to real-time traffic conditions, promoting smoother traffic flow and enhancing overall safety on the roads. Ultimately, this technology aligns with our aspirations for smarter and more livable cities, where transportation systems are not only functional but also attuned to the evolving needs of a rapidly changing world.

1.8 Organization of the Report

Chapter 1 begins with an introduction that offers the background,problem definition,the scope and its motivation as well as the motivation behind this project.It also includes the challenges,assumptions and the various societal and industrial relevance. The second chapter then examines the current status of research, covering existing literature, challenges, and potential applications in the field of traffic light optimization using different techniques which involves various methodologies like using Reinforcement learning techniques utilizing Q learning as well as deep Q learning. The third chapter delves into the requirements, the hardware as well as software requirements and the functional requirements.The fourth chapter includes the overall system architecture including the systematic architecture diagram,the module division and gantt chart representing the work schedule.The fifth chapter covers the system implementation,including the datasets that had been identified.This chapter also comprises of the proposed methodology and inclusion of explanation to various modules.The Database scheme has also been provided in this chapter. The different implementation strategies has been considered along with the source code.Chapter 6 delves into the culminated results of the implementing strategies.Chapter 7 involves the conclusion and future scope of this area,following the references made in the literature survey.

In conclusion, this chapter sets the stage for addressing the pressing issue of traffic congestion through an AI-assisted traffic light simulation, using data fusion techniques. With conventional traffic control methods proving inefficient, the need for smarter solutions becomes imperative. By integrating artificial intelligence and data analytics, this project aims to develop a comprehensive approach to optimize traffic flow, minimize travel time, and enhance fuel efficiency in major metropolitan areas and cities. Despite facing challenges such as data availability, traffic pattern consistency, and model adaptability, the project holds significant societal and industrial relevance by offering potential solutions to alleviate traffic congestion and improve overall transportation efficiency. Through thorough literature review, clear problem definition, and delineation of objectives, the groundwork is laid for further exploration into traffic optimization methodologies, culminating in the proposed system architecture and implementation strategies outlined in subsequent chapters.

Chapter 2

Literature Survey

2.1 An Information Fusion Approach to Intelligent Traffic Signal Control Using the Joint Methods of Multiagent Reinforcement Learning and Artificial Intelligence of Things[1]

2.1.1 Multiagent Reinforcement Learning for Traffic Signal Control

The MARL4TS (Multiagent Reinforcement Learning for Traffic Signal Control) project presents an innovative approach to traffic signal management using a multiagent reinforcement learning framework. The primary goal of MARL4TS is to optimize traffic signal control through intelligent decision-making in real-time. The project uses multiagent reinforcement learning techniques to enable traffic signal agents to learn and adapt their strategies based on the dynamic traffic environment. MARL4TS adopts a multiagent framework where individual traffic signals operate as autonomous agents. These agents collaborate to collectively enhance traffic flow at intersections. The use of multiple agents enables decentralized decision-making, allowing for adaptability to changing traffic conditions. At the core of MARL4TS is the Q-learning algorithm, a reinforcement learning technique. This algorithm equips traffic signal agents with the capability to learn optimal control policies by interacting with the environment and receiving feedback. Q-values are updated iteratively, allowing agents to make informed decisions based on past experiences.

2.1.2 AIoT based Framework

The AIoT-based framework described integrates a range of sensors, such as velocity sensors, passive infrared (PIR) sensors, target-tracking video cameras, and traffic density sensors. These sensors are strategically positioned at intersections equipped with traffic signals. Their collective role is pivotal in augmenting the system's capacity to perceive the traffic environment effectively.

At these intersections, velocity sensors, PIR sensors, target-tracking video cameras, and traffic density sensors are deployed. They actively gather real-time data, enhancing the system's awareness of the traffic environment.

Agents, equipped with learning and computing capabilities, are stationed at intersections. The agents collaborate to respond to dynamic changes in traffic flows and make optimal decisions for controlling traffic signals.

Every agent initially reads environmental information to determine the state space at the beginning of the simulation. Utilizing a reward function, the agents proceed with a strategy selection process, choosing an appropriate action from the action space based on the current state. The reinforcement learning algorithm calculates the return value Q for each action.

Subsequently, each agent selects an action to execute at its respective traffic intersection, guided by the determined strategy. These steps are iterated continuously, allowing the agents to refine and output the optimal control strategy.

Effectively, the signal control process at intersections transforms into a dynamic training process with real-time feedback.

Q learning algorithm: The training process utilizes the Q-learning algorithm with the objective of maximizing the expected cumulative reward. At each time step, the value Q, derived from reinforcement learning, is pivotal. The algorithm iteratively selects actions based on policy until the loop concludes or the final state is reached.

Through the Q-learning algorithm, the aim is to optimize the expected cumulative reward. Rewards are assigned after each update, resulting in a sequence of rewards. These cumulative values serve as knowledge to determine the appropriate strategy to employ.

Fundamentally, this research centers on constructing a system wherein intelligent agents, driven by the Q-learning algorithm, dynamically learn and adjust based on environmental feedback. The overarching objective is to enhance traffic signal control by identifying strategies that yield the highest cumulative reward over time.

2.1.3 Agent Learning Application

In the implemented learning process, each intersection is represented by an agent for reinforcement learning.The traffic volume, saturation, and signal phase, as detailed in the traffic environment information section, are encoded to reflect the environment. The agent determines the appropriate action, i.e., the green signal ratio, crucial for minimizing delay time.

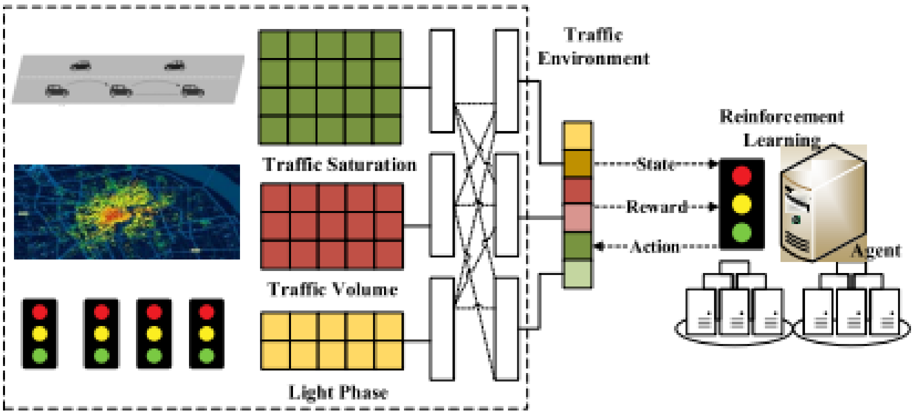


Figure 2.1: Learning process based on information fusion for traffic signal control.

In this learning approach, the current traffic state serves as input, representing the traffic environment. Initially, an action is randomly chosen from the action set A(s) to be applied to the current state. Subsequently, based on the feedback received, adjustments are made to transition the traffic state to the next state in accordance with updated Q-values, traffic conditions, and time. This iterative decision-making process is outlined in Algorithm 1, Agent Learning for a Single Intersection (ALSI).[6]

Algorithm 1 Agent Learning for a Single Intersection (ALSI)

Input: Current traffic state s, action set $A(s)$
Output: Action a_t for handling traffic signal control
while T in episode:
    Initialize s and $Q(s,a) \leftarrow 0$
    for step in each T
        Choose action a_t from $A(s_t)$ via the ϵ-greedy method
        Take this action a_t and obtain feedback r_t
        $Q_t(s_t, a_t) = (1 - \beta_t) Q_t(s_t, a_t) + \beta_t \left[r_t + \gamma \max_a Q_t(s_{t+1}, a_t) \right]$
        $s_t \leftarrow s_{t+1}$ && $t \leftarrow t + 1$
    end for
end while

Figure 2.2: Algorithm 1-Agent Learning for a Single Intersection(ALSI)

The ALSI algorithm outlines the steps for each episode. Firstly, the environmental state s is reset, and the $Q(s, a)$ value is initialized to 0. Then, at each step of the episode, the current traffic state s_t is observed, and an appropriate action a_t is chosen according to the current strategy using the -greedy method. Action a_t is selected from the action set $A(s)$ to be applied to the current state s_t, resulting in feedback r_t. The environment's state transitions to the next state s_{t+1} based on the estimated value $Q_t(s_t, a_t)$.

The update of the Q value follows the temporal difference (TD) method, as expressed by:

$$Q_t(s_t, a_t) = (1 - \beta_t)Q_t(s_t, a_t) + \beta_t \left(r_t + \gamma \max_a Q_t(s_{t+1}, a) \right)$$

Given that the reward function is based on delay, the strategy aims to maximize rewards. Subsequently, the state space and time phase are updated, where β_t in the range (0, 1) denotes the update rate. Then, s is updated to s_{t+1}, and t is incremented by 1. This process iterates through the described steps until the episode concludes.

Accounting for the interconnected nature of traffic systems, controlling multiple intersections requires agents to balance traffic flow across the entire area, aiming for minimal wait times and queue lengths. To address this, we employ multiagent reinforcement learning, enhancing our understanding of the traffic environment.

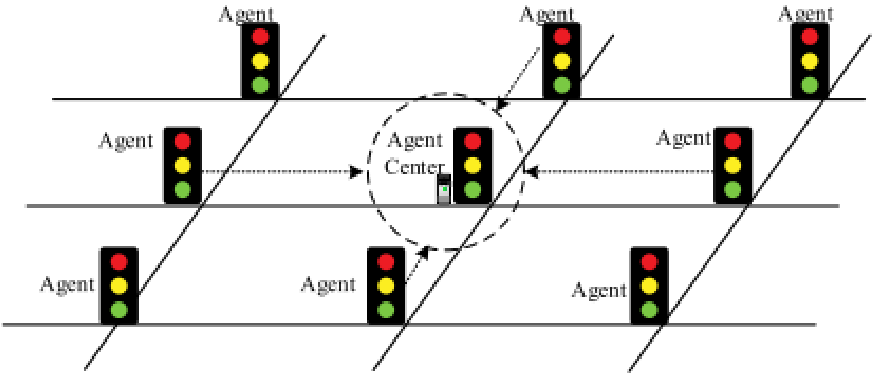


Figure 2.3: Agent learning for multiple intersections.

Figure 2.5 illustrates the topology of agent learning for multiple intersections. Each agent learns its local traffic dynamics while also interacting with neighboring agents. One agent, designated as the central agent, engages in learning through information fusion, wherein all agents share their knowledge. This approach, resembling the cognition, learning, feedback, and evolution (CLFE) loop, enhances the central agent's perception of the traffic environment by integrating a wealth of information.

Inter-agent communication is pivotal for regional traffic control, effectively creating a collective agent for reinforcement learning. As depicted in Figure 2.6, interactions between adjacent intersections, such as A and B, facilitate the exchange of traffic flow information. This shared information underpins robust control strategies based on community consensus, shaping traffic signal control actions in multiagent reinforcement learning scenarios.

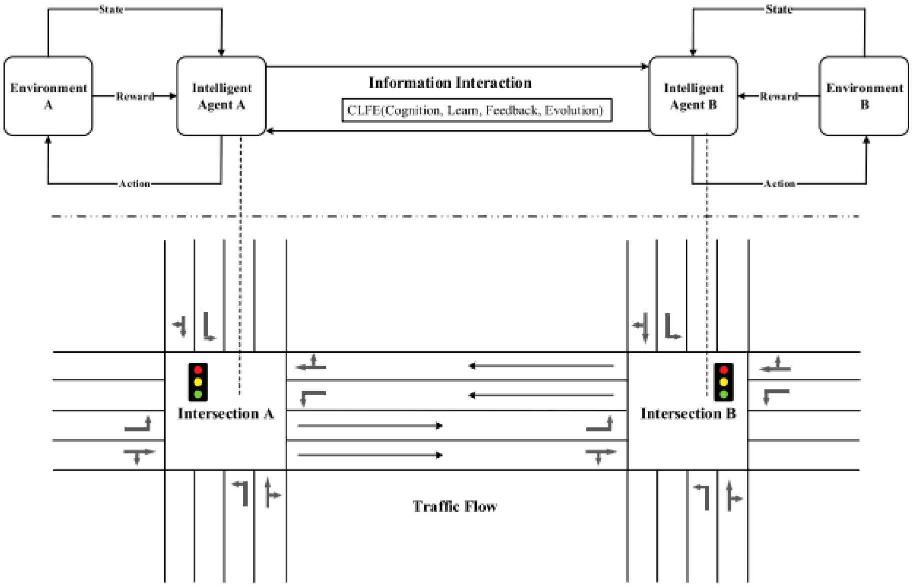


Figure 2.4: Traffic flow information exchange between intersections.

The algorithm for learning across multiple intersections can be viewed as an expanded reinforcement learning framework with richer environmental information. Algorithm 2, Agent Learning for Multiple Intersections (ALMI), outlines this process. Given the need to manage interactions among agents, each agent learns and stores optimal values. We introduce the global action function $V(s, a)$ for multiple agents, aligning it with the $Q(s, a)$ table for the central agent's learning. The update of $V(s, a)$ relies on the values derived from $Q(s, a)$. Subsequently, setting parameters such as the discount factor and learning rate before reiterating the Q-learning algorithm. ALMI encapsulates these methods for effective learning across multiple intersections.[7]

Algorithm 2 Agent Learning for Multiple Intersections (ALMI)

Input: Current traffic state $\{s1, s2, s3 \ldots, sn\}$, one agent selected as the center agent

Output: Action a_t for handling traffic signal control

1: Initialize $Q(s,a) \leftarrow 0$ and $V(s,a) \leftarrow 0$

2: Read the traffic states $\{s1, s2, s3 \ldots, sn\}$ and corresponding action sets *A(s1), A(s1)..., A(sn)* from neighboring traffic intersections

3: **while** (*compare (Q, V)*)

4: Generate the maximum value *Q* in each state using the ALSI algorithm

5: Compute the reward based on the delay when each action is performed

6: Update the *Q value*, *status* and *time*

7: Store the maximum *Q* value for each intersection and update it to *V(s, a)*

8: Detect changes in *Q(s, a)* and *V(s, a)* that support *compare (Q, V)*

9: **end while**

Figure 2.5: Algorithm 2-Agent Learning for Multiple Intersections (ALMI)

The ALMI algorithm commences by initializing $Q(s, a)$ and $V(s, a)$ based on the observed traffic state. If a discrepancy exists between $Q(s, a)$ and $V(s, a)$, the algorithm employs a comparison method (Q, V) to assess changes, indicating that ALMI has yet to achieve a balanced, globally optimal state.

Next, the algorithm proceeds to determine the queue length and feasible actions for a single phase. Subsequently, it establishes the transition function from queue length to the corresponding state.

In the learning process's fourth step, the maximum Q value associated with a specific state is identified. The reward value is then computed based on the queue length before and after each action is taken.

Following this, the estimated $Q(s, a)$ value is updated according to the state-action pair. Finally, the algorithm stores the maximum Q value obtained at each intersection and updates it to derive $V(s, a)$. This iterative process aims to achieve global optimization of the traffic signal control strategy, persisting until $Q(s, a)$ and $V(s, a)$ converge.

2.2 Traffic signal control with reinforcement learning based on region-aware cooperative strategy[2]

2.2.1 Reinforcement Learning

Reinforcement learning, a subset of machine learning, involves the interaction between an agent and its environment to maximize a reward signal through iterative trial and error. The agent learns to make decisions based on observations from the environment, receiving feedback in the form of rewards or penalties, with the ultimate aim of discovering the optimal policy for maximizing cumulative rewards.

In the domain of traffic light control, reinforcement learning algorithms, such as Q-table learning, linear regression Q-learning, deep Q-learning, dueling Q-learning, deep deterministic policy gradient (DDPG), and meta reinforcement learning, are utilized to optimize traffic signal switching times. These algorithms are continuously trained as they interact with the traffic environment, with the agent learning to select actions based on observed traffic conditions at intersections.

One notable approach, the Regional Cooperative Strategy (RACS), dynamically integrates neighboring observations to coordinate traffic light control across multiple intersections. Performance evaluations, measured in terms of wait time and queue length, demonstrate RACS's superiority over other reinforcement learning methods. This improvement is attributed to enhanced spatial and temporal coordination and the implementation of cooperative strategies among neighboring intersections.

2.2.2 Advantage Actor-Critical (A2C) Algorithm

The A2C algorithm presents a decentralized framework merging a policy function and a value function, ideal for handling continuous state spaces. It establishes global control for each local RL agent or junction, enabling independent decision-making at each junction. This methodology addresses the issue of partial observability by incorporating spatial information gleaned from neighboring agents. The A2C algorithm aims to ensure stability and dependability in multi-agent traffic signal control scenarios.

2.2.3 Region-Aware Cooperative Strategy (RACS) based on Graph Attention Network (GAT)

RACS extends the A2C algorithm by incorporating information from neighboring intersections and the state of each local intersection. It employs a graph attention network (GAT) to dynamically learn the weighted influence exerted by surrounding intersections. This approach enables each intersection to discern the state of neighboring intersections and generate distinct weight distributions. Additionally, RACS introduces an attention-based discount factor that attenuates the influence of state and reward signals from neighboring intersections. By integrating observations and policies from adjacent intersections with the current intersection's status, RACS determines optimal intersection actions and traffic light operations. Experimental findings validate the effectiveness and scalability of the RACS model, showcasing its superior performance compared to traditional and RL-based methods.

