Real World Data-Based Deep Reinforcement Learning for Traffic Management and Emissions Reduction in a Low Emission Zone

Laura Saldarriaga Higuita

Tesis de maestría presentada para optar al título de Magíster en Ingeniería

Director
Gustavo Adolfo Patiño Álvarez, Doctor (PhD)

Universidad de Antioquia
Facultad de Ingeniería
Maestría en Ingeniería
Medellín, Antioquia, Colombia
2024

Cita		Saldarriaga Higuita [1]
Referencia Estilo IEEE (2020)	[1]	L. Saldarriaga Higuita, “Real World Data-Based Deep Reinforcement Learning for Traffic Management and Emissions Reduction in a Low Emission Zone”, Tesis de maestría, Maestría en Ingeniería, Universidad de Antioquia, Medellín, Antioquia, Colombia, 2024.

Maestría en Ingeniería, Cohorte XXXVI.

Grupo de Investigación Sistemas Embebidos e Inteligencia Computacional (SISTEMIC).

El contenido de esta obra corresponde al derecho de expresión de los autores y no compromete el pensamiento institucional de la Universidad de Antioquia ni desata su responsabilidad frente a terceros. Los autores asumen la responsabilidad por los derechos de autor y conexos.

Real World Data-Based Deep Reinforcement Learning for Traffic Management and Emissions Reduction in a Low Emission Zone

UNIVERSIDAD
DE ANTIOQUIA
1803

A dissertation submitted for the degree of Master of Science in Engineering

Laura Saldarriaga Higuita

Director: Prof. Dr.-Eng. Gustavo Adolfo Patiño Alvarez

Faculty of Engineering
Department of Electronics and Telecommunications Engineering
University of Antioquia
2024

Acknowledgments

I would like to express my gratitude to my parents Martha and Jorge. Without their love, encouragement, and support, none of what I have achieved would be possible. They have been my constant pillars of strength and their sacrifices, and belief in my abilities have shaped me into the person I am today.

I would also like to express my sincere thanks to my dear friends, especially Stiven, Maria José, Luis Felipe, Juliana, and Maria Fernanda, for their friendship, support and understanding throughout this journey. Their trust in me and their company have been a constant source of motivation, even when things seemed difficult. There is no one I would rather go through with during this journey and share our joys and sorrows. To all my other wonderful friends, your presence and friendship have also meant the world to me during this time.

I am deeply thankful to my professor and advisor Gustavo Patiño for his invaluable guidance and mentorship. His dedication and commitment to my academic growth have been really important in the completion of this work. Moreover, he proved to be not only an exceptional mentor but also a genuinely good person, always willing to offer support.

I extend my appreciation to this University, a place I have come to love for all that it has allowed me to learn and to live since I was a little girl. Also, the resources, funding, facilities, and opportunities provided by the University, the SISTEMIC research group, and CODI have been essential for this project.

A special thanks goes to the members of CITRA, especially engineer Christian Quintero, an old friend who not only collaborated with us but also provided invaluable data crucial for the development of our project.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

In recent years, vehicular congestion has become one of the major issues for large cities, as it brings about a series of negative consequences that affect both the economy of the cities and the health of their residents. In economic and environmental terms, the high costs associated with traffic congestion result from increased fuel consumption, greater wear and tear on road infrastructure, disruptions to public transportation services, and the deterioration of air quality due to the emission of pollutants. This leads to an increase in respiratory and cardiovascular diseases that impact various population groups.

For example, the cities that make up the Metropolitan Area of the Aburrá Valley (AMVA - *Área Metropolitana del Valle de Aburrá*) [1], in the state of Antioquia (Colombia), are not immune to the effects that the transportation sector has on air quality (mainly in the capital, Medellin). According to the entity, in the emissions inventory for the base year 2018, 91% of PM2.5[1] emissions came from mobile sources such as trucks, buses, 4-stroke motorcycles, and private vehicles [3].

