Faculty Of Engineering, Alexandria University

Graduation Project submitted in partial fulfilment of the B. Sc. Degree

July 2023

Intelligent Traffic Analysis (2022-2023)

A system that aims for less congestion, less pollution and better transportation

---

Authors:

Nouran Hisham

Mennatallah Moustafa

Rawan Hindawy

Seif Mohamed

Kareem Sabra

Supervisors:

Prof. Dr. Mohamed S. Abougabal

Prof. Dr. Shaimaa Lazem

Abstract

Over the years, we can all realize how much our population keeps exponentially increasing which leads to a problem that we're all aware of having which is major traffic congestion. Traffic congestion is a world-wide problem that affects our lives negatively every day. How many times has any of us been late to an important appointment just because the cars won't budge. Let alone all the gas exhaust that harm us and our environment greatly from unnecessarily waiting for a traffic light to open.

Due to all these reasons and even more, we felt that there is a need for a better way to control traffic, a way that makes better use of all the available resources and technologies we currently have, a way that can reflect positively on our times, health and environment, a way that grants everyone a better quality of life.

Our solution is an intelligent traffic system that controls traffic lights in a way dependent on the current state of the road instead of having static timed traffic lights that don't take into consideration rush hours, accidents, etc... It can decide which intersection is more critical and will have the greater effect on the road's state to facilitate cars' trips and help everyone reach their destinations efficiently and comfortably.

Our system uses different machine learning and reinforcement learning techniques, along with SUMO Logic simulation software and different datasets to encounter various traffic scenarios to be able to reach better results.

Acknowledgement

First, we would like to thank Allah for helping us to complete this project successfully.

We are deeply thankful to Prof. Dr. Mohamed Abougabal, and Prof. Dr. Shaimaa Lazem who not only served as our supervisors, but also guided, encouraged, and challenged us throughout the project, never accepting less than our best efforts, teaching us how to work under methodologies and in a more professional way.

We also owe our deepest gratitude to our families for their great support, without it our work would not have been as successful.

Table of Contents

List of Definitions and Acronyms

Term / Symbol	Meaning	Page
AI	Artificial Intelligence	10
AlST	Alexandria SUMO Traffic	41
ANN	Artificial Neural Network	54
CMD	Command Definition	50
CSV	Comma-Separated Values	32
DQN	Deep Q-Network	22
GUI	Graphical User Interface	35
IoT	Internet of Things	10
LuST	Luxembourg SUMO Traffic	57
MOM	Minutes Of Meeting	43
PCA	Principal Component Analysis	13
RL	Reinforcement Learning	18
RMSProp	Root Mean Squared Propagation	54
SMEs	Small Medium Enterprises	43
SPI	Software Process Improvement	34
SUMO	Simulation of Urban MObility	2
SVD	Singular Value Decomposition	13
SVM	Support Vector Machine	13
TraCI	Traffic Control Interface	37
XML	Extensible Markup language	32

List of Figures

List of Tables

Chapter 1

Introduction

In this section, we will talk generally about the problem and why we chose it.

1.1 General

Traffic congestion is a major issue in urban cities leading to aggregated traffic. With the advancement in intelligent internet of vehicles, new technologies and protocols have been developed to predict traffic congestion and utilize this traffic-related knowledge for congestion prediction and identification [1].

1.2 Motivation

In recent years, there has been a drop in technology prices such as smart cameras, along with the drastic development of the Internet of Things (IoT) and Artificial Intelligence (AI) technologies. These have opened the doors for the large-scale implementation of such systems, and Egypt should be at the forefront of such an exciting journey [2].

Egypt is known to be one of the most populated countries in the world and its population keeps increasing as time passes which results in more vehicle usage, resulting in more congestion. All these reasons make traffic congestion in Egypt a pressing problem that needs immediate attention to provide a better quality of life in many different ways; faster transportation, less pollution due to less fuel consumption, saving a lot of wasted commuting time.

Our main focus would be on Alexandria as it's known for its relatively small area and large population, and as our citizens of Alexandria, we all find it really hard to reach our destinations, especially during rush hours, which in turn makes us either too early or too late to any meeting/appointment we have, decreasing the quality of life for all of us.

1.3 Conclusion

Traffic is a major everyday problem that each one of us has suffered because of it at least once in our lives so that's what makes our intelligent traffic system a pressing need to be able to help ourselves and the environment from unnecessary waiting times, fuel consumption and car exhausts that harm us all.

Chapter 2
Background and Related Work

2.1 Introduction

A general introduction along with the project motivations is presented in the previous chapter. In the current chapter, the necessary background about Traffic Analysis as well as a survey of related work are introduced.

In this chapter, the idea on hand becomes clearer by introducing other related materials that resemble ours either in the technology phase or in the application phase.

In Section 2.2.1, an overview on Traffic Analysis is introduced. In Section 2.2.2, a background on Classical Machine Learning Classifiers. In Section 2.2.3, a background on Reinforcement Learning. An introduction to simulation and why we need it for our project in section 2.2.4.

Section 2.3 and section 2.4 discuss and compare a literature review that tackles the results of applying Traffic Analysis using different techniques and systems that are similar to what we aimed to implement.

In Section 2.5, we will discuss the need for our new system and its added value. Setting our scope of work and concluding this chapter takes place in sections 2.6 and 2.7, respectively.

2.2 Scientific Background

2.2.1 Overview

Our main aim is creating a system that gives a simple, accurate and early prediction of traffic congestion and can give high accurate results. This system is a fully automated system that can replace the conventional predetermined fixed-time based traffic system with a dynamically managed traffic system. This can be beneficial in not making congested roads more congested, which will be of great use to control traffic in many different cities and save many long needless waiting hours.

2.2.2 Classical Machine Learning Classifiers

2.2.2.1 Introduction to Machine Learning

Machine learning is a branch of AI and computer science which focuses on the use of data and algorithms to imitate the way that humans learn, gradually improving its accuracy.

Machine learning is an important component of the growing field of data science. Through the use of statistical methods, algorithms are trained to make classifications or predictions, and to uncover key insights in data mining projects. These insights subsequently drive decision making within applications and businesses, ideally impacting key growth metrics. As big data continues to expand and grow, the market demand for data scientists will increase. They will be required to help identify the most relevant business questions and the data to answer them [3].

2.2.2.2 Machine learning methods

2.2.2.2.1 Supervised Learning

Supervised Learning, also known as supervised machine learning, is defined by its use of labelled datasets to train algorithms to classify data or predict outcomes accurately. As input data is fed into the model, the model adjusts its weights until it has been fitted appropriately. This occurs as part of the cross-validation process to ensure that the model avoids overfitting or underfitting. Supervised learning helps organizations solve a variety of real-world problems at scale, such as classifying spam in a separate folder from your inbox. Some methods used in supervised learning include neural networks, naïve bayes, linear regression, logistic regression, random forest, and support vector machine (SVM) [3].

2.2.2.2.2 Unsupervised Learning

Unsupervised Learning, also known as unsupervised machine learning, uses machine learning algorithms to analyse and cluster unlabelled datasets. These algorithms discover hidden patterns or data groupings without the need for human intervention. This method's ability to discover similarities and differences in information make it ideal for exploratory data analysis, cross-selling strategies, customer segmentation, and image and pattern recognition. It's also used to reduce the number of features in a model through the process of dimensionality reduction. Principal component analysis (PCA) and singular value decomposition (SVD) are two common approaches for this. Other algorithms used in unsupervised learning include neural networks, k-means clustering, and probabilistic clustering methods [3].

2.2.2.2.3 Semi-supervised Learning

Semi-supervised learning offers a happy medium between supervised and unsupervised learning. During training, it uses a smaller labelled data set to guide classification and feature extraction from a larger, unlabelled data set. Semi-supervised learning can solve the problem of not having enough labelled data for a supervised learning algorithm. It also helps if it's too costly to label enough data [3] [4].

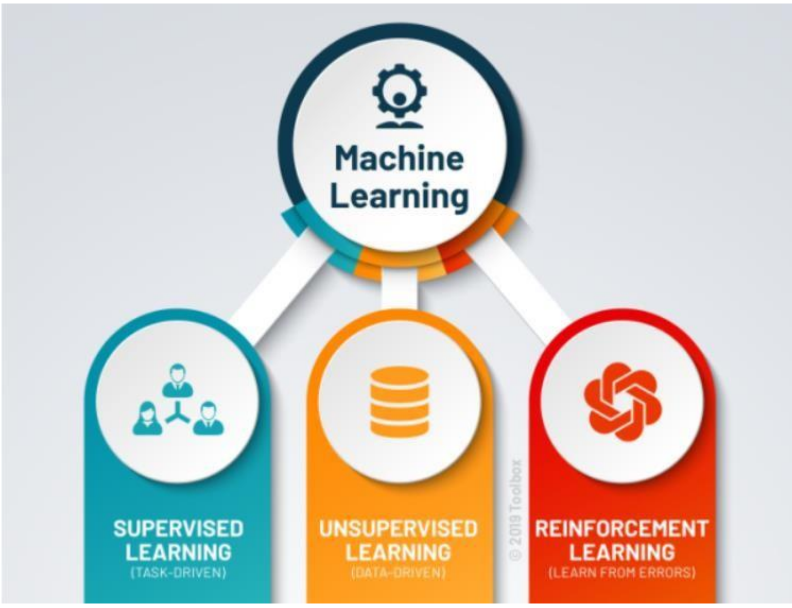


Figure 2.2.2.2

2.2.2.3 Feature Scaling

Feature scaling is a data pre-processing technique that involves transforming the values of features or variables in a dataset to a similar scale. This is done to ensure that all features contribute equally to the model and to prevent features with larger values from dominating the model. Feature scaling is essential when working with datasets where the features have different ranges, units of measurement, or orders of magnitude. Common feature scaling techniques include standardization, normalization, and min-max scaling. By applying feature scaling, the data can be transformed to a more consistent scale, making it easier to build accurate and effective machine learning models [5] [6].

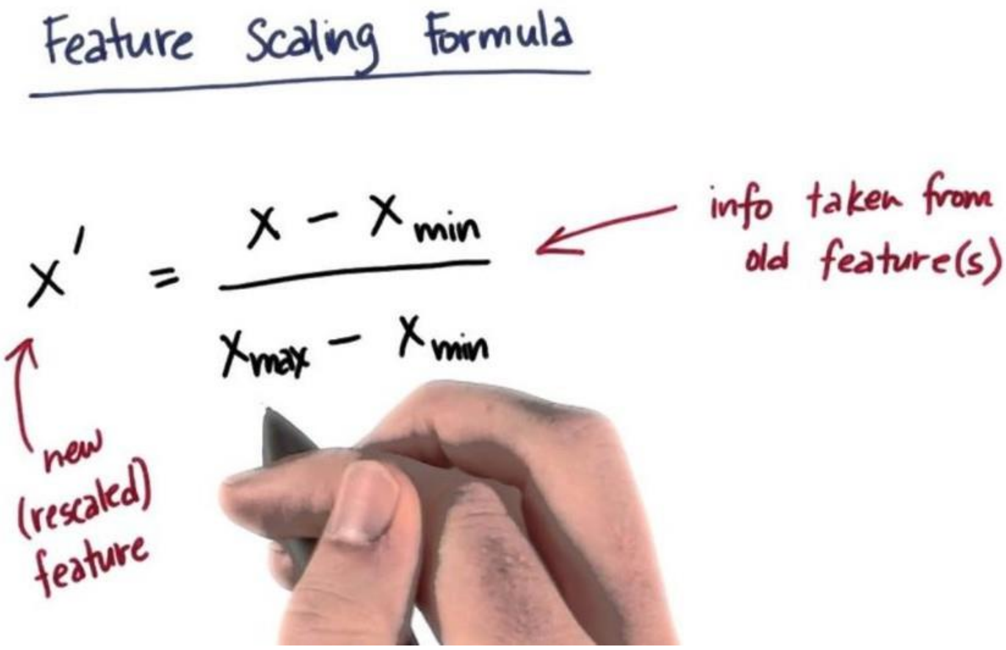


Figure 2.2.2.3

2.2.2.4 Principal Component Analysis (PCA)

The Principal Component Analysis is a popular unsupervised learning technique for reducing the dimensionality of data. It increases interpretability yet, at the same time, it minimizes information loss. It helps to find the most significant features in a dataset and makes the data easy for plotting in 2D and 3D. PCA helps in finding a sequence of linear combinations of variables.

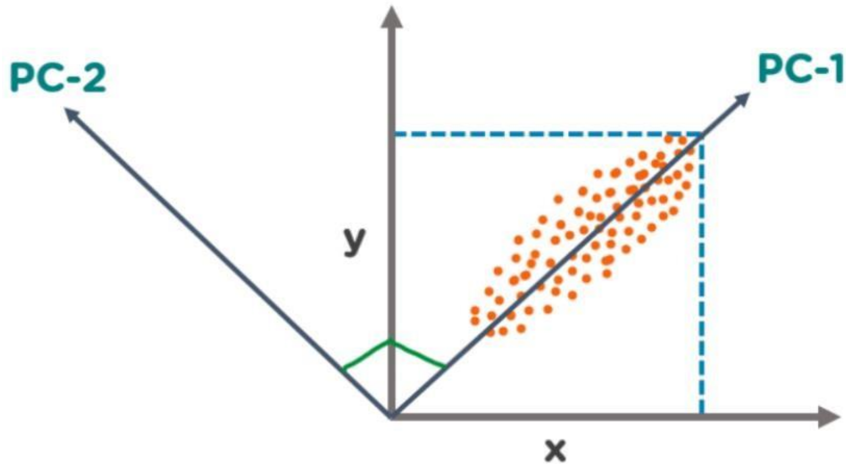


Figure 2.2.2.4

In the above figure, we have several points plotted on a 2-D plane. There are two principal components. PC1 is the primary principal component that explains the maximum variance in the data. PC2 is another principal component that is orthogonal to PC1.

The Principal Components are a straight line that captures most of the variance of the data. They have a direction and magnitude. Principal components are orthogonal projections (perpendicular) of data onto lower-dimensional space.

The term "dimensionality" describes the quantity of features or variables used in the research. It can be difficult to visualize and interpret the relationships between variables when dealing with high-dimensional data, such as datasets with numerous variables.

While reducing the number of variables in the dataset, dimensionality reduction methods like PCA are used to preserve the most crucial data.

The original variables are converted into a new set of variables called principal components, which are linear combinations of the original variables, by PCA in order to accomplish this. The dataset's reduced dimensionality depends on how many principal components are used in the study.

The objective of PCA is to select fewer principal components that account for the data's most important variation. PCA can help to streamline data analysis, enhance visualization, and make it simpler to spot trends and relationships between factors by reducing the dimensionality of the dataset [7].

2.2.2.5 Clustering

Clustering is the act of organizing similar objects into groups within a machine learning algorithm. Assigning related objects into clusters is beneficial for AI models. Clustering has many uses in data science, like image processing, knowledge discovery in data, unsupervised learning, and various other applications. Cluster analysis, or clustering, is done by scanning the unlabelled datasets in a machine learning model and setting measurements for specific data point features. The cluster analysis will then classify and place the data points in a group with matching features. Once data has been grouped together, it will be assigned a cluster ID number to help identify the cluster characteristics. Breaking down large, intricate datasets in a machine learning model using the clustering technique can alleviate stress when deciphering complex data [8].

2.2.3 Reinforcement Learning

2.2.3.1 Introduction to Reinforcement Learning

Reinforcement Learning (RL) is a type of machine learning technique that enables an agent to learn in an interactive environment by trial and error using feedback from its own actions and experiences.

Though both supervised and reinforcement learning use mapping between input and output, unlike supervised learning where the feedback provided to the agent is correct set of actions for performing a task, reinforcement learning uses rewards and punishments as signals for positive and negative behaviour.

As compared to unsupervised learning, reinforcement learning is different in terms of goals. While the goal in unsupervised learning is to find similarities and differences between data points, in the case of reinforcement learning the goal is to find a suitable action model that would maximize the total cumulative reward of the agent. The figure below illustrates the action-reward feedback loop of a generic RL model [9].

Some key terms that describe the basic elements of an RL problem are:

- **Environment** – Physical world in which the agent operates.
- **State** – Current situation of the agent
- **Reward** – Feedback from the environment
- **Policy** – Method to map agent’s state to actions.
- **Value** – Future reward that an agent would receive by taking an action in a particular state.

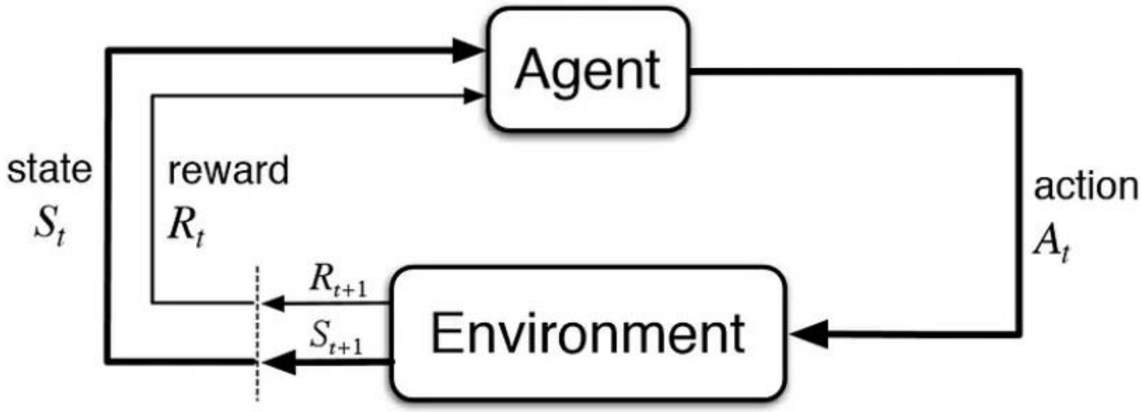


Figure 2.2.3.1

2.2.3.2 Q-Learning agent

Q-Learning is a Reinforcement learning policy that will find the next best action, given a current state. It chooses this action at random and aims to maximize the reward.

Q-learning is a model-free, off-policy reinforcement learning that will find the best course of action, given the current state of the agent. Depending on where the agent is in the environment, it will decide the next action to be taken.

The objective of the model is to find the best course of action given its current state. To do this, it may come up with rules of its own or it may operate outside the policy given to it to follow. This means that there is no actual need for a policy, hence we call it off-policy.

Model-free means that the agent uses predictions of the environment's expected response to move forward. It does not use the reward system to learn, but rather, trial and error [10].

Important Terms in Q-Learning

- **States:** The State, S, represents the current position of an agent in an environment.
- **Action:** The Action, A, is the step taken by the agent when it is in a particular state.
- **Rewards:** For every action, the agent will get a positive or negative reward.
- **Episodes:** When an agent ends up in a terminating state and can't take a new action.
- **Q-Values:** Used to determine how good an Action, A, taken at a particular state, S, is. Q (A, S).
- **Temporal Difference:** A formula used to find the Q-Value by using the value of current state and action and previous state and action.

Let's say we know the expected reward of each action at every step. This would essentially be like a cheat sheet for the agent! Our agent will know exactly which action to perform.

It will perform the sequence of actions that will eventually generate the maximum total reward. This total reward is also called the Q-value and we will formalize our strategy as:

$$Q(s, a) = r(s, a) + \gamma \max_{a} Q(s', a)$$

Equation 2.2.3.2.1

The above equation states that the Q-value yielded from being at state *s* and performing action *a* is the immediate reward r(s,a) plus the highest Q-value possible from the next state*s'*. Gamma here is the discount factor which controls the contribution of rewards further in the future.

Q(s',a) again depends on Q(s",a) which will then have a coefficient of gamma squared. So, the Q-value depends on Q-values of future states as shown here:

$$Q(s,a) \rightarrow \gamma Q(s',a) + \gamma^2 Q(s'',a) \ldots \ldots \ldots \gamma^n Q(s''^{\ldots n},a)$$

Equation 2.2.3.2.2

Adjusting the value of gamma will diminish or increase the contribution of future rewards.

Since this is a recursive equation, we can start with making arbitrary assumptions for all q-values. With experience, it will converge to the optimal policy. In practical situations, this is implemented as an update:

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha \left[R_{t+1} + \gamma \max_a Q(S_{t+1}, a) - Q(S_t, A_t) \right]$$

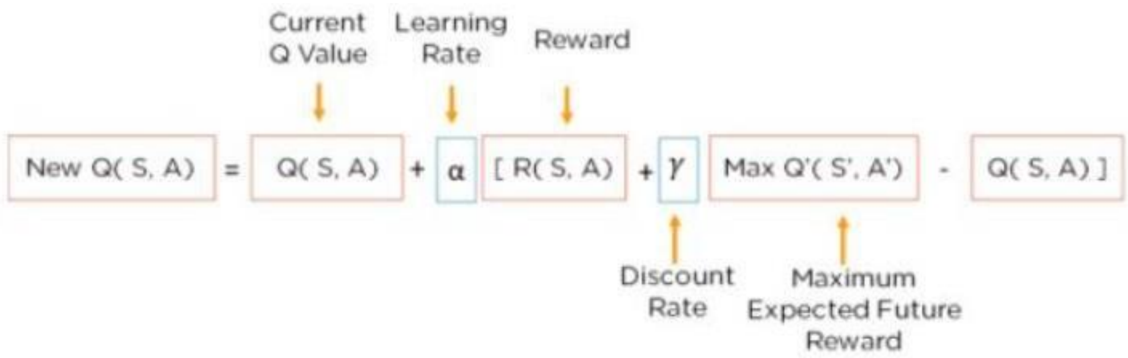
Equation 2.2.3.2.3

where alpha is the learning rate or step size. This simply determines to what extent newly acquired information overrides old information [11].

2.2.3.2.1 What Is the Bellman Equation?

The Bellman Equation is used to determine the value of a particular state and deduce how good it is to be in/take that state. The optimal state will give us the highest optimal value.

The equation is given below. It uses the current state, and the reward associated with that state, along with the maximum expected reward and a discount rate, which determines its importance to the current state, to find the next state of our agent. The learning rate determines how fast or slow, the model will be learning [10].



Equation 2.2.3.2.4

2.2.3.3 Deep Q-Network (DQN) agent

In deep Q-learning, we use a neural network to approximate the Q-value function. The state is given as the input and the Q-value of all possible actions is generated as the output. The comparison between Q-learning & deep Q-learning is wonderfully illustrated below:

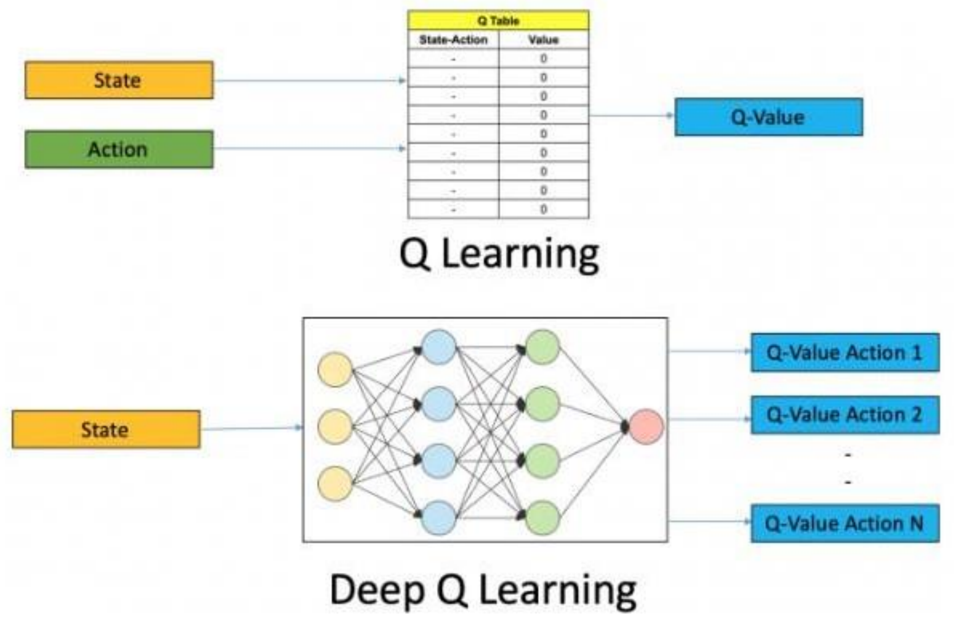


Figure 2.2.3.3

So, what are the steps involved in reinforcement learning using deep Q-learning networks (DQNs)?

1. All the past experience is stored by the user in memory.

2. The next action is determined by the maximum output of the Q-network.

3. The loss function here is mean squared error of the predicted Q-value and the target Q-value – Q*. This is basically a regression problem. However, we do not know the target or actual value here as we are dealing with a reinforcement learning problem. Going back to the Q-value update equation derived from the Bellman equation. we have:

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha\left[\boxed{R_{t+1} + \gamma \max_a Q(S_{t+1}, a)} - Q(S_t, A_t)\right]$$

Equation 2.2.3.3.1

The section in green represents the target. We can argue that it is predicting its own value, but since R is the unbiased true reward, the network is going to update its gradient using backpropagation to finally converge [10].

2.2.3.4 Reward Functions

In reinforcement learning, rewards are a key part of the learning process. Rewards signal to an agent what controls it has taken that are valuable, indicating which ones should be repeated when the same state is visited in the future.

Rewards play a critical role in reinforcement learning by providing a motivation for agents to learn and explore. Without rewards, agents would have no way of knowing which controls are important and would be unlikely to make any progress in completing its assigned task or solve a certain control problem. Rewards provide feedback to the agent, allowing it to learn from its controls and adapt its behavior over time.

In reinforcement learning, the agent is rewarded for taking controls that lead to successful states. The rewards can be immediate, such as receiving a point for each step taken in the right direction, or they can be delayed, such as receiving a point at the end of the episode if the goal was reached. There are many different ways to define rewards, and the choice of reward function can have a substantial impact on the learning process. For example, rewards that are too small may not provide enough motivation for the agent to learn, while rewards that are too large may result in the agent taking too many risks. Finding the right balance of rewards is an important part of designing a successful reinforcement learning algorithm.

A rewards function is used to define what constitutes a successful or unsuccessful outcome for an agent. Different rewards functions can be used depending on the goal of the agent. For example, if an agent is trying to maximize its score in a game, then a rewards function could give positive rewards for winning and negative rewards for losing. On the other hand, if an agent is trying to reach a target location as quickly as possible, the rewards function could be the negative of the number of steps taken so far to encourage the agent to reach the goal in minimum time. Similarly, the reward function could consider distance from the goal to where the agent currently is too as an inverse function [12].

2.2.4 Simulation

2.2.4.1 Introduction to Simulation

A simulation is a model that mimics the operation of an existing or proposed system, providing evidence for decision-making by being able to test different scenarios or process changes. This can be coupled with virtual reality technologies for a more immersive experience.

Simulations can be used to tune up performance, optimize a process, improve safety, testing theories, training staff and even for entertainment in video games! Scientifically modelling systems allows a user to gain an insight into the effects of different conditions and courses of action.

Simulation can also be used when the real system is inaccessible or too dangerous to assess or when a system is still in the design or theory stages.

Key to any simulation is the information that is used to build the simulation model and protocols for the verification and validation of models are still being researched and refined, particularly with regard to computer simulation.

Simulation works through the use of intuitive simulation software to create a visual mock-up of a process. This visual simulation should include details of timings, rules, resources, and constraints, to accurately reflect the real-world process [13].

2.2.4.2 Why do we need simulation?

Traffic Analysis is not an easy problem which makes it really hard to be able to see any results of any experiments we make in real life, so we felt the need to use simulation.

Simulation helped us in modelling how the real-life traffic in certain countries or cities normally take place, so we were able to analyze its shortcomings and work on improving them.

Whenever we make an improvement, we apply it to our simulation scenario to see its effect which definitely is infeasible to be done in the real world. This way whatever model we come up with to improve traffic, like we intend to do, will be well tested on a simulation model mimicking real traffic.

2.3 Literature Review

We studied the related surveyed papers (14-21) which are summarized, and the detailed summaries can be found in appendix A.

A comparison between the different papers based on the main features is presented in table 2.3 literature review of traffic analysis.