A. Definition of State, Action, and Reward:

To ensure the safe passage of vehicles through the intersection without collisions, it's crucial to provide sufficient time for the remaining vehicles to clear the intersection when switching the traffic light from green to yellow. This time difference between the action taken by the agent and the corresponding change in the environment is denoted as δt. A previous work $\delta t = 10s$ and $t_y = 5s$ were suggested. However, for this study, choose to set $\delta t = 5s$ and $t_y = 2s$. This adjustment ensures smooth information exchange between the agent and the environment, allowing vehicles to safely pass through the intersection when the traffic light changes.

1) Action: The action taken by an intersection is determined by the combination of feasible signal phases without conflicts, as proposed . This approach ensures both the safety of vehicles passing through the intersection and the maximization of vehicle throughput. Denoting the action taken by the i-th agent or intersection at time step t as $a_{t,i}$, it represents one of all feasible actions available at the intersection, with an execution time of δt.

2) State: Define the state of the intersection i as $s_{t,i} = [wait_t[l], wave_t[l]]_{j\in e, l\in L_{ji}}$, where l represents each incoming lane of the intersection i. $wave_t$ indicates the number of vehicles at the intersection, calculated using induction-loop detectors in Simulation of Urban MObility (SUMO). These detectors are positioned in each lane 50 meters away from the central intersection. $wait_t$ measures the cumulative wait time of all vehicles at the intersection. It is calculated by summing up the number of vehicles at the intersection per second. During the simulation experiment, the laneAreaDetector API in SUMO [36] is utilized to subscribe to vehicle state.

3) Reward: The reward definition should intuitively evaluate the quality of actions taken by the agent. While some research defines the reward solely based on the cumulative wait time of vehicles at the intersection during $t+\delta t$ periods [7], [13], in this article, we define the reward following the approach in [35]. $r_{t,i} = -\sum_t^{t+\delta t}(queue[l] + p\cdot wait[l])_{j\in e, l\in L_{ji}}$, where p [veh/s] is the tradeoff factor. $queue$ represents the queue length at each incoming lane. This reward definition is directly related to the state and can intuitively indicate whether an action is beneficial or detrimental. In summary, the reward definition emphasizes both traffic congestion and trip delay.

2.2.4 Independent Advantage Actor-Critic

For a multi-agent system $G = (V, E)$, the neighborhood of agent i is denoted as N_i, and a local region that consists of agent i is represented by $v_i = N_i \cup \{i\}$. For any neighbor agent $j \in N_i$, d_{ij} measures the length of the connected edge between agent i and agent j, where edge $E_{ij} \in E$. d_{ij} represents the distance between two connected intersections on the synthetic traffic grid or real-world traffic network. In this study, each agent or intersection i updates its own policy π_{θ_i}, and the critic network is also updated based on the corresponding state-value V_{w_i}.

Independent Q-learning (IQL) treats other agents' states as part of the environment, while other agents' policies are continuously updating during model training. Extend this concept to independent advantage actor-critic (IA2C). Assuming that the reward and state are shared among other agents, for centralized A2C, the agent can receive feedback from the environment as shown where $|B|$ is the size of one mini-batch.

$$Q(s,a) = R_t + \gamma^n V_{w_{-i}}(s_t|\pi_{\theta_{-j}})$$

where $n = tB - t$, R_t is from the same sampling policy $\pi_{\theta_{-j}}$, and $\pi_{\theta_{-j}}$ is the current policy, and $V_{w_{-i}}(s_t|\pi_{\theta_{-j}})$ is the state-value estimation at the last state.

Suppose there exists limited information exchange between local agents in IA2C. In other words, each agent can consider its neighbor's state as its partial environment instead of one state s_t. At time step t, the input of agent i is represented as $s_{t,v_i} = \{s_{t,j}|j \in v_i\}$. Then, the loss function of the critic network and equation becomes

$$L(w_i) = \frac{1}{2|B|}\sum_{t\in B}\left(Q(s,a) - V_{w_i}(s_{t,v_i})\right)^2$$

where V_{w_i} is based on the local region state s_{t,v_i}, however $Q(s,a)$ relies on state s_t. This property may result in nonstationarity during model training. Moreover, we can update the actor network according to the following equation:

$$L(\theta_i) = -\frac{1}{|B|}\sum_{t\in B}\log \pi_{\theta_i}(a_{t,i}|s_{t,v_i})Advt_i$$

where $Advt_i = Q(s,a) - V_{w_{-i}}(s_{t,v_i})$, the nonstationarity problem still exists since $Q(s,a)$ is based on the current policy $\pi_{\theta_{-i}}$, while w_{-i} is updated by the previous policy $\pi_{\theta_{-i}}$.

2.3 Collaborative Traffic Signal Automation Using Deep Q-Learning[3]

Multi-agent deep reinforcement learning (MDRL) has gained traction as a favored method for coordinating traffic signal control across numerous intersections, fostering decentralized cooperation within specific traffic networks. Despite its widespread adoption, current MDRL algorithms encounter certain limitations. Firstly, the complexities inherent in multi-agent setups impede the transferability and generalization of traffic signal policies across diverse traffic networks. Secondly, existing MDRL algorithms face hurdles in adapting to fluctuating vehicle volumes traversing the traffic networks.

To tackle these challenges, this research introduces a novel approach called Cooperative Multi-Agent Deep Q-Network (CMDQN) for traffic signal control, aimed at alleviating traffic congestion. The CMDQN incorporates innovative elements such as signal state at preceding junctions, inter-junction distances, visual cues, and average vehicle speeds. Employing a Decentralized Multi-Agent Network (DMN) framework and a Markov Game abstraction, this approach facilitates collaboration and information exchange among agents to reduce waiting times. Leveraging Reinforcement Learning (RL) and a Deep Q-Network (DQN) for adaptive traffic signal control, the method integrates Deep Computer Vision for real-time traffic density analysis. Furthermore, it proposes intersection-specific and network-wide reward functions to evaluate performance and optimize traffic flow.

To assess the efficacy of the system, experiments were conducted using both synthetic and real-world traffic scenarios. Synthetic network simulations were carried out using the Simulation of Urban Mobility (SUMO) traffic simulator, while real-world traffic data collected from cameras installed at actual traffic signals was employed for validation.

2.3.1 Problem Formulation

The architecture and components of a Multi-Agent System (MSA) designed for traffic management includes :

1. **Action Space:**

Each signal at each time-step in the MSA has a discrete action set. The intersection can choose an action from this action space. The potential action space is defined by a set of individual actions available for each agent of the MSA. The duration of actions at each intersection varies between L_{min} and L_{max}.

2. **Observation Space:** Each agent observes the agent state as their observation. The agent state includes three blocks: Node features set (including intersection, traffic flow, and road nodes) Adjacency matrix (links between intersections) Intersection mask (filters the intersection graph embeddings) Node features include attributes such as traffic flow position, actual speed, and traffic flow location.

3. **Reward Function:** The MSA utilizes two types of reward functions: intersection reward and network-wide reward function. Intersection reward aims to minimize waiting time and increase throughput at each signal. A local reward function is defined for each intersection, considering factors such as queue length and traffic flow delay. The reward function for each intersection is influenced by the actions of its neighbors and the duration of actions at each intersection.

Overall, the system aims to optimize traffic flow by allowing intersections to take actions based on observed states, with the goal of maximizing rewards defined by minimizing waiting times and delays, and increasing overall throughput. The system also emphasizes collaboration between agents to achieve these goals efficiently.

2.3.2 Deep Q Network

The implementation of a traffic control system using a combination of Deep Q-Network (DQN) and reinforcement learning (RL) methods.

1. **Deep Q-Network (DQN):** DQN is a combination of Convolutional Neural Network (CNN) and Q-learning. It is effective for processing high-dimensional inputs like traffic images. DQN eliminates the need for environment simulation and learns the optimal policy directly from the physical environment. The DQN receives the current signal state and traffic flow information to determine the next action. The objective of the controller is to select actions at each time step that maximize future rewards.

2. **Q-Function and Bellman Equation:** The Q-function represents the expected cumulative future reward for taking action A_t in state S_t. The Bellman equation expresses the optimal Q-function recursively. The discount factor ρ balances immediate and future rewards. The Q-function is approximated by a CNN with parameters θ, denoted as $Q(S, A; \theta)$. The CNN is trained to approximate the Q-function by minimizing the loss function $L(\theta)$.

3. **Stability Improvement Techniques:** Reinforcement learning methods with nonlinear approximation functions can suffer from instability due to data correlation. To mitigate this, experience replay and periodic updates of the network parameters are introduced. Experience replay stores the agent's interactions with the environment in a memory buffer, reducing correlation between training samples. The Q-network parameters are updated with a lower frequency compared to the actual Q-network to stabilize training. Target parameters of the Q-network are updated only periodically (every C steps) to prevent rapid fluctuations.

In summary, the combination of DQN and RL methods allows for the learning of optimal traffic control policies directly from observed traffic states, with stability improvements achieved through techniques like experience replay and target network updates.

2.3.3 Training Mechanism

The models were trained and evaluated on three synthetic traffic scenarios, each featuring different road network structures and traffic signal programs:

1) Scenario S1: This scenario consists of 2 phases and 2x2 inbound lanes. 2) Scenario S2: Designed with 4 phases and 3x3 inbound lanes. 3) Scenario S3: Configured with 4 phases and 4x4 inbound lanes.

In Scenarios S1 and S2, a single inbound lane is designated for permissive and protected left turns respectively, while in Scenario S3, two inbound lanes are allocated for protected left turns. The signal timings for change, clearance, and minimum green intervals are consistent across all scenarios, with Ty = 5 seconds, Tr = 3 seconds, and Tg, min = 15 seconds respectively. Each approach spans 500 meters, and considering vehicle size and inter-vehicle gap in the SUMO simulation environment, cells of 10 meters were used in CMDQN, with a detection range of 300 meters to accommodate up to 30 connected vehicles per lane.

SUMO simulations were conducted for 5000 seconds per episode, with randomly generated traffic demand creating a heterogeneous traffic environment. Each episode featured a random traffic penetration rate within the range $[0, 1]$ and random traffic flows for insertion within $[10, 2000]$ vehicles per hour for each entry approach. The traffic demand followed a Poisson process with a parameter set to 3 times the inverse of the flow. The CMDQN agent underwent training for 6 million timesteps, equivalent to 50 hours on a GPU, in each scenario following an ϵ-greedy policy. Subsequently, in the deployment phase, the CMDQN agents were evaluated in each scenario using the trained neural network weights, adhering to the optimal policy learned during training.

The ϵ-greedy action selection policy effectively balanced exploration and exploitation during learning, addressing the exploration-exploitation trade-off challenge in the SUMO-based CMDQN traffic controller. This policy gradually reduced randomness over time, aiding the CMDQN's training and enhancing adaptability in diverse traffic scenarios, as demonstrated in both synthetic and real-world networks.[8]

2.4 A Deep Q Learning Network for Traffic Lights' Cycle Control in Vehicular Networks[4]

Current traffic light control methods often lead to inefficiencies, causing long delays and unnecessary energy consumption. To address these issues, it's essential to utilize real-time traffic data for dynamically adjusting traffic light durations. Existing approaches typically either allocate equal durations to traffic signals or rely on limited information from collected data. In this study, explores a novel approach to determining traffic signal durations using data gathered from various sensors and vehicular networks.

The research introduces a deep reinforcement learning model for traffic light control, which conceptualizes the intricate traffic environment through states generated from data collection and division of intersections into smaller grids. Timing adjustments of traffic lights are viewed as actions within a high-dimensional Markov decision process. The reward function is defined based on the cumulative disparity in waiting time between two cycles.

To address the complexity of the model, employed a convolutional neural network to map states to rewards. The model incorporates several components to enhance performance, including dueling networks, target networks, double Q-learning networks, and prioritized experience replay.

The effectiveness of the model is assessed through simulation using the Simulation of Urban Mobility (SUMO) platform within a vehicular network. The simulation outcomes illustrate the efficacy of our method in managing traffic signals.

2.4.1 The Reinforcement Learning Model

A.States:

Defining states based on the position and speed of vehicles at an intersection requires gathering vehicle position and speed data using a vehicular network or similar tools. The traffic light generates a virtual snapshot image of the intersection, partitioned into small square-shaped grids of uniform size. The length of these grids, denoted as c, ensures that no two vehicles can occupy the same grid, with each grid accommodating a single vehicle to streamline computation. Within each grid, the state is represented as a two-value vector $< position, speed >$ of the vehicle inside. The position dimension is binary, indicating vehicle presence (1 for occupied, 0 for empty), while the speed dimension is an integer denoting the vehicle's current speed in meters per second.

Figure illustrates this process: Fig. 2.8 depicts a snapshot of the traffic status at a simple one-lane four-way intersection, while Fig. 2.9 shows the corresponding position matrix representing the presence of vehicles on the road.[9]

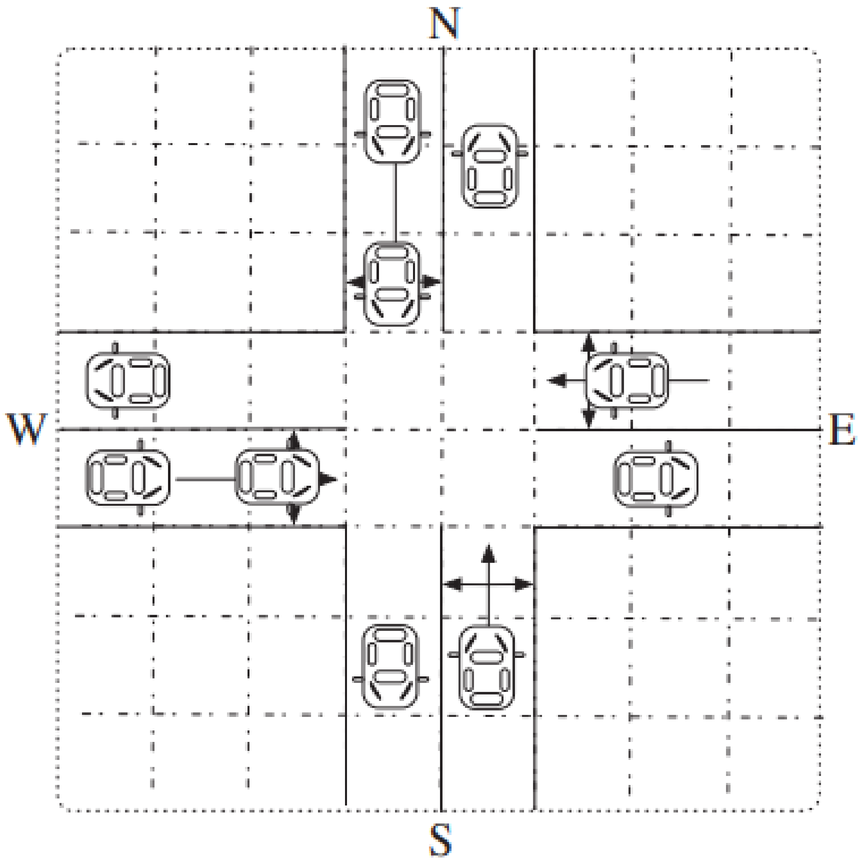


Figure 2.6: The snapshot of traffic on a road at one moment.

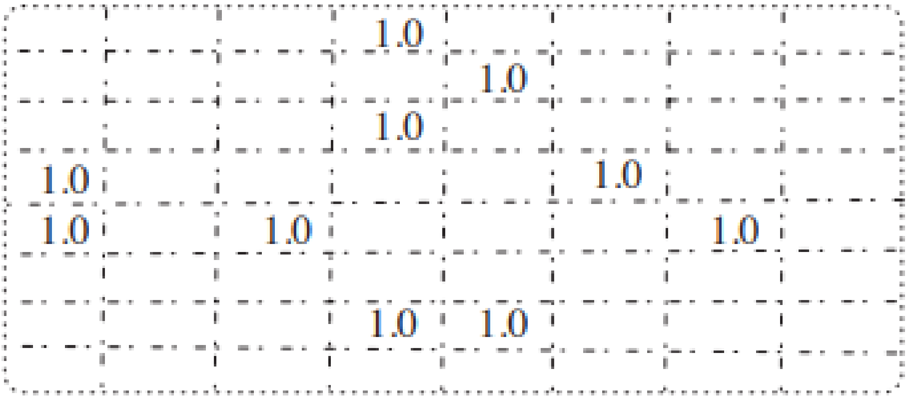


Figure 2.7: The corresponding position matrix on this road.

B. Action Space:

In the model, the action space defines how to update the duration of each phase in the next cycle. To prevent instability, specifying a change step of 5 seconds between cycles. Each phase's duration change between neighboring cycles is modeled as a high-dimensional Markov decision process (MDP). The traffic light adjusts only one phase's duration by 5 seconds if there is any change. [10]

Consider the intersection in Fig. 2.8, where four phases exist: north-south green, east-north west-south green, east-west green, and east-south west-north green. The unmentioned directions default to red. Yellow signals, not discussed here, are inserted for safety. Each phase's duration in the current cycle is denoted as $< t1, t2, t3, t4 >$. Fig. 2.10 illustrates the legal actions for the next cycle, where each circle represents the durations of the four phases. The duration of one phase in the next cycle is the current duration plus or minus 5 seconds. After selecting the phase durations for the next cycle, the current durations are updated accordingly. The traffic light selects actions based on this procedure, with phase durations constrained between 0 and 60 seconds.

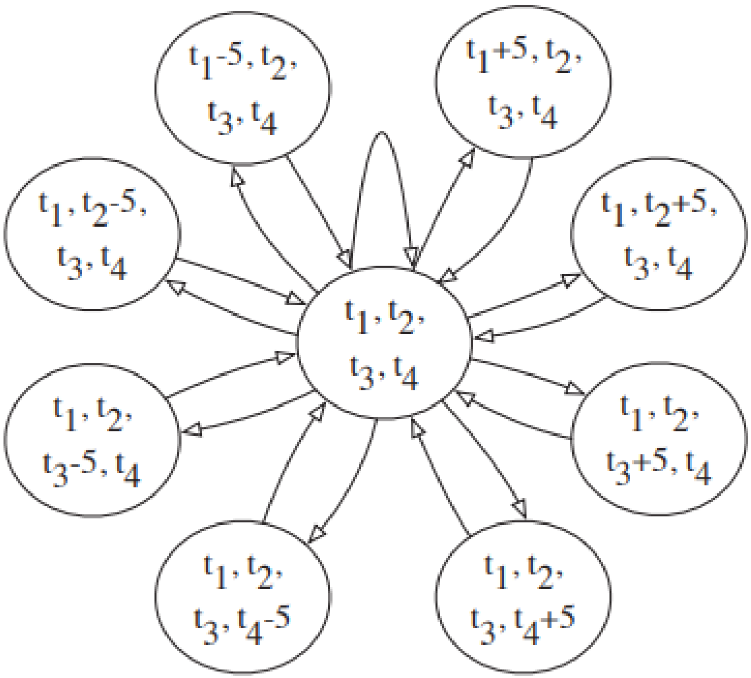


Figure 2.8: Part of the Markov decision process in a multiple traffic lights scenario.

The MDP can be extended to more complex intersections with additional phases or irregular configurations. The circle's dimension in the MDP equals the number of phases at the intersection.

C. Rewards:

Rewards provide feedback to the reinforcement learning model about previous actions' performance. In the system, the primary goal is to increase intersection efficiency and reduce vehicle waiting time. Thus, define rewards as the change in cumulative waiting time between neighboring cycles.

Let W_t represent the cumulative waiting time up to the starting time of cycle t, and $w_{it,t}$ denote the waiting time of vehicle i until cycle t. The reward in cycle t is defined as $r_t = W_t - W_{t+1}$. This reward reflects the change in cumulative waiting time before and after the action. By maximizing this reward, the aim is to reduce waiting time at the intersection.

2.4.2 Double Dueling Deep Q Network

[11] The number of states is prohibitively large, making it challenging to directly solve Bellman equation. To address this, propose the use of a Convolutional Neural Network

(CNN) to approximate the Q value. Using state-of-the-art techniques, the proposed network is termed the Double Dueling Deep Q Network (3DQN).

A. Convolutional Neural Network

The architecture of our CNN comprises three convolutional layers and several fully-connected layers. The input to the CNN consists of small grids representing the position and speed information of vehicles at an intersection. Each intersection grid is 60×60, resulting in input data dimensions of $60 \times 60 \times 2$ (for position and speed). These data are processed through three convolutional layers, each containing convolution, pooling, and activation components. Leaky ReLU is employed as the activation function to accelerate convergence and prevent 'dead' neurons.

The architecture specifics include:

First convolutional layer with 32 filters of size 4×4, moving with a stride of 2×2.

Second convolutional layer with 64 filters of size 2×2, moving with a stride of 2×2.

Third convolutional layer with 128 filters of size 2×2, moving with a stride of 1×1. Following the convolutional layers, a fully-connected layer transforms the tensor into a 128×1 matrix, which is then divided into two parts: one for value calculation and the other for advantage estimation.

B. Dueling DQN

This network incorporates a dueling DQN, where the Q value is derived from the value of the current state and the advantage of each action relative to others.

C. Target Network

A target network is utilized to guide parameter updates in the primary network. The target Q value is calculated separately and updated using Mean Square Error (MSE).

D. Double DQN

Adopt the double Q-learning algorithm to mitigate overestimations and enhance performance.

E. Prioritized Experience Replay

During the update process, a prioritized experience replay strategy is employed to accelerate learning and improve the final policy.

F. Optimization

Optimize the neural networks using the Adam optimization algorithm, which efficiently updates learning rates based on both first-order and second-order moments.

G. Overall Architecture

The entire architecture integrates the CNN, dueling DQN, target network, and prioritized experience replay to train an adaptive traffic light system. The agent learns to dynamically adjust phase durations based on traffic scenarios to maximize rewards. The pseudocode of the 3DQN with prioritized experience replay is provided in Algorithm 1.

Algorithm 1 Dueling Double Deep Q Network with Prioritized Experience Replay Algorithm on a Traffic Light

Input: replay memory size M, minibatch size B, greedy ϵ, pre-train steps tp, target network update rate α, discount factor γ.
Notations:
θ: the parameters in the primary neural network.
θ^-: the parameters in the target neural network.
m: the replay memory.
i: step number.

Initialize parameters θ, θ^- with random values.
Initialize m to be empty and i to be zero.
Initialize s with the starting scenario at the intersection.
while there exists a state s **do**
  Choose an action a according to the ϵ greedy.
  Take action a and observe reward r and new state s'.
  if the size of memory $m > M$ **then**
    Remove the oldest experiences in the memory.
  end if
  Add the four-tuple $\langle s, a, r, s' \rangle$ into M.
  Assign s' to s: $s \leftarrow s'$.
  $i \leftarrow i + 1$.
  if $|M| > B$ and $i > tp$ **then**
    Select B samples from m based on the sampling priorities.
    Calculate the loss J:
    $$J = \sum_s \frac{1}{B}[r + \gamma Q(s', \arg\max_{a'}(Q(s', a'; \theta)), \theta^-) - Q(s, a; \theta)]^2.$$
    Update θ with ∇J using Adam back propagation.
    Update θ^- with θ:
    $$\theta^- = \alpha\theta^- + (1 - \alpha)\theta.$$
    Update every experience's sampling priority based on δ.
    Update the value of ϵ.
  end if
end while

Figure 2.9: Algorithm 1-Dueling Double Deep Q Network with Prioritized Experience Replay Algorithm on a Traffic Light

This paper introduces a novel approach to addressing the traffic light control problem through the utilization of a deep reinforcement learning framework. Traffic data is acquired either from vehicular networks or through camera surveillance. The states are represented as two-dimensional values containing information on the position and speed of vehicles. Actions are formalized as a Markov decision process, while rewards are defined as the cumulative waiting time difference between successive cycles.

To tackle the intricacies of our traffic scenario, we propose the utilization of a double dueling deep Q network (3DQN) alongside prioritized experience replay. This model exhibits the capability to learn effective policies under varying traffic conditions, including both rush hours and normal traffic flow rates. Notably, the model demonstrates a remarkable reduction of over 20 percent in average waiting time compared to initial training stages. Extensive simulations conducted in SUMO and TensorFlow validate the superiority of our proposed approach, particularly in terms of learning speed.

2.5 Network-wide traffic signal control optimization using a multi-agent deep reinforcement learning[5]

An approach to achieving optimal traffic control of networks through research and network-wide deep reinforcement learning is proposed, which involves using information to optimize traffic lights. It presents the KS-DDPG algorithm based on multi-agent reinforcement learning (MARL) and using deep deterministic policy gradient (DDPG) to optimize traffic signal control. The algorithm also includes an information exchange mechanism that allows agents to share information and coordinate actions to improve the efficiency and safety of traffic flows.

The KS-DDPG algorithm is designed to solve traffic signal control optimization problems in a network-wide environment. Traditional methods such as fixed time and Max-Pressure controls cannot learn from environmental feedback and hardly achieve satisfactory results. DQN and DDPG also have limited potential to reduce congestion, especially in dynamic traffic situations, mainly because they independently optimize a single intersection strategy. The proposed KS-DDPG algorithm overcomes these limitations by enabling explicit communication between agents through a data sharing mechanism.

The KS-DDPG sharing mechanism allows agents to access a collective representation of the traffic environment, which improves coordination between agents. Each agent has his own way of interpreting and constructing collective knowledge, and agents must consider both their personal observation and understanding of collective knowledge when making decisions. Agents use a communication protocol to retrieve and update information during the cycle. The algorithm uses deep neural networks as feature estimators for policies and functions with learnable gates to facilitate the interaction of each agent with the data.

To evaluate the KS-DDPG algorithm using the Simulation of Urban Mobility (SUMO) platform, an open source traffic simulation package designed to manage large road networks. SUMO provides a realistic environment for simulating traffic conditions and interactions, which allows to evaluate the effectiveness of the algorithm in controlling traffic signals in different network scenarios.

Demonstration results show that the KS-DDPG algorithm achieves consistent performance improvement over various road networks and traffic conditions. The advantage increases further when the estimated traffic scenario changes from a synthetic normal network to a real irregular and dynamic environment. The algorithm also achieves control improvements of at least 10% compared to the state-of-the-art multi-agent RL, i.e., MADDPG. The growing differences between KS-DDPG and MADDPG from the Grid experiment to the MoCo experiment also indicate that direct cooperation through information sharing is better than indirect cooperation based on other policies.

2.5.1 Multi-Agent Reinforcement Learning (MARL)

MARL (Multi-Agent Reinforcement Learning) is a subfield of reinforcement learning where multiple agents interact with each other and the environment to achieve individual and/or collective goals. In MARL, each agent is an independent entity that perceives the environment, acts, receives rewards, and learns from its interactions. The main aspects of MARL are:

Decentralized Decision Making: Each MARL agent makes decisions independently based on its own observations and local information. Agents may have different goals or modes of action and must learn to coordinate their actions to achieve optimal results.

Centralized learning, Decentralized execution: Although agents learn independently, they can share information during the learning process to improve coordination. However, during execution, each agent typically operates under its own policy without centralized control.

Emergent behavior: MARL systems can experience interactions with the environment and other factors. This means that the collective behavior of multiple agents can lead to results that cannot be achieved by individual agents alone.

Communication and Coordination: Effective communication and coordination between agents is critical to achieving cooperative goals in MARL. Agents can share observations, actions or learned practices to improve collaboration and improve overall performance.

Challenges: MARL presents challenges such as non-stationarity (the environment changes as agents learn), credit distribution (rewarding actions), exploration-exploitation trade-offs (balancing new actions and using known strategies), and scalability (dealing with large problems). quantities of matter and complex interactions).

MARL has applications in many fields, including robotics, autonomous driving, traffic control, gaming, and multi-agent systems. By enabling agents to learn and adapt in dynamic environments by interacting with other agents, MARL provides an efficient framework for solving complex decision problems where multiple actors must cooperate and compete to achieve desired outcomes.

2.5.2 Deep Deterministic Policy Gradient (DDPG)

Deep Deterministic Policy Gradient (DDPG) is a model-free, policy-free reinforcement learning algorithm that combines the advantages of deep learning with deterministic policy gradients. DDPG is particularly suitable for continuous operation in reinforcement learning tasks. Here are the main components and features of DDPG:

Actor-critic architecture: DDPG uses an actor-critic architecture where a network of actors learns policy and a network of critics evaluates actor actions. An actor network produces the best action in a given state, while a critic network analyzes a value function to evaluate an actor's action.

Deterministic policy: Unlike stochastic policies that generate a probability distribution of actions, DDPG uses a deterministic policy that directly produces the best action for a given state. This deterministic nature simplifies the trade-off between research and use and makes learning more stable in continuous operational mode.

Experience replay: DDPG uses experience replay, a technique that stores past experiences (state, action, reward, next state) in a replay buffer and tries a minibatch during practice. Experience repetition helps correlate experience, reduce update variability, and improve sampling efficiency.

Target Networks: DDPG uses target networks for both actor and critical networks to stabilize training. Target networks are copies of master networks that are slowly updated using a soft update mechanism. This helps prevent target values from changing during training.

Q-function approximation: The DDPG critical network approximates an action value function (Q-function) to estimate the expected reward of an action in a given state. By learning the Q function, the DDPG can evaluate the quality of the operations performed by the operator.

Policy gradient updates: DDPG uses a deterministic policy gradient statement to update the actor network. The policy gradient is calculated based on a deterministic action value gradient that directs the actor to actions that maximize expected reward.

Continuous Modes: DDPG is well suited for continuous mode tasks such as robot control, autonomous driving, and continuous control problems in reinforcement learning. A deterministic policy simplifies the choice of actions in such environments.

In general, DDPG is an efficient algorithm for solving reinforcement learning problems with continuous operation. Its combination of deep neural networks, deterministic practice gradients, repetition of experience and goal networks make it effective in learning complex practices for continuous control tasks. [12]

2.5.3 Simulation of Urban MObility (SUMO)

SUMO, Urban Traffic Simulation, is an open source microscopic traffic simulation software designed for comprehensive urban traffic analysis. It provides detailed modeling and research on the behavior of vehicles, pedestrians and other road users in complex transport networks. Key features of SUMO include microscopic simulation at the level of individual vehicles, scalability for managing large-scale road networks, TraCI Control Interface (TraCI) for real-time external control and monitoring, realistic modeling of vehicle behavior, integration with reinforcement learning frameworks. , extensibility of custom traffic models and control algorithms, and visualizers to display traffic networks and perform detailed analysis. SUMO is a valuable tool for researchers, transport planners and decision makers to analyze and optimize urban traffic, traffic management strategies, intelligent transport systems and the impact of infrastructure changes on traffic flows and congestion. Its ability to simulate realistic traffic scenarios makes it widely used in urban traffic research and development.

2.5.4 Knowledge Sharing

Information sharing refers to the exchange of knowledge, skills or expertise between individuals or groups within an organization or community. This includes disseminating information and best practices to improve learning, problem solving, and decision making. In relation to the document you sent, the term "knowledge sharing" apparently refers to the exchange of insights, observations and methods related to traffic light optimization and urban traffic research. This process allows researchers, practitioners and decision makers to benefit from each other's experiences and contribute to the development of effective traffic management strategies and intelligent transport systems. The document's data sharing flowchart fig. 2.12, describes a likely systematic approach to the dissemination and use of data to improve understanding and optimization of traffic light management in the urban environment.

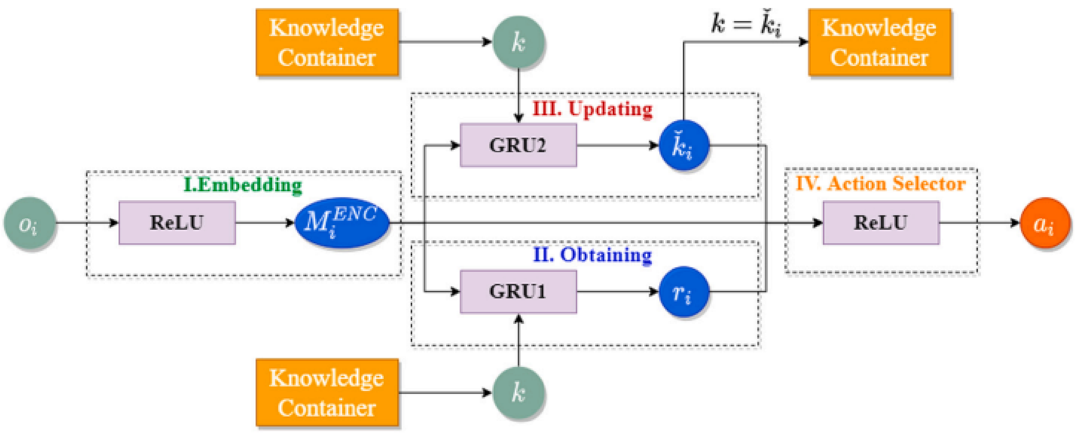


Figure 2.10: flowchart for knowledge sharing

2.6 Summary and Gaps Identified

Reinforcement learning (RL), including Q-learning and deep Q-learning, offers a dynamic approach to traffic optimization by learning optimal control policies through trial and error. RL algorithms can adapt to changing traffic conditions and learn from past experiences to make better decisions over time. Q-learning, with its simplicity and clear convergence properties, provides a solid foundation for RL-based traffic optimization. Deep Q-learning enhances this approach by using deep neural networks to handle high-dimensional state spaces and complex decision-making. However, RL methods require

SL NO	TITLE	METHODOLOGY	ADVANTAGE	DISADVANTAGE
1	An information fusion approach to intelligent traffic signal control using the joint methods of MARL and AIoT	Multiagent reinforcement learning with AIoT framework	Decentralized coordination, handle large scale traffic accomodation, adaptive to heterogenous environment, continuous learning and improvisation	Management of complex interacting agents, coordination challenges, high computational resources, sensitive to dynamic traffic environment
2	Traffic signal control with reinforcement learning based on RACS	Aware cooperative strategy	Innovation in traffic signal, improved perfomance, scalable, incorporation of surrounding intersections	More complex, limited real world validation, assumptions and simplifications in training time
3	Collaborative Traffic signal automation using deep Q learning	Reinforcement learning involving deep Q learning	Flexible representation of state-action pair, generalized policy updation, handle large scale traffic, enables continues learning and improvisation	Training complexity, requires large amount of training data, challenging in dynamic traffic environment, sensitivity to hyperparameters
4	A Deep Q learning network for traffic ligths' cycle control in vehicular network	Double Dueling Deep Q learning network	Enhanced learning ability, effective representation of state-action pair, reduced variance, flexible architecture	More complex, hyperparameter sensitivity, requires more computational resources, extensive data requirements
5	Network-wide traffic signal control optimization using a multi-agent deep reinforcement learning	A Deep Deterministic Policy Gradient Algorithm	Continuos action-space handling, handles high dimensional data spaces, action-critic policy so it can learn effectively from past experience, adapts to a dynamic environment.	High computational cost , limitations in transferability , Training instability, difficulty in reward designing.