Comparison points		[14]	[15]	[16]	[17]	[18]	[19]	[20]	[21]	**Ours**
1.Year		**2018**	**2019**	**2020**	**2020**	**2021**	**2022**	**2022**	**2022**	**2023**
2.Objectives	**2.1. Predict traffic congestion**	✓	✓	✓	✓	✓	✓	✓	✓	✓
	2.2. Automated traffic lights	✓	✓	✓	✓	✓	✓	✓	✓	✓
	2.3. Handling accidents	✓	✗	✗	✗	✗	✗	✓	✗	✗
	2.4. Congestion detection	✓	✓	✓	✓	✓	✓	✓	✓	✓

	2.5. Provide emergency help	✓	+	+	+	+	+	+	+	+
	2.6. Single intersections	✓	✓	✓	✓	✓	✓	✓	✓	✓
	2.7. Multiple intersections	■	+	■	+	+	✓	+	+	✓
	2.8. Notify people with certain news	✓	+	+	+	+	+	+	■	+
3. Simulation		+	+	✓	+	+	+	+	✓	✓
4. Implemented in form of application		+	+	+	+	+	+	+	+	+
5. Open source		+	+	+	+	+	+	+	+	✓

6. Classifiers (classical ML)	**6.1. SVMs**	+	+	+	motor cycles	motor cycles	+	motor cycles	+	+
	6.2. KNNs	+	+	+	+	congestion	+	congestion	+	+
	6.3. Linear regression	+	+	+	car count	+	car count	+	car count	+
	6.4. Random forest	+	+	+	congestion	+	congestion	+	+	+
	6.5. PCA	+	✓	+	+	✓	✓	+	+	✓
	6.6. Sequential Minimal Optimized Regression	+	+	+	+	+	congestion	+	+	+
	6.7. Data preprocessing	✓	+	+	motor cycles	+	✓	+	+	✓
	6.8. Decision tree	+	+	+	congestion	+	+	+	+	+
	6.9. Data cleaning	+	✓	+	✓	+	+	+	+	✓
7. Classifiers (deep learning)	**7.1. MLP**	+	car count	+	+	+	+	+	+	+
	7.2. CNN	+	+	cars	+	cars	+	cars	+	+
	7.3. RNN	+	+	+	+	+	cars	+	+	+
	7.4. LSTM	+	+	+	+	+	cars	+	+	+

	7.5. GRU	+	+	+	+	+	cars	+	+	+
	7.6. MLFNN	+	+	+	+	car count	+	+	+	+
	7.7. RBFNN	+	+	+	+	conge stion	+	+	+	+
	7.8. WNN	+	+	+	+	conge stion	+	+	+	+
	7.9. FNN	+	+	+	+	car count	+	+	+	+
	7.10. YOLO	+	+	+	+	+	+	cars	+	+
	7.11. RL	+	+	✓	+	+	+	+	+	✓
8.Datasets availability		+	+	+	✓	+	✓	+	✓	✓
9. Accuracies (classical ML)	**9.1. Gradient boosting**	+	+	+	+	+	**0.9306**	+	+	+
	9.2. Random forest	+	+	+	**0.91**	+	**0.9297**	+	+	+
	9.3. Linear regression	+	+	+	+	+	**0.9264**	+	+	+
	9.4. Stochastic regression	+	+	+	+	+	**0.9004**	+	+	+
	9.5. Decision tree	+	+	+	**0.88**	+	+	+	+	+
	9.6. SVMs	+	+	+	**0.88**	+	+	+	+	+
10. Accuracies (deep learning)	**10.1. MSR2C-AB PPN**	+	**0.97**	+	+	+	+	+	+	+
	10.2. Time series model	+	**0.95**	+	+	+	+	+	+	+
	10.3. MLP-NN	+	+	+	+	+	**0.9307**	+	+	+
	10.4. GRU	+	+	+	+	+	**0.9295**	+	+	+
	10.5. LSTM	+	+	+	+	+	**0.987**	+	+	+
11. Applied in Alexandria		+	+	+	+	+	+	+	+	✓

Table 2.3 Literature review of traffic analysis

2.4 Review of similar systems

We studied the related implemented systems (22-25) which are summarized, and the detailed summaries can be found in appendix B.

A comparison between the different systems based on the main features is presented in table 2.4 Comparison between similar traffic analysis implemented systems.

Comparison points		Intellias [22]	Telegra [23]	Chetu [24]	Q-free [25]	Ours
1.Founded in		2002	N/A	2000	1984	2023
2.Services offered	2.1. Traffic data fusion	✓	■	✓	✓	✓
	2.2. Traffic data visualization	✓	✓	✓	✓	✓
	2.3. Traffic data delivery	✓	■	✓	✓	✓
	2.4. Traffic centers and complete solutions	✓	■	■	■	✓
3.Services' outcomes	3.1. Guidance, navigation, and control	■	■	■	■	✓
	3.2. Traffic modeling and simulation	✓	✓	+	■	✓
	3.3. Video analytics	✓	✓	✓	✓	+
	3.4. Road safety	✓	✓	✓	✓	✓
4.Google Rating		4.8/5	3.5/5	5/5	3.1/5	N/A
5.Free		+	+	+	+	✓
6.Target Audience	6.1. Business Related	✓	✓	✓	■	✓
	6.2. Single/multiple users	+	+	✓	✓	✓
7.Mobile app		✓	✓	+	■	+

8. Technologies	**8.1. Connectivity & IoT**	✓	✓	+	✓	+
	8.2. Cloud & DevSecOps	✓	✓	+	+	+
	8.3. Applications & integrations	✓	✓	■	■	+
	8.4. Big data & data science	✓	✓	+	+	+
	8.5. Edge processing	✓	✓	+	+	+
	8.6. Embedded development	✓	✓	+	+	+
9. Documented		+	+	+	+	✓
10. Classifiers		+	+	+	+	+

Table 2.4 Comparison between similar traffic analysis implemented systems.

Notations:

- ✓ : this feature exists/is implemented
- + : this feature doesn't exist/isn't implemented

2.5 The Need to Extend Related Work

Due to a close study for table 3.1, we found that only 2 papers, [16] and [21], addressed intelligent traffic analysis simulation, the rest of the papers did not.

Following a close study for table 3.2, we concluded that there is no implemented open-source system with proper documentation explaining the technologies or methods used to implement the system.

None of the existing systems or papers addressed the Egyptian environment in particular at all.

There is a need to address the following features in table 3.1: 2.1, 2.2, 2.4, 3, 5 and the following features in table 3.2: 2.3, 3.2, 3.3 and 9.

In conclusion, our intelligent traffic analysis simulation added value is outlined in the three points.

1. It's going to be regional. None of the researched papers above and no known systems works in the city of Alexandria, so we hope to be first paper or system to be directly linked to this region.

2. It's aimed to be open source. None of the researched papers above are open source and the system we strive to achieve will be open source for the whole world to alter or add and for more research to be worked on it with our fellow colleagues.

3. We plan to make it well documented. The used code will be illustrated thoroughly while opening the door for future updates and enhancements which we could not found in any of the papers or systems we researched.

2.6 Scope of Work

1) Identify simulation model, this includes:

- Identify traffic analysis as a simulation problem.
- Identify its events.
- Events' probability distributions

2) Simulation for specific area in Alexandria, this includes:

- Identify this area’s traffic problems and when they happen.
- How these problems can be fixed

3) Data processing, this includes:

- Reading data
- Converting XML files to CSV files
- Analyzing Data
- Feature Scaling
- Dimensionality Reduction using PCA.
- Clustering

4) Reinforcement learning, this includes:

- Agent
- Reward function
- Memory handling

5) Reinforcement learning result analysis, this includes:

- Benchmarking
- Visualization

6) Generate intelligent traffic simulation, this includes:

- Link reinforcement learning with simulation.
- Analyse how it differs than basic simulation.

7) Open source, this includes:

- Publish public repo on GitHub.

8) Documentation, this includes:

- Report writing
- References for all resources used.

2.7 Conclusion

This chapter included all the background needed to be able to understand our project. Additionally, it shows you the research process we went through to be able to understand the whole problem better.

By the end of this chapter, we were able to decide exactly what we want to do in our project presented in our scope of work and how we decided on these points on particular.

An overall view of our intelligent traffic analysis systems is presented in chapter 3. More details about our system's implementation are introduced in chapter 4.

Chapter 3

Requirements, design, and implementation environment of the system

3.1 Introduction

In the previous chapter, all the scientific background and related work needed for the project was discussed.

In this chapter, we will go through the proposed workflow in the system with a flow chart that summarizes the sequence of events in section 3.2.

We will also be discussing the requirements and implementation environment needed to be able to implement this project in section 3.3.

In the following section 3.4, we will decompose the systems components and explain briefly why we used each of them including system and data decomposition along with the Process Activity Task Matrix.

Section 3.5 will cover the Software Process Improvement (SPI) model and which parts we were able to follow according to the limited scope of the project which are, development process and peer review process, that are discussed in section 3.6.

3.2 Workflow Overview

3.2.1 Proposed Workflow

This section will mainly discuss the major steps needed to run the project and reach the results we reached.

- To start running the project, there is a need to decide whether the reinforcement learner will be used or not.
- In case it's decided not to use the reinforcement learner, the basic SUMO traffic system is initiated, and its results are collected for analysis.
- If the reinforcement learner is chosen, another decision needs to be made which is to use the simple DQN agent or the full DQN agent.
- To be able to link between the reinforcement learner and SUMO's GUI, a command in the terminal has to be written which is parsed to know which of the previous decisions were made, along with the maximum number of iteration steps to execute.
- After SUMO's GUI starts, it keeps iterating to dictate the reinforcement learning – generated traffic light states until the maximum number of iterations.
- Finally, all simulation results are collected for analysis, which includes benchmarking and visualization, to compare between simple DQN, full DQN agent and basic SUMO traffic system.

3.2.2 High-level Workflow flowchart

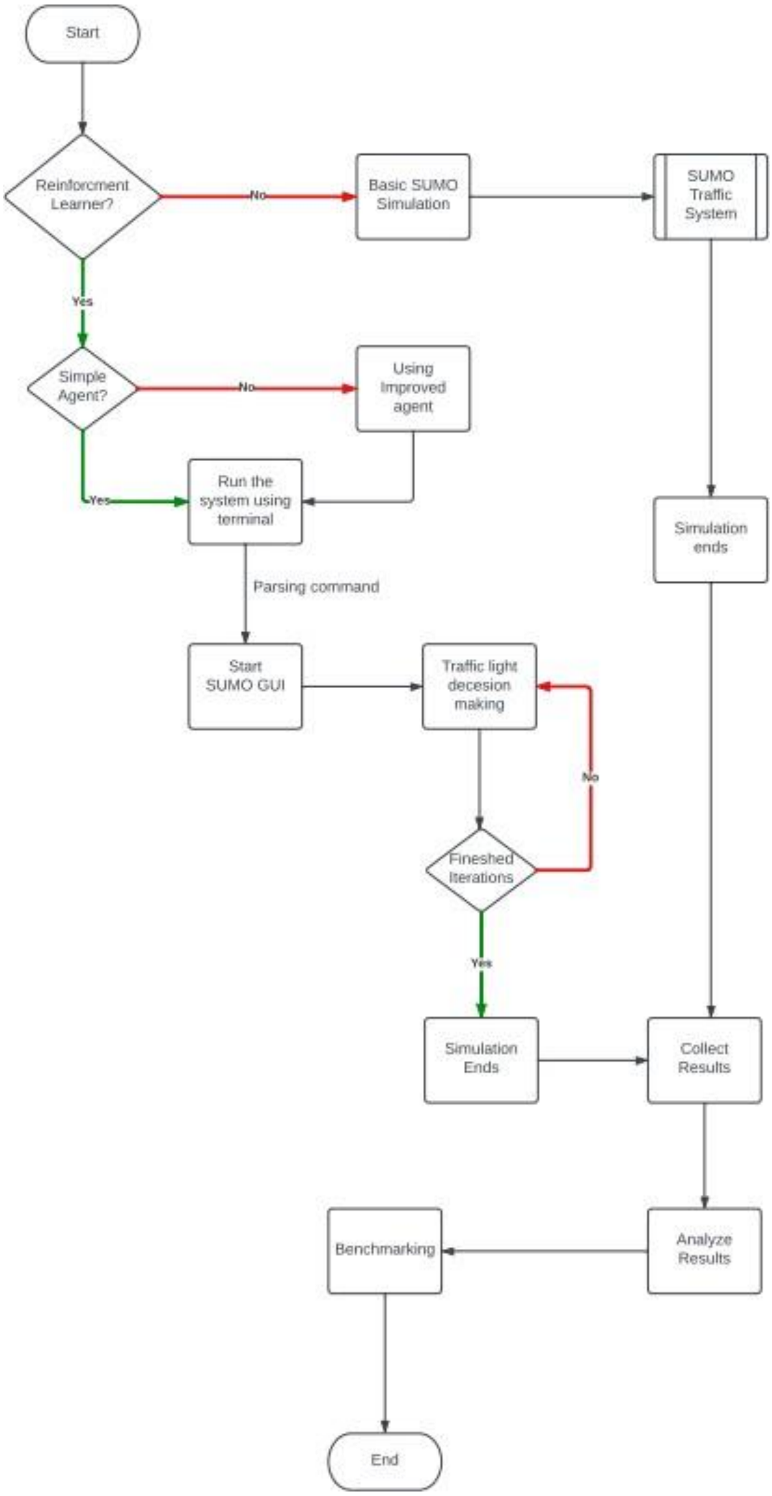


Figure 3.2.2

3.3 Software Requirements

3.3.1 Needed Environment

To run and test the project it was required to implement a simulation environment as well as the machine learning environment. The project was written in **python 3.8** for its support for most reinforced learning libraries, its stability and efficiency.

Simulation environment

As a result of its simplicity to communicate with python libraries and the readily available maps, SUMO was chosen as the main simulation environment through SUMO GUI and Traci.

Reinforced learning environment

Reinforced learning is supported by a wide range of python libraries. Keras was the library of choice for having built in functions to implement the features needed.

Data set manipulation

We needed to convert the data gathered from the simulation from .xml to .csv files to be more user-friendly when integrated with machine learning and reinforcement learning, this was mainly achieved using Lxml.

General purpose libraries

- numpy: used to perform a wide variety of mathematical operations on arrays.
- csv: implements classes to read and write tabular data in CSV format.
- sys: provides various functions and variables that are used to manipulate different parts of the Python runtime environment.
- os: provides functions for interacting with the operating system.
- random: used to generate random numbers.
- math: provides standard mathematical constants and functions.
- json: parses JSON from strings or files.
- argparse: makes it easy to write user-friendly command-line interfaces.
- signal: allows you to use various signals provided by the underlying operating system.
- pandas: facilitates working with data sets.
- ast: helps Python applications to process trees of the Python abstract syntax grammar.
- itertools: iterates over data structures that can be stepped over using a for-loop.
- __future__: inherit new features that will be available in the new Python versions.
- matplotlib: creates static, animated, and interactive visualizations in Python.

3.3.2 Tools Used

Python 3.8

Python is an easy to learn, powerful programming language. It has efficient high-level data structures and a simple but effective approach to object-oriented programming. Python's elegant syntax and dynamic typing, together with its interpreted nature, make it an ideal language for scripting and rapid application development in many areas on most platforms [26].

Sumo

SUMO (Simulation of Urban MObility), an open source, highly portable, microscopic and continuous multi-modal traffic simulation package designed to handle large networks [27].

Traci

TraCI is the short term for "Traffic Control Interface". Giving access to a running road traffic simulation, it allows to retrieve values of simulated objects and to manipulate their behavior "on-line" [28].

Keras

Keras is an API designed for human beings, not machines. Keras follows best practices for reducing cognitive load: it offers consistent & simple APIs, it minimizes the number of user actions required for common use cases, and it provides clear & actionable error messages. Keras also gives the highest priority to crafting great documentation and developer guides [29].

Lxml

lxml is one of the fastest and feature-rich libraries for processing XML and HTML in Python. This library is essentially a wrapper over C libraries libxml2 and libxslt. This combines the speed of the native C library and the simplicity of Python [30].

3.3 Software Design Description

In this section, we are breaking down the structure of the traffic system we implemented into system decomposition in section 3.3.1 and data decomposition in section 3.3.2.

3.3.1 System Decomposition

The systems can be divided into two modules:

- Reinforcement Learning module
- SUMO simulation module

The following subsection will describe each of the two modules briefly:

Reinforcement Learning module

The reinforcement learning module implements the reinforcement learning models and the reward functions we decided to use for the project.

To be able to try different combinations to know which would give us the best results, we tried 2 reinforcement learning models, simple and full DQN, and two reward functions. Then, we tried each model with each reward function and gathered the data accordingly.

SUMO simulation module

As mentioned before, we had to set an environment for SUMO to be able to integrate it with the project which is what the simulation module is mainly about, it sets the needed environment variables for SUMO, along with the paths needed to run it.

Further details will be discussed in the next chapter.

3.3.2 Data Decomposition

The data gathered for this project are:

- LuST dataset
- AlST dataset

The following subsection will describe each of the two datasets briefly:

LuST dataset

LuST Scenario has been generated with SUMO version 0.26 and validated with real data [31]. The reason we chose this dataset is that it's a 24-hour simulation run for the whole city of Luxembourg and contains various different scenarios, yet it's not too complicated.

As a result, we thought it might be a good starting point to test our different models on and observe how they behave to be able to know which model works best to run it on the main dataset, AlST dataset.

AlST dataset

We generated this dataset to test how our best model, as observed using LuST dataset, acts on Alexandria's roads and whether it would be suitable for traffic in Alexandria and actually improve it or not.

This dataset wasn't available online like LuST dataset, we generated it ourselves using SUMO's with Open Street Map (OSM) web wizard.

3.3.3 Process Activity Task Matrix (PATM)

Process	Activity	Task
Simulation	Identify simulation model	Identify traffic analysis as asimulation problem.
		Identify events.
		Identify probabilitydistributions.
	Simulationfor specific area in Alexandria	Identify this area's traffic problems and when they happen.
		How these problems can befixed
Machine Learning	Data Processing	Reading Data
		Converting XML files toCSV files
		Analyzing Data
		Dimensionality Reductionusing PCA
		Clustering
	Reinforcement Learning	Agent
		Reward Function
		Memory Handling
	Reinforcement learning results analysis	Benchmarking
		Visualization
Integration	Generate intelligent traffic simulation	Link reinforcement learningwith simulation
		Analyze how it differs thanbasic simulation
Publication	Open source	Publish public repo onGitHub.
	Documentation	Report writing
		References for all resources used

Table 3.3.3 PATM

3.4 Software Engineering Process Followed

Since the project had multiple stages and different processes, it was crucial to follow a standard methodological software engineering model to keep the work organized and fully utilize the efforts of the whole team, the model we used was Software Process Improvement (SPI) model discussed in the following subsection.

3.4.1 Software Process Improvement model (SPI model)

Software Process Improvement (SPI) for Small Medium Enterprises (SMEs) model is a model developed by Software Engineering Competence Centre whose goal is to help small and medium enterprises to raise the quality of their products by using modern software development processes and practices.

consists of several processes: project management , development, peer-review, quality assurance and configuration management.

Since the SPI model targets mainly enterprise and business, it was required to slightly modify it to fit the scientific nature of the project. Three of these processes were followed in the work:

- Project management process: this was done in the form of a GANTT chart implementation plan, minutes of meeting (MOM) document after every meeting and weekly timesheets.
- Development process: the development processes contained several phases so it will be discussed in the next section 3.5
- Peer-review Process: having peers reviewing. code provides strength to the implementation and saves a lot of time in detecting errors early as discussed in section 3.6.

3.5 Development Process

The development process is an important aspect of the SPI methodology, which includes the following phases:

- Planning
- Design
- Implementation
- Result Analysis

3.5.1 Planning

To achieve the main requirements of the project, extensive research was done on traffic analysis scientific papers and implemented systems.

Out of this research, it was applicable to identify the scope of work, the need-to-extend features, and future work, in addition to using the proper frameworks and programming language.

3.5.2 Design

Since there are many approaches and features that are compared and analyzed, a flexible design is required to enable easy modifications and integration between the different components.

3.5.3 Implementation

It consists of 3 phases:

- Simulation which includes identifying the simulation model and the specific area in Alexandria to work on.
- Machine Learning including data processing and reinforcement Learning.
- Integration which helped us achieve generating an intelligent traffic simulation to link reinforcement learning with simulation.

3.5.4 Result Analysis

After gathering the different implemented scenarios' results, a comparison was done between them and the basic simulation. Then, a visualization of the results showed the best model to choose among them to apply it on Alexandria using AIST dataset.

3.6 Peer Review Process

Peer review aims to detect errors early and rapidly remove defects early in the development cycle. All team members share their ideas on each point which helps to provide many points of view for each point.

- Peer reviews are planned, scheduled, and monitored.
- Defects are identified, reworked, and removed during the peer review process.

3.7 Conclusion

In this chapter, the development process was briefly discussed. The project environment was introduced. In the next chapter, implementation details, analysis and results will be provided in detail in the next chapter.

Chapter 4

Implementation, Results, Discussion

4.1 Introduction

In the last chapter, we discussed the nature and environment required to implement this intelligent traffic management system.

In this chapter, we will go through the most important and most challenging phase of the project, which is the implementation phase in section 4.2.

Following the implementation phase, we had only one thing left to do which is to finally analyze the results of the models we tried and observe them to know what we can improve in the future, results and analysis are discussed in section 4.3.

4.2 Implementation Details

Throughout this section, we will go through all the phases we need to implement this project, starting from defining the simulation model to the final testing phase.

4.2.1 Identify Simulation Model

As we discussed before in chapter 2, section 2.2.4.2, why we need simulation for the traffic problem we're attempting to solve, it was found necessary to identify a clear and simple traffic simulation model for us to follow and use it later on to validate the results obtained.

4.2.1.1 Identify traffic analysis as a simulation problem

There are mainly 2 simulation modelling approaches, event-scheduling approach, and process approach [32].

Event-scheduling approach to discrete-event simulation modelling

System is modelled by identifying its characteristic events and then writing a set of event routines that give a detailed description of the state changes taking place at the time of each event. The simulation evolves over time by executing the events in increasing order of their time of occurrence.

Process approach to simulation modelling

Process is a time ordered sequence of interrelated events separated by intervals of time, which describes the entire experience of an "entity" as it flows through a "system." A system or simulation model may have several different types of processes. Simulation using the process approach also evolves over time by executing the events in order of their time of occurrence. Unlike an event routine, the process routine has multiple entry points.

4.2.1.2 Identify traffic events

Common traffic events to simulate are:

- Car arrival (arrival time)
- Car waiting (waiting time)
- Car leaving (response time)

4.2.1.3 Identify events' probability distributions

Probability distribution for cars' arrival times and waiting times events can be an exponential distribution [33] as shown in the below figure.

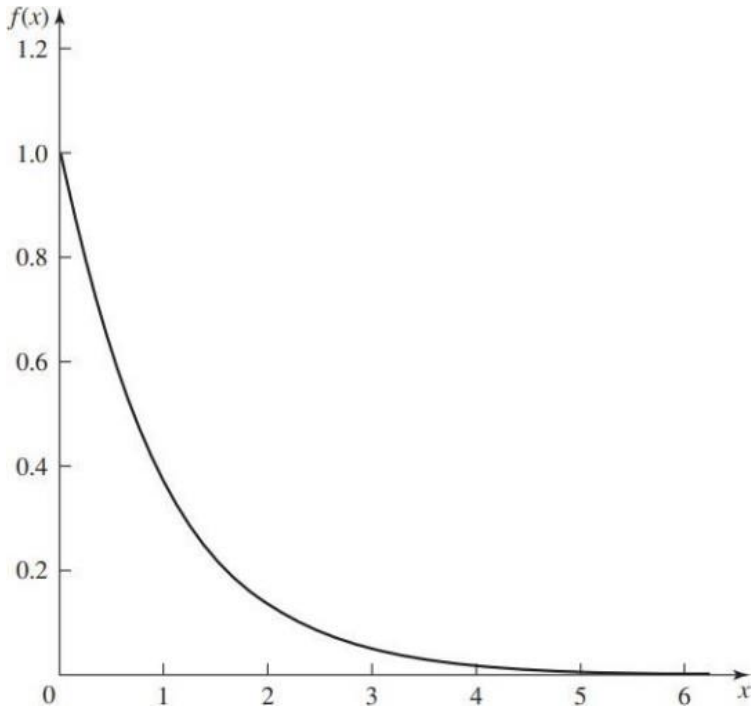


Figure 4.2.1.3

The response time is the addition of waiting time, represented by the time a car spends waiting for a red traffic light to turn green, and the service time, represented by the time a car spends driving towards its destination.

4.2.2 Datasets

As was mentioned before in chapter 3, section 3.3.2, we used two datasets for this project, LuST and AlST datasets, and we discussed the reasons why it was chosen to work with them.

In this section, we will be discussing in detail the process we went through to prepare these datasets for the nature of the project.

4.2.2.1 Pre-processing Phase

The data generated by SUMO is always in an XML format which isn't very user friendly and hard to read so in order to facilitate its usage, we converted it into CSV format, but some steps were taken in advance.

1) A command needs to be written in the cmd, inside the folder where the scenario is, to import digital road networks from different sources and generates road networks that can be used by other tools from the package. The command is "netconvert -s <name>.net.xml –plain-output-prefix". This helps in converting the data from .xml to .csv. After finishing this command, a success message appears, and some extra files will be created.
2) Then, we start parsing the .xml files to gather the needed data and place it in a .csv file. The parsing takes place through the following functions:
   - isRoundAbout(): determines if a given lane is part of a roundabout. A roundabout is a road junction at which traffic moves in one direction round a central island to reach one of the roads converging on it.
   - appendEmptyValuesToRow(): for values that are null or equal zero

- processLanes(): processes all info about lanes like; lane lengths, lane speeds, lane ids, etc... and appends them to the row.
- findConnections(): Find connections using the global nodes_table and connection_tl table, we need to find the "tl"-id for a corresponding junction id. Using this id, we can figure out how many traffic lights this junction controls because number of connections with tl == tl_id of junction equals the number of traffic lights.
- evaluateJunction(): Reads data from xml for one junction.
- createXmlEntityCaches(): This method creates hash tables for faster searching in the xml data by building connection, node, and lane tables.
- runDataExtraction(): Run the extraction process for; xml: scenario.net.xml file, node_xml: a scenario.nod.xml file, that was generated using netconvert and output_filename: the name of the csv file to export.

3) After the having the .csv file, we apply one of the most important steps which is data cleaning, and it will be discussed in the following section 4.2.2.2.

4.2.2.2 Data Cleaning Phase

In this phase, we work on understanding the features we have and dropping any redundant featured to make sure that all the following steps will be efficient.

1) We have the following features:

- junction_id: the id of the intersection
- junction_type: is always traffic_light5
- x,y,z: the physical coordinates of the junction in the simulations

- isRoundabout: Whether the intersection is part of a roundabout.
- trafficlight_count: an array containing the number of traffic lights the intersection controls, trafficlight's ID and a list of all the connections to the intersection.
- avg_lane_speed, avg_lane_length & standard deviations: Because every intersection has a different number of lanes, the mean was used to aggregate their speed limits and lengths. To mitigate the information loss the standard deviation was also computed.
- edge_types: the different types of edges6 connected to this intersection.
- edge_priorities: The average of the edge's priorities. Because these values are very close together, the standard deviation was not included.
- number of lanes: the total amount of lanes in the junction

2) For the data to be more efficient during preprocessing, the following columns will be dropped:

- junction_id: because it is a unique id, it won't be of any value for clustering.
- junction_type: this is always the same value "traffic_light".
- x,y: they won't be of any importance to the results.
- z: in these example scenarios, there are no z-values.

3) The roundabout field will be converted from {true|false} to {0|1} because dealing with numbers is easier than dealing with string values.

4.2.2.3 Dataset analysis Phase

After cleaning the data and making sure it's ready for processing, one last step is required which is analyzing and understanding this data to prepare it for the reinforcement learning model because the quality of the data fed into the models can greatly affect the obtained results.

1) Feature scaling, as explained in section 2.2.2.3, was applied to the data to normalize the range of independent features present. For example, in our datasets, length of lanes and maximum speed are on different scales and vary greatly in range, feature scaling had to be implemented before running PCA so that we're sure that no feature affects the results more than the others.

2) Dimensionality reduction using PCA, as discussed in section 2.2.2.4, was applied to reduce the number of random variables in the problem by obtaining a set of principal variables. Dimension reduction eliminates noisy data dimensions and thus and improves accuracy in classification and clustering, in addition to reduced computational cost, so this step was needed before the next step, clustering.