Table 2.1: Comparison of Different Traffic Signal Optimization Approaches

extensive training and may struggle with scalability and generalization to new environments. They also rely heavily on accurate modeling of the traffic system, which can be challenging in real-world scenarios with uncertain dynamics and interactions.

Reinforcement learning methods like Q-learning and deep Q-learning provide a more adaptive and learning-based approach to traffic optimization. Combining the strengths of these approaches could lead to more robust and efficient traffic management systems, utilizing the adaptive decision-making capabilities of RL algorithms. However, addressing challenges such as scalability, generalization, and real-world implementation remains crucial for realizing the full potential of these techniques in traffic optimization.

Chapter 3

Requirements

3.1 Hardware Requirements

PC/Laptop Specifications:

- Processor: Intel Core i7-3770K or equivalent
- RAM: Minimum 8GB DDR3/DDR4 (16GB recommended for optimal performance)
- Graphics Card: NVIDIA GTX 970 or higher with CUDA support
- Storage: Solid State Drive (SSD) with a minimum of 256GB capacity for faster data access

Networking:

- Ethernet/Wi-Fi: Stable network connectivity for data transfer and model updates

3.2 Software Requirements

SUMO (Simulation of Urban MObility):

- Description: Open-source microscopic traffic simulation software for modeling and analyzing traffic flow
- Version: Latest stable release (current version: SUMO 1.10.0)

Anaconda Distribution:

- Description: Open-source distribution platform with Python and a collection of pre-installed libraries for easier package management
- Version: Latest stable release (current version: Anaconda 2022.x)

TensorFlow:

- Description: Machine learning framework for building and training AI models
- Version: Latest stable version (current version: TensorFlow 2.x)

Python/Python3:

- Description: Programming language used for scripting, data manipulation, and integration with libraries and frameworks
- Version: Python 3.8 or later

Additional Tools (Optional but Recommended):

- Pandas: Data manipulation and analysis library
- Matplotlib/Seaborn: Data visualization libraries for generating plots and charts

3.3 Functional Requirements

3.3.1 Traffic Data Acquisition

The system must gather real-time traffic data from multiple sources, including road cameras, sensors, and connected vehicles. This data will encompass vehicle count, speed, types, and traffic density.

3.3.2 Simulation

Upon acquiring data, the system will integrate information from various sources. This process involves aligning and processing disparate data streams of vehicles and roadmaps to generate a comprehensive understanding of traffic conditions.

3.3.3 AI-driven Traffic Pattern Analysis

Using machine learning algorithms, the system must analyze traffic patterns, predict traffic flow, and identify congestion-prone areas. This analysis will aid in dynamically optimizing signal timings.

3.3.4 Dynamic Signal Optimization

The system should adaptively adjust traffic signal timings in real-time based on the analyzed traffic patterns. It must optimize signal phases to reduce wait times and alleviate congestion while prioritizing safety.

3.3.5 Logging and Reporting

The system should maintain logs of traffic data, signal optimization decisions, and user interactions. It should generate reports summarizing system performance, traffic trends, and signal optimization effectiveness for periodic review and analysis.

3.3.6 Scalability and Extensibility

The system architecture should be scalable to accommodate additional sensors, cameras, or future enhancements. It should also allow for the integration of new data sources and technologies seamlessly.

3.3.7 Security and Privacy Measures

Ensuring data security and privacy, the system should implement robust measures to protect sensitive traffic data. Access controls, encryption, and anonymization techniques should safeguard data integrity and privacy.

Chapter 4

System Architecture

4.1 System Overview

The system architecture for traffic signal optimization integrates sophisticated reinforcement learning methodologies within the SUMO (Simulation of Urban MObility) simulation environment. This holistic approach is designed to effectively minimize average vehicle waiting time by dynamically adjusting traffic signal timings based on real-time traffic conditions and vehicle attributes. Different agents are there for each measuring constraints. After simulation testing, this integrated information is then utilized to calculate optimal green signal times. While our predictions are primarily analysis-based, we recognize the need for dynamic decision-making. To address this, we employ a real-time traffic agent, which continuously analyzes live traffic data to learn and adapt to evolving patterns.

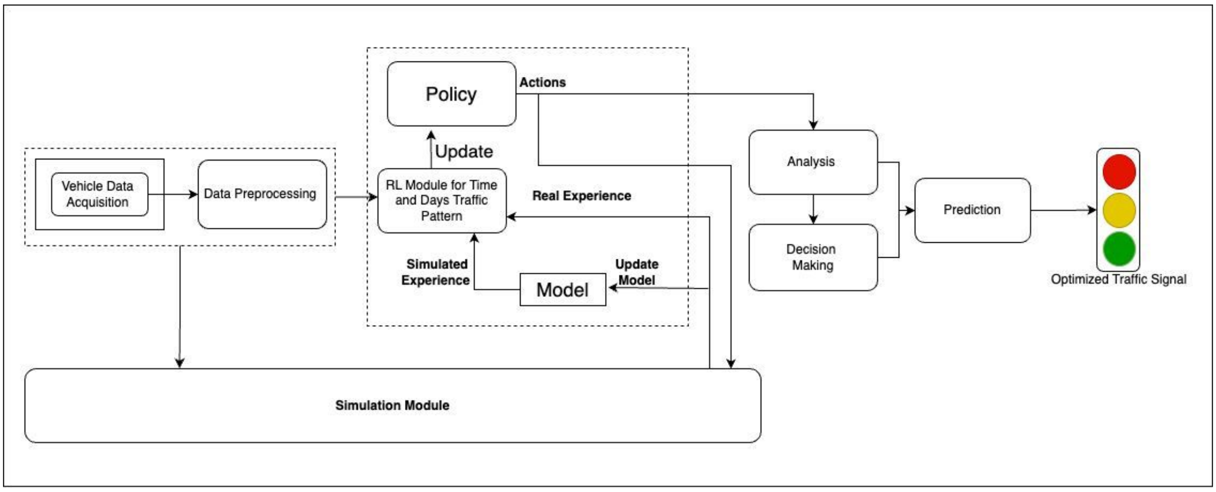


Figure 4.1: System Architecture of AI-assisted Smart Signal Traffic Management System

4.2 Module Division

4.2.1 Integrated Data Acquisition Module

The Integrated Data Acquisition Module is a critical component of the traffic management system, responsible for gathering vehicle count data from various sources to facilitate comprehensive analysis. This module operates as follows:

- **Data Sources**: The module accesses databases containing historical vehicle count data and weather information. Additionally, it interfaces with sensors installed in strategic locations to collect real-time data on vehicle count and weather conditions.

- **Vehicle Count Monitoring**: Utilizing cameras installed at key traffic points, the module monitors traffic flow to accurately count the number of vehicles in different scenarios. This involves real-time processing of video streams to detect and track vehicles, ensuring precise vehicle count data.

- **Data Preprocessing**: Before storing the collected data, the module preprocesses it to ensure consistency and accuracy. This involves filtering out noise, correcting anomalies, and standardizing the format of the data for seamless integration into the analysis pipeline as XML files.

- **Storage and Management**: Once the data is collected and preprocessed, the module stores it in a structured manner, ensuring easy accessibility and retrieval for analysis purposes. The data is organized in a data warehouse, allowing for efficient querying and analysis.

Overall, the Integrated Data Acquisition Module plays a pivotal role in gathering diverse data streams related to traffic flow. The module provides comprehensive analysis and informed decision-making in traffic management.

4.2.2 Reinforcement Learning

Reinforcement Learning (RL) is a machine learning paradigm focused on training agents to make sequential decisions in an environment to maximize cumulative rewards. Unlike supervised learning, where the model learns from labeled input-output pairs, and unsupervised learning, where the model learns from unlabeled data, RL agents learn through

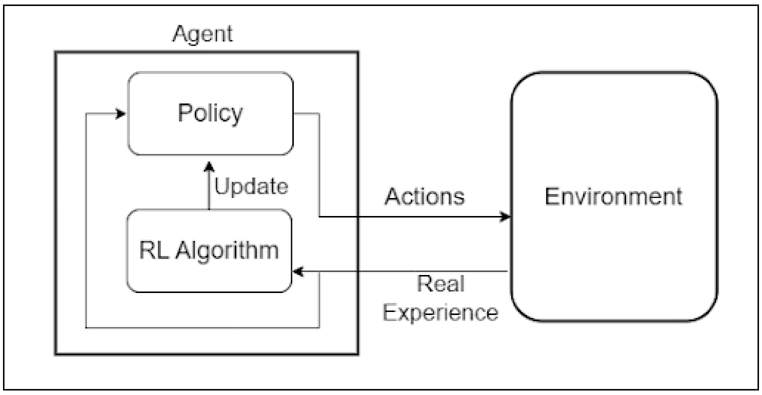


Figure 4.2: Reinforcement Learning Overview

trial and error by interacting with their environment. The core idea behind RL is to find the optimal policy, or strategy, for an agent to take actions in different states of the environment to achieve its goals over time. RL algorithms, such as Q-Learning and Deep Q-Learning, utilize concepts from psychology and control theory to enable agents to learn from feedback (rewards or penalties) received from their actions, ultimately learning to make better decisions over time.

States as Inputs

The data acquisition module collects and preprocesses various traffic-related data, including vehicle count, weather conditions, and real-time traffic data from sensors and cameras. These data are used to define the states of the environment. States may include factors such as the current traffic density, weather conditions, time of day, and historical traffic patterns.

Actions or Policies as Outputs

In Deep Q-Learning, the agent (traffic signal controller) selects actions based on the current state to optimize traffic flow. Actions represent decisions the agent can take, such as adjusting the duration of green signals for different lanes or intersections. The agent learns policies, which are mappings from states to actions, aiming to maximize a cumulative reward over time.

Deep Q-Learning Algorithm

Deep Q-Learning combines reinforcement learning with deep neural networks to approximate the Q-function, which represents the expected cumulative reward for taking an action in a given state. The Q-function is updated iteratively based on the Bellman equation, using a loss function to minimize the error between the predicted Q-values and the target Q-values. The agent interacts with the environment by selecting actions based on an exploration-exploitation strategy (e.g., -greedy), updating the Q-values, and learning the optimal policy over time.

1. **Neural Network (Q-Network):**
   - The Q-function $Q(s, a; \theta)$ is approximated using a deep neural network parameterized by weights θ.
   - The neural network takes the state s as input and outputs the Q-values for all possible actions a in that state.
2. **Experience Replay:**
   - Experience replay improves learning efficiency and stability.
   - It maintains a replay buffer of past experiences (state, action, reward, next state) and samples mini-batches during training.
3. **Target Network:**
   - A separate target network $Q(s, a; \theta^-)$ is used to compute target Q-values during updates.
   - The target network's weights θ^- are periodically updated with the main Q-network's weights θ.

Training Process

1. **Initialization:**
   - Initialize the Q-network with random weights.
2. **Interaction with Environment:**

- Observe the current state s from the environment.
- Use an -greedy policy to select an action a based on the Q-network's current Q-values.
- Execute the action a and observe the reward r and next state s'.

3. **Experience Replay and Q-Network Update:**
   - Store the experience (s, a, r, s') in the replay buffer.
   - Sample a mini-batch of experiences from the replay buffer.
   - Compute target Q-values y_i for each experience using the target network:

$$y_i = r_i + \gamma \max_{a'} Q(s'_i, a'; \theta^-)$$

   - Update the Q-network's weights θ by minimizing the loss:

$$L(\theta) = \frac{1}{N} \sum_i (y_i - Q(s_i, a_i; \theta))^2$$

   - Periodically update the target network's weights θ^- with the Q-network's weights θ.

4.2.3 Benefits of DQN for Traffic Signal Optimization

- DQN can handle complex state spaces and high-dimensional inputs, making it suitable for modeling intricate traffic conditions.
- By learning from past experiences and leveraging neural networks, DQN adapts and optimizes traffic signal timings dynamically based on real-time traffic data.
- The use of experience replay and target networks enhances stability and accelerates learning convergence, leading to more effective traffic signal control strategies.

Traffic Signal Optimization

The trained Deep Q-Learning agent learns to optimize traffic signal timings based on the collected data and current traffic conditions. By selecting actions that minimize congestion, reduce waiting times, and improve overall traffic flow, the agent dynamically adjusts signal timings to adapt to changing traffic patterns.

Integration with Data Acquisition Module

The data acquisition module continuously provides real-time traffic data to the Deep Q-Learning agent, enabling it to make informed decisions about traffic signal optimization. States are updated based on the latest data, and actions are selected accordingly to optimize traffic flow in response to current conditions.

Overall, Deep Q-Learning connects the data acquisition module to learn optimal traffic signal optimization policies by iteratively exploring and exploiting traffic patterns to maximize cumulative rewards. This approach enables adaptive and efficient traffic signal control in real-world environments.

Integration of RL Policy

The learned RL policy, derived from the RL module, provides insights into optimal decision-making strategies for traffic signal optimization. These policies, based on reinforcement learning algorithms such as Deep Q-Learning, guide the traffic signal controller in selecting actions to optimize traffic flow, such as adjusting signal timings based on current traffic conditions.

4.2.4 Simulation Module

The Simulation Module is a vital component of the traffic management system, responsible for simulating traffic scenarios in a given road network and similar conditions using SUMO. This module enables the evaluation of new traffic strategies and the analysis of their impact before implementing them in real-world situations. The Simulation Module operates in conjunction with the Data Acquisition Module, where results from the q-learning process serve as a base to the simulation, and the optimized timings for the current time slice of the particular road network are generated as output.

Simulation Process

Using SUMO, the Simulation Module simulates traffic flow, vehicle movements, and interactions within the specified road network based on the input data. SUMO's capabilities allow for the creation of detailed simulations, considering factors such as lane configurations, traffic light timings, vehicle speeds, and environmental conditions.

Evaluation of New Traffic Strategies

The Simulation Module facilitates the evaluation of new traffic strategies by simulating their implementation in the virtual environment. This enables traffic engineers and planners to assess the effectiveness of different strategies, such as adjusting signal timings, implementing lane management policies, or introducing traffic flow optimization measures, before deploying them in real-world scenarios.

Analysis of Results

After simulating various traffic scenarios, the Simulation Module analyzes the results to determine the impact of different strategies on traffic flow, congestion levels, travel times, and overall transportation efficiency. By comparing simulated outcomes under different conditions, traffic management stakeholders can make informed decisions about the most effective strategies for optimizing traffic flow and mitigating congestion.

Output: Optimized Timings

The output of the Simulation Module is the optimized timings for the current time slice of the particular road network. These optimized timings are generated based on the simulated traffic scenarios and are designed to maximize traffic flow, minimize congestion, and improve overall transportation efficiency. These optimized timings serve as valuable input for real-world traffic signal controllers to implement adaptive signal timing strategies based on current traffic conditions.

Overall, the Simulation Module facilitates the evaluation and analysis of traffic management strategies, providing valuable insights for optimizing traffic flow and enhancing transportation efficiency in urban environments under different scenarios. In traffic management systems, exceptional cases such as accidents, road protests, construction works resulting in immediate road blockages, and other unforeseen events require special handling.

4.3 Exceptional Cases Handling

Exceptional cases in traffic management refer to unforeseen events such as accidents, road protests, construction works leading to immediate road blockages, and other emergencies that disrupt normal traffic flow. These cases require special handling to ensure the safety of road users, minimize traffic congestion, and restore normal traffic operations as quickly as possible.

4.3.1 Adaptive Traffic Control

Traffic signal controllers and adaptive traffic management systems adjust signal timings and traffic flow strategies in real-time to accommodate the changing traffic conditions resulting from exceptional cases. Dynamic rerouting of traffic and temporary signal adjustments help alleviate congestion and facilitate the movement of vehicles around the affected area.