3) Clustering, as elaborated in section 2.2.2.5, has been proposed to improve the learning speed of reinforcement learning models [34]. Clusters generated represent different traffic scenarios as dealing with each of them would be done differently, so dividing the data into clusters of similar states will facilitate the learning process of the reinforcement learning model we implement.

4) Finally, the datasets are exported to csv files again, but with the clustering results appended to them.

4.2.3 Traffic System Implementation Details

Implementing this intelligent traffic management system can be mainly split into two parts, the reinforcement learning model, along with the reward function, and setting the simulation environment in a way suitable to be integrated with the reinforcement learning model.

4.2.3.1 Model Implementation Details

For the initial DQN model created, we followed the steps mentioned here [35], which we will explain throughout this section. In section 2.2.3, we thoroughly covered reinforcement learning, the need to use it and why DQN model was chosen, so in this section, we'll talk more about the steps taken to implement it.

The initial simple DQN model consists of only two dense layers, with learning rate = 0.00025 and RMSProp as the optimiser. Let's break this down a little bit.

1) Dense layer: a dense layer is a layer that is deeply connected with its preceding layer which means the neurons of the layer are connected to every neuron of its preceding layer [36].
2) Learning rate: the learning rate is a tuning parameter in an optimization algorithm that determines the step size at each iteration while moving toward a minimum of a loss function [37].
3) RMSProp: root mean squared propagation is the optimization machine learning algorithm to train the Artificial Neural Network (ANN) by different adaptive learning rate. RMSProp can reach the faster convergence rate than the original optimizer [38].

Some modifications were made to improve this model and reach the full DQN model. These modifications include, raising the number of layers to three instead of two because adding more layers allows for learning the data fed to the model better.

Another improvement was degrading the learning rate from 0.00025 to 0.000001 as high learning rates may result in taking very large steps and missing the maximum point we're looking for.

4.2.3.2 Reward Function Details

For the reward functions, we needed some guidance on what parameters we can use for efficient reward functions, our references for this part were [39] and [40]

First reward function

The idea for the reward function was to encourage normal situations with an equal amount of green and red lights.
Additionally, the average waiting time should be minimized. The reward function is;

$$R(s, a) = + W(s) + 0.1 * \sum \text{green lights} - 0.1 * \sum \text{red lights}$$

Equation 4.2.3.2.1

It's a negative reward function that depends mainly on the last term which is the average waiting time, if the waiting time exceed a certain limit, the model is penalized and if the waiting time is zero, the model is rewarded.

Second reward function

The second reward function's main idea in to stop sudden actions that fluctuate quickly by calculating the hamming distance between every consecutive actions to try and minimize it. Moreover, the model is penalized for emergency stops. The reward function is;

$$R(s,a,a') = (occupancy/haltingNumber) – hamming(a,a') - emergencyStops$$

Equation 4.2.3.2.2

The first term of the equation mainly focuses on the reward not the penalty, it rewards the model for high traffic flows which means the higher the occupancy and lower the halting number, the more the reward it collects.

4.2.3.3 Simulation Implementation Details

Setting up SUMO environment, that will be able to be integrated with reinforcement learning, requires states, actions and rewards as mentioned before [41].

Also setting SUMO_HOME is important to be able to visualize our improved traffic simulation on SUMO [42].

Main functions used in setting the SUMO environment and integrating it with the reinforcement learning models that were created are:

- step(): which moves to the next simulation step (iteration) after gathering the data of the previous step.
- storeLastActions(): which acts as a memory for the actions taken in the latest simulation step to be able to learn from them in the following step.

Figure 4.3.2.1

- computeRewards(): which is used to apply either of the above reward functions and calculate the final reward for every simulation step.
- performActions(): which actually controls the traffic light and change it according to the decision taken by the reinforcement learning model.

4.2.4 Testing Phase

This is the last phase in our implementation process, this phase is divided into two separate steps:

- Testing the project with LuST dataset
- Testing the project with AlST dataset

Testing the project with LuST dataset

This step is mainly taken to be able to compare between the results obtained from our 2 different models with our 2 different reward functions, along with SUMO's basic simulation, so the cases we have for this step are:

1) Simple DQN model with first reward function
2) Simple DQN model with second reward function
3) Full DQN model with first reward function
4) Full DQN model with second reward function
5) SUMO's basic simulation

Testing the project with AlST dataset

In this step, we only compared between Full DQN model with second reward function, as it gave the best results when tried with LuST dataset, and SUMO's basic simulation to be able to observe how much we improved the traffic flow in Alexandria.

4.3 Results & Discussion

In this section, we will discuss all the cases we tested, the results we obtained for each case and the observations we were able to make.

4.3.1 Experiments

The experiments we could think of consist of the following:

1) 5 different models to compare between as mentioned in section 4.2.4
2) Changing number of iterations (simulation steps) between 500, 1000, 1500, 2000, 3000, 4000 for LuST dataset and 250, 500, 750, 1000, 1500, 2000 for AlST dataset.
3) Changing traffic scale to try and monitor different congestion levels at different times of the day, so we changed traffic scale between 25, 50, 75 and 100.
4) The parameters that were found to be the most important to decide which is the best model were [43]:
    - Waiting time: The average time spent standing (involuntarily) (seconds).
    - Waiting cars: Number of vehicles with delayed insertion that were still waiting for insertion at simulation end.
    - Speed: The average trip speed (m/s).
    - Depart delay: The average time vehicles had to wait before starting their journeys (seconds).
    - Time loss: The average time lost due to driving slower than desired (includes waiting time) (seconds).

4.3.2 Results

Results for LuST dataset

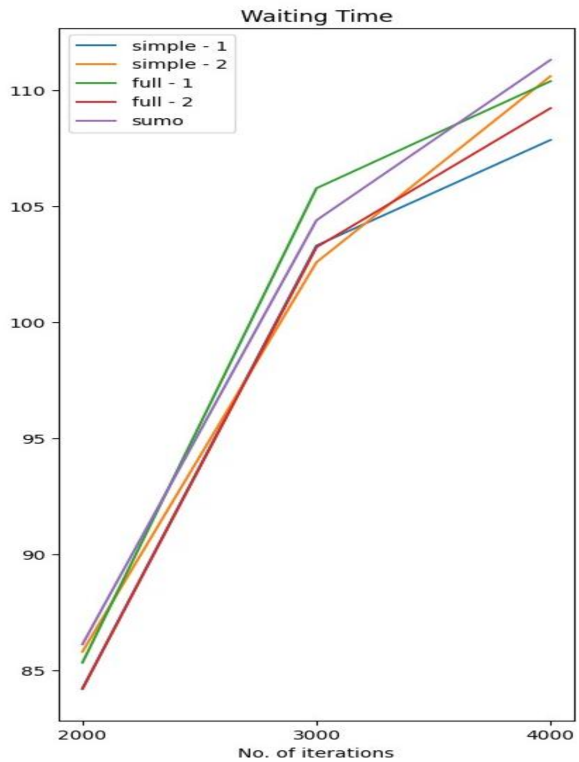
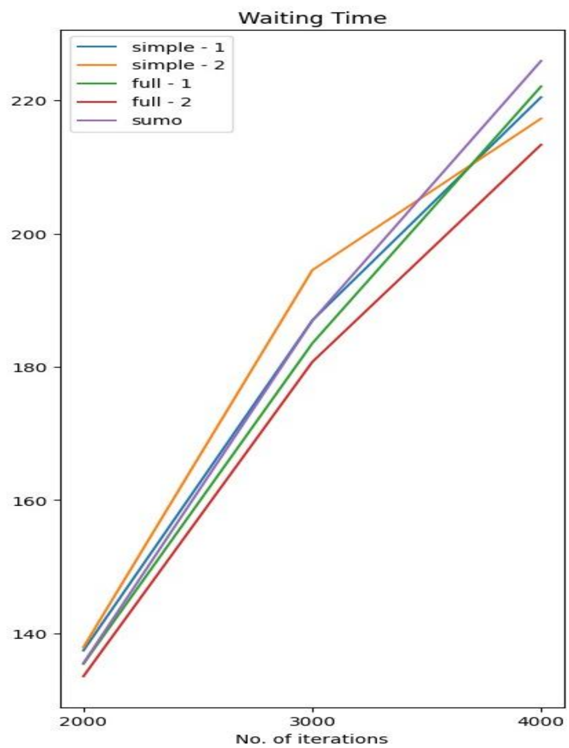


Figure 4.3.2.1

Waiting time for scale traffic = 25. In case of light traffic, simple DQN with first reward function behaves well, and second-best model is full DQN with second reward function.



Waiting time for scale traffic = 50. In case of moderate traffic, best two models are full DQN with first and second reward function.

Figure 4.3.2.2

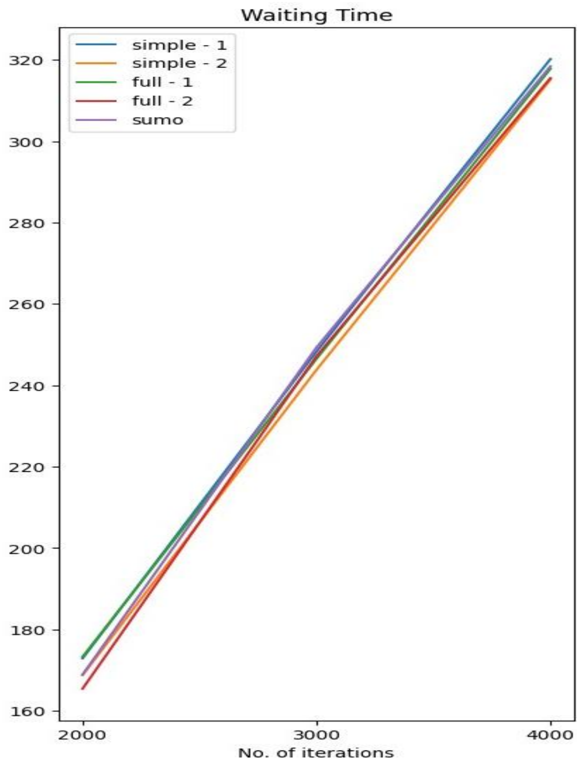


Figure 4.3.2.3

Waiting time for scale traffic = 75. In case of heavy traffic, the 4 models we implemented beat SUMO's basic simulation.

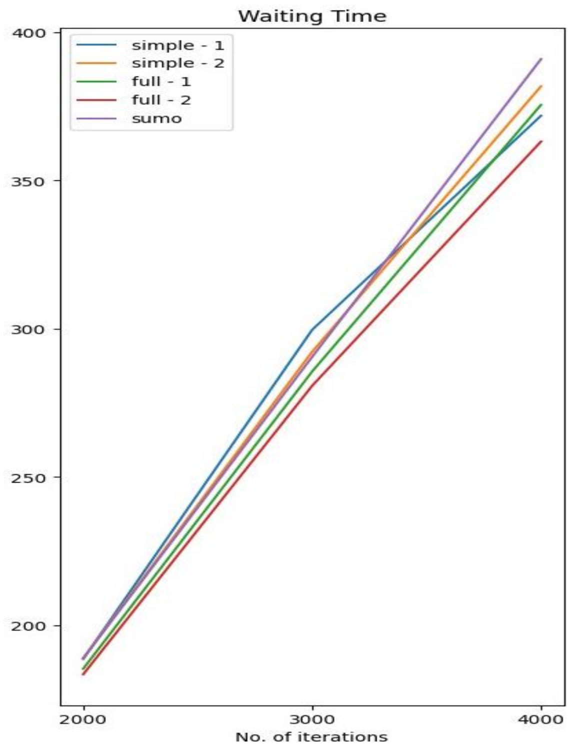


Figure 4.3.2.4

Waiting time for scale traffic = 100. In case of extreme traffic, the best model is full DQN with second reward function which means it behaves really well even with extreme traffic.

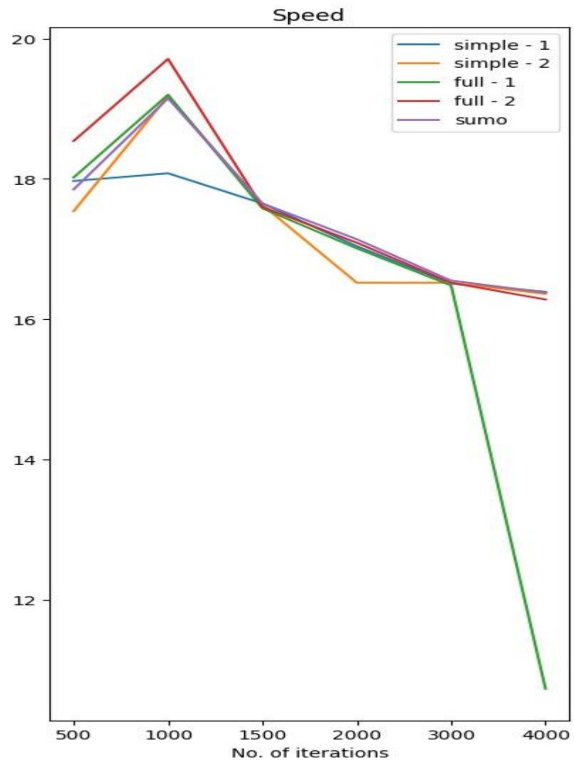


Figure 4.3.2.5

Speed for scale traffic = 25. In case of light traffic, best speed was obtained by the full DQN with second reward function and worst one is full DQN with first reward function.

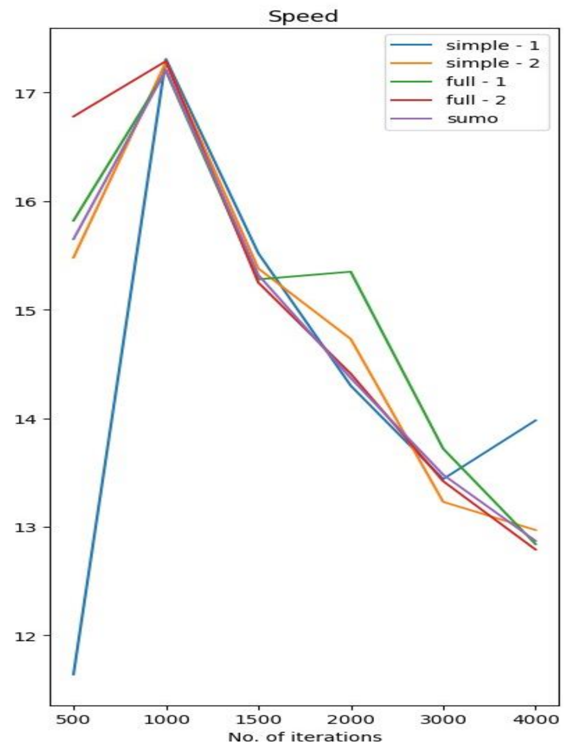


Figure 4.3.2.6

Speed for scale traffic = 50. In case of moderate traffic, all models obtained approximately equal results.

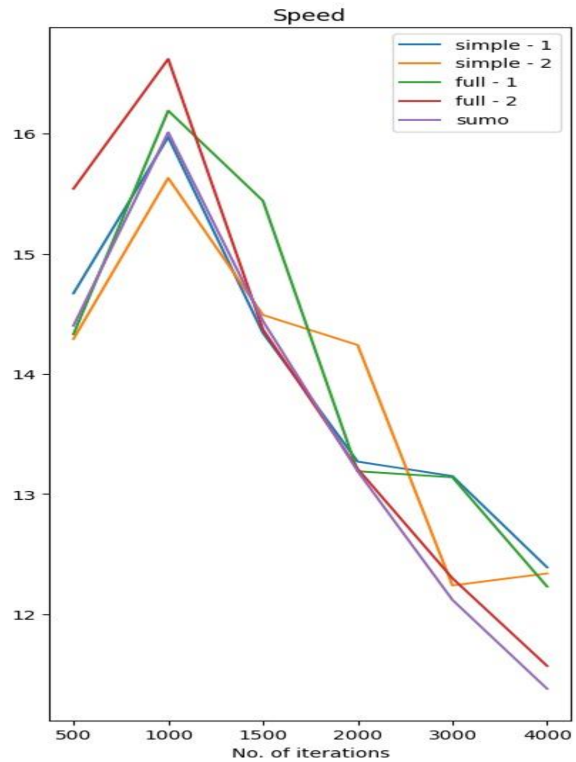


Figure 4.3.2.7

Speed for scale traffic = 75. In case of heavy traffic, all models obtained approximately equal results.

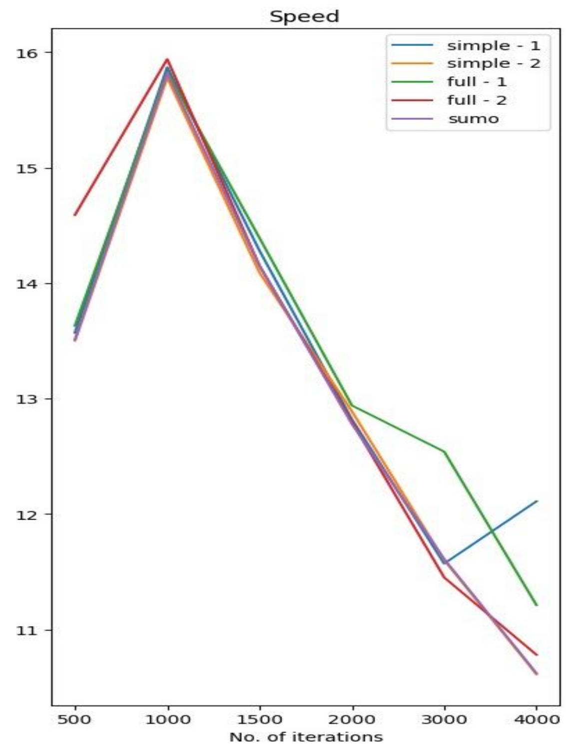


Figure 4.3.2.8

Speed for scale traffic = 100. In case of extreme traffic, all models obtained approximately equal results.

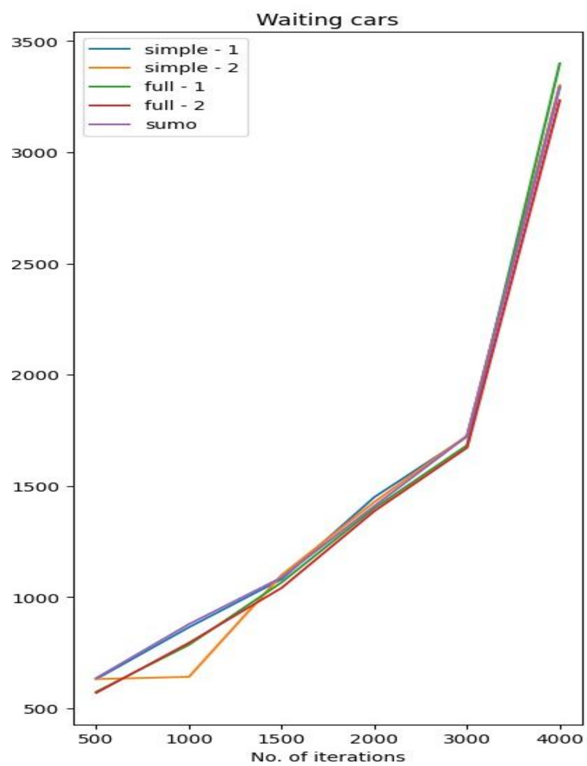


Figure 4.3.2.9

Number of waiting card for scale traffic = 25. In case of light traffic, all models obtained approximately equal results.

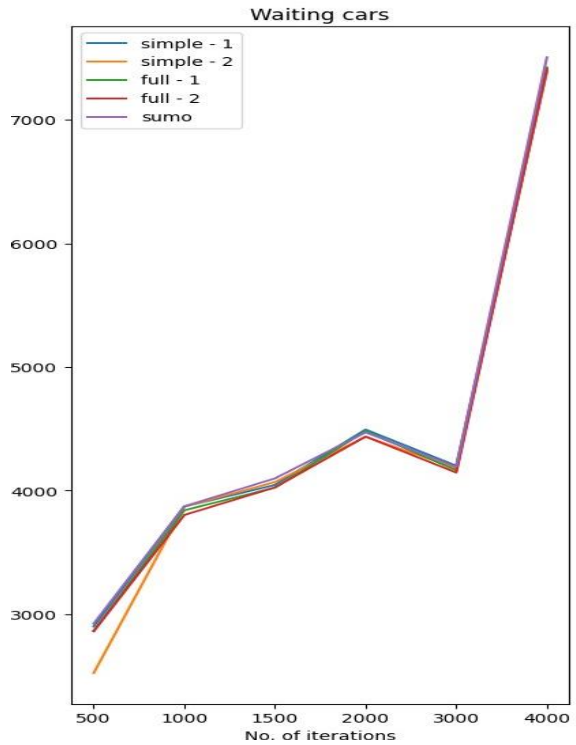


Figure 4.3.2.10

Number of waiting card for scale traffic = 50. In case of moderate traffic, all models obtained approximately equal results, but simple DQN with second reward function performed slightly worse.

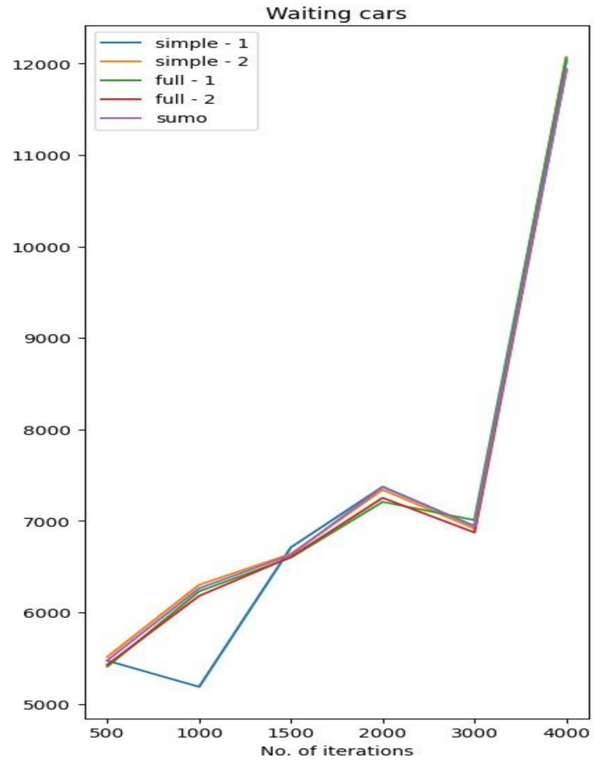


Figure 4.3.2.11

Number of waiting card for scale traffic = 75. In case of moderate traffic, all models obtained approximately equal results, but simple DQN with first reward function performed slightly worse at the beginning.

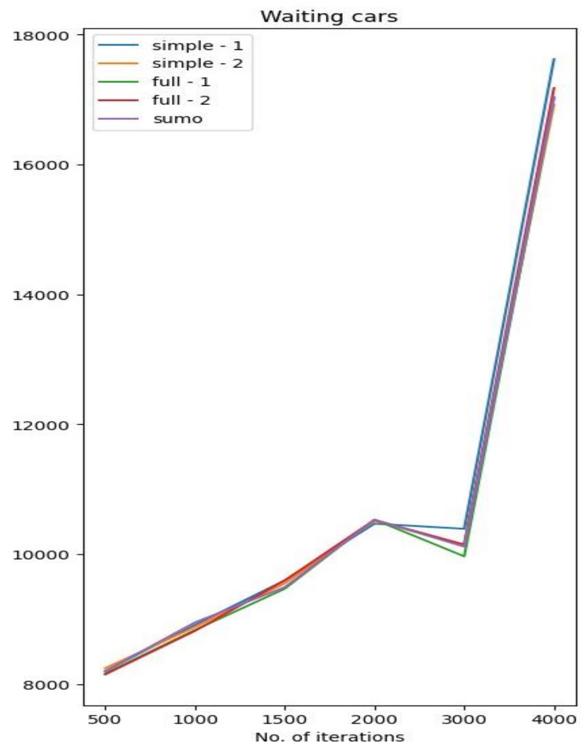


Figure 4.3.2.12

Number of waiting card for scale traffic = 100. In case of moderate traffic, all models obtained approximately equal results.

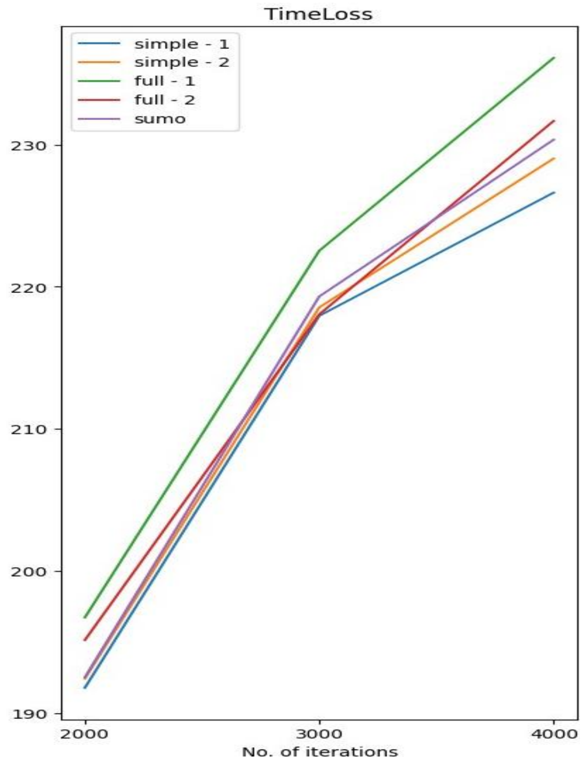


Figure 4.3.2.13

Time loss for scale traffic = 25. In case of light traffic, all models obtained approximately equal results, but simple DQN with first reward function performed slightly better than the rest.

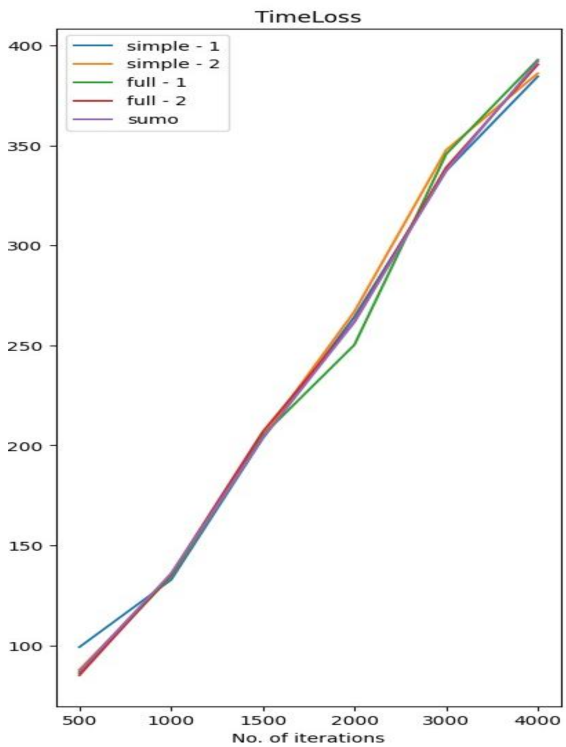


Figure 4.3.2.14

Time loss for scale traffic = 50. In case of moderate traffic, all models obtained approximately equal results.

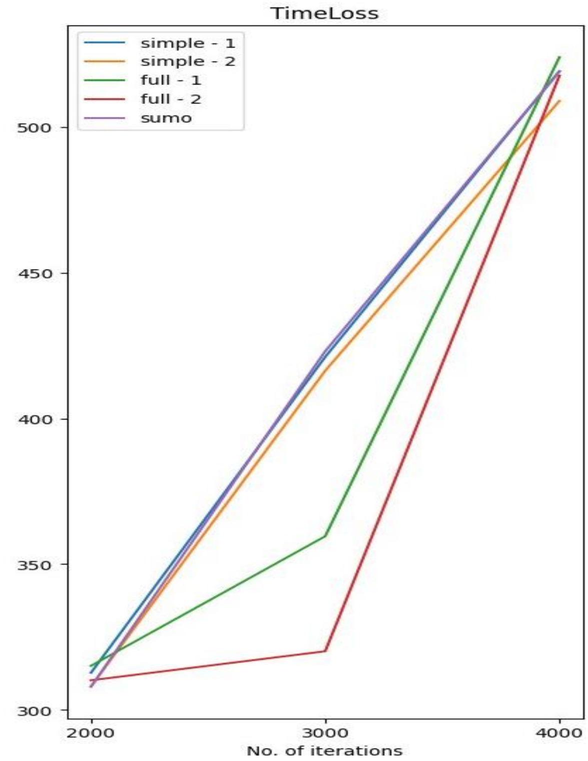


Figure 4.3.2.15

Time loss for scale traffic = 75. In case of heavy traffic, full DQN with second reward function performed better than the 4 models.

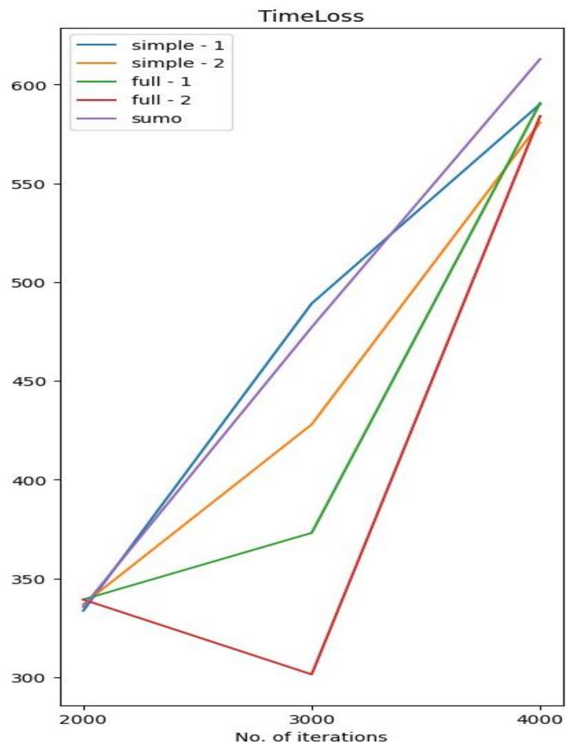
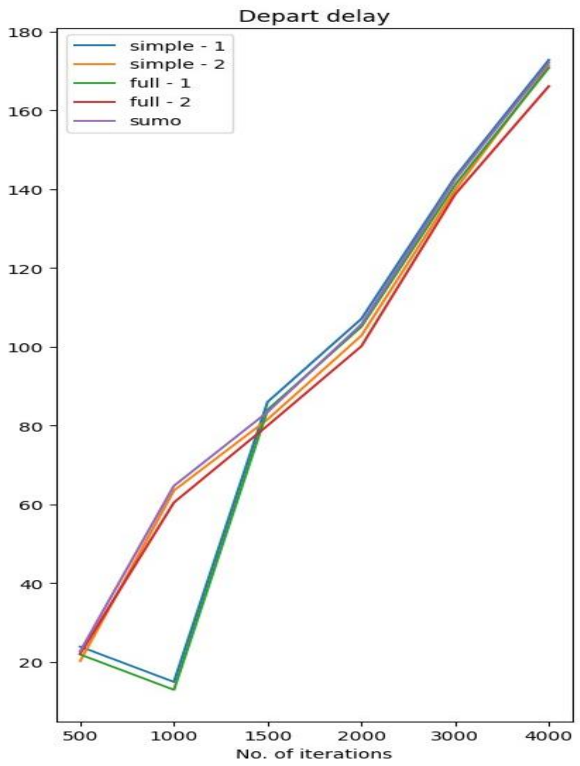


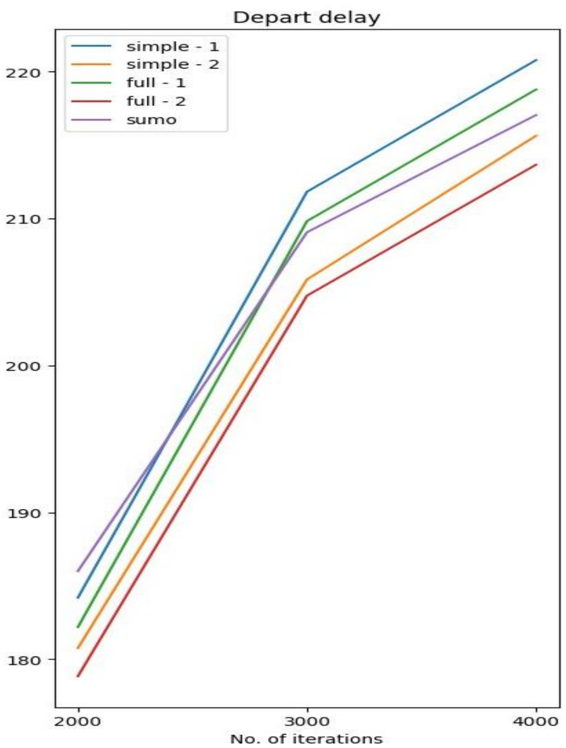
Figure 4.3.2.16

Time loss for scale traffic = 100. In case of extreme traffic, full DQN with second reward function performed better than the 4 models.



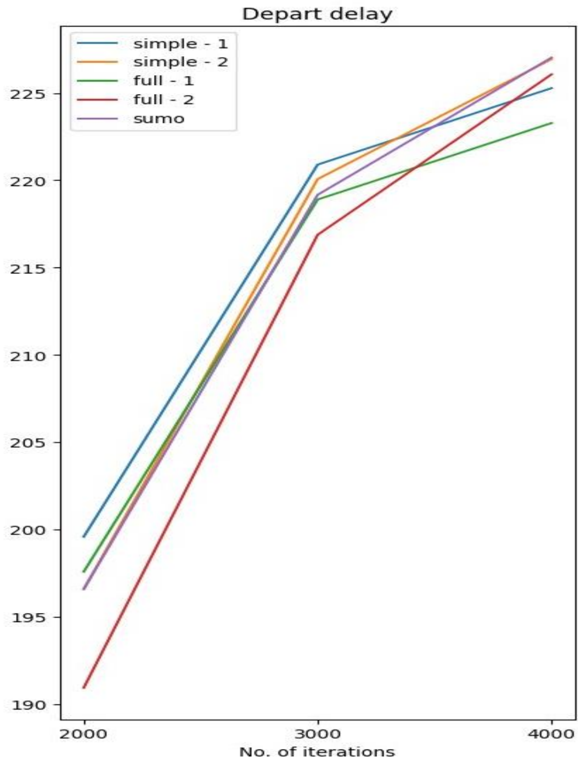
Depart delay for scale traffic = 25. In case of light traffic, full DQN with first reward function and simple DQN with first reward function performed slightly worse than the rest at the beginning.

Figure 4.3.2.17



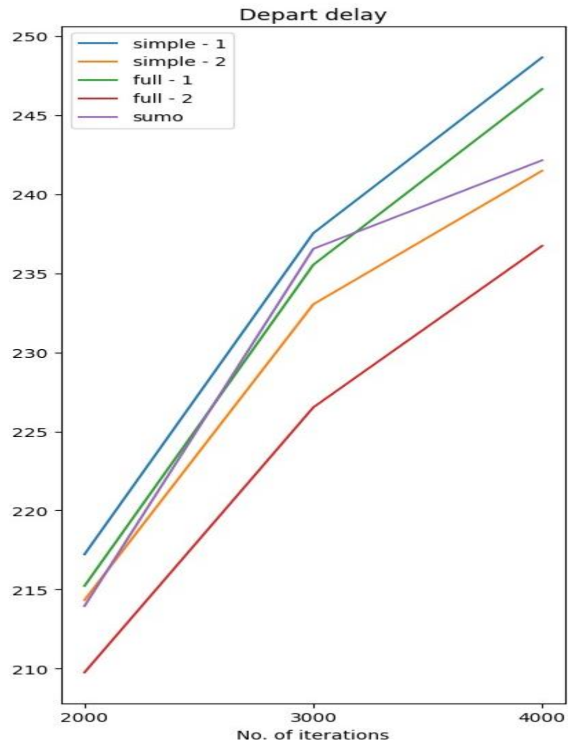
Depart delay for scale traffic = 50. In case of moderate traffic, full DQN with second reward function performed better than the other 4 models.

Figure 4.3.2.18



Depart delay for scale traffic = 75. In case of heavy traffic, all models performed approximately the same.

Figure 4.3.2.19



Depart delay for scale traffic = 100. In case of extreme traffic, full DQN with second reward function performed better than the other 4 models.

Figure 4.3.2.20

Average results for LuST dataset

- **When taking the average of the previous parameters, full DQN with second reward function performed the best.**

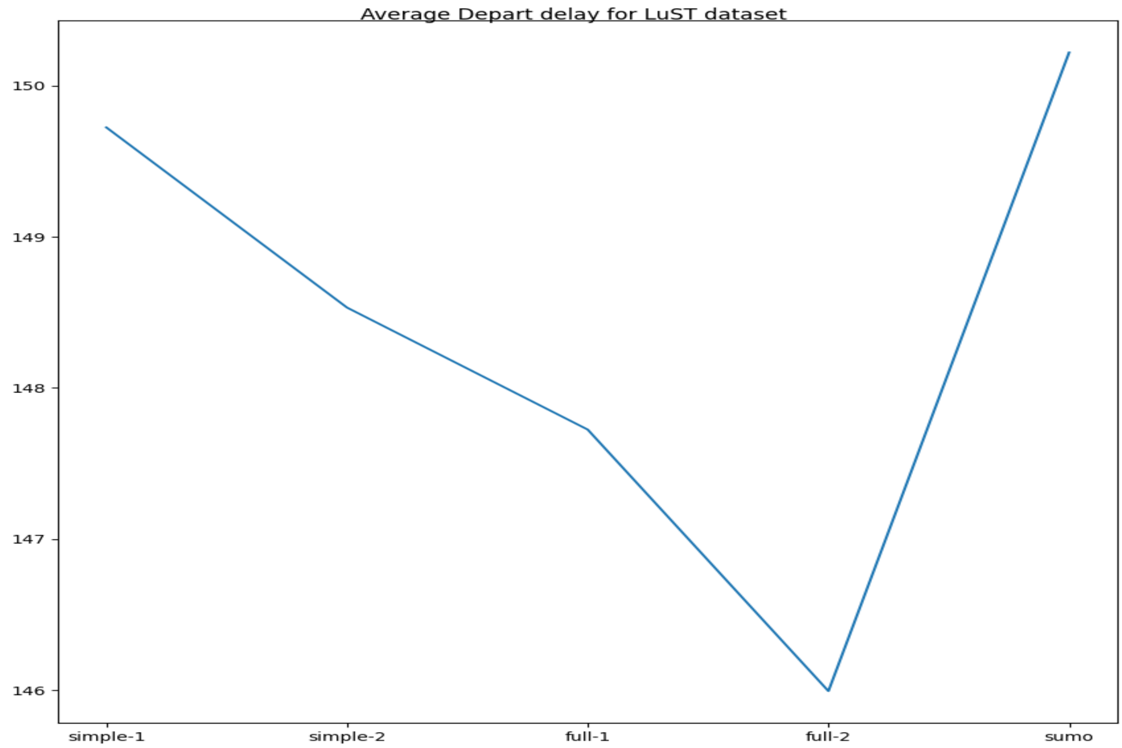


Figure 4.3.2.21

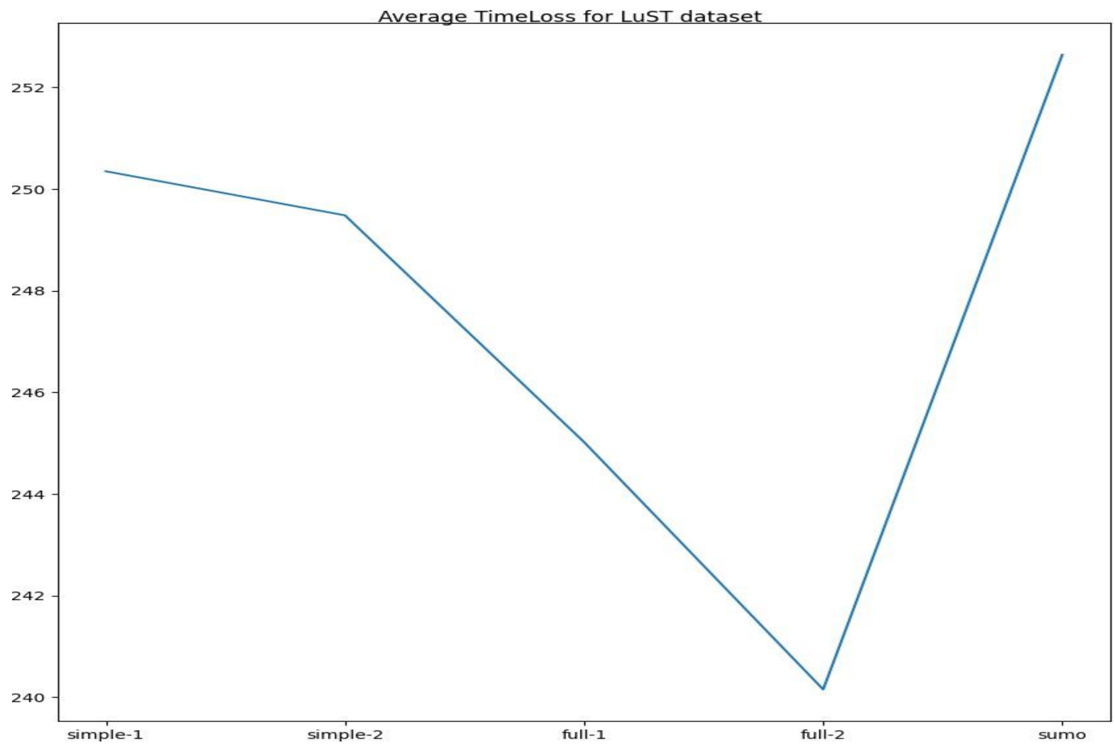


Figure 4.3.2.22

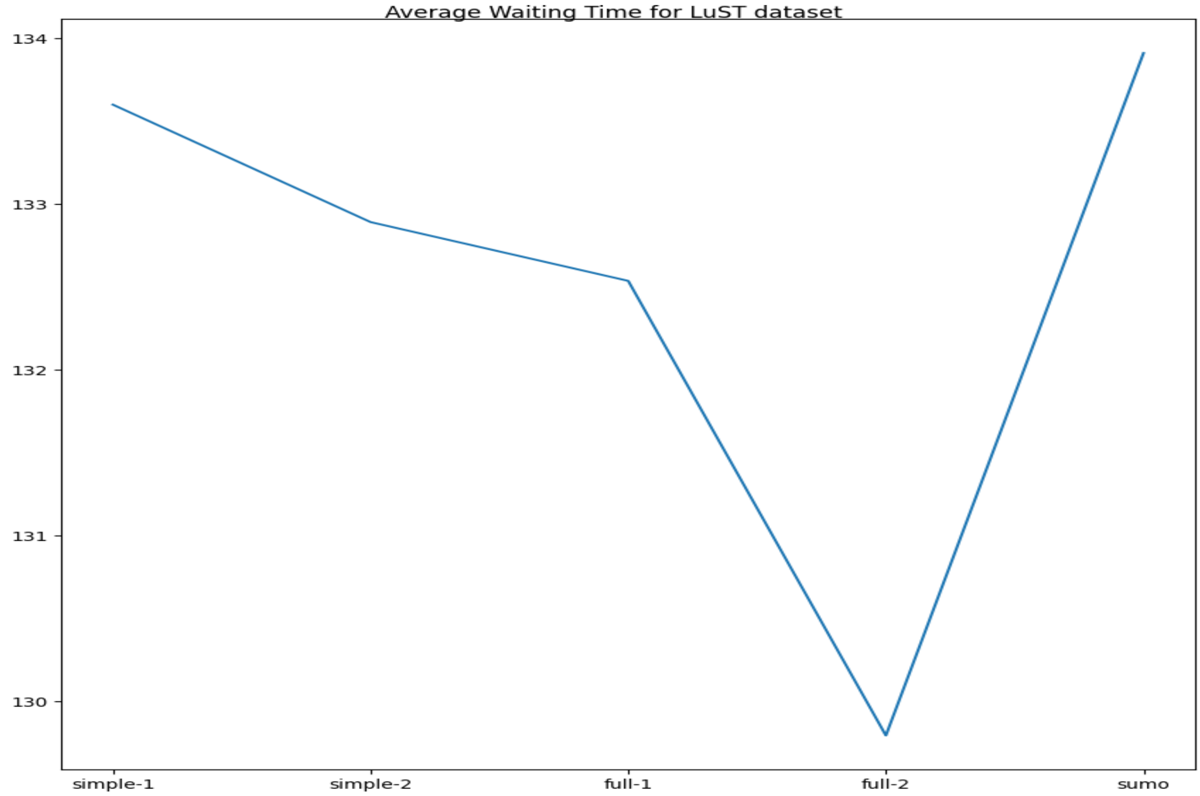


Figure 4.3.2.23

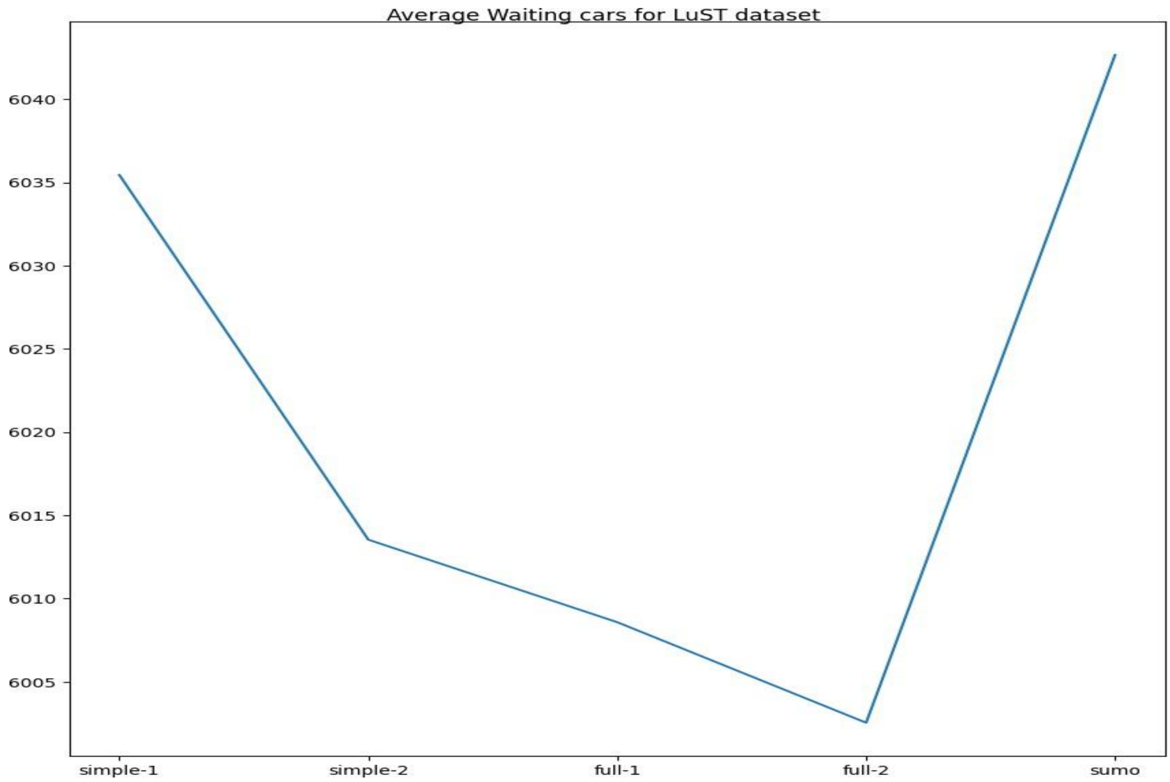


Figure 4.3.2.24

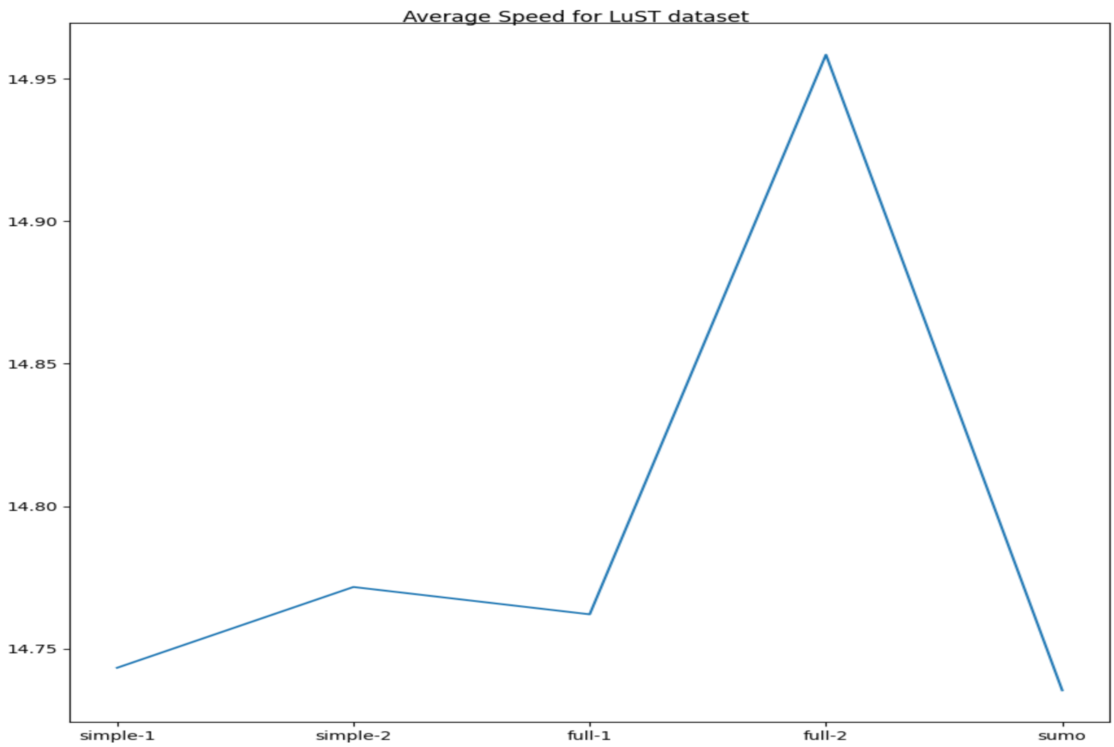


Figure 4.3.2.25

Results for AlST dataset

- **Full DQN with second reward function performed better than SUMO's basic simulation.**