4.4 Work Schedule - Gantt Chart

The Figure 4.2 showcases the overall tentative work schedule of the project development in the form of gannt chart

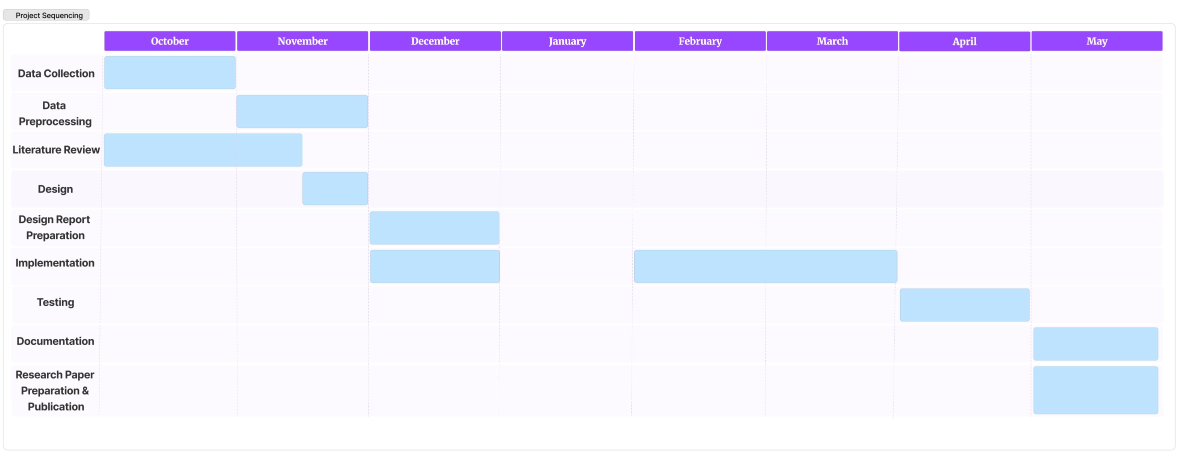


Figure 4.3: Gannt Chart

Chapter 5

System Implementation

In this chapter, process of implementing an AI-assisted traffic signal optimization system through the innovative technique of data fusion is explained . This chapter outlines an approach to revolutionize traffic signal management by utilizing the power of artificial intelligence through Reinforcment Learning techniques.

The primary objective of this implementation is to enhance traffic flow efficiency, minimize congestion, and optimize signal timings by combining data from diverse sources. By incorporating information from multiple sensors, cameras, and datasets, we aim to create a robust system capable of comprehensively analyzing traffic patterns, vehicle behaviours, and environmental conditions.

The focus is mainly on AI algorithms, advanced data processing techniques, and real-time data integration. Through the fusion of information gained from various sources such as vehicle detection, road monitoring cameras, and environmental sensors aim is to build an understanding of traffic dynamics.

This chapter will detail the system implementation and methodologies adopted to develop and deploy this AI-driven traffic signal optimization system. Additionally, it will highlight the intricacies of data fusion techniques employed, the integration of machine learning models, and the strategies devised to ensure real-time adaptability and responsiveness to changing traffic scenarios.

5.1 Simulation in SUMO

Simulate identified traffic patterns in SUMO (Simulation of Urban MObility). Utilize the estimated traffic parameters as inputs to replicate real-world scenarios within the simulation environment. Validate the simulation against observed traffic patterns for accuracy.

5.1.1 How SUMO Works

SUMO (Simulation of Urban MObility) is an open-source traffic simulation software designed to model and simulate urban traffic networks. It provides a platform for researchers and practitioners to analyze, develop, and test various traffic management strategies and intelligent transportation systems (ITS) solutions.

At its core, SUMO operates by simulating the movement of vehicles and other traffic entities within a network of roads and intersections.

1. **Network Definition**: SUMO uses XML-based files to define the road network, including the layout of roads, intersections, traffic signals, lanes, and connections between different elements. These network files specify the geometry, properties, and relationships among various components within the simulated environment.

2. **Vehicle Definition**: Vehicles are defined and controlled using XML files called "routes" and "additional files". The routes file specifies the origin, destination, and specific routes that vehicles will follow during the simulation. Additional files can specify vehicle characteristics such as type, speed, acceleration, deceleration, and behavior.

3. **Simulation Initialization**: To start a simulation, SUMO reads the network and vehicle definition files and initializes the simulation environment based on this information. It sets up the virtual urban environment with the specified road layout, traffic flow, and initial vehicle states.

4. **Simulation Execution**: During simulation execution, SUMO advances the simulation in discrete time steps. At each time step, it computes vehicle movements, interactions, and traffic dynamics based on various factors such as vehicle speeds, accelerations, lane changes, traffic rules, and control systems (e.g., traffic signals).

5. **Traffic Control**: SUMO supports different traffic control strategies, including fixed-time traffic signals, actuated signals, and adaptive signal control. Traffic lights and control logic can be defined and configured within the simulation to manage traffic flow at intersections.

6. **Data Collection and Analysis**: Throughout the simulation, SUMO collects detailed data on vehicle movements, traffic conditions, delays, emissions, and other performance metrics. This data can be analyzed using built-in tools or exported for further analysis using external software.

7. **Visualization**: SUMO provides visualization tools to display the simulated traffic network and vehicle movements in real-time or post-simulation. This allows users to visually inspect the behavior of vehicles, traffic flows, and control systems within the simulated environment.

Overall, SUMO serves as a versatile and extensible platform for studying and optimizing urban mobility and traffic systems. Its ability to simulate complex traffic scenarios, integrate different control strategies, and analyze performance metrics makes it a valuable tool for transportation research, planning, and development.

5.2 Traffic Signal Optimization using Deep Q-Learning (DQN) in SUMO

The following Python code snippet demonstrates an implementation of traffic signal optimization using Deep Q-Learning (DQN) within the SUMO traffic simulation environment. This approach leverages reinforcement learning techniques to dynamically adjust traffic signal timings based on real-time traffic conditions.

5.2.1 Code Overview

The Python code utilizes the 'traci' library to interact with SUMO (Simulation of Urban MObility) and implements a Deep Q-Network (DQN) agent for traffic signal control. Key components of the code include:

- **DQNAgent Class:**
  - Implements a Deep Q-Learning agent with methods for initializing the neural network model, selecting actions based on -greedy policy, and training the model using experience replay.
- **SumoEnvironment Class:**

- Manages the SUMO simulation environment and integrates the DQNAgent for traffic signal optimization.
- Defines methods for running the simulation, extracting state information, performing simulation steps, and tracking waiting times.
- Includes a method for plotting and visualizing the average waiting time per time step.

The main execution of the code initializes an instance of 'SumoEnvironment' and runs the traffic simulation using the DQN-based traffic signal control strategy. The simulation progresses in time steps, with the DQN agent making decisions to optimize traffic flow and reduce average waiting times at intersections.

5.2.2 System Implementation

The implemented Deep Q-Learning (DQN) algorithm operates within a reinforcement learning framework to optimize traffic signal timings. Here's an overview of how the code works:

1. **Initialization:**
   - The SUMO simulation environment is initialized, and a DQNAgent instance is created with specified state and action sizes.
   - The DQNAgent initializes a neural network model to approximate the Q-function.
2. **Simulation Loop:**
   - The simulation progresses in discrete time steps, where the DQNAgent observes the current state from the environment.
   - Based on the observed state, the DQNAgent selects an action using an -greedy policy to balance exploration and exploitation.
   - The selected action is applied to the SUMO simulation, and the resulting next state, reward, and termination status (done) are obtained.
3. **Agent Training:**

- The DQNAgent updates its Q-network based on the observed experience (state, action, reward, next state) using the Bellman equation:

$$Q(s,a) \leftarrow Q(s,a) + \alpha \left[r + \gamma \max_{a'} Q(s',a') - Q(s,a) \right]$$

- Experience replay is used to sample and train the model with batches of past experiences, enhancing learning efficiency and stability.

5.2.3 Policy Update and Exploration-Exploitation Tradeoff

The DQNAgent employs an -greedy policy to balance exploration and exploitation during action selection. The policy gradually shifts from exploration (random actions) to exploitation (actions based on learned Q-values) over time. The -greedy policy is defined by:

$$Action(s) = \{\ randomactionwithprobability\epsilon \arg\max_a Q(s,a) otherwise$$

where ϵ is the exploration rate, which decays over episodes to prioritize exploitation of learned knowledge.

Optimized Signal Timing

Extract data from SUMO simulations to analyze traffic behavior and congestion points. Employ AI-driven algorithms to optimize signal timings based on observed traffic patterns and simulation outcomes. Adjust signal phases dynamically to reduce congestion and improve traffic flow efficiency.The system implements Reinforcement Learning techniques to optimize traffic signal timings adaptively. RL algorithms, such as Deep Q-Networks (DQN) algorithms are utilized to learn optimal signal timing policies by interacting with the traffic environment. The system receives rewards or penalties based on traffic flow improvements, aiming to dynamically adjust signal timings to minimize congestion and maximize traffic flow efficiency.

5.3 Conclusion: Comprehensive Implementation Strategies

Throughout this project, a diverse array of implementation strategies were deployed, encompassing various technical domains and cutting-edge methodologies:

5.3.1 Multifaceted Implementation Approach

The project employed a multifaceted approach, integrating state-of-the-art techniques:

Reinforcement Learning for Traffic Control

Using Reinforcement Learning, specifically Deep Q-Learning, the project optimized traffic signal timings dynamically. The system learned optimal signal control policies, adapting to changing traffic conditions and maximizing traffic flow efficiency.

Evaluation and Validation Techniques

Multiple evaluation techniques were employed to assess the performance of the implemented model. This included accuracy evaluation and visualizations using libraries like ‘Matplotlib‘.

5.3.2 Robustness and Integration

The project showcased versatility, adaptability, and robustness across various technical domains. The methods of Reinforcement Learning for traffic signal control demonstrated the system’s ability to amalgamate cutting-edge methodologies for comprehensive traffic management.

5.3.3 Future Considerations

As technology evolves, the project remains open to incorporating new methodologies and enhancements. Future considerations may include advancements in real-time data processing, AI algorithms, and system scalability to further refine traffic management strategies. The fusion of diverse and advanced implementation strategies underscored its commitment to using innovative technical solutions for comprehensive traffic management and optimization.

Chapter 6

Results and Discussions

This chapter presents the culmination of the implemented strategies and methodologies, delving into the outcomes and their implications. The chapter commences by outlining the results obtained from the various implemented techniques, the Q learning of reinforcement learning.

The obtained results are meticulously analyzed, scrutinized, and interpreted in light of the project's objectives. The discussions delve into the effectiveness, limitations, and implications of each implemented strategy in achieving the predetermined goals of the project. Furthermore, comparisons between the anticipated outcomes and the actual results are thoroughly examined, providing insights into the efficacy and potential areas of improvement for future iterations.

6.1 Overview

In this study, we extended the application of traffic signal regulation using q-learning to encompass various types of vehicles, including cars, buses, trucks, and bikes. Each vehicle was assigned a weightage based on several parameters such as the acceleration and deceleration capabilities, maximum speed that can be attained by a type of vehicle, and length of the vehicle.By incorporating diverse vehicle types and considering their specific characteristics, our model aimed to provide a more comprehensive and realistic approach to traffic management.

The agent was trained using information gathered from SUMO simulations, allowing it to anticipate the optimal course of action given the current traffic situation. Through dynamic modification of signal timing, the model aimed to reduce vehicle waiting times and enhance overall traffic flow.

In scenarios with fixed-time signal phases, predetermined signal timings were employed to regulate traffic flow at intersections. During these fixed-time phases, the green signal remained active for a fixed duration of ten seconds, while the red signal persisted for five seconds. Despite the fixed nature of these signal phases, the average vehicle waiting time varied, as depicted in Fig.6.1

Each discrete simulation time interval, or time step, facilitated the updating and simulation of vehicle and traffic behavior, enabling the model to adapt to changing traffic conditions and optimize signal timing accordingly.

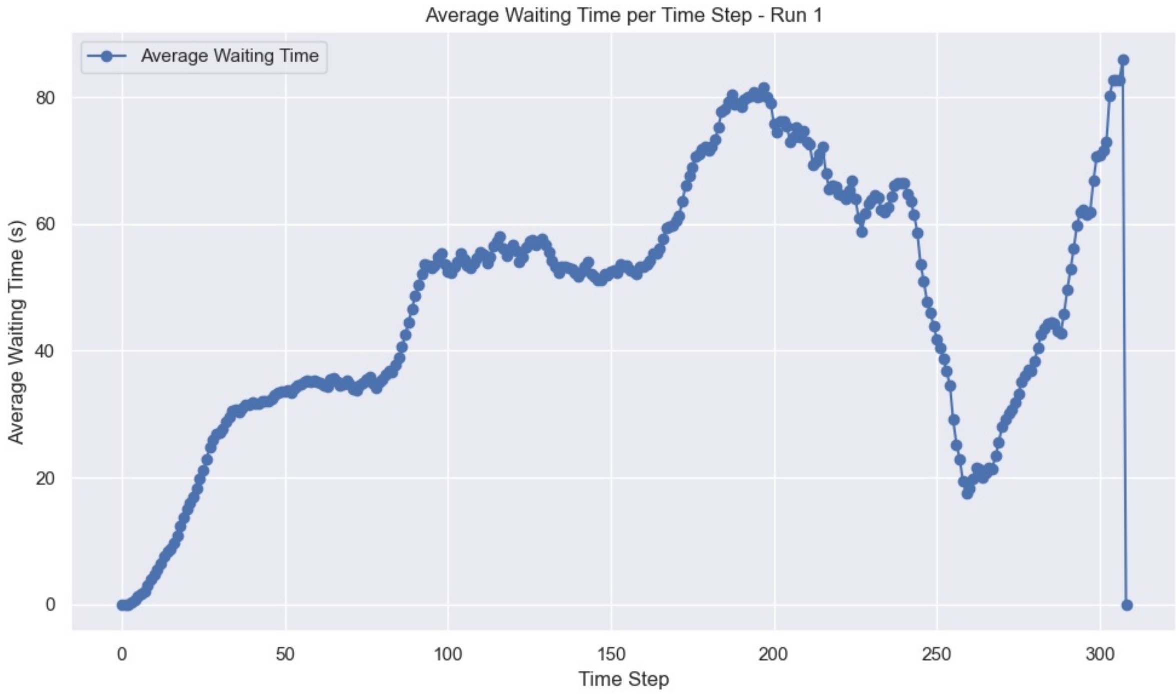


Figure 6.1: Average Waiting Time of Vehicles in a Fixed Time Traffic Signal

A predefined signal phase duration is created for fixed traffic phase control in order to handle signals. Red has a set length of five seconds, and green has a fixed duration of ten. The average vehicle wait time at the intersection is shown in Fig.6.1. Because the signal phases have a set duration, the average waiting time varies in this case.
Traffic data is gathered and trained in q-learning to forecast the least amount of waiting time and enhance traffic flow. The signal phase in this case has no set length since iteratively gathers data, trains it for the prediction, and repeats the process. The average waiting time for a vehicle that stays the same is displayed in Fig.6.2.

Figure 6.2: Average Waiting Time of Vehicles based on Q-learning

6.2 Testing

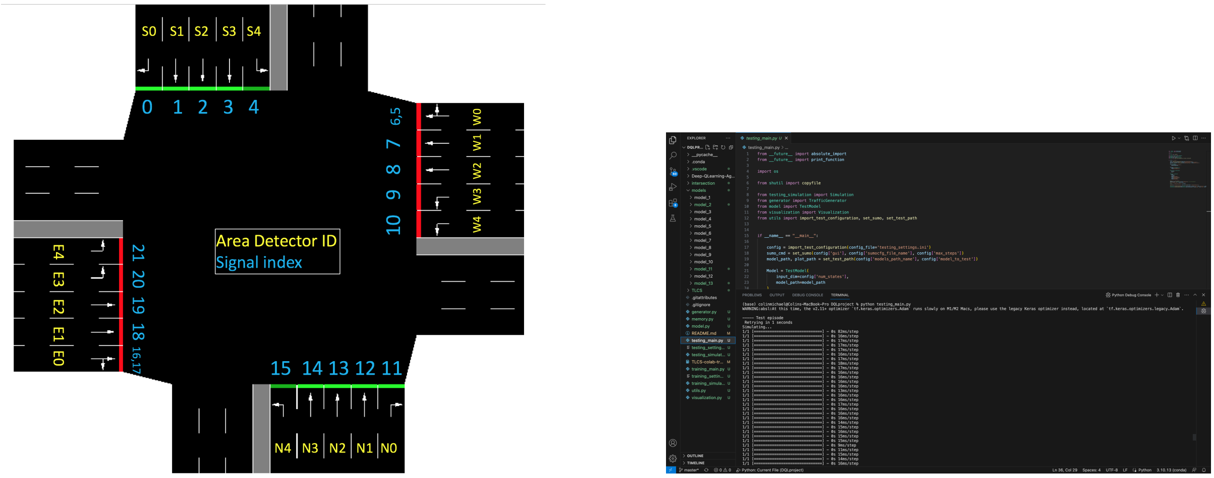
In this section, we present the testing methodology and results of our traffic signal optimization project utilizing Q-learning within the SUMO simulation framework. Our testing process involved progressively scaling our approach from a single intersection simulation to a larger 4x4 intersection network, culminating in the implementation of optimized traffic signal control on the Thrissur road network. Through these simulations, we aimed to assess the efficiency and scalability of our Q-learning-based approach in improving traffic flow and reducing congestion in urban environments.

Initially, we focused on validating our Q-learning algorithm's performance at a basic level, analyzing key metrics such as average vehicle delay and traffic throughput. Building upon these findings, we expanded our testing to more complex traffic scenarios, evaluating the algorithm's adaptability and effectiveness across interconnected intersections.

Finally, we applied our optimized traffic signal control system to a real-world road network, demonstrating its practical utility and potential impact on urban traffic management.

6.2.1 Simulation and Testing of Single Intersection

To evaluate our traffic signal optimization approach using Q-learning, we initially focused on a single intersection simulation. This allowed us to develop and test the Q-learning algorithm within a controlled environment before scaling up to more complex networks. Using Python and VS Code, we implemented the SUMO simulation platform to model the intersection and integrated our Q-learning-based traffic signal control system. We measured performance metrics such as average vehicle delay, queue lengths, and traffic throughput to assess the effectiveness of our approach.