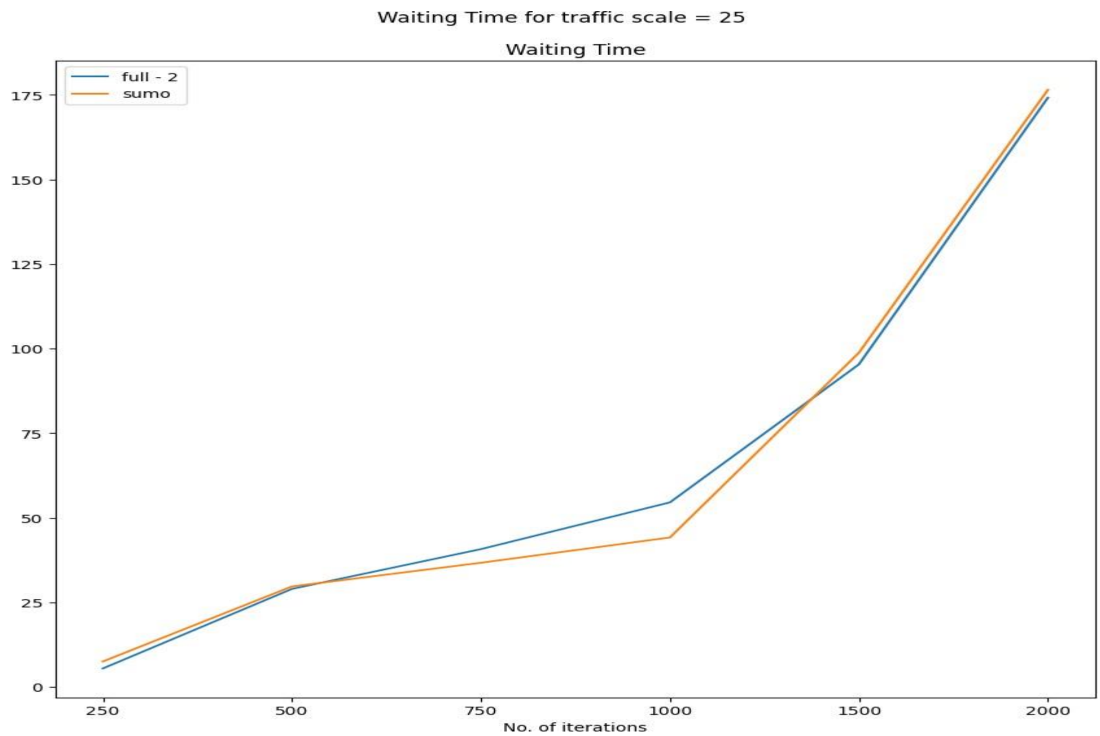


Figure 4.3.2.26

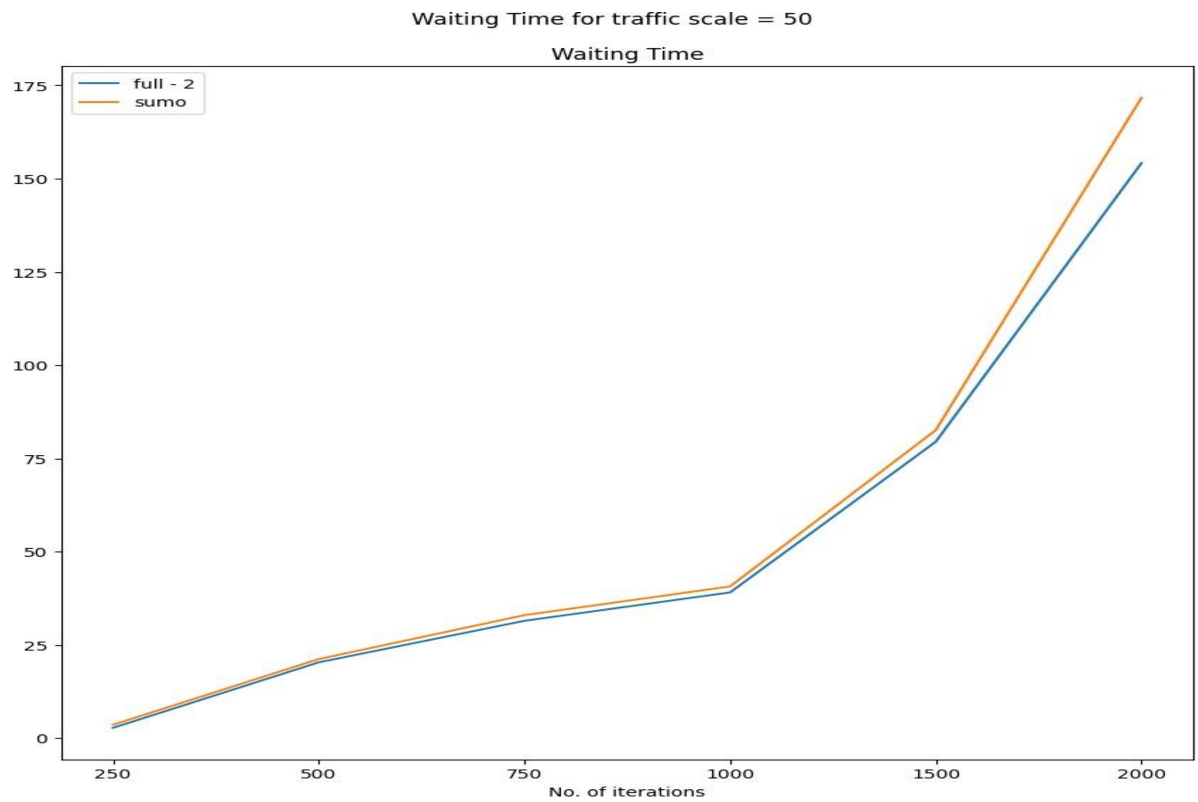


Figure 4.3.2.27

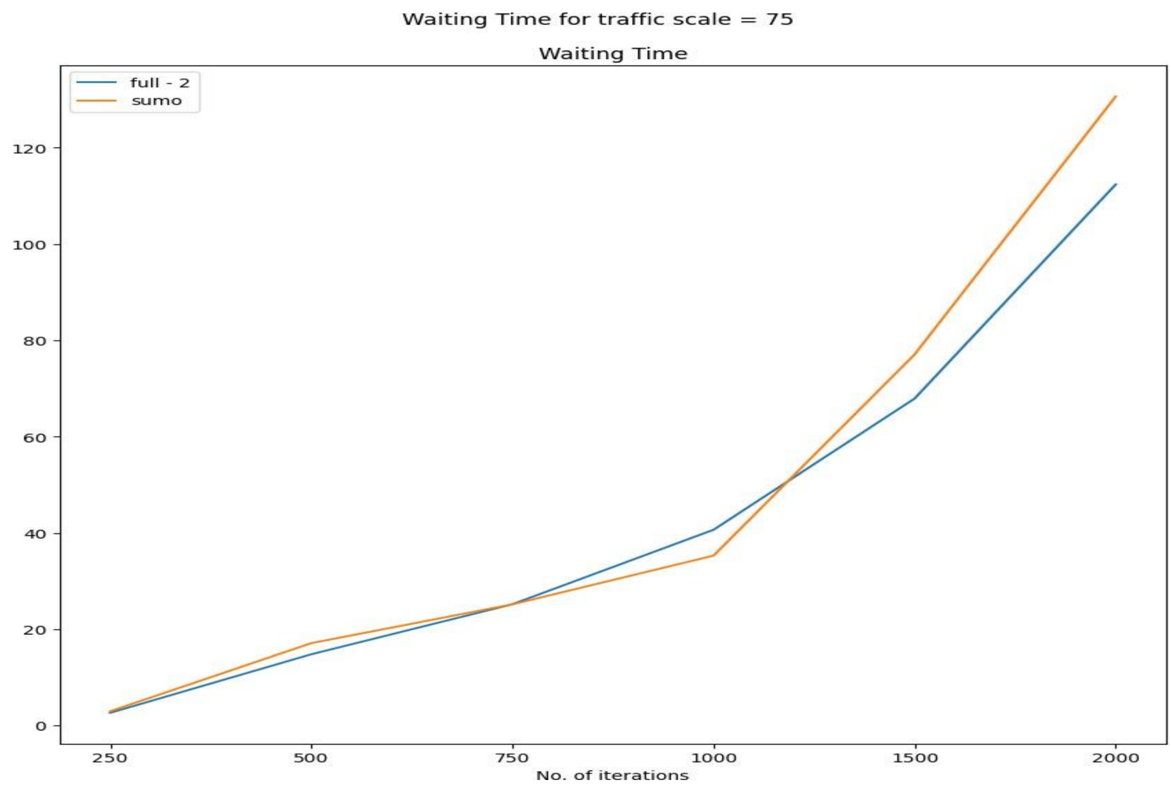


Figure 4.3.2.28

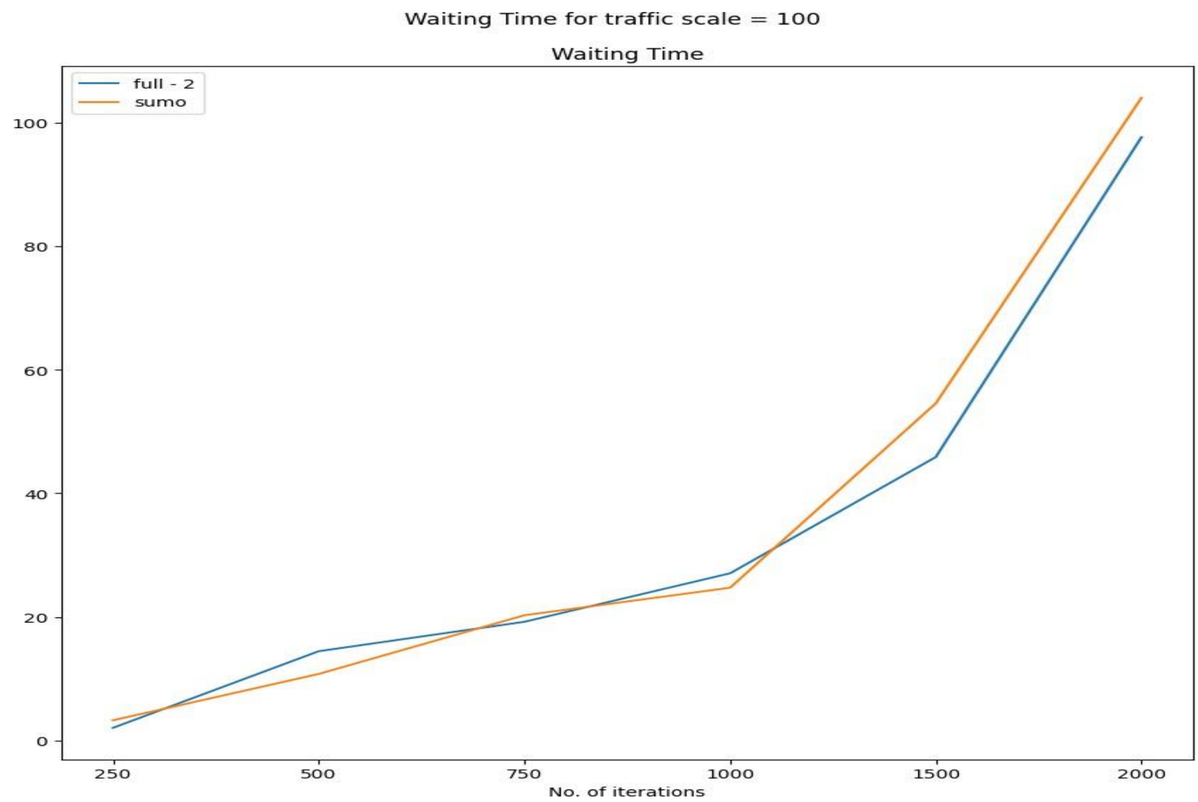


Figure 4.3.2.29

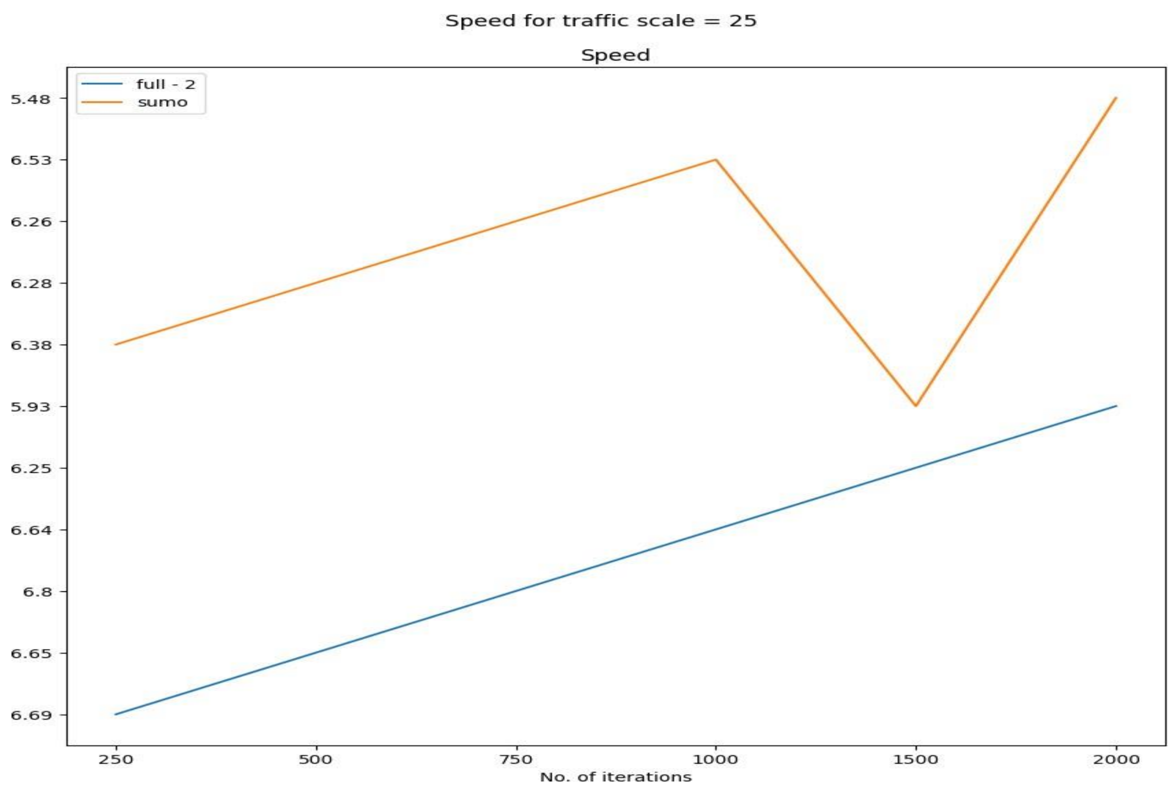


Figure 4.3.2.30

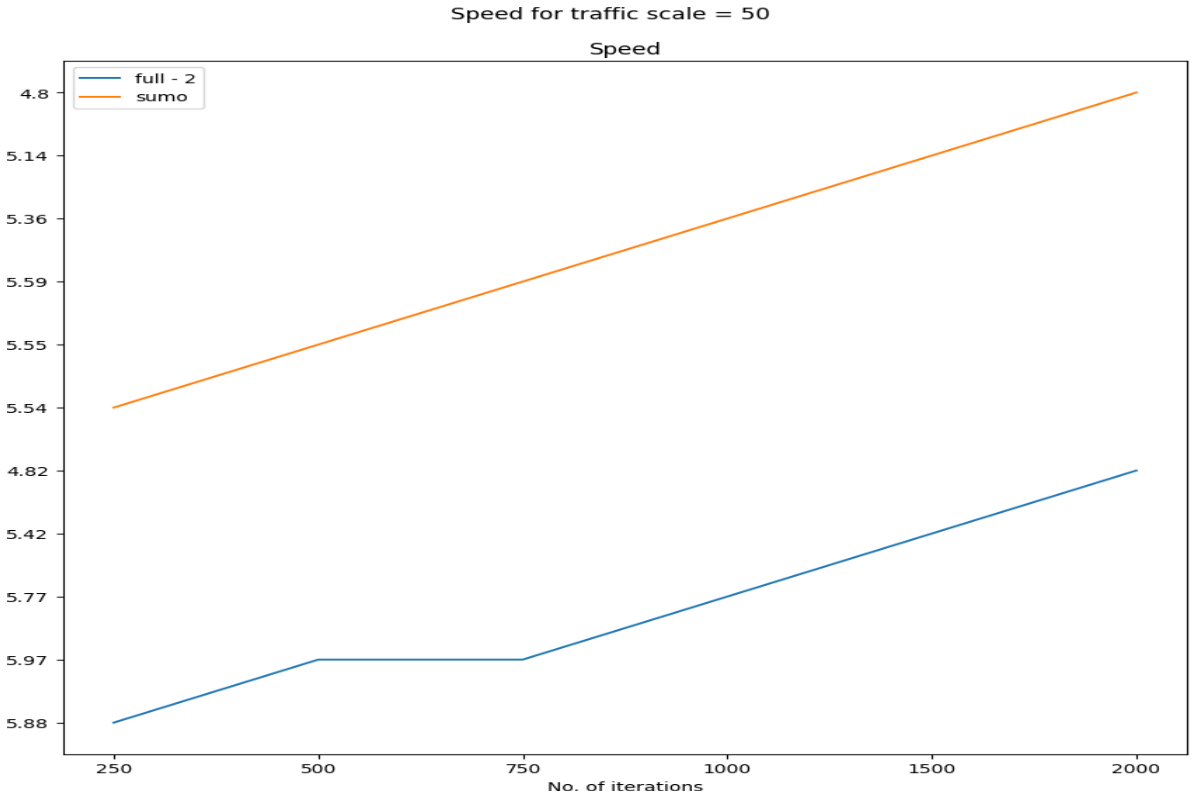


Figure 4.3.2.31

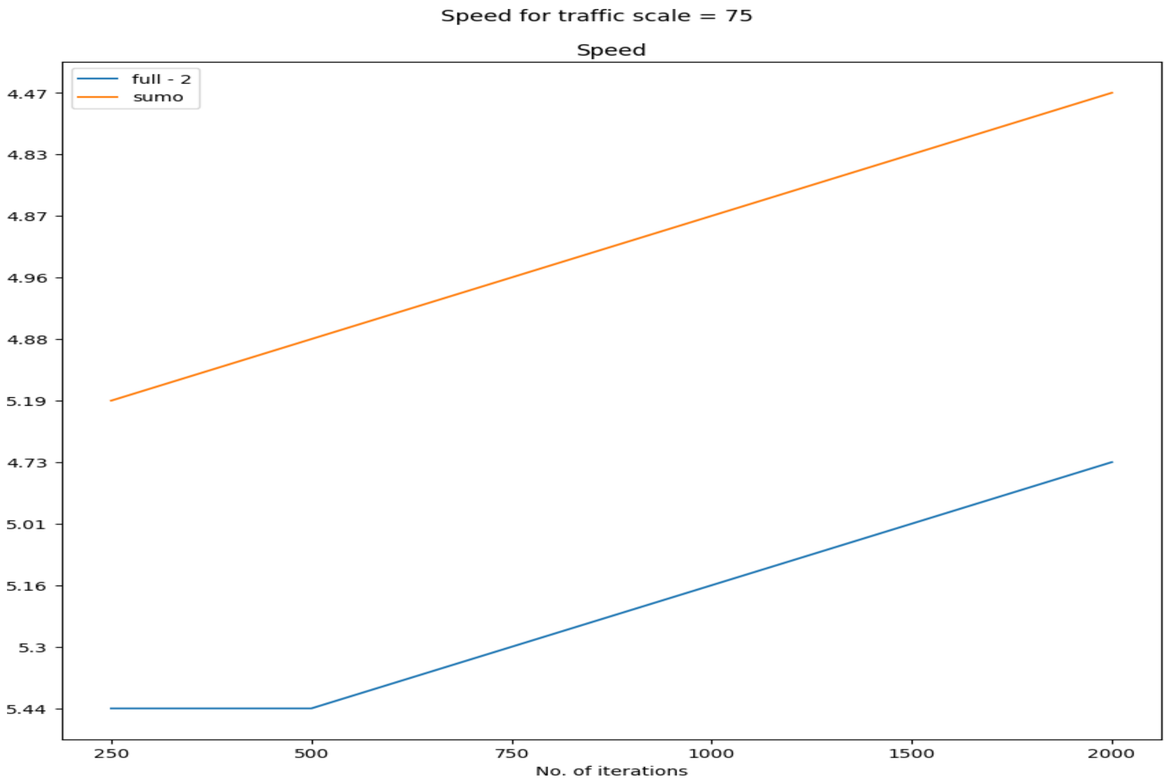


Figure 4.3.2.32

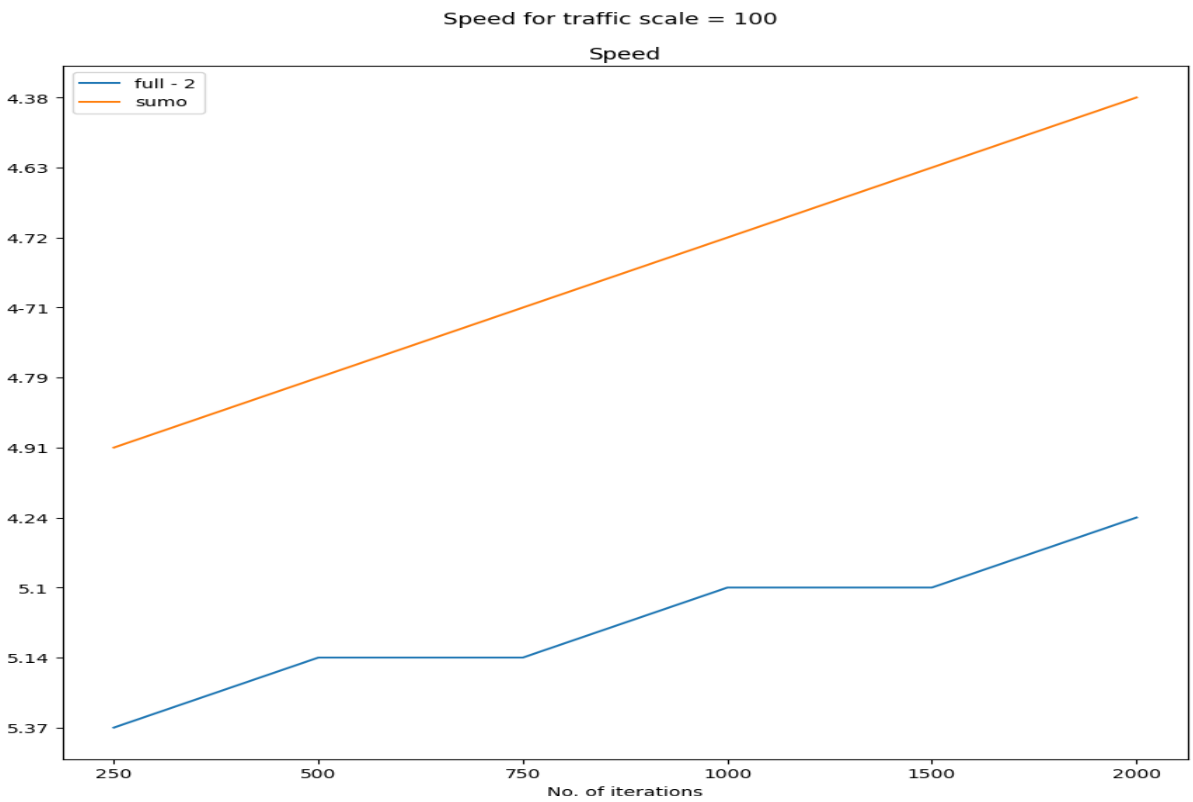


Figure 4.3.2.33

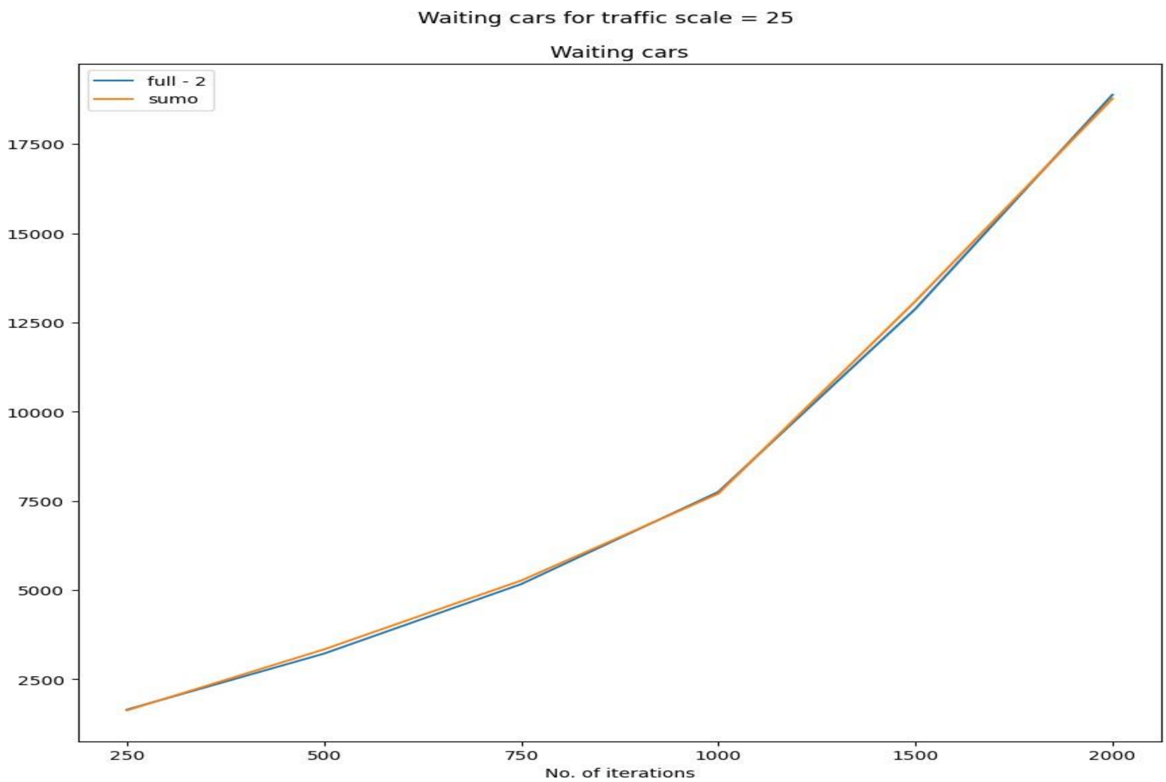


Figure 4.3.2.34

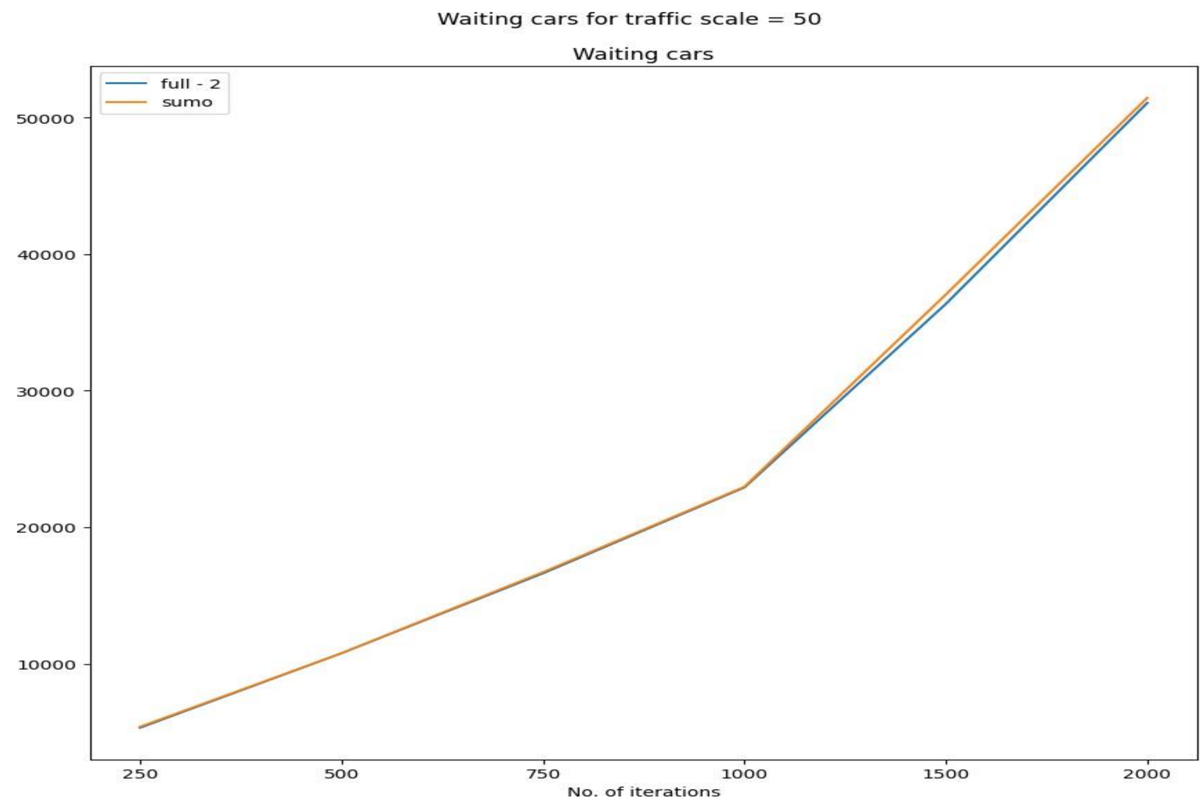


Figure 4.3.2.35

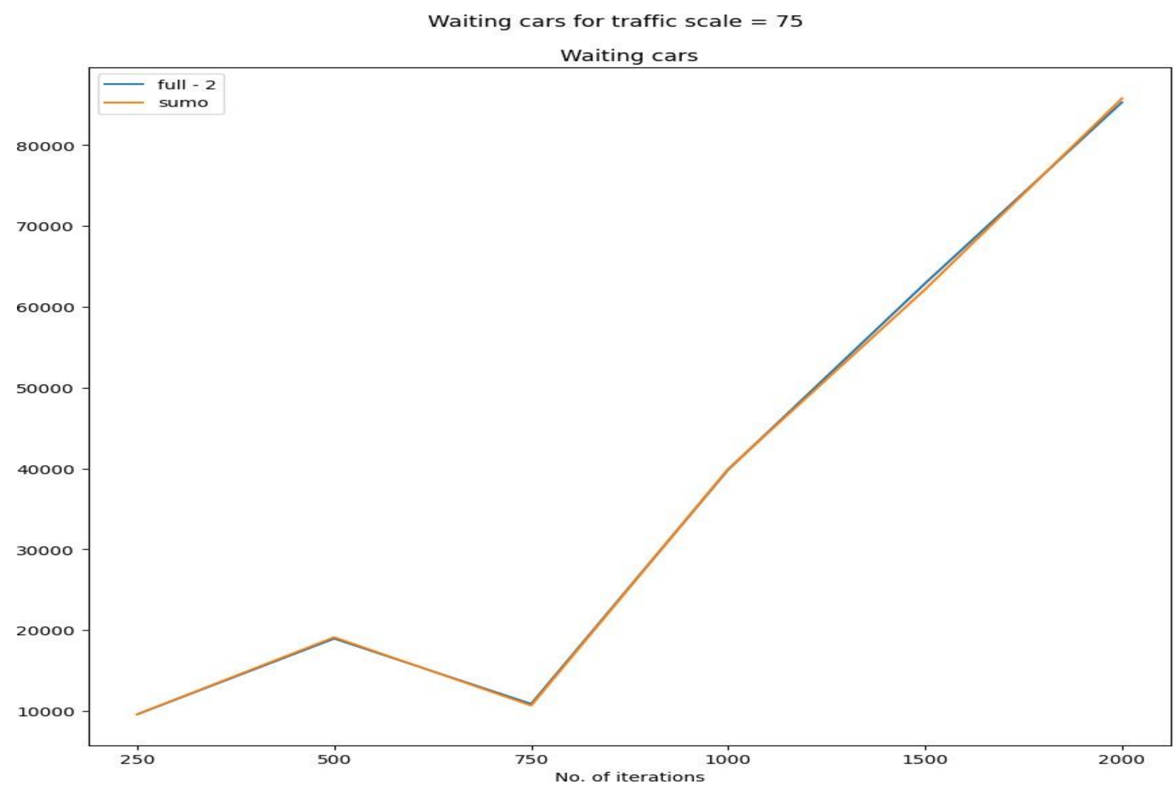


Figure 4.3.2.36

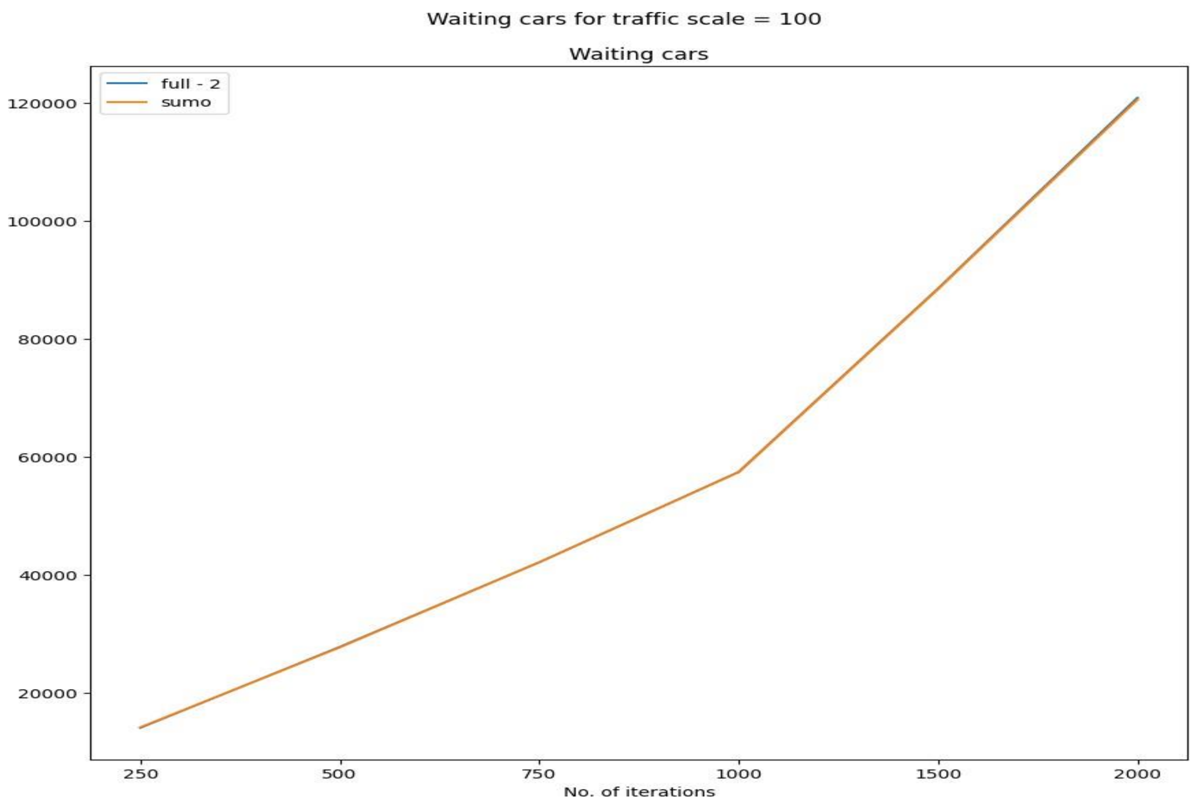


Figure 4.3.2.37

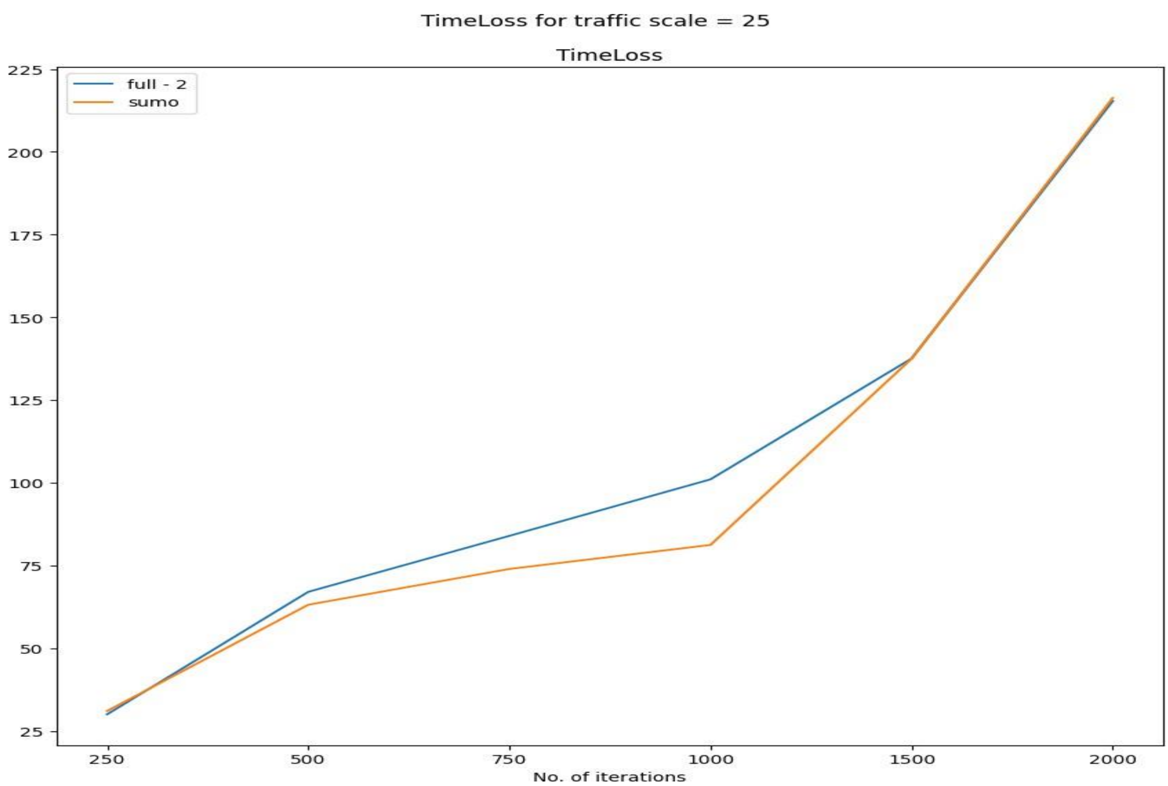


Figure 4.3.2.38

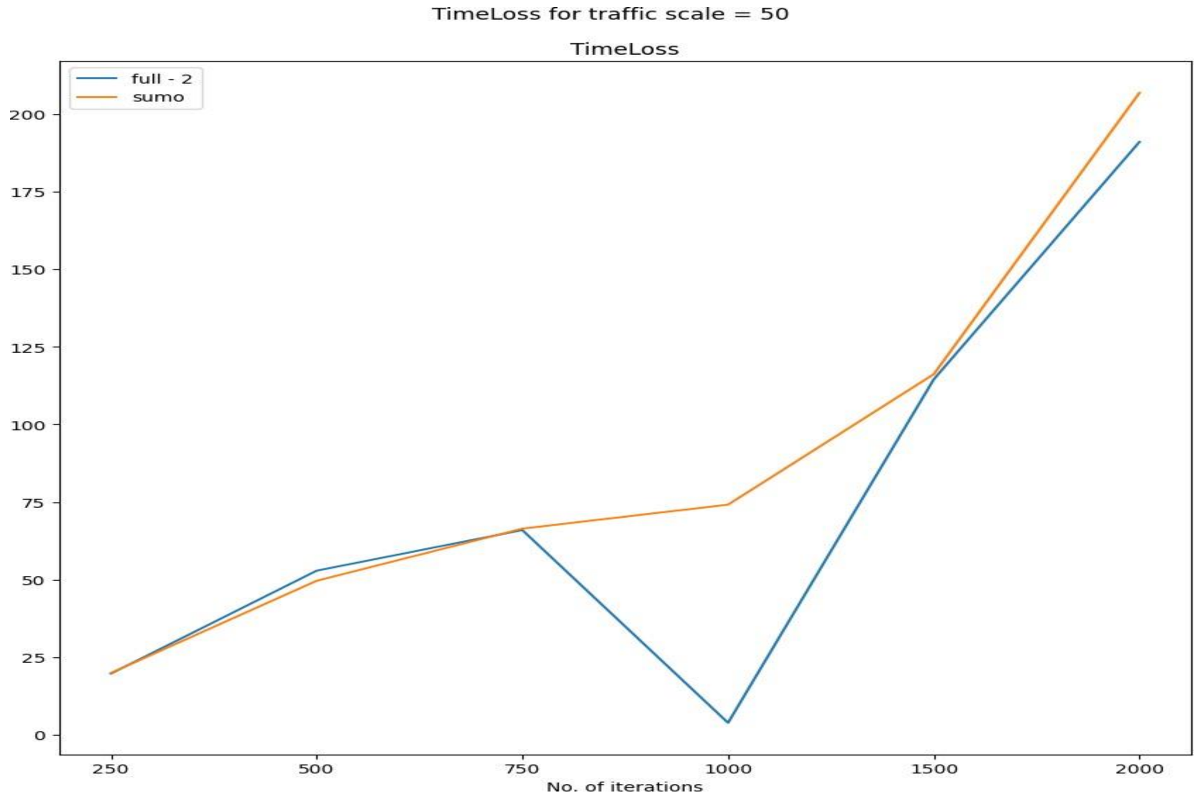


Figure 4.3.2.39

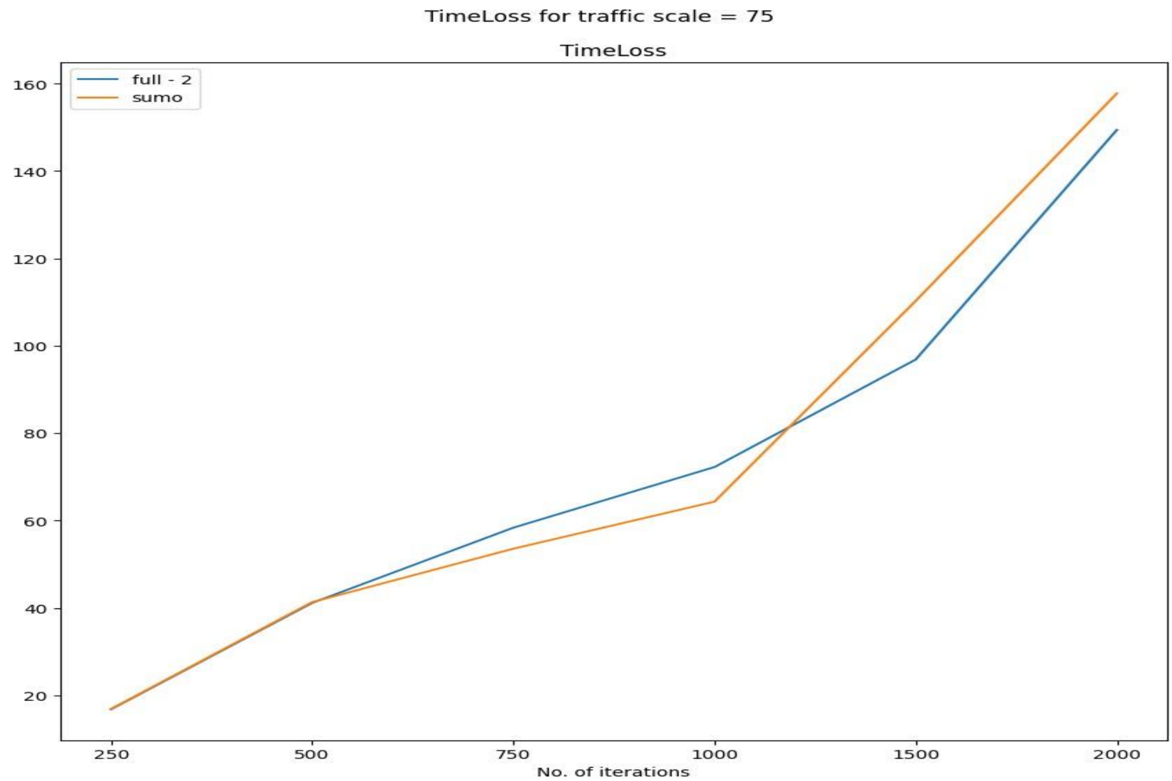


Figure 4.3.2.40

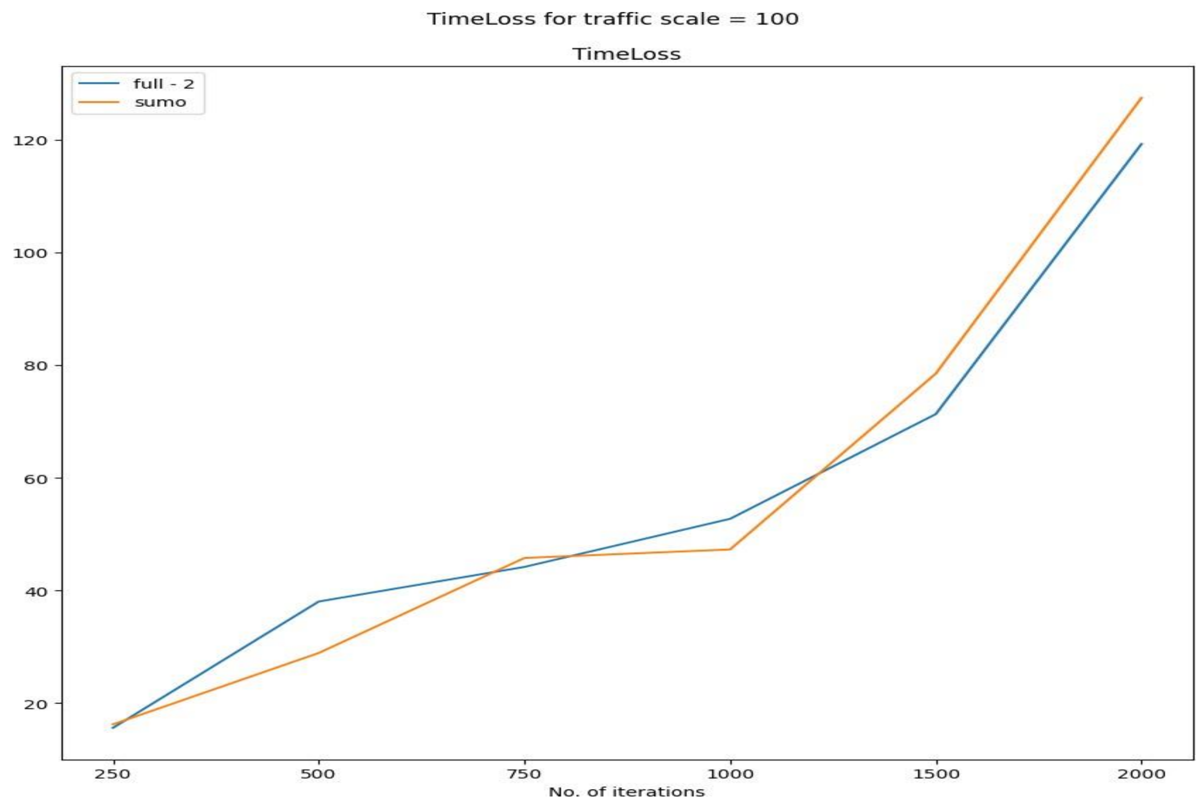


Figure 4.3.2.41

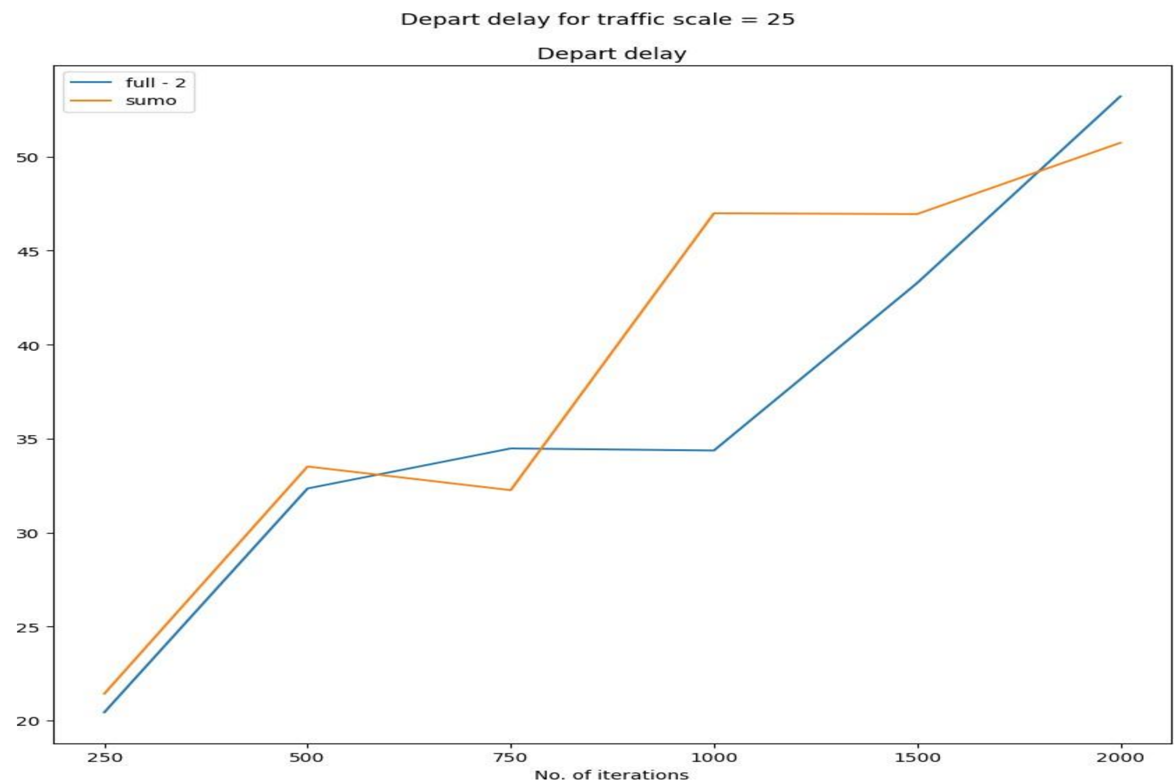


Figure 4.3.2.42

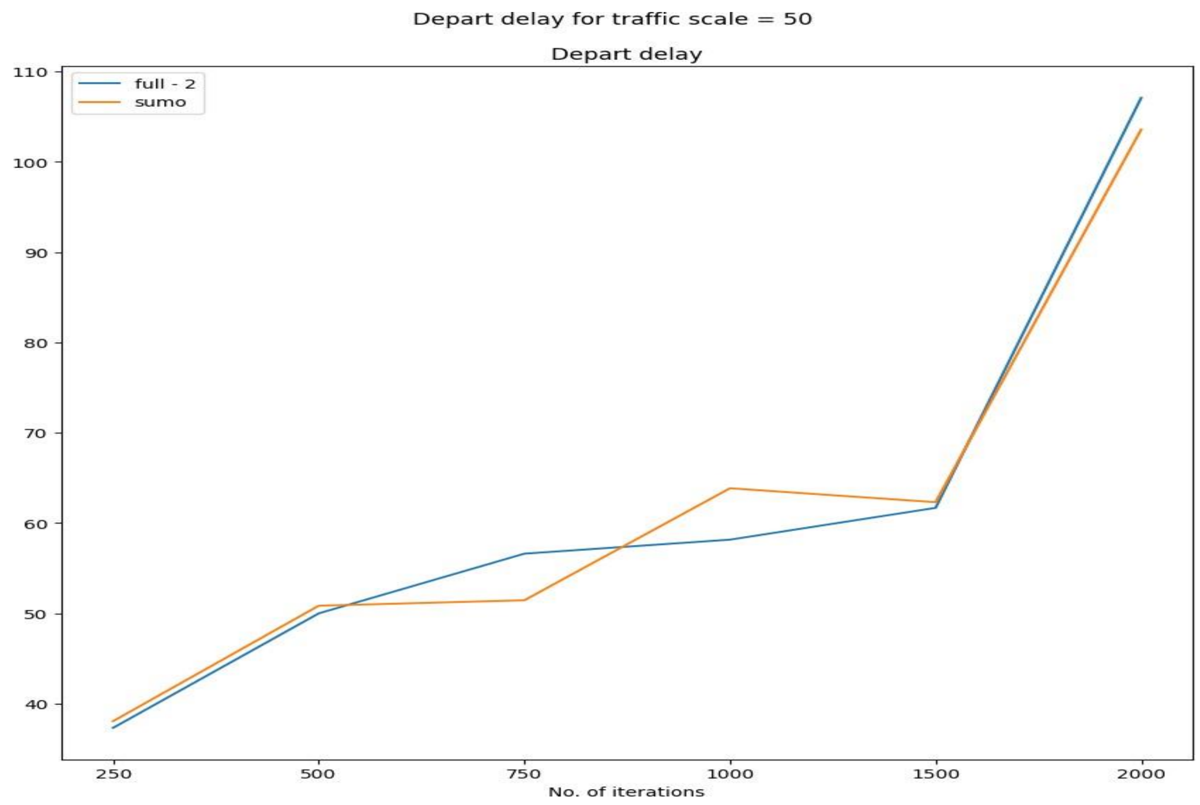


Figure 4.3.2.43

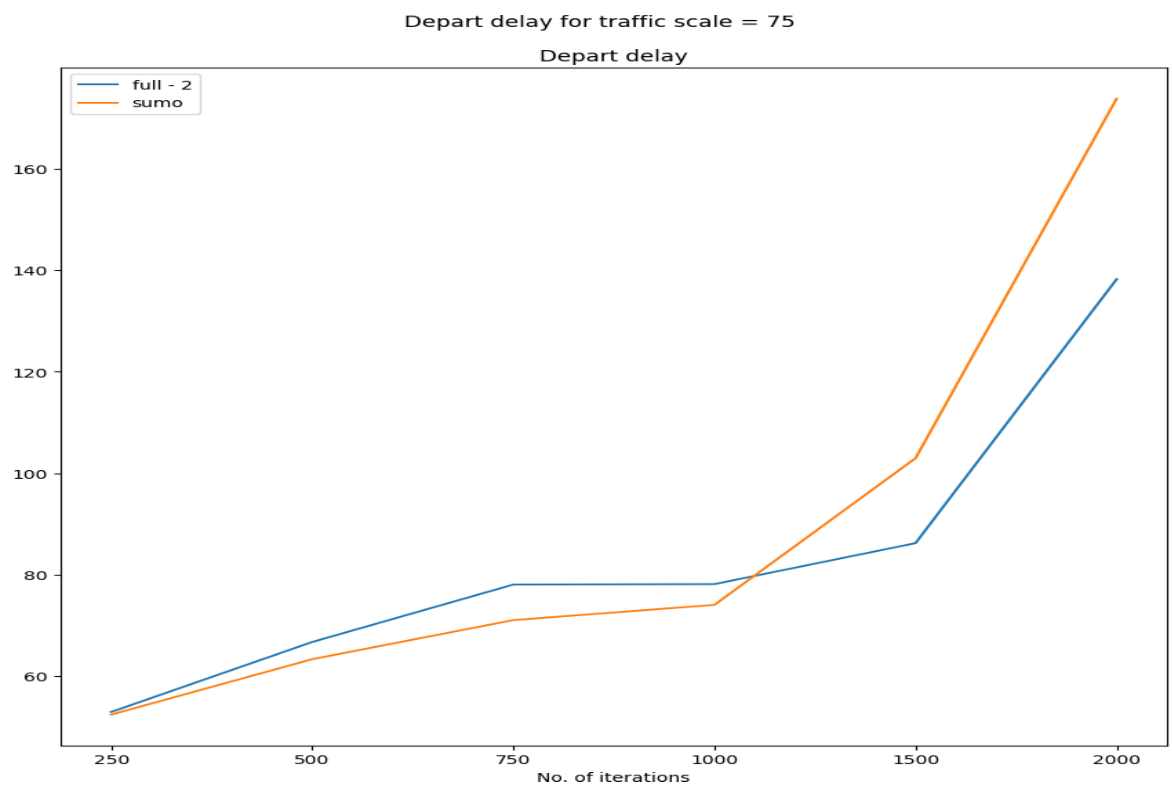


Figure 4.3.2.44

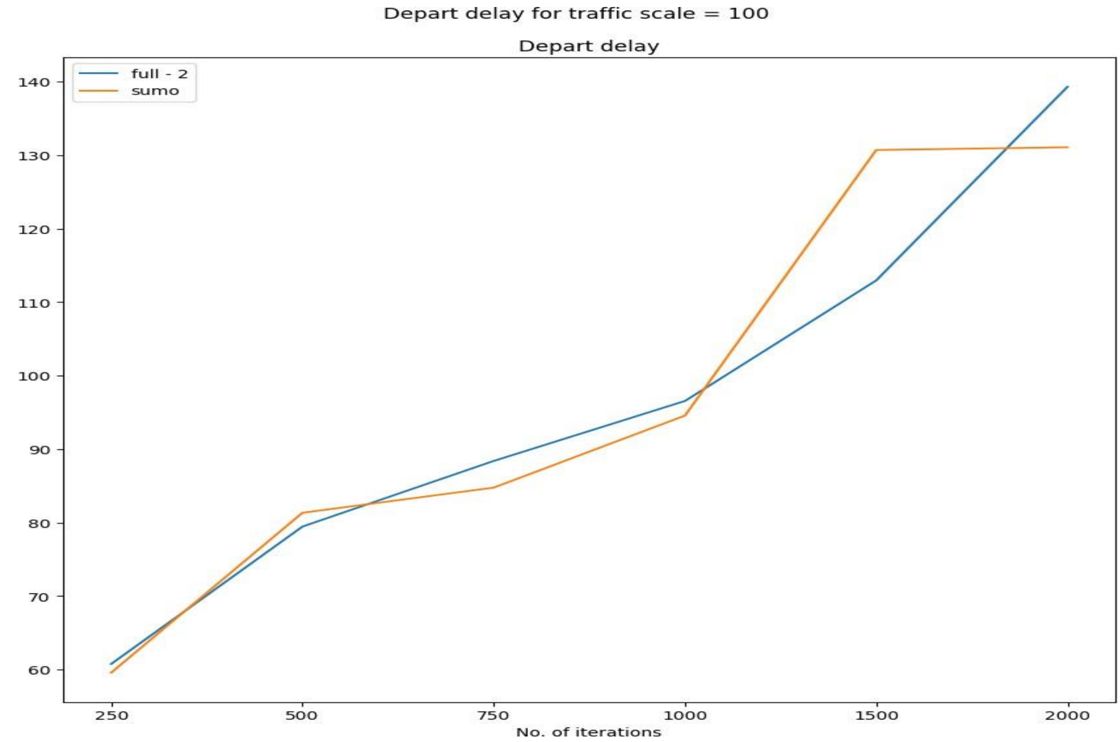


Figure 4.3.2.45

Average results for AIST dataset

- **When taking the average of the previous parameters, full DQN with second reward function performed the best.**