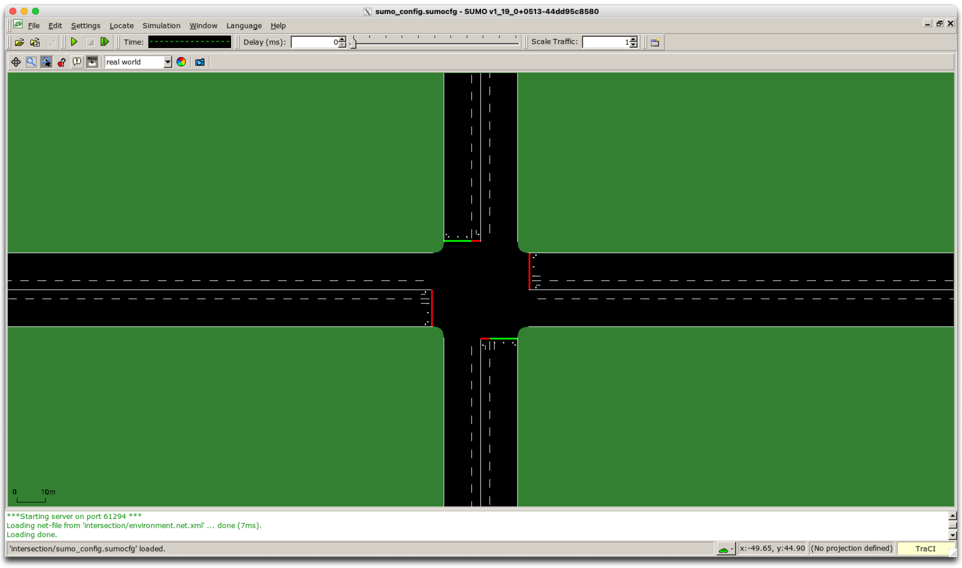
(a) Building of 4 lane intersection

(b) Simulation

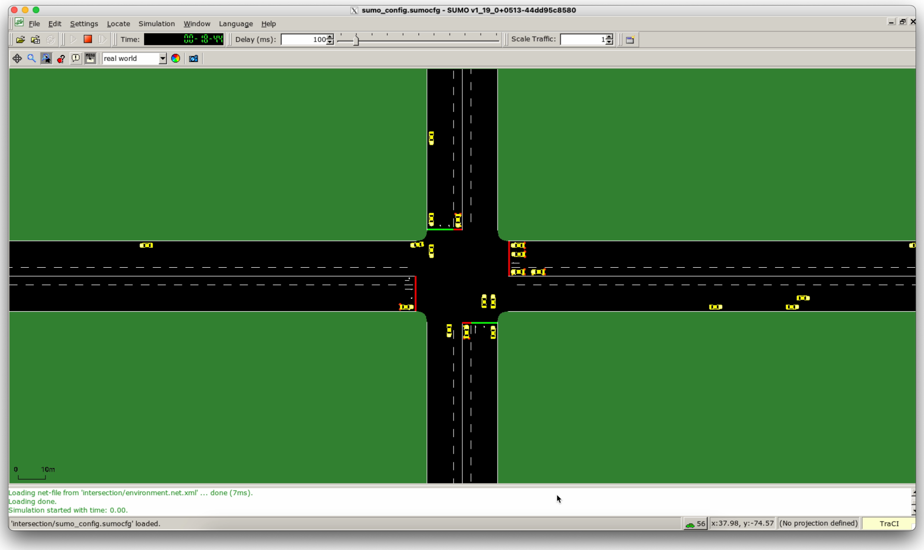
Figure 6.3: Traffic signal controlling of a single 4-way intersection

6.2.2 Building of road network

We constructed a comprehensive network using XML files that define the road layout, traffic flow, and vehicle characteristics. The network design was implemented through SUMO-specific XML codes, which allowed us to specify road segments, intersections, traffic lanes, and traffic control elements. Additionally, we configured vehicle types—including cars, buses, bikes, and trucks—using the ‘rou.xml‘ file, which defines the routes and attributes for each vehicle type. This setup enabled us to simulate diverse vehicle behaviors and interactions within the traffic network, crucial for assessing the performance and scalability of our Q-learning-based traffic signal optimization system. By customizing vehicle types and routes through SUMO’s XML files, we created a realistic and dynamic simulation environment representative of real-world traffic conditions, facilitating accurate evaluation and refinement of our traffic management algorithms.



(a) 4 way intersection and road network

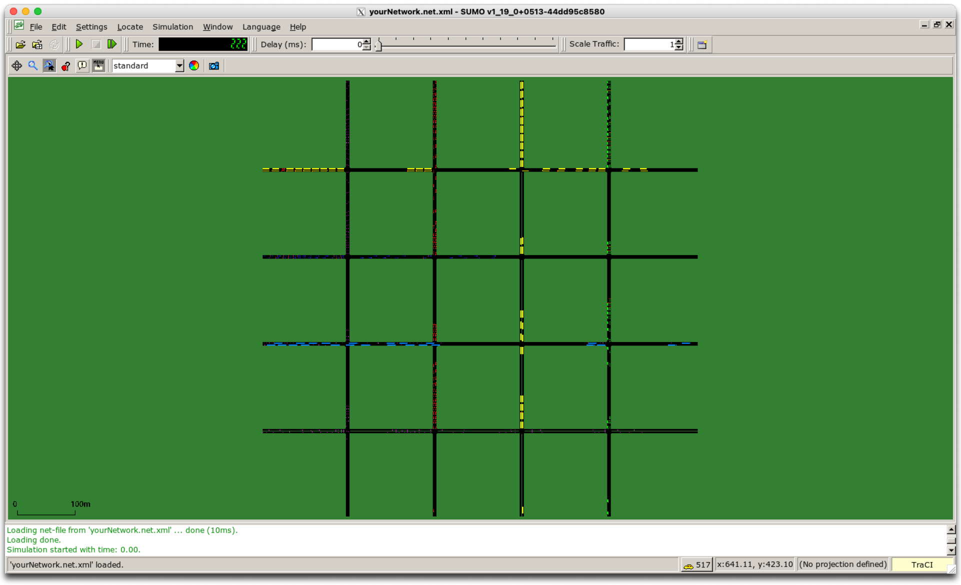


(b) Simulation of a single junction using only cars

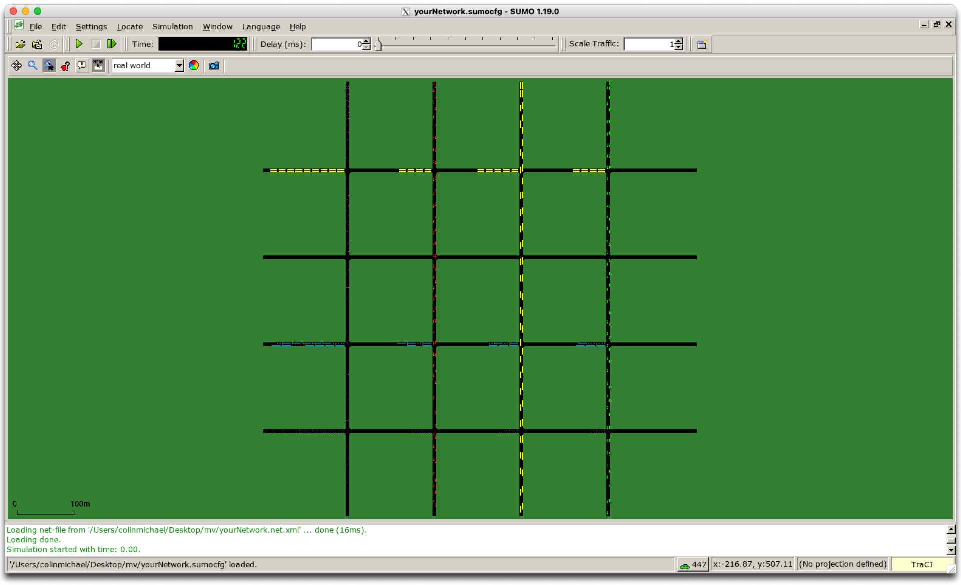
Figure 6.4: Traffic Signal Simulation in a Single Intersection

6.2.3 Scaling Up: Simulation of a 4x4 Intersection Network

Building upon the success of the single intersection simulation, we expanded our testing to a larger network comprising a 4x4 grid of intersections. This network represented a more complex traffic scenario with increased interactions between intersections. By extending our Q-learning model to control traffic signals across this network, we aimed to optimize traffic flow and minimize congestion. Through extensive simulations and data collection, we analyzed the impact of our Q-learning-based approach on overall traffic efficiency and intersection coordination.



(a) Fixed time traffic control



(b) Q-learning based traffic control

Figure 6.5: Comparison of simulations in a multiple vehicle environment using a 4*4 grid

6.2.4 Implementation on Thrissur Road Network with Optimized Traffic Signal Control

We implemented our optimized traffic signal control system on the Thrissur road network — a real world urban environment with complex traffic patterns and multiple intersections. Leveraging the insights gained from our previous simulations, we tailored our Q-learning algorithm to adapt dynamically to the unique traffic conditions of the Thrissur road network. This implementation involved integrating real-time traffic data and fine-tuning our model parameters to achieve optimal traffic signal timings.

In this extended testing phase, we conducted comprehensive evaluations of our system's performance using field data and simulated scenarios. We collected and analyzed key performance indicators, including travel times, congestion levels, and intersection efficiency, to validate the effectiveness of our Q-learning-based traffic signal optimization approach in a practical setting. By demonstrating improvements in traffic flow and reduction in delays compared to traditional traffic signal control methods, we highlighted the potential of our approach to enhance urban mobility and traffic management strategies.

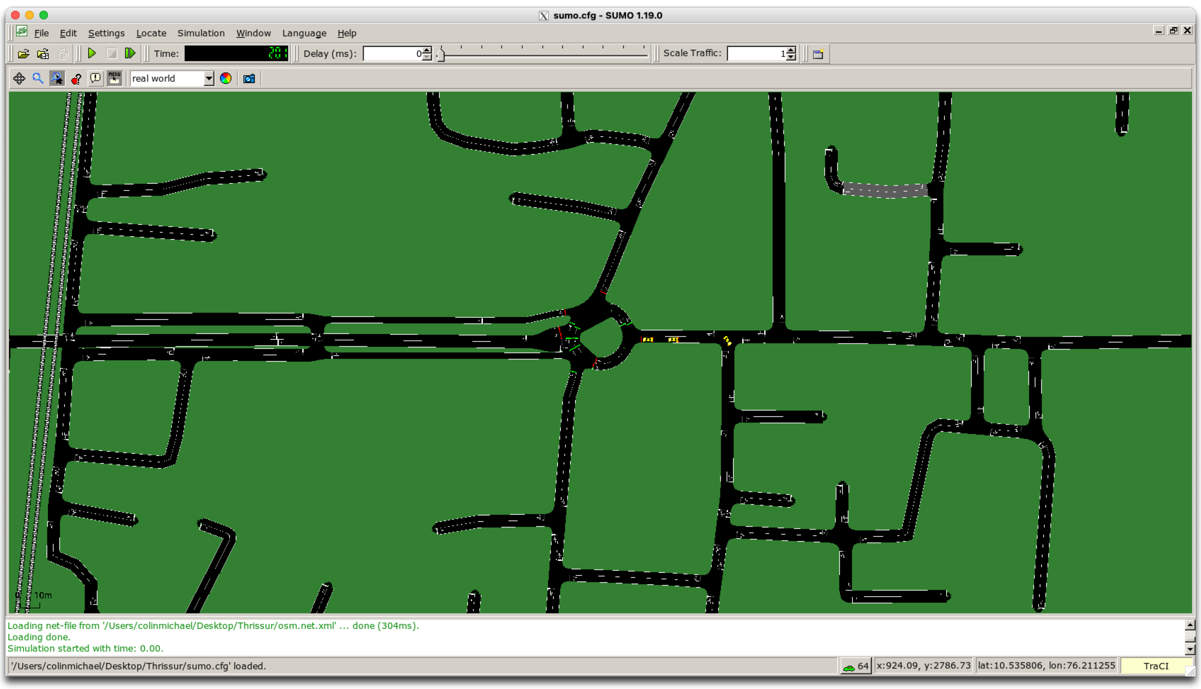


Figure 6.6: Roadnetwork of Thrissur Roundabout

6.3 Discussion

In addition to the simulation-based analysis conducted on signal regulation using q-learning, we extended our study to real-world scenarios by implementing and testing our model on the road network of Thrissur roundabout. This road network comprises various intersections and traffic signals, providing a diverse and challenging environment for traffic management systems as depicted in

Our model was applied to the Thrissur road network to evaluate its effectiveness in optimizing signal timings and reducing vehicle waiting times in a real-world setting. By integrating the q-learning approach with the traffic dynamics of Thrissur, the aim is to address the complexities of urban traffic management, including varying traffic volumes, heterogeneous vehicle types, and diverse road geometries. Graphical representations, akin to those presented in our simulated scenarios, illustrated the impact of our q-learning approach on traffic performance within the Thrissur road network as depicted in Fig These visualizations provided tangible evidence of the benefits afforded by our adaptive signal regulation strategy, further validating its utility in addressing real-world traffic congestion challenges.

The study demonstrates how complicated traffic control issues in urban settings can be addressed by combining RL approaches with dynamic modelling. RL agents can improve traffic signal timings and coordination by continuously learning from real-time traffic data, which results in more efficient traffic flow and reduction. The new method advances the field of intelligent transportation systems by offering a viable method for improving traffic control tactics via reinforcement learning and dynamic simulation. According to results, the proposed approach enhances traffic management, lowers congestion, and improves traffic flow. RL-based control strategies outperform conventional static or rule-based systems by dynamically adjusting to changing traffic circumstances, which results in more efficient use of road infrastructure and shorter commute times for commuters.

Chapter 7

Conclusions & Future Scope

In conclusion, the development of an AI-assisted traffic light simulation utilizing data fusion marks a significant step towards resolving the complexities of urban traffic congestion. By integrating artificial intelligence and data analytics, the system offers a dynamic and adaptive approach to traffic signal control, addressing the limitations of manual and conventional methods. Through the utilization of advanced algorithms such as reinforcement learning, the system accurately predicts traffic flow patterns, enabling optimal decision-making for signal timings.

This project presents a novel strategy to improve traffic control techniques using Reinforcement earning (RL) and dynamic simulation. By using RL's ability to learn adaptive control rules in dynamic traffic contexts, the rproject tackles the enduring problems associated with urban traffic congestion. Through the incorporation of multiple vehicle types such as cars, buses, trucks, and bikes, each assigned weightage based on parameters like space occupancy, acceleration, deceleration capacity, and maximum speed, the proposed approach offers a comprehensive solution to traffic management.

Moving forward, there are several avenues for enhancing the capabilities and effectiveness of the AI-assisted traffic light simulation system. Firstly, further refinement and optimization of prediction algorithms can improve the accuracy and reliability of traffic forecasts, leading to more precise signal timings. Additionally, the integration of real-time data from emerging technologies such as connected vehicles and smart infrastructure can provide additional insights for traffic management.

Moreover, the scalability of the system can be expanded to encompass a wider geographical area, extending its benefits to smaller cities and rural areas facing traffic congestion issues. Collaborations with transportation authorities and urban planners can facilitate the implementation of the system on a broader scale, promoting sustainable and efficient traffic management practices.

Furthermore, ongoing advancements in artificial intelligence and data analytics offer opportunities for continuous improvement and innovation in traffic control systems. Research into novel techniques such as machine learning and deep reinforcement learning can further enhance the capabilities of the system, enabling more sophisticated decision-making processes and adaptive control strategies.

Overall, the AI-assisted traffic light simulation system presents a promising solution to the challenges of urban traffic congestion, with ample opportunities for future development and expansion. Through ongoing research, collaboration, and innovation, the system has the potential to significantly improve the efficiency, safety, and sustainability of urban transportation networks.

References

[1] X. Yang, Y. Xu, L. Kuang, Z. Wang, H. Gao, and X. Wang, "An information fusion approach to intelligent traffic signal control using the joint methods of multiagent reinforcement learning and artificial intelligence of things," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 7, pp. 9335–9345, 2021.

[2] M. Wang, L. Wu, J. Li, and L. He, "Traffic signal control with reinforcement learning based on region-aware cooperative strategy," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 7, pp. 6774–6785, 2021.

[3] M. A. Hassan, M. Elhadef, and M. U. G. Khan, "Collaborative traffic signal automation using deep q-learning," *IEEE Access*, vol. 11, pp. 136 015–136 032, 2023.

[4] X. Liang, X. Du, G. Wang, and Z. Han, "A deep q learning network for traffic lights' cycle control in vehicular networks," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 2, pp. 1243–1253, 2019.

[5] Z. Li, H. Yu, G. Zhang, S. Dong, and C.-Z. Xu, "Network-wide traffic signal control optimization using a multi-agent deep reinforcement learning," *Transportation Research Part C: Emerging Technologies*, vol. 125, p. 103059, 2021.

[6] Qi, Liang, M. Zhou, and W. Luan, "Emergency traffic-light control system design for intersections subject to accidents," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 1, pp. 170–183, 2015.