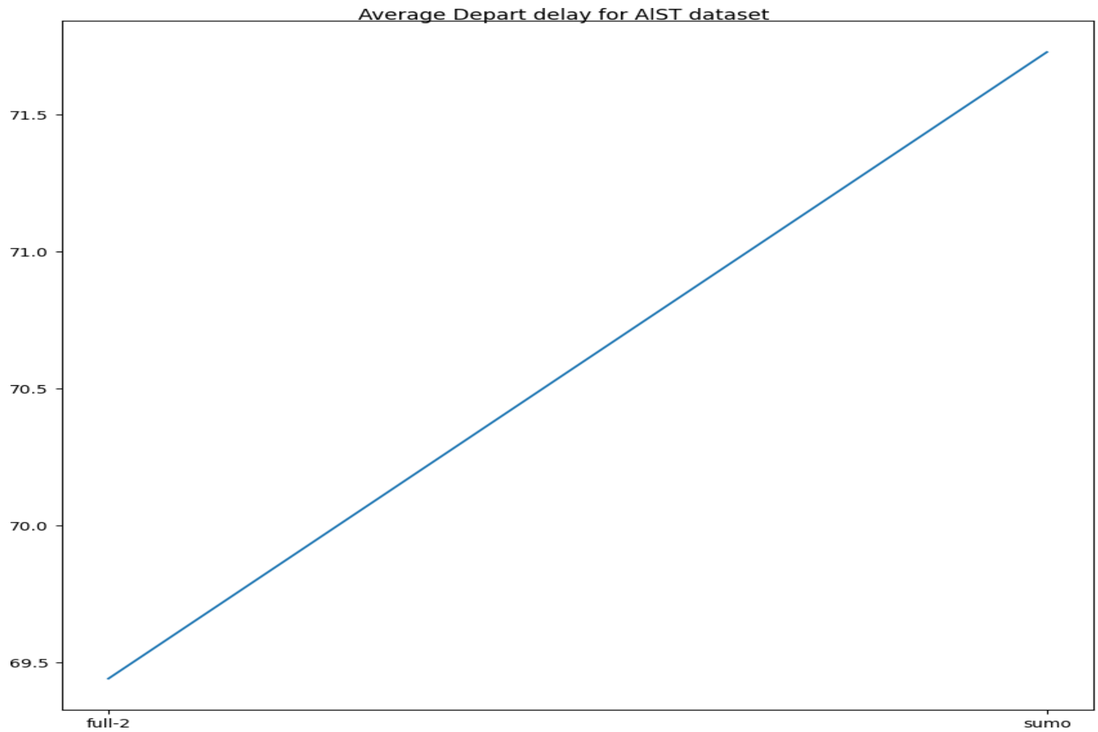


Figure 4.3.2.46

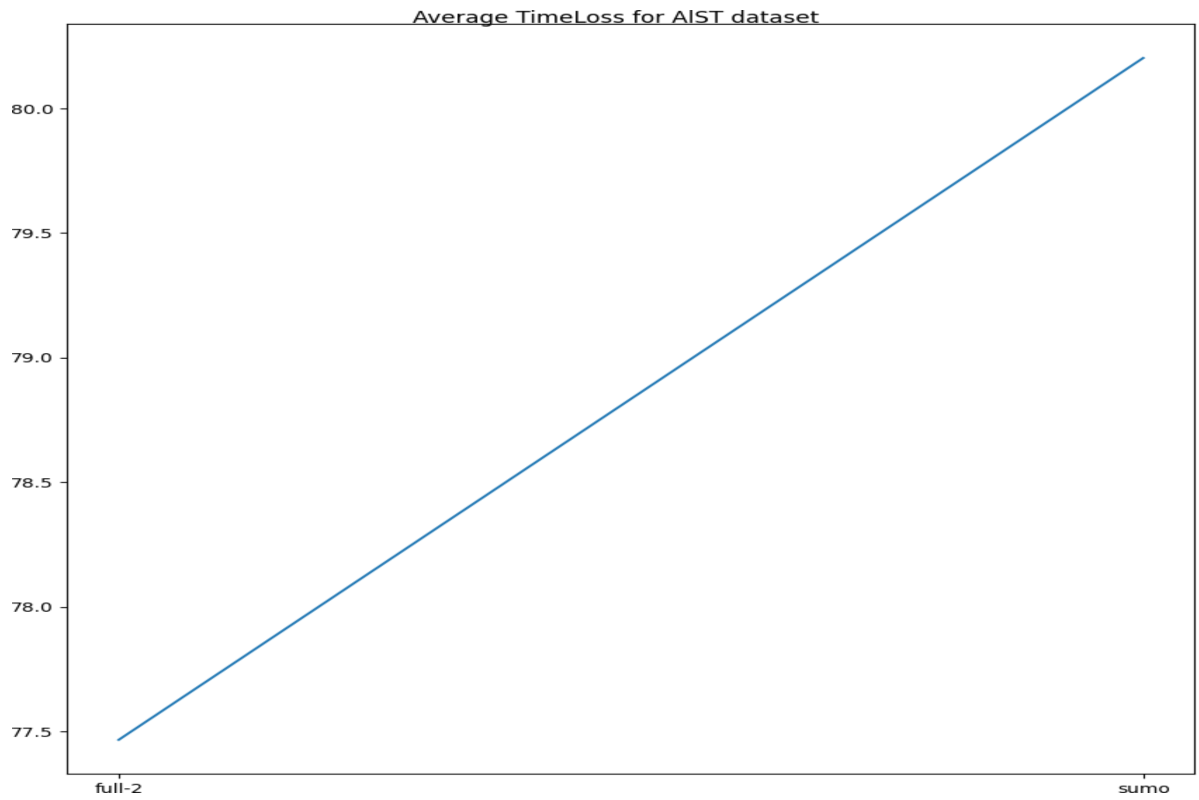


Figure 4.3.2.47

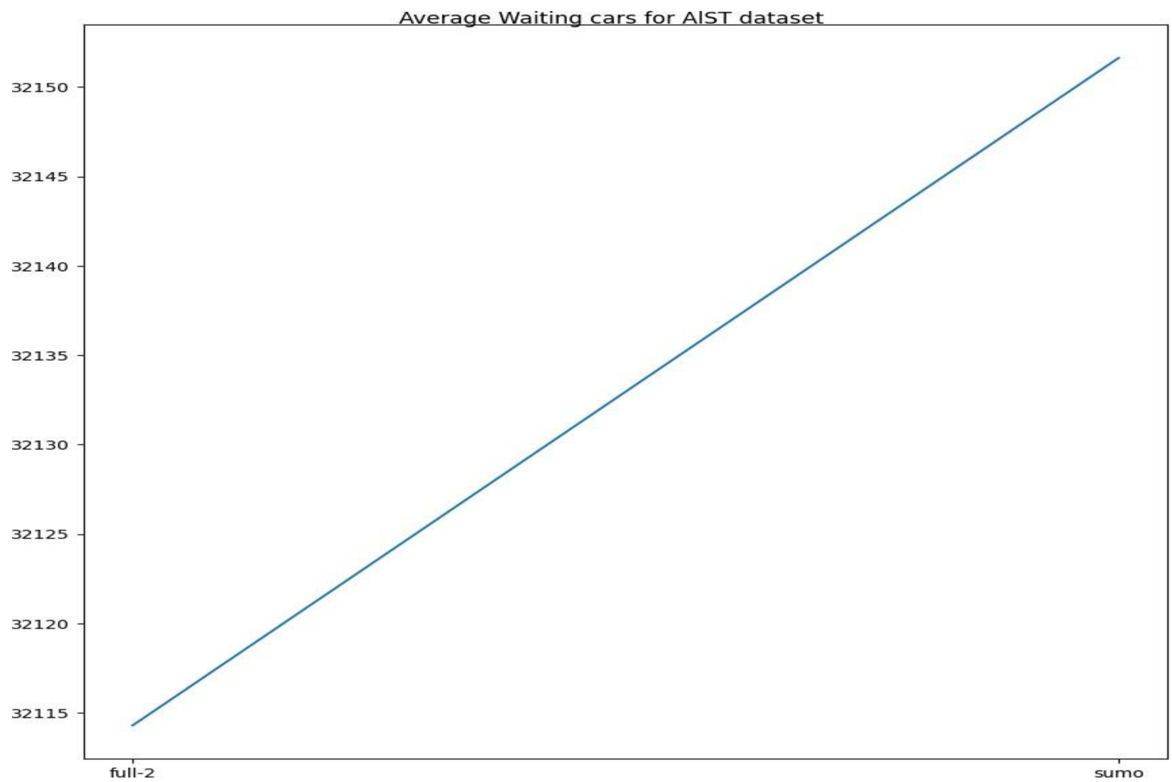


Figure 4.3.2.48

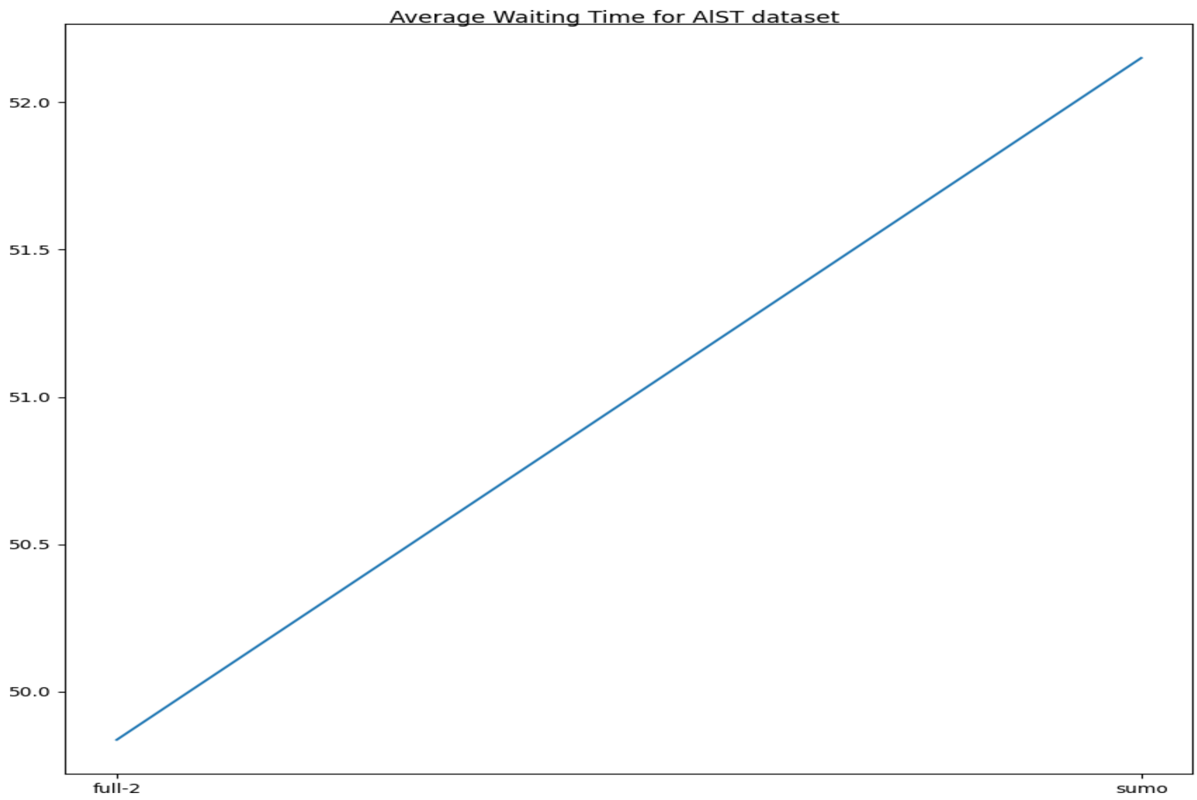


Figure 4.3.2.49

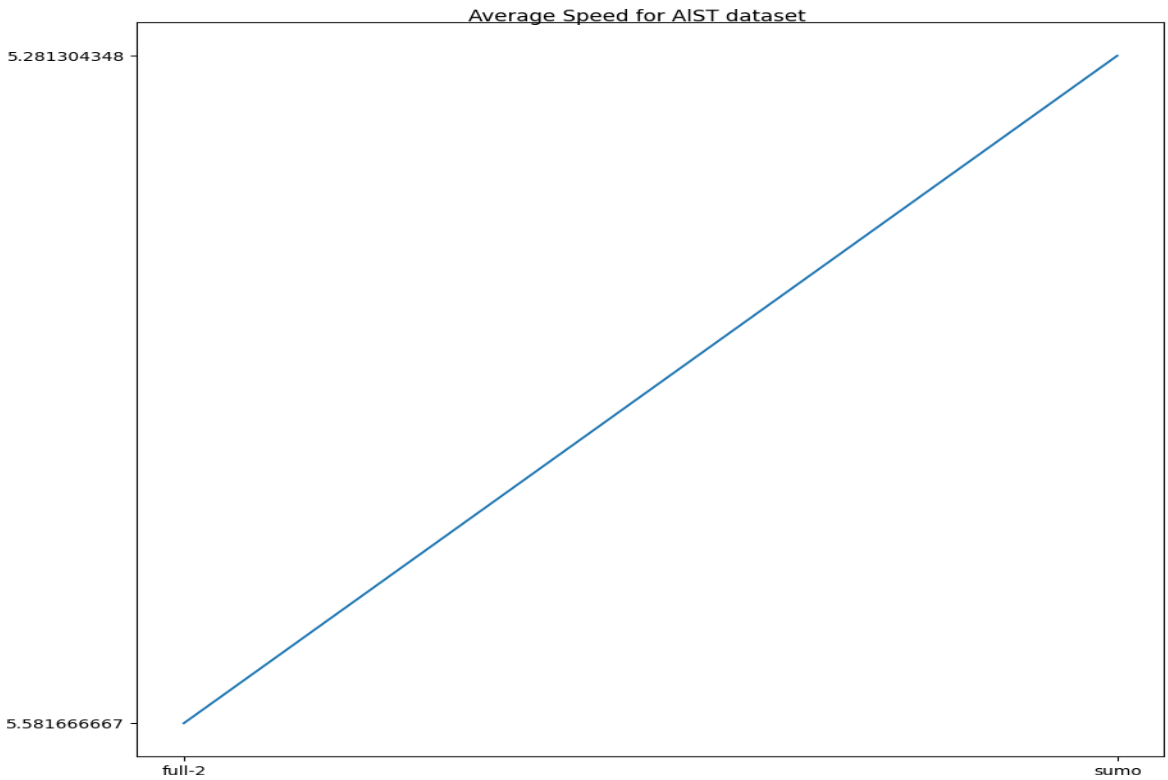


Figure 4.3.2.50

4.3.3 Discussion and results comparisons

Observations for LuST dataset

1) Full DQN with second reward function yielded the best results withing the 5 experimented models.
2) Results obtained from simple DQN with second reward function and full DQN with first reward function are approximately close to each other.
3) All our models succeeded in beating SUMO's basic simulation but with different amounts, starting with our weakest model, simple DQN with first reward function, to our strongest model, full DQN with second reward function.
4) Speed was approximately equal between all the models.

Observations for AlST dataset

1) Full DQN with second reward function managed to beat SUMO's basic simulation with Alexandria's dataset as well.
2) Results yielded from comparing the winning model we implemented, and SUMO's basic simulation seem to have been better, especially in cases of extreme traffic, than when we compared this model on Luxembourg's dataset.

Observations between the different datasets

1) It was observed that the nature of roads in Alexandria made the simulation itself harder and slower than in Luxembourg, given how greatest speed in LuST scenario was about 18 m/s, while it barely reached 7 m/s in AlST scenario.
2) It was also noticed that certain areas in Alexandria's map were extremely congested and other areas were almost completely empty, unlike in Luxembourg where we could easily notice that

traffic was distributed almost equally throughout the whole map.

3) Running the simulation using AlST dataset was much slower than running it using LuST dataset, which is why we had to decrease the simulation steps from 4000 to 2000 which took approximately 10 hours to be done.

General observations

1) The models we implemented seem to be behaving better as we increased number of iterations which makes sense because increasing number of iterations means more time to learn and more decisions to learn from.
2) An important observation is that even in cases of extreme unrealistic levels of traffic, in traffic scale 100 for instance, the models we implemented didn’t break and managed to yet beat SUMO’s basic simulation.

4.4 Conclusion

In this chapter, we went through the thorough details of implementing this intelligent traffic management system. Additionally, the results obtained from both our datasets were compared and analysed.

Chapter 5
Contribution And Future Work
5.1 Contribution of the project

After going through all the phases of this project, we can finally discuss how it contributed to solving traffic problems and why it is beneficial to anyone who desires to tackle down the same problem in the future.

1) Intelligent traffic management system where we integrate simulation with reinforcement learning to improve current traffic systems.

2) First system of this type to be tried on Alexandria streets to be able to analyze the results and focus more on the problems of traditional traffic management systems.

3) The project will be well documented to make it easier for anyone who wants to start working on the same problem in the future as this was the main problem we faced at the beginning of the project.

4) The project is going to be an open-source project to always be open for future improvements to reach better results unlike similar projects.

5.2 Future Work

This project can be the starting point for many other great projects, nevertheless, the project itself can be improved to reach better results which is something we, unfortunately, couldn’t afford given our tight schedule.

1) A stronger DQN reinforcement learning model can be implemented.
2) Another reinforcement learning models can be attempted to know how they will behave in the given environment.
3) The logic of this system can be integrated with a much more complex system where it can work with real live roads instead of a simulation, the simulation was only a step to be able to try the system, given its nature and how hard it is to try it in real life.

References

[1] S. J. Kamble and M. R. Kounte, "Machine Learning Approach on Traffic Congestion Monitoring System in Internet of Vehicles," Procedia Computer Science, vol. 171, pp. 2235–2241, 2020, doi: 10.1016/j.procs.2020.04.241.

[2] "Road technology can improve Egypt's traffic safety: EJB Representative - Daily News Egypt," dailynewsegypt.com, Oct. 31, 2020. https://dailynewsegypt.com/2020/10/31/road-technology-canimprove- (accessed Dec. 25, 2022).

[3] https://www.ibm.com/topics/machine-learning

[4]https://www.potentiaco.com/what-is-machine-learning-definition-types-applications-and-examples/

[5]https://www.analyticsvidhya.com/blog/2020/04/feature-scaling-machine-learning-normalization-standardization/

[6] https://vitalflux.com/python-improve-model-performance-using-feature-scaling/

[7] https://www.simplilearn.com/tutorials/machine-learning-tutorial/principal-component-analysis

[8] https://h2o.ai/wiki/clustering/

[9] https://towardsdatascience.com/reinforcement-learning-101-e24b50e1d292

[10] https://www.simplilearn.com/tutorials/machine-learning-tutorial/what-is-q-learning

[11] https://www.analyticsvidhya.com/blog/2019/04/introduction-deep-q-learning-python/

[12] https://medium.com/mlearning-ai/characteristics-of-rewards-in-reinforcement-learning-f5722079aef5

[13] https://www.twi-global.com/technical-knowledge/faqs/faq-what-is-simulation#WhatDoesitMean

[14] P. A. Mandhare, V. Kharat, and C. Y. Patil, “Intelligent Road Traffic Control System for Traffic Congestion A Perspective,” International Journal of Computer Sciences and Engineering, vol. 6, no. 7, pp. 908–915, Jul. 2018, doi: 10.26438/ijcse/v6i7.908915.

[15] A. Ata, M. A. Khan, S. Abbas, G. Ahmad, and A. Fatima, “MODELLING SMART ROAD TRAFFIC CONGESTION CONTROL SYSTEM USING MACHINE LEARNING TECHNIQUES,” Neural Network World, vol. 29, no. 2, pp. 99–110, 31 2019, doi: 10.14311/nnw.2019.29.008.

[16] J. H. A. et al. Josin Hippolitus A et al., “Traffic Signal Control Using Machine Learning,” International Journal of Mechanical and Production Engineering Research and Development, vol. 10, no. 3, pp. 10849–10872, 2020, doi: 10.24247/ijmperdjun20201040.

[17] 3rd International Conference on Emerging Technologies in Computer Engineering: Machine Learning and Internet of Things (ICETCE-2020), 07-08 February 2020, (IEEE Conference Record 48199)

[18] S. Reza, H. S. Oliveira, J. J. M. Machado, and J. M. R. S. Tavares, “Urban Safety: An Image-Processing and Deep-Learning-Based Intelligent Traffic Management and Control System,” Sensors, vol. 21, no. 22, p. 7705, Nov. 2021, doi: 10.3390/s21227705.

[19] A. Navarro-Espinoza et al., “Traffic Flow Prediction for Smart Traffic Lights Using Machine Learning Algorithms,” Technologies, vol. 10, no. 1, p. 5, Jan. 2022, doi: 10.3390/technologies10010005.

[20] Ritu, Ritu. (2022). Traffic Management using Machine Learning.

[21] H. R. Deekshetha, A. V. Shreyas Madhav, and A. K. Tyagi, “Traffic Prediction Using Machine Learning,” Evolutionary Computing and Mobile Sustainable Networks, pp. 969–983, 2022, doi: 10.1007/978-981-16-9605- 3_68.

[22] “Traffic Management Software Development,” Intellias. https://intellias.com/traffic-management-software-development/ (accessed Dec. 25, 2022).

[23] Advanced traffic management system software TOPXVIEW | Telegra (telegra-europe.com)

[24] “Traffic Management Software Development,” Chetu. https://www.chetu.com/transportation/traffic-management.php (accessed Dec. 25, 2022).

[25] “Advanced Traffic Management Systems by Q-Free,” Q-Free. https://www.q-free.com/solution/traffic-management-systems/ (accessed Dec. 25, 2022).

[26] https://apps.microsoft.com/store/detail/python-38/9MSSZTT1N39L

[27] https://eclipse.dev/sumo/

[28] https://sumo.dlr.de/docs/TraCI.html

[29] https://keras.io/

[30] https://oxylabs.io/blog/lxml-tutorial

[31] L. Codeca, R. Frank, S. Faye and T. Engel, *"Luxembourg SUMO Traffic (LuST) Scenario: Traffic Demand Evaluation"* in IEEE Intelligent Transportation Systems Magazine, vol. 9, no. 2, pp. 52-63, Summer 2017. DOI: 10.1109/MITS.2017.2666585

[32] A. M. Law, Simulation modeling and analysis. New York, Ny: Mcgraw-Hill Education, 2015.

[33] https://towardsdatascience.com/what-is-exponential-distribution-7bdd08590e2a#:~:text=Exponential%20distribution%20is%20often%20used,purchase%20in%20your%20store%20(success)

[34] E. M. Grua and M. Hoogendoorn, Exploring Clustering Techniques for Effective Reinforcement Learning based Personalization for Health and Wellbeing. 2018. doi: 10.1109/ssci.2018.8628621.

[35] https://jaromiru.com/2016/10/03/lets-make-a-dqn-implementation/

[36] https://keras.io/api/layers/core_layers/dense/

[37] https://towardsdatascience.com/https-medium-com-dashingaditya-rakhecha-understanding-learning-rate-dd5da26bb6de

[38] https://optimization.cbe.cornell.edu/index.php?title=RMSProp#:~:text=RMSProp%2C%20root%20mean%20squared%20propagation,of%20gradients%20descent%20and%20RProp.

[39] Stevens, Matt and Christopher Yeh. “Reinforcement Learning for Traffic Optimization.” (2016).

[40] Genders, Wade and Saiedeh Navabzadeh Razavi. “Using a Deep Reinforcement Learning Agent for Traffic Signal Control.” *ArXiv* abs/1611.01142 (2016): n. pag.

[41] https://github.com/LucasAlegre/sumo-rl/blob/main/sumo_rl/environment/env.py

[42] https://sumo.dlr.de/docs/Basics/Basic_Computer_Skills.html

[43] https://sumo.dlr.de/docs/Simulation/Output/StatisticOutput.html#:~:text=Statistic%20output%20is%20activated%20by,sumocfg%20file

Appendix A

Appendix A.1

Summary for Intelligent Road traffic control system for traffic congestion: a perspective [14]

- **What is your take-away message from this paper?**

  - With the help of Intelligent Transportation Systems (ITS), several attempts were made to automate the traffic lights based on the density of vehicles on the road. This paper reviews different sensor frameworks by analyzing the pros and cons of each in cost, reliability, accuracy, efficiency, and maintenance overhead.

- **What is the motivation for this work (both people's problems and technical problem), and its distillation into a research question? Why doesn't the people problem have a trivial solution? What are the previous solutions and why are they inadequate?**

  - The main reason can be defined as, to increase in the population which in turn has caused a rise in the number of vehicles on the road.
  - Also, there are several other issues for traffic congestion like insufficient infrastructure, ineffective management of capacity (i.e., poor traffic timing), work zones, special events, emergencies, unconstraint demands etc.

- **What is the proposed solution (hypothesis, idea, design)? Why is it believed it will work? How does it represent an improvement? How is the solution achieved?**

  - ITS is defined as the application of advanced sensors, computer, electronics and telecommunication technologies and management strategies in an integrated way to improve the safety and efficiency of the transportation system.
  - The major goal of ITS is to evaluate, develop, analyze, and integrate the sensors, information communications technologies, and concept to make efficient traffic flow to improve environmental quality, save energy, conserve time such that enhance the comfort of drivers, pedestrian, and other traffic groups.

- **What is the author's evaluation of the solution? What logic, argument, evidence, artifacts (e.g., a proof-of-concept system), or experiments are presented in support of the idea?**

  - Most video analytics systems on traffic congestion focus on counting and doing classifications for more statistics. The vehicle identification is used with self-adaptive windows to estimate the mean travel timeunder traffic demand and supply uncertainty.
  - Demonstrated motion-based tracking with trajectory analysis method is to improve intersection behavior analysis for accurate turning movement count at the intersection.
  - Image processing algorithm is used to estimate traffic density using cameras. Based on analysis of traffic images from live traffic evidence of congestion collapse which lasts for the extended time period.
  - There are many lighting and weather conditions effects on vision-based systems. Such system must adopt all these lighting conditions.

- **What is your analysis of the identified problem, idea, and evaluation? Is this a good idea? What flaws do you perceive in the work? What are the most interesting or controversial ideas? For work that has practical implications, ask whether this will work, who would want it, what it will take to give it to them, and when might it become a reality?**
  - This paper is very good at analyzing each and every side of the problem and all its possible solutions.
  - It specifies many sections to discuss previous systems that were actually implemented in real life and how effective they were.
  - Another important section was all the possible obstacles that can meet whoever tries to approach this problem.
- **What are the paper's contributions (author's and your opinion)? Ideas, methods, software, experimental results, experimental techniques...?**
  - Advanced Traveler Information System (ATIS): It implements a broad range of technologies, such as internet, telephones, cellular phones, television, radio, etc. to help travelers and drivers in making informed decisions regarding trip departures, optimum routes, and available modes of travel.
  - Advanced Traffic Management System (ATMS): It is used by traffic police department and traffic regulation authorities as a tool to manage and control traffic by monitoring the flow of traffic and making appropriate decisions in a timely manner. Traffic management systems optimize the movement of vehicles by using real-time information to interfere with and adjust controls such as traffic signals to improve traffic flow.
  - Advanced Public Transportation System (APTS): It is concerned with increasing operational efficiency of all public transportation modes and increasing condition by making the transportation system more reliable. With the help of APTS the way public transportation systems function is transformed and the nature of the transportation services that can be offered by public transportation systems are changed.
  - Emergency Management System (EMS): It is the newest research field in the intelligent transportation system. EMS is mainly concerned with the application of different intelligent transportation system technologies to develop a transport system which can provide help in emergency conditions.
- **What are future directions for this research (author's and yours, perhaps driven by shortcomings or other critiques)?**
  - In the future, the systems are often further improved using more factors that affect traffic management using other methods like deep learning, artificial neural network, and even big data
  - Increasing the accuracy of forecasting methods which in return increases the accuracy
- **What questions are you left with? What questions would you like to raise in an open discussion of the work (review interesting and controversial points, above)? What do you find difficult to understand? List as many as you can**
  - Would any new systems actually be able to beat the previously implemented ones?

Appendix A.2

Summary for modelling smart road traffic congestion control system using machine learning techniques [15].

- **What is your take-away message from this paper?**

  - The smart transportation system has multiple applications like enhancing road safety and security, monitoring of traffic flow, saturation detection, reduce travel time, alternate routing, route weather condition, reducing pollution and greenhouse gas emission, efficient fuel consumption, emergency management, noise monitoring, etc.

- **What is the motivation for this work (both people's problem and technical problem), and its distillation into a research question? Why doesn't the people problem have a trivial solution? What are the previous solutions and why are they inadequate?**

  - The main reason can be defined as, to increase in the population which in turn has caused a rise in the number of vehicles on the road.
  - Also, there are several other issues for traffic congestion like insufficient infrastructure, ineffective management of capacity (i.e., poor traffic timing), work zones, special events, emergencies, unconstraint demands etc.

- **What is the proposed solution (hypothesis, idea, design)? Why is it believed it will work? How does it represent an improvement? How is the solution achieved?**

  - This paper aims to provide a mechanism to predict the traffic congestion with the help of Artificial Neural Networks (ANN) using Artificial Back Propagation Neural Networks (MSR2C-ABPNN) for road traffic increase transparency, availability and efficiency in services offered to the citizens.

- **What is the author's evaluation of the solution? What logic, argument, evidence, artifacts (e.g., a proof-of-concept system), or experiments are presented in support of the idea?**

  - The accuracy of the neural network depends upon different iterations which include three different parts of the neural network structure, and it's working, i.e., input layer, hidden layers, and an output layer. This paper concludes the accuracy of the neural network when there is one hidden layer including many neurons in the hidden and input layer is equal
  - (SERSU) Scalable Enhanced Roadside Unit, having contamination recognition, the arrangement of versatile traffic control and climate data framework, SERSU utilized radio recurrence, and remote correspondence. SERSU modules were put on the roadsides at various interims, catching created sensor motions by another module of vehicle sensors
  - Proposed a comparative framework which controls the traffic lights progressively as indicated by the mass of traffic utilizing the normal speed of vehicles? In the proposed framework, every vehicle is fitted out with locally available gadgets (OBD) which gain vehicular speed information process and disperse information to unify server utilizing Zigbee convention

- **What is your analysis of the identified problem, idea, and evaluation? Is this a good idea? What flaws do you perceive in the work? What are the most interesting or controversial ideas? For work that has practical implications, ask whether this will work, who would want it, what it will take to give it to them, and when might it become a reality?**

  - This paper made use of many different approaches to solve the congestion problem, many of which are different neural networks and machine learning algorithms which they shared their results to.

- **What are the paper's contributions (author's and your opinion)? Ideas, methods, software, experimental results, experimental techniques...?**

  - Multi-Layer Perceptron (MLP) was the best model for prediction. With the help of a neural network, the mapping can be done effortlessly for input and output to calculate and predict future trends, Multilayerperceptron consists of an input and output layer
  - Back Propagation Algorithm was used to train multilayer perceptron, while the weather information and traffic flows were used to predict congestion
  - This paper predicts the traffic flow by using artificial neural network model named as Back Propagation (BP) Neural network. It is being used for gaining maximum accuracy in the prediction of the traffic congestion with a defined structure of a neural network having an input layer, hidden layer, and output layer — the structure composed of the forward propagation and back propagation of error.
  - In the forward propagation, information is being processed from the input layer to the hidden layer, then finally transmitted to output layer and if the output layer cannot accept then it sends back to the process of back propagation error, in which different values of weights are adjusted in such a way to minimize error and again transferred to the forward propagation

- **What are future directions for this research (author's and yours, perhaps driven by shortcomings or other critiques)?**

  - It is detected that the smart transportation system has a significant impact on smart city whereas the traditional systems do not have the flexibility to automatically control the adjacent signals timers tominimize the congestion in traffic.
  - Therefore, a Smart traffic congestion control system was presented to manage the traffic signal timer automatically by using machine learning techniques.
  - Different sensors implemented on different adjacent signals that collect, share traffic data, and sent to acontroller through IoT enabled devices.
  - In soft real-time systems like smart traffic, time is critical. Therefore, if the data received by the signal sensors is delayed or there is too much noise, in that sense proposed (MSR2C- ABPNN) solution performance may be affected.