[7] J. A. Calvo and I. Dusparic, "Heterogeneous multi-agent deep reinforcement learning for traffic lights control," in *AICS*, 2018, pp. 2–13.

[8] P. Fazzini, I. Wheeler, and F. Petracchini, "Traffic signal control with communicative deep reinforcement learning agents: a case study," *arXiv preprint arXiv:2107.01347*, 2021.

[9] L. Song and W. Fan, "Traffic signal control under mixed traffic with connected and automated vehicles: a transfer-based deep reinforcement learning approach," *IEEE Access*, vol. 9, pp. 145 228–145 237, 2021.

[10] Y. Wang, X. Yang, H. Liang, and Y. Liu, "A review of the self-adaptive traffic signal control system based on future traffic environment," *Journal of Advanced Transportation*, vol. 2018, 2018.

[11] H. Wei, G. Zheng, V. Gayah, and Z. Li, "A survey on traffic signal control methods," *arXiv preprint arXiv:1904.08117*, 2019.

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

Appendix A: Presentation

AI-assisted Smart Signal Traffic Management System

Athulram KR
Alona Mary Sebastian
GUIDE: DR.Preetha K.G
Colin Michael
Diya Anna Sunil

CONTENTS

INTRODUCTION

- As our cities grow and congestion of urban traffic is becoming one of the critical issues with increasing population and automobiles in cities.
- Traffic jams not only cause **extra delay** and **stress for drivers**, but also **increase fuel consumption**, and **increase carbon dioxide emission.**
- The traffic lights are one of the critical factors affecting traffic flow.

Conventional Traffic Systems

Features

- **Manual Controlling**
- **Automatic Controlling**
- **Electronic Sensors**

Drawback

- The manual controlling system requires a **large number of manpower.**
- Conventional traffic lights uses a timer for every phase, which is fixed and **does not adapt according to the real-time traffic on that road.**
- Electronic sensors i.e proximity sensors or loop detectors, the accuracy and coverage are often in conflict because the collection of high-quality information is usually based on sophisticated and expensive technologies and thus **limited budget will reduce the number of facilities.**

PROBLEM DEFINITION

Design a Smart traffic light simulation using integrated Artificial Intelligence and Reinforcement Learning Techniques

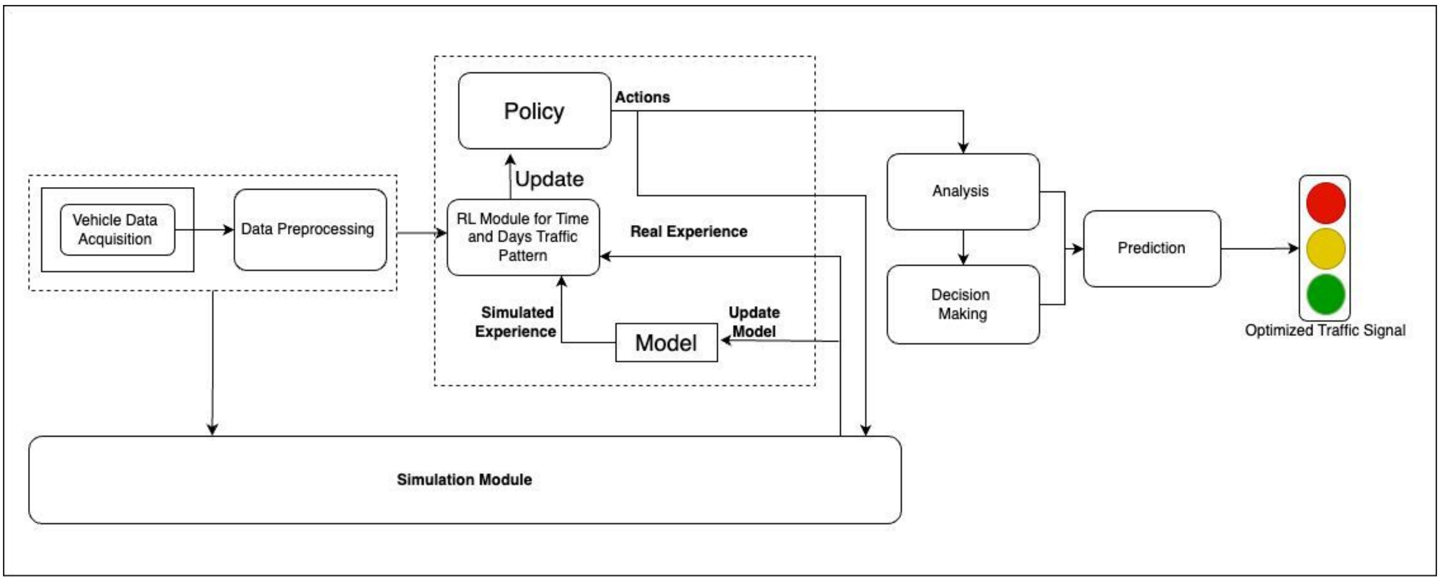
PURPOSE AND NEED

- Traffic jams not only cause extra delay and stress for drivers, but also increase fuel consumption, add transportation costs, and increase carbon dioxide air pollution.
- Traditional traffic control methods are becoming inadequate to handle the increasing and dynamic nature of traffic demands.
- Development of the economy and technology demands an improvement in the current traffic signal control (TSC) systems.
- To solve real-world traffic congestion challenges effectively by using traffic flow prediction as well as analyzing the real time traffic congestion.

PROJECT OBJECTIVE

- Simulate a smart method to resolve the traffic congestion in major metropolitan as well as big cities by predicting traffic to increase fuel efficiency and prevent driver frustration.
- Minimize the average waiting time the signal by predicting the traffic.

Architecture Diagram



Architecture Diagram- Q Learning

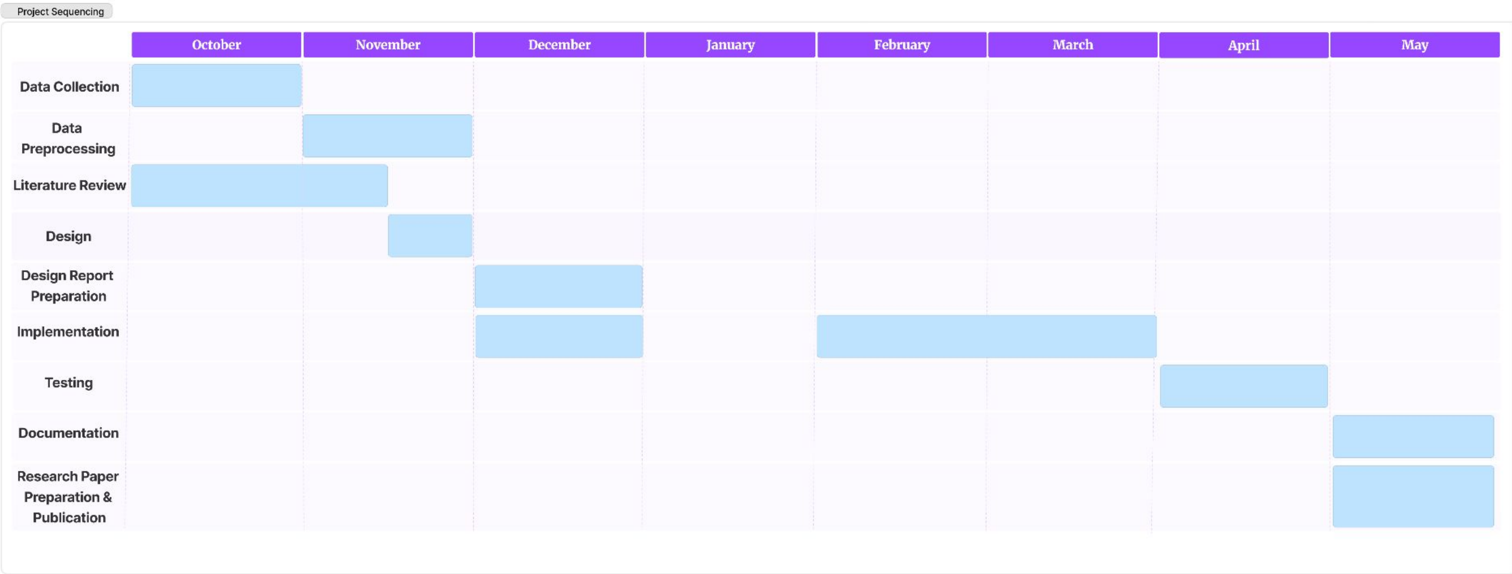
The proposed system worked based on **Q learning**, a type of **RL** Q learning consists of 3 major concepts

- **State(input) :** the state of the intersection is represented as a vector and the state space represents a segment of the lanes approaching the intersection.
- **Actions :** 2 possible actions constitutes the action space each corresponding to a different traffic light phase configuration and at each step the traffic agent selects an action based on the current state.

Architecture Diagram- Q Learning

- **Policy :** The policy used is the epsilon greedy which maximizes the **q value**.Each q value is updated based on the rewards.
- Reward Calculation is based after selecting an action.It calculates the reward based on the change in cumulative waiting time of vehicles at the intersection caused by an action.

Gantt Chart



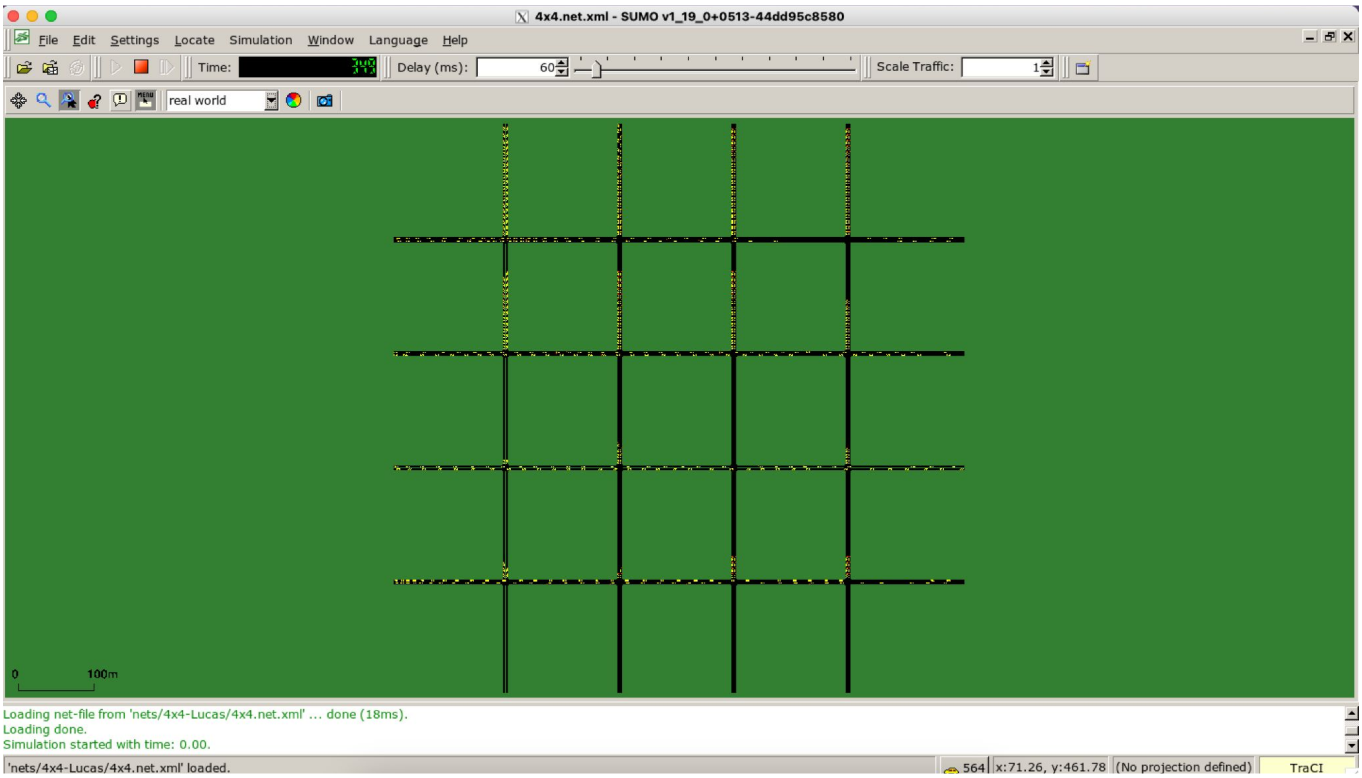
Work done during 30% Evaluation

- An environment consisting of a 4 way intersection with 4 incoming lanes and 4 outgoing lanes has been visualized using the SUMO simulation tool and TraCI
- The generated visualization only constituted a traffic system consisting only of cars,a random count was given as input in each epoch

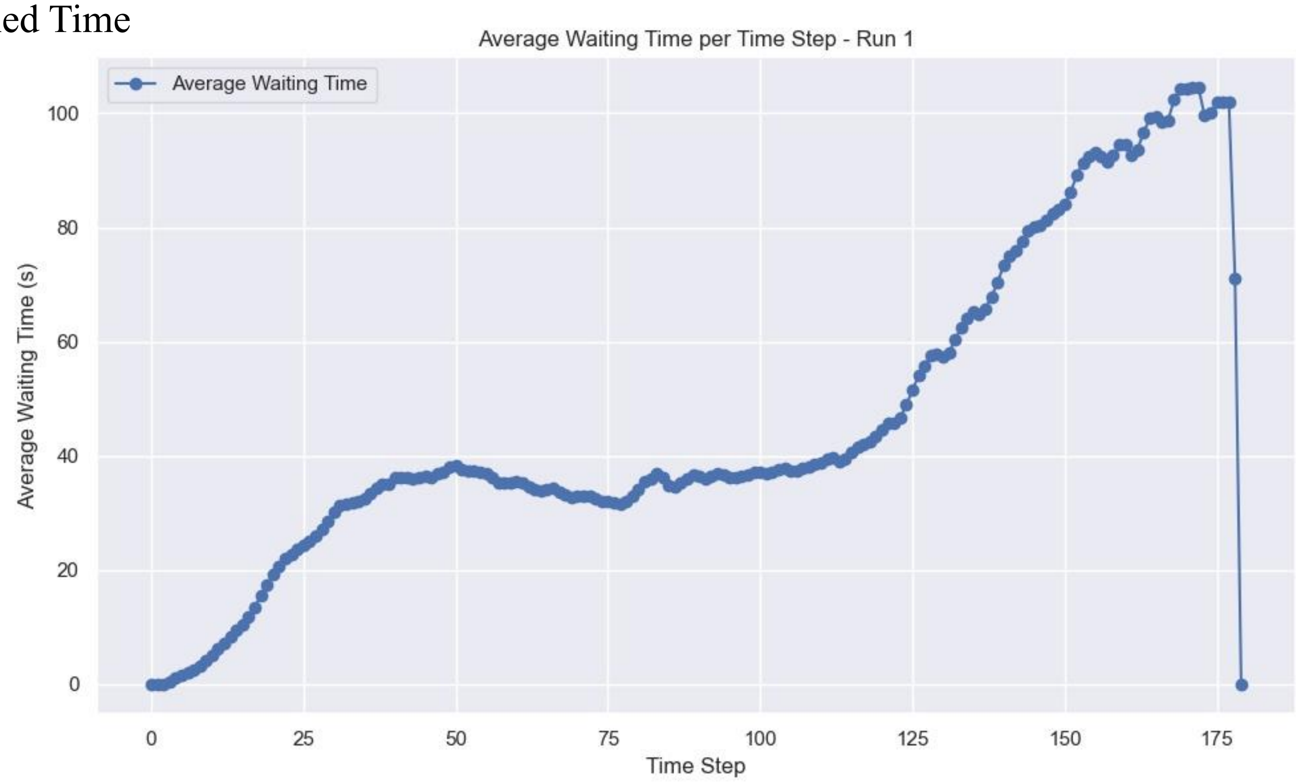
Work done during (60% evaluation)

- In 30% we have done traffic signal optimization of a single 4 way intersection road which only consist of cars using reinforcement learning.
- In 60% we are including Multiple intersections

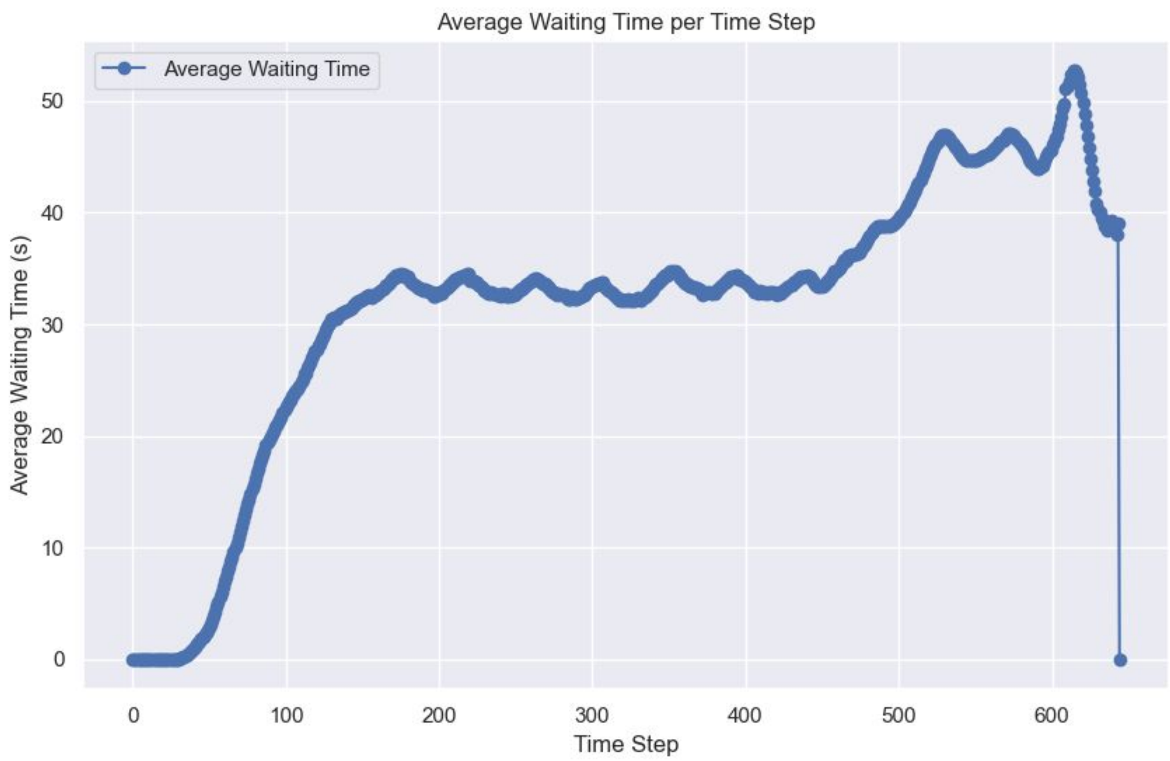
Interim Result



Fixed Time



QL Based



Work Progress(100% evaluation)

In 100% we are including different types of vehicles such as bike, bus and truck.