- **What questions are you left with? What questions would you like to raise in an open discussion of the work (review interesting and controversial points, above)? What do you find difficult to understand? List as many as you can**

  - It made me curious to try this new neural network they tried out, (MSR2C-ABPNN).

Appendix A.3

Summary for traffic signal control using machine learning [16]

- **What is your take-away message from this paper?**
  - This project uses neural networks and reinforced learning to create an intelligent traffic signal controller that takes into consideration the characteristics of urban traffic flows. Q-learning network is proposed to extract the information from the state space in order to derive the optimal signal control policy and to perform with large state space which consists of real-time vehicle position and speed. Four traffic patterns are used in SUMO simulation.
- **What is the motivation for this work (both people's problem and technical problem), and its distillation into a research question? Why doesn't the people problem have a trivial solution? What are the previous solutions and why are they inadequate?**
  - Traffic signals have been a long-standing topic in urban traffic control. Ineffective and inflexible traffic control at urban intersections can often lead to an obstruction in traffic flow and will lead to congestion of traffic.
  - Every year almost 253 million vehicles are manufactured and sold to customers. As the roads cannot expand, we need to manage the number of vehicles on the road. Providing real-time traffic control based on the density of vehicles could be the best solution so far.
  - The main purpose of controlling traffic signals is to simultaneously increase the intersection capacity, decrease delays and guarantee the safety of people and vehicles. It can also reduce the wasted time while staying at signals. The main purpose for the accumulation of traffic is due to fixed-time traffic control, every part of the road is given an equal time for the movement of vehicles so that the traffic congestion does not reduce the amount of traffic.
  - Conventional traffic signal timing plan management using statistical information of traffic lacks the ability to rapidly adapt into the dynamic traffic flow. Thus, we need to develop intelligent traffic signal timing plan management to continuously learn from the dynamic traffic environment for adaptability.
  - The problem with the existing traffic system is that for every minute the vehicles at the 4-way cross-roads will be large in number resulting in a lot of congestion. Even though there are no vehicles at a particular side, the traffic signal will still glow green for a fixed time.
- **What is the proposed solution (hypothesis, idea, design)? Why is it believed it will work? How does it represent an improvement? How is the solution achieved?**
  - With recent advances in machine learning, especially reinforcement learning (RL), traffic signal control represents a promising solution to tackle this problem.
  - Adjacent traffic light intersections will work independently and yet cooperate with each other to ensure the fluency of the traffic flow within traffic network. The experimental results show that the Q-Learning algorithm can learn from the dynamic traffic flow and optimize the traffic flow.
  - Density-based traffic control system using reinforced learning. Priority is based when there is a huge amount of traffic and then switching it back to the normal sequence after there is less amount of traffic.

- **What is the author's evaluation of the solution? What logic, argument, evidence, artifacts (e.g., a proof-of-concept system), or experiments are presented in support of the idea?**
  - The Performance of Reinforced Learning Approach is evaluated in three areas: Training convergence, Comparison with benchmarks. Generalization across different traffic patterns
  - Average queue length and average wait time are used as a performance criterion. At the beginning of thetraining process, the Q-learning network explores the control policy by selecting random action with high probability.
  - As training goes on, the Q-learning network gets positive or negative rewards depending upon the weather corrective action has been taken to reduce the number of halting vehicles.
  - The Q-learning Network gradually exploits the control policy and reduces the average queue length and average wait time. Finally, the Q-learning Network succeeds the stabilized performance with respect tothe average queue length and the average wait time.

- **What is your analysis of the identified problem, idea, and evaluation? Is this a good idea? What flaws do you perceive in the work? What are the most interesting or controversial ideas? For work that has practical implications, ask whether this will work, who would want it, what it will take to give it to them, and when might it become a reality?**
  - I personally think that the proposed system is actually good and may actually be a good solution to the proposed problem with constant improvement to adapt to the constant traffic variations.
  - What made this solution really good is mixing machine learning with simulation to be able to actually see a near real-time model to what would happen if this system were implemented in real life.
- **What are the paper's contributions (author's and your opinion)? Ideas, methods, software,experimental results, experimental techniques...?**
  - This project designs a system which uses deep neural network algorithm, which will provide intelligence to the current traffic control system present at a four-way junction.
  - It aimed to replace the conventional timer traffic control system with our artificial intelligence system. Nowadays most cities are equipped with CCTV cameras, the idea is to collect the live video from them and detect the number of vehicles on each lane and feed the data into another machine learning algorithm which according changes the light phase of the signal.
  - This system mainly aims to increase the traffic efficiency by increasing vehicle flow which will reduce waiting time.
  - One of neural networks chosen to design an algorithm to detect vehicles instead of the fully connected neural network because it will be impossible to connect neurons to all the neurons of the previous volume. The network is designed by adding layers who are used in feature extraction: Convolution layer - Pooling layer – Activation - Fully connected layer.

- Experience replay allows a reinforcement learning agent to recollect information from its memory and reprocess these past experiences. It uses various methods of sampling to help group up elements of its memory. It improves the learning rate and performance of the agent. The dataset that comprises of the experiences for each time step is known as the replay memory which consists of every sample collectedduring the training session.
- Simulation of Urban Mobility. This platform is widely used for traffic simulations so that any changes to the traffic setup can be evaluated and implemented. This model uses a four-way intersection to determine the efficiency of this model in different conditions.

• **What are future directions for this research (author's and yours, perhaps driven by shortcomings or other critiques)?**

- There is scope for an improvement of the Q-learning network by changing the performance in convergence and stability. Advanced techniques such as dulling network and double Q- learning network can be employed.
- Future works at investigating what would be the implications of introducing multiple reinforced learning agents within a road network and what would be the possibility to coordinate their efforts for achieving global improvements over local ones, and the implications on the vehicle population.
- It is important to understand the plausibility, potential advantages, or even unintended negative implications of self-adaptive systems in the real world. The time constraints have prevented us from being able to analyze our approach when multiple intersections are present.
- It would be interesting to see if the same state-action pairs would be learned or if the presence of multiple nodes would cause change.

• **What questions are you left with? What questions would you like to raise in an open discussion of the work (review interesting and controversial points, above)? What do you find difficult to understand? List as many as you can**

- Would this system actually work in real life?
- Are there better ways to do this than using machine learning and simulation?
- Would IoT yield better results?

Appendix A.4

Summary for 3rd International Conference on Emerging Technologies in Computer Engineering: Machine Learning and Internet of Things [17].

- **What is your take-away message from this paper?**

- This paper aims to develop a tool for predicting accurate and timely traffic flow Information. Traffic Environment involves traffic signals, accidents, rallies, even repairing of roads that can cause a jam. If we have prior information about all the above, then a driver or rider can make an informed decision. Also, it helps in the future of autonomous vehicles. In the current decades, we have moved towards thebig data concepts for transportation.

- **What is the motivation for this work (both people's problems and technical problem), and its distillation into a research question? Why doesn't the people problem have a trivial solution? What are the previous solutions and why are they inadequate?**

- The development and deployment of Intelligent Transportation System (ITSs) provides better accuracy for Traffic flow prediction to alleviate traffic congestion, improve traffic operation efficiency, and reduce carbon emissions. It is a crucial element for the success of advanced traffic management systems, advanced public transportation systems, and traveler information systems.

- **What is the proposed solution (hypothesis, idea, design)? Why is it believed it will work? How does it represent an improvement? How is the solution achieved?**

- Different machine algorithms are applied and tested for achieving higher efficiency and accurate results. To identify classification and regression we have used a Decision Tree Algorithm (DT). The goal is to predict the value of the target variables.

- Decision tree learning represents a function that takes as input a vector of attributes value and return a (Decision) a single output value. It falls under the category of supervised learning algorithm and can be used to solve both regression and classification problems.

- DT identifies its results by performing a set of tests on the training dataset. Outliers' detection is another critical step for an accurate result, and for this, we have used Support vector machines (SVMs), which is a set of supervised learning methods that can also be used for classification and regression.

- The SVM is beneficial for high dimensional spaces and helps when the number of samples are less than the number of dimensions.

- The random forest algorithm is a robust machine learning algorithm, defined as bootstrap aggregation. It is based on forecasting models, and mostly used to classify the data. The bootstrap algorithm is used togenerate multiple models from a single training data set. It used a sample to estimate statistical quantities.

- **What is the author's evaluation of the solution? What logic, argument, evidence, artifacts (e.g., a proof-of-concept system), or experiments are presented in support of the idea?**

- Intelligent Transportation system (ITS) has used the application of computer, electronics, and communication technology to provide information that increases the safety and efficiency of the road transportation systems and reduces carbon emission. Irrespective of vehicles

- increases on roads, the traffic also increases, and the available road network capacity is not feasible to handle this heavy load.
- There are two possible approaches to resolve this issue. The first one is to make new roads and new highway lanes, but this requires extra land, extensive infrastructure to maintain it, and high cost of expenditure. But already the land facility is not available for the expansion of the roads and lanes. The second approach uses some control strategies to use the existing road network efficiently.
- The expenditure will reduce, and it is cost-effective models for the government or the traffic managers. In this control, strategies identify the potential congestion on the roads, and it directed to the passengers to take some alternative routes to their destinations.
- Deep learning is a part of machine learning algorithms, and it is a compelling tool to handle a large amount of data. Deep learning provides a method to add intelligence in the wireless network with complex radio data and large- scale topology.
- In DL, using concepts of a neural network, it is beneficial to find network dynamics (such as spectrum availability, congestion points, hotspots, traffic bottleneck. Travel time is the essential aspect in ITS and the exact travel time forecasting also is very challenging to the development of ITS.Support Vector Machine (SVM) is one of the most effective classifiers among those which are sort of linear.
- It is advantageous to prevent overfitting of data. SVM is great for relatively small data sets with fewer outliers. Another algorithm (Random Forest, Deep Neural Network, etc.) requires more data but always came up with very robust models.
- SVM support linear and nonlinear regression that we can refer to as support vector regression, instead of trying to fit the most significant possible roads between two classes while limiting margin violation. Support Vector Regression (SVR) tries to fit as many instances as possible on the road while limiting margin violations.

- **What is your analysis of the identified problem, idea, and evaluation? Is this a good idea? What flaws do you perceive in the work? What are the most interesting or controversial ideas? For work that has practical implications, ask whether this will work, who would want it, what it will take to give it to them, and when might it become a reality?**
  - I think this paper wrote many important details regarding the actual implementation not just the general idea which I found to be lacking on some other papers.
- **What are the paper's contributions (author's and your opinion)? Ideas, methods, software,experimental results, experimental techniques...?**
  - Created an application which can provide us the GPS coordinates.
  - Perform the proposed algorithm
  - Evaluate the matrix for the dataset
  - Divide the dataset into training and testing.
  - Analyze different machine learning algorithms.

- Predict the 45 min interval parameters through machine learning algorithm
- Conclude about the traffic congestion

- **What are future directions for this research (author's and yours, perhaps driven by shortcomings or other critiques)?**
  - Although deep learning and genetic algorithm is an important problem in data analysis, it has not been dealt with extensively by the ML community. The proposed algorithm gives higher accuracy than the existing algorithms. It improves the complexity issues throughout the dataset. Algorithms will be further improved to much higher accuracy.
- **What questions are you left with? What questions would you like to raise in an open discussion of the work (review interesting and controversial points, above)? What do you find difficult to understand? List as many as you can**
  - I'd love to try their proposed system some time.

Appendix A.5

Summary for Urban Safety: An Image-Processing and Deep-Learning-Based Intelligent Traffic Management and Control System [18]

- **What is your take-away message from this paper?**
  - Traffic state prediction aims to forecast future traffic variables such as flow or speed in the road network based on historical or observed traffic data and other supporting information relevant to the demand
- **What is the motivation for this work (both people's problem and technical problem), and its distillation into a research question? Why doesn't the people problem have a trivial solution? What are the previous solutions and why are they inadequate?**
  - Due to the increasing population all over the world, and the constant increase in the number of vehicles used, congestion has been realized to be one of the most important and pressing problems nowadays which need attention and a solution other than the traditional manually controlled ones.
- **What is the proposed solution (hypothesis, idea, design)? Why is it believed it will work? How does it represent an improvement? How is the solution achieved?**
  - Autoregressive Integrated Moving Average
  - Support Vector Machines
  - K-Nearest Neighbors
  - Neural Networks
- **What is the author's evaluation of the solution? What logic, argument, evidence, artifacts (e.g., a proof-of-concept system), or experiments are presented in support of the idea?**
  - The most widely used method to tackle this problem is FT controllers, which use historical data to determine the appropriate timing of traffic signals. However, this approach cannot meet the current traffic stochastic demands and handle unexpected traffic situations.
  - Due to the limitations of the Fixed Time (FT) controllers, Webster's method was introduced, where inductive detectors are employed to observe the actual traffic conditions and efficiently extend or terminate the green signal time by measuring the gap between vehicles.
  - However, accumulative information is neglected, reducing the overall performance.
- **What is your analysis of the identified problem, idea, and evaluation? Is this a good idea? What flaws do you perceive in the work? What are the most interesting or controversial ideas? For work that has practical implications, ask whether this will work, who would want it, what it will take to give it to them, and when might it become a reality?**
  - I think this paper proposed many different solutions to the problem by referencing many previous works done on the same problem.

- **What are the paper's contributions (author's and your opinion)? Ideas, methods, software,experimental results, experimental techniques...?**
  - The Sydney Coordinated Adaptive Traffic System (SCAT) and Split, Cycle, and Offset Optimization Technique (SCOOT) adopt adaptive systems to suppress the drawbacks of the previous methods by gathering the data of the traffic flow in real-time at each intersection to control the timing of traffic lights effectively.
- **What are future directions for this research (author's and yours, perhaps driven by shortcomings or other critiques)?**
  - Neuro-Fuzzy-Based Systems
  - Deep Reinforcement Learning
  - Combination of QL, NNs, and FL
  - Hybrid Deep Q-Networks
  - Boosted Genetic Algorithm
- **What questions are you left with? What questions would you like to raise in an open discussion of the work (review interesting and controversial points, above)? What do you find difficult to understand? List as many as you can**
  - This paper had numerous different techniques to tackle the traffic congestion problem which makes me eager to be able to try at least some of them.

Appendix A.6

Summary for Traffic Flow Prediction for Smart Traffic Lights Using Machine Learning Algorithms [19]

- **What is your take-away message from this paper?**
- Traffic prediction can be divided into two types of techniques: parametric, including stochastic and temporal methods, and non-parametric, such as machine-learning (ML) models, recently used to solve complex traffic problems. The review made in found that non-parametric algorithms outperform parametric algorithms due to their ability to deal with a large number of parameters in massive data.

- **What is the motivation for this work (both people's problem and technical problem), and its distillation into a research question? Why doesn't the people problem have a trivial solution? What are the previous solutions and why are they inadequate?**
- The purpose of traffic forecasting is to predict future traffic conditions on a transportation network based on historical observations. This data can be helpful in ITS applications such as traffic congestion control and traffic light control. For example, it can calculate the likelihood of congestion on the corresponding road segment and prepare for it in advance.

- **What is the proposed solution (hypothesis, idea, design)? Why is it believed it will work? How does it represent an improvement? How is the solution achieved?**
- In this paper, five ML models: MLP-NN, Gradient Boosting Regressor, Random Forest Regressor, Linear Regressor and Stochastic Gradient Regressor, and two DL models based on RNNs: GRU and LSTM; are compared in the task of traffic flow prediction of each lane of an intersection.

- **What is the author's evaluation of the solution? What logic, argument, evidence, artifacts (e.g., a proof-of-concept system), or experiments are presented in support of the idea?**
- In this paper, they proposed several ML and DL models for the traffic flow prediction at an intersection of vehicular traffic, thus laying the groundwork for adaptive traffic control. Two public datasets were used to train, validate, and test the proposed models. Experimental results showed that Multilayer Perceptron Regressor has better performance and takes less processing time to train (18 s).
- Gradient Boosting Regressor has a similar performance but takes more processing time (28 s). Both RNNs and Random Forest Regressor have similar scores. However,RNNs are slow to train (between 250 and 321 s). Finally, Linear Regression and Stochastic
- Gradient Regressor has good processing time (20 s) but are the worst performance between these models. All ML and DL models achieved an explained variance score(EV Score) and R-squared (R 2) greater than 0.90, MAE near to 10; the RMSE is near15 and the MAPE is between 20 and 30 percent.
- Actually, the performance of these seven algorithms does not differ significantly. In conclusion, the results were satisfactory for predicting traffic flow in the four lanes of an intersection, demonstrating the feasibility of being implemented on smart traffic light controllers.

- **What is your analysis of the identified problem, idea, and evaluation? Is this a good idea? What flaws do you perceive in the work? What are the most interesting or controversial ideas? For work that has practical implications, ask whether this will work, who would want it, what it will take to give it to them, and when might it become a reality?**

- I think this paper proposed many different solutions to the problem by referencing many previous works done on the same problem and by doing many themselves.

- **What are the paper's contributions (author's and your opinion)? Ideas, methods, software,experimental results, experimental techniques...?**

- Datasets→ The dataset contains recorded data from six crosses in the urban area for56 days, in the form of flow time series, depicting the number of vehicles passing every five minutes for a whole day, which is recommended for short-term predictions. It is common to find missing values in databases represented by zeros,probably due to sensor failures.
- RNNS Designs→ Two recurrent neural networks are designed: GRU and LSTM The input layer with shape equal to the number of time steps per the number of lanes.Then two recurrent layers with 64 neurons, 20% dropout, and finally, an output layer with neurons equal to the number of lanes and sigmoid activation function.
- Data split→ 1. The database is split into 75% of the data (42 days) for training and 25% (14 days) for testing. 2. The data are scaled in a range from 0 to 1, following the standard normal distribution using MinMaxScaler
- The previous hour's traffic flow is used, which is a time sequence of 12 data points, to predict the traffic flow coming in the next five minutes.
- RNNs Training→ mean squared error (MSE) is used as loss function, and the optimizer is RMSprop. For training, a batch size of 128 and 50 epochs are used, five percent of the training data is used for validation.

- **What are future directions for this research (author's and yours, perhaps driven by shortcomings or other critiques)?**

- These models can be used in a smart traffic light controller, fed by traffic sensors that count the number of vehicles passing through a lane every certain period; with these readings, a database similar to the one used in this paper can be created. Once the database is generated, the ML model can be trained for each intersection.Then the traffic flow for the next period can be predicted by using a given number of past readings.
- Once the prediction is made, it will allow better programming of the times of each state, either manually by an operator or automatically using an algorithm to calculate the optimal times of the traffic light states. The whole process can be carried out by wirelessly communicating the traffic light with a central station or at the controller itself.

- **What questions are you left with? What questions would you like to raise in an open discussion of the work (review interesting and controversial points, above)? What do you find difficult to understand? List as many as you can**

- - I'd love to see how this would actually work in real life.

Appendix A.7

Summary for Traffic Management using Machine Learning [20]

- **What is your take-away message from this paper?**
- Machine Learning Algorithms are formed based on a mathematical model and help in the creation of the Intelligent Traffic Management System. The adaptation of traffic management systems has been explored with the help of Machine learning.Secondly, the study has attempted to discuss the application of the YOLO algorithm and the proposed system. Traffic Management System (TMS) monitors the vehicles to decrease the time spent in traffic signals along with suggesting alternative routes to the vehicles.
- **What is the motivation for this work (both people's problem and technical problem), and its distillation into a research question? Why doesn't the people problem have a trivial solution? What are the previous solutions and why are they inadequate?**
- Due to the increasing population all over the world, and the constant increase in the number of vehicles used, congestion has been realized to be one of the most important and pressing problems nowadays which need attention and a solution other than the traditional manually controlled ones.
- **What is the proposed solution (hypothesis, idea, design)? Why is it believed it will work? How does it represent an improvement? How is the solution achieved?**
- Adaption of Traffic Management System with help of Machine Learning
- Application of YOLO Algorithm and Proposed System
- Smart Traffic Management System with help of Machine Learning
- The role of Machine learning in traffic prediction
- Machine Learning and management of traffic congestion
- **What is the author's evaluation of the solution? What logic, argument, evidence, artifacts (e.g., a proof-of-concept system), or experiments are presented in support of the idea?**
- The efficient prediction of broad traffic has the capability of creating an intelligent transportation system. Collecting information related to traffic, such as location measured with GPS, direction, Start-End junction, and speed need to be analyzed.
- Calculation of these data by calculating the distance between each vehicle within the specific junction provides a clear prediction. It is imperative to consider that for using ML in traffic prediction, creating an application for providing GPS coordinates is crucial.
- On the other hand, due to a lack of availability of historical data, impact predictions of traffic are often conducted using mathematical model-based simulations. There are often considerations of suboptimal situations as real-world traffic conditions can differ from the considered assumptions.
- The prediction of the traffic in a road network depends on different aspects such as the general traffic level of that area, the situation of the road.

- **What is your analysis of the identified problem, idea, and evaluation? Is this a good idea? What flaws do you perceive in the work? What are the most interesting or controversial ideas? For work that has practical implications, ask whether this will work, who would want it, what it will take to give it to them, and when might it become a reality?**

- I believe that this paper has successfully tackled the traffic congestion problem using machine learning as it was detailed and clear with its descriptions on how to solve the problem.
- I also believe that this paper paid attention to details that weren't mentioned in other papers like how to detect accidents, overspeed limit and wrong lane detection.

- **What are the paper's contributions (author's and your opinion)? Ideas, methods, software,experimental results, experimental techniques...?**

- Multi-Layer Perceptron (MLP) was the best model for prediction. With the help of a neural network, the mapping can be done effortlessly for input and output to calculate and predict future trends, Multilayer perceptron consists of an input and output layer
- Back Propagation Algorithm was used to train multilayer perceptron, while the weather information and traffic flows were used to predict congestion
- This paper predicts the traffic flow by using artificial neural network model named as Back Propagation (BP) Neural network. It is being used for gaining maximum accuracy in the prediction of the traffic congestion with a defined structure of a neural network having an input layer, hidden layer, and output layer — the structure composed of the forward propagation and backpropagation of error.
- In the forward propagation, information is being processed from the input layer to the hidden layer, then finally transmitted to output layer and if the output layer cannot accept then it sends back to the process of back propagation error, in which different values of weights are adjusted in such a way to minimize error and again transferred to the forward propagation

- **What are future directions for this research (author's and yours, perhaps driven by shortcomings or other critiques)?**

- There has been a proposal for an Adaptive Traffic Management System (ATMS) that makes use of IoT and Machine Learning Camera Sensors along with two control boards. Camera Sensors captures the details from the lane and pass the data to the first controller board. The controller board uses this to adjust the traffic signals and congestion lights accordingly. Min- Max Fairness algorithm is used after estimating thedifference between the counts of the two lanes. Priority is given to the lane having a low average waitingtime.
- With help of YOLO, the advantage that is obtained is of having a high accuracy while being able to run in real-time. The algorithm works by looking at the object once and the image is sensed while proceeding with the only one forward propagation. It passes through the neural network and makes predictions. It has beenobserved that with help of YOLO, single CNN simultaneously predicts the multiple bounding boxes and trains on complete images.
- The YOLO algorithm is run it helps in counting the vehicles. The wrong way vehicle image is captured, and an alert message is sends to manage the traffic. This has been observed to prevent traffic congestion and hence, traffic management takes place

effectively.

- Accidents of motorcycles have been on the rise in recent years and the use of bicycles is also on the rise. It has been necessary to detect the helmet as objects so that road safety is obtained. It has been observed that Support Vector Machines (SVM) has been made to use to classify helmets and human heads without helmets.It has been proposed by Waranusat to detect moving objects with the help of a k- NN classifier so that the head of the motorcyclists can be detected.

- **What questions are you left with? What questions would you like to raise in an open discussion of the work (review interesting and controversial points, above)? What do you find difficult to understand? List as many as you can**

- What change would it make to implement all these multiple systems in real life?

Appendix A.8

Summary for Evolutionary Computing and Mobile Sustainable Networks. Lecture Notes on Data Engineering and Communications Technologies, vol 116 [21]

- **What is your take-away message from this paper?**

- Congestion could be predicted by taking two datasets, one with the past year and one with the recent year's data set
- To overcome the problem of traffic congestion, traffic prediction using machine learning which contains regression model and libraries like pandas, os, numpy, matplotlib. pyplot are used to predict the traffic.

- **What is the motivation for this work (both people's problem and technical problem), and its distillation into a research question? Why doesn't the people problem have a trivial solution? What are the previous solutions and why are they inadequate?**

- Conventional traffic signal timing plan management using statistical information of traffic lacks the ability to rapidly adapt into the dynamic traffic flow. Thus, we need to develop intelligent traffic signal timing plan management to continuously learn from the dynamic traffic environment for adaptability.
- The problem with the existing traffic system is that for every minute the vehicles at the 4-way cross-roads will be large in number resulting in a lot of congestion. Even though there are no vehicles at a particular side, the traffic signal will still glow green for a fixed time.

- **What is the proposed solution (hypothesis, idea, design)? Why is it believed it will work? How does it represent an improvement? How is the solution achieved?**

- A person developed the LSTM based prediction models by using machine learning approaches, which involve structure designing or network training designing and prediction
- The next goal is to deal with prediction errors that may occur during the prediction process with deep learning methods. This method is applied to big data that has collected from the performance measurement system

- **What is the author's evaluation of the solution? What logic, argument, evidence, artifacts (e.g., a proof-of-concept system), or experiments are presented in support of the idea?**

- LSTM model performed well

- **What is your analysis of the identified problem, idea, and evaluation? Is this a good idea? What flaws do you perceive in the work? What are the most interesting or controversial ideas? For work that has practical implications, ask whether this will work, who would want it, what it will take to give it to them, and when might it become a reality?**

- I don't think this paper talked about the topic or their proposed system in depth, they only tried some neural network models and showed some simulation commands and that's it

- **What are the paper’s contributions (author’s and your opinion)? Ideas, methods, software,experimental results, experimental techniques...?**

- This paper used an LSTM model
- This paper used a regression model
- This paper showed some simulation commands for a simulation software

- **What are future directions for this research (author’s and yours, perhaps driven by shortcomings or other critiques)?**

- In the future, the systems are often further improved using more factors that affect traffic management using other methods like deep learning, artificial neural network, and even big data
- Increasing the accuracy of forecasting methods which in return increases the accuracy
- There will be GPS that's the road and accident-prone areas will be highlighted in order that people wouldn't prefer using the paths which aren't safe.

- **What questions are you left with? What questions would you like to raise in an open discussion of the work (review interesting and controversial points, above)? What do you find difficult to understand? List as many as you can**

- This paper could’ve used many other neural network models that I believe can yield better results than the one they used.

Appendix B

Summary for all implemented traffic analysis systems [22] [23] [24] [25].

Services offered:

- Traffic data fusion: Integrate multiple data sources to produce consistent, accurate, and insightful traffic views, applying traffic flow analytics, data validation and filtering, map matching, closure detection, and incident extraction.

- Traffic data visualization: Create easy-to-use dashboards that visualize a wealth of information collected from existing road infrastructure, provide reporting tools and live alerts concerning all traffic participants, and analyze traffic volumes over a long period of time.

- Traffic data delivery: Maximize the use of traffic data by easily delivering it to critical applications, customizing products and services, integrating APIs, creating traffic tiles, and broadcasting traffic data to a wide audience.

- Traffic centers and complete solutions: Apply collected, analyzed, and visualized data to create real-time traffic monitoring systems and traffic centers for managing and optimizing traffic networks that influence infrastructure development and increase quality of life in cities.

- Road safety: Recognize a road incident right after it occurs; identify the guilty party based on video registration, computer vision, and data analytics tools; and make automated emergency calls to save lives and accelerate the clearing of roadways.

Technologies used:

- Connectivity and IoT: Unite the latest technologies like 5G and connected sensors to provide sufficient data for real-time traffic monitoring systems with centralized device management and stable access to online traffic services.

- Clouds and DevSecOps: Build and nurture cloud-based infrastructure for traffic management technologies, reinforced with modern DevSecOps practices for secure and stable performance of your traffic management program.

- Applications and integrations: Connect your traffic management application with platforms and services for effortless data exchange to give businesses, transport operators, and drivers easy access to recent traffic information.

- Big data & data science: Consolidate a huge amount of data collected from connected vehicles, mobile devices, online services, weather stations, and road infrastructure to get actionable insights and automate traffic decisions.

- Embedded development: Install smart sensors in the core of vehicles and infrastructure to monitor and manage constantly changing traffic flows, comply with emissions standards, increase road safety, and promote sustainability.