1. Defining weightage factors:
   a. Space occupied by each vehicle
   b. Acceleration and deceleration capabilities
   c. Length of vehicles
   d. Max speed

2. Calculating queue length of each lane:

$$Q \propto W_{avg}$$

$$W_{\text{AVERAGE}} = \frac{W_{\text{TOTAL}}}{\text{TOTAL VEHICLES}}$$

$$W_{total} = W_c*N_c+W_B*N_B+W_t*N_T+W_U+N_u$$

$$N_{total} = N_C+N_B+N_T+N_U$$

$$Q_i = W_{total} \text{ of } i^{th} \text{ lane}$$

3. Waiting time:

average_waiting_time = total_waiting_time / number_of_vehicles

4. MARL (Multi-agent reinforcement learning):

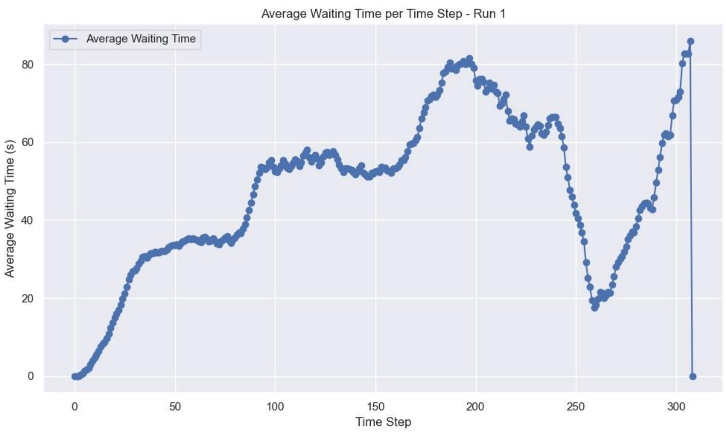
- MARL algorithm used here is Q learning.

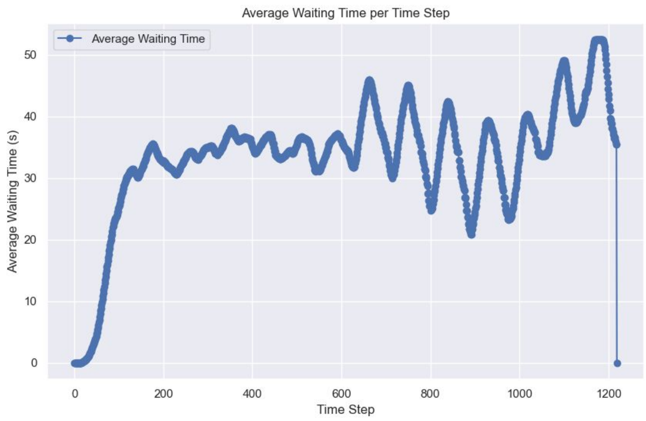
$$Q(s_t^i, a_t^i) \leftarrow Q(s_t^i, a_t^i) + \alpha \left[r_t^i + \gamma \max_a Q(s_{t+1}^i, a) - Q(s_t^i, a_t^i) \right]$$

- Determines whether the current time corresponds to a green or red phase and assigns the appropriate duration accordingly.

- **phase_duration =**
  green_duration if (current_time // (green_duration + red_duration)) % 2 == 0
  else
  red_duration

Results





Conclusion

- Traditional traffic signal control employes an incredible amount of man labor.
- A novel strategy to improve traffic control techniques using Reinforcement Learning and dynamic simulation using the SUMO tool is implemented.
- RL uses its ability to learn adaptive control rules in dynamically changing environments that may lead to traffic congestion
- With the integration of the sumo traffic environment with the RL module,the latter has the ability to continuously learn and coordinate from this environment.
- The proposed approach can lower congestion and improve traffic conditions by adjusting dynamically to the current scenario

Future Scope

- We can simulate this code in any map in real world with the help of SUMO and by this we can resolve the traffic congestion in major metropolitan as well as big cities

Status of Paper Publication

Enhancing Traffic Control Strategies through dynamic simulation and Reinforcement Learning- **Paper has been accepted and recommended for publication at the 3rd International Conference on Applied Artificial Intelligence and Computing – ICAAIC 2024**

References

1. Ren, Y., Jiang, H., Zhang, L., Liu, R., & Yu, H. (2022). HD-RMPC: A Hierarchical Distributed and Robust Model Predictive Control Framework for Urban Traffic Signal Timing. Journal of Advanced Transportation, 2022.
2. Li, Zhenning, ao Yu, Guohui Zhang, Shangjia Dong, and Cheng-Zhong Xu. "Network-wide traffic signal control optimization using a multi-agent deep reinforcement learning." Transportation Research Part C: Emerging Technologies 125 (2021).
3. Wang, Tong, Jiahua Cao, and Azhar ussain. "Adaptive Traffic Signal Control for large-scale scenario with Cooperative Group-based Multi-agent reinforcement learning." Transportation research part C: emerging technologies 125 (2021): 103046.
4. Cheng-Jian Lin, Shiou-Yun Jeng, Hong-Wei Lioa, "A Real-Time Vehicle Counting, Speed Estimation, and Classification System Based on Virtual Detection Zone and YOLO", Mathematical Problems in Engineering, vol. 2021, Article ID 1577614, 10 pages, 2021.
5. K. -F. Chu, A. Y. S. Lam and V. O. K. Li, "Traffic Signal Control Using End-to-End Off-Policy Deep Reinforcement Learning," in IEEE Transactions on Intelligent Transportation Systems, vol. 23, no. 7, pp. 7184-7195, July 2022, doi: 10.1109/TITS.2021.3067057.

Appendix B: Vision, Mission, Programme Outcomes and Course Outcomes

Vision, Mission, Programme Outcomes and Course Outcomes

Institute Vision

To evolve into a premier technological institution, moulding eminent professionals with creative minds, innovative ideas and sound practical skill, and to shape a future where technology works for the enrichment of mankind.

Institute Mission

To impart state-of-the-art knowledge to individuals in various technological disciplines and to inculcate in them a high degree of social consciousness and human values, thereby enabling them to face the challenges of life with courage and conviction.

Department Vision

To become a centre of excellence in Computer Science and Engineering, moulding professionals catering to the research and professional needs of national and international organizations.

Department Mission

To inspire and nurture students, with up-to-date knowledge in Computer Science and Engineering, ethics, team spirit, leadership abilities, innovation and creativity to come out with solutions meeting societal needs.

Programme Outcomes (PO)

Engineering Graduates will be able to:

1. Engineering Knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

2. Problem analysis: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

3. Design/development of solutions: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

4. Conduct investigations of complex problems: Use research-based knowledge including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

5. Modern Tool Usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.

6. The engineer and society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal, and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

7. Environment and sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

8. Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

9. Individual and Team work: Function effectively as an individual, and as a member or leader in teams, and in multidisciplinary settings.

10. Communication: Communicate effectively with the engineering community and with society at large. Be able to comprehend and write effective reports documentation. Make effective presentations, and give and receive clear instructions.

11. Project management and finance: Demonstrate knowledge and understanding of engineering and management principles and apply these to one's own work, as a member and leader in a team. Manage projects in multidisciplinary environments.

12. Life-long learning: Recognize the need for, and have the preparation and ability to engage in independent and lifelong learning in the broadest context of technological change.

Programme Specific Outcomes (PSO)

A graduate of the Computer Science and Engineering Program will demonstrate:

PSO1: Computer Science Specific Skills
The ability to identify, analyze and design solutions for complex engineering problems in multidisciplinary areas by understanding the core principles and concepts of computer science and thereby engage in national grand challenges.

PSO2: Programming and Software Development Skills
The ability to acquire programming efficiency by designing algorithms and applying standard practices in software project development to deliver quality software products meeting the demands of the industry.

PSO3: Professional Skills
The ability to apply the fundamentals of computer science in competitive research and to develop innovative products to meet the societal needs thereby evolving as an eminent researcher and entrepreneur.

Course Outcomes (CO)

Course Outcome 1: Model and solve real world problems by applying knowledge across domains (Cognitive knowledge level: Apply).

Course Outcome 2: Develop products, processes or technologies for sustainable and socially relevant applications (Cognitive knowledge level: Apply).

Course Outcome 3: Function effectively as an individual and as a leader in diverse teams and to comprehend and execute designated tasks (Cognitive knowledge level: Apply).

Course Outcome 4: Plan and execute tasks utilizing available resources within timelines, following ethical and professional norms (Cognitive knowledge level: Apply).

Course Outcome 5: Identify technology/research gaps and propose innovative/creative solutions (Cognitive knowledge level: Analyze).

Course Outcome 6: Organize and communicate technical and scientific findings effectively in written and oral forms (Cognitive knowledge level: Apply).

Appendix C: CO-PO-PSO Mapping

CO-PO AND CO-PSO MAPPING

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PSO 1	PSO 2	PSO 3
CO 1	2	2	2	1	2	2	2	1	1	1	1	2	3		
CO 2	2	2	2		1	3	3	1	1		1	1		2	
CO 3									3	2	2	1			3
CO 4					2			3	2	2	3	2			3
CO 5	2	3	3	1	2							1	3		
CO 6					2			2	2	3	1	1			3

3/2/1: high/medium/low

JUSTIFICATIONS FOR CO-PO MAPPING

MAPPING	LOW/MEDIUM/ HIGH	JUSTIFICATION
100003/ CS722U.1-PO1	M	Knowledge in the area of technology for project development using various tools results in better modeling.
100003/ CS722U.1-PO2	M	Knowledge acquired in the selected area of project development can be used to identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions.
100003/ CS722U.1-PO3	M	Can use the acquired knowledge in designing solutions to complex problems.
100003/ CS722U.1-PO4	M	Can use the acquired knowledge in designing solutions to complex problems.
100003/ CS722U.1-PO5	H	Students are able to interpret, improve and redefine technical aspects for design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
100003/ CS722U.1-PO6	M	Students are able to interpret, improve and redefine technical aspects by applying contextual knowledge to assess societal, health and consequential responsibilities relevant to professional engineering practices.
100003/ CS722U.1-PO7	M	Project development based on societal and environmental context solution identification is the need for sustainable development.
100003/ CS722U.1-PO8	L	Project development should be based on professional ethics and responsibilities.
100003/ CS722U.1-PO9	L	Project development using a systematic approach based on well defined principles will result in teamwork.
100003/ CS722U.1-PO10	M	Project brings technological changes in society.

100003/ CS722U.1- PO11	H	Acquiring knowledge for project development gathers skills in design, analysis, development and implementation of algorithms.
100003/ CS722U.1- PO12	H	Knowledge for project development contributes engineering skills in computing & information gatherings.
100003/ CS722U.2- PO1	H	Knowledge acquired for project development will also include systematic planning, developing, testing and implementation in computer science solutions in various domains.
100003/ CS722U.2- PO2	H	Project design and development using a systematic approach brings knowledge in mathematics and engineering fundamentals.
100003/ CS722U.2- PO3	H	Identifying, formulating and analyzing the project results in a systematic approach.
100003/ CS722U.2- PO5	H	Systematic approach is the tip for solving complex problems in various domains.
100003/ CS722U.2- PO6	H	Systematic approach in the technical and design aspects provide valid conclusions.
100003/ CS722U.2- PO7	H	Systematic approach in the technical and design aspects demonstrate the knowledge of sustainable development.
100003/ CS722U.2- PO8	M	Identification and justification of technical aspects of project development demonstrates the need for sustainable development.
100003/ CS722U.2- PO9	H	Apply professional ethics and responsibilities in engineering practice of development.
100003/ CS722U.2- PO11	H	Systematic approach also includes effective reporting and documentation which gives clear instructions.
100003/ CS722U.2- PO12	M	Project development using a systematic approach based on well defined principles will result in better teamwork.
100003/ CS722U.3- PO9	H	Project development as a team brings the ability to engage in independent and lifelong learning.
100003/ CS722U.3- PO10	H	Identification, formulation and justification in technical aspects will be based on acquiring skills in design and development of algorithms.
100003/ CS722U.3- PO11	H	Identification, formulation and justification in technical aspects provides the betterment of life in various domains.
100003/ CS722U.3- PO12	H	Students are able to interpret, improve and redefine technical aspects with mathematics, science and engineering fundamentals for the solutions of complex problems.

100003/ CS722U.4-PO5	H	Students are able to interpret, improve and redefine technical aspects with identification formulation and analysis of complex problems.
100003/ CS722U.4-PO8	H	Students are able to interpret, improve and redefine technical aspects to meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
100003/ CS722U.4-PO9	H	Students are able to interpret, improve and redefine technical aspects for design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
100003/ CS722U.4-PO10	H	Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools for better products.
100003/ CS722U.4-PO11	M	Students are able to interpret, improve and redefine technical aspects by applying contextual knowledge to assess societal, health and consequential responsibilities relevant to professional engineering practices.
100003/ CS722U.4-PO12	H	Students are able to interpret, improve and redefine technical aspects for demonstrating the knowledge of, and need for sustainable development.
100003/ CS722U.5-PO1	H	Students are able to interpret, improve and redefine technical aspects, apply ethical principles and commit to

		professional ethics and responsibilities and norms of the engineering practice.
100003/ CS722U.5-PO2	M	Students are able to interpret, improve and redefine technical aspects, communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
100003/ CS722U.5-PO3	H	Students are able to interpret, improve and redefine technical aspects to demonstrate knowledge and understanding of the engineering and management principle in multidisciplinary environments.
100003/ CS722U.5-PO4	H	Students are able to interpret, improve and redefine technical aspects, recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.
100003/ CS722U.5-PO5	M	Students are able to interpret, improve and redefine technical aspects in acquiring skills to design, analyze and develop algorithms and implement those using high-level programming languages.
100003/ CS722U.5-PO12	M	Students are able to interpret, improve and redefine technical aspects and contribute their engineering skills in computing and information engineering domains like network design and administration, database design and

		knowledge engineering.
100003/ CS722U.6-PO5	M	Students are able to interpret, improve and redefine technical aspects and develop strong skills in systematic planning, developing, testing, implementing and providing IT solutions for different domains which helps in the betterment of life.
100003/ CS722U.6-PO8	H	Students will be able to associate with a team as an effective team player for the development of technical projects by applying the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
100003/ CS722U.6-PO9	H	Students will be able to associate with a team as an effective team player to Identify, formulate, review research literature, and analyze complex engineering problems
100003/ CS722U.6-PO10	M	Students will be able to associate with a team as an effective team player for designing solutions to complex engineering problems and design system components.
100003/ CS722U.6-PO11	M	Students will be able to associate with a team as an effective team player, use research-based knowledge and research methods including design of experiments, analysis and interpretation of data.
100003/ CS722U.6-PO12	H	Students will be able to associate with a team as an effective team player, applying ethical principles and
		commit to professional ethics and responsibilities and norms of the engineering practice.
100003/ CS722U.1-PSO1	H	Students are able to develop Computer Science Specific Skills by modeling and solving problems.
100003/ CS722U.2-PSO2	M	Developing products, processes or technologies for sustainable and socially relevant applications can promote Programming and Software Development Skills.
100003/ CS722U.3-PSO3	H	Working in a team can result in the effective development of Professional Skills.
100003/ CS722U.4-PSO3	H	Planning and scheduling can result in the effective development of Professional Skills.
100003/ CS722U.5-PSO1	H	Students are able to develop Computer Science Specific Skills by creating innovative solutions to problems.
100003/ CS722U.6-PSO3	H	Organizing and communicating technical and scientific findings can help in the effective development of Professional Skills.