Given the relationship between transportation and environmental pollution, major cities like Rome, Milan, and London have started to engage in more conscious mobility planning through strategies such as higher parking fees to encourage the use of public transportation, standardization of speeds

[1] Complex mixture with diverse chemical components, containing aerodynamic particles with diameters of less than or equal to 2.5μm. These particles are mainly produced by combustion processes of gasoline, oil, and diesel fuels [2]

for specific urban areas, the establishment of restricted traffic zones during specific hours [4], and the labeling of Low Emission Zones (LEZ) [5][6].

Moreover, improvements have not only been sought through policies and infrastructure but also through academic and private sector research and development of traffic management systems. From these research and experiments, strategies heavily reliant on technological and computational inputs have emerged, such as route planning, Vehicle-to-Infrastructure (V2I)[2] communication, and traffic light control, taking into account weather conditions, traffic history, and information collected from sensors and communication systems.

Currently, with the rise of artificial intelligence (AI), various machine learning techniques have been explored as potential solutions that, in conjunction with the aforementioned strategies, could drive the development of more robust and dynamic systems capable of adapting to traffic conditions. Among these techniques are traffic prediction models [8], traffic lights optimization, traffic simulation, reinforcement learning, and computer vision.

This research work aims to address traffic management from a multi-agent perspective, considering the existing road infrastructure in the city of Medellin, to execute traffic light control in a simulated environment and evaluate the impact that an intelligent management system could have on the city's traffic, which could positively affect the air quality in the study region.

1.2 Research Question

Considering the above, the research questions is defined:

✓ How can a Machine Learning-based model approach be effectively applied in urban traffic management to enhance traffic efficiency and reduce emissions within a Low Emission Zone?

1.3 Contribution of the Research Work

This research represents a contribution to the field of traffic management and emissions reduction in urban areas. Addressing a gap in the literature,

[2] Wireless data communication between vehicles and highway infrastructure such as traffic lights, road signs, and sensors [7]

this study focuses on the implementation of a Deep Reinforcement Learning (DQN) model in the context of a Low Emission Zone, using a multi-agent architecture for the purpose of traffic light control. Unlike previous research, which has mostly focused on simulations with synthetic data and primarily single-agent approaches, this research is based on real traffic data captured in the city of Medellin to simulate a scenario with a real map of the study area, containing multiple intersections. In this simulation, a novel weighted multi-objective reward function is proposed, incorporating both traffic efficiency and emissions reduction as key objectives. This multi-objective approach takes into account both the need to reduce congestion and the importance of minimizing emissions, which is essential for creating sustainable urban environments. The inclusion of traffic characterization and a more realistic simulation scenario provides a foundation for modeling and evaluating traffic congestion and emissions control policies. The results obtained show improvements in both traffic efficiency and emissions reduction, validating the effectiveness of the applied methodology.

1.4 Objectives

1.4.1 General Objective

To design, implement, and evaluate a dynamic model for intelligent traffic management with the aim of reducing vehicular congestion in areas of Medellin where the Air Quality Index (AQI) indicates high levels of harmfulness. This will be based on machine learning techniques and multi-agent systems.

1.4.2 Specific Objectives

✓ Characterize vehicular traffic and the road infrastructure of the city by exploring, analyzing, and interpreting maps and traffic data available in the databases of the control entities in Medellin and the Metropolitan Area.

✓ Design the architecture of a computational model based on intelligent agents that can identify and respond to changes and trends in vehicular congestion in the region.

- ✓ Develop a software algorithm that implements the designed model based on intelligent agents, with adaptability rooted in machine learning techniques according to current mobility conditions.
- ✓ Validate and simulate various vehicular traffic scenarios to analyze the scalability and adaptability of the proposed model, as well as the achieved reduction in congestion, considering traffic and air quality variables.
- ✓ Evaluate the implementation of the computational system using performance metrics such as execution times, error percentages, memory usage, complexity, among others.

1.5 Structure of the Research Work

Chapter 2 delves into the intricate relationship between traffic and air quality. This chapter explores various traffic management strategies and initiatives aimed at mitigating air pollution, including Low Emission Zones. It provides an overview of the state of the art in this field.

Chapter 3 focuses on reinforcement learning and deep reinforcement learning. It introduces the fundamental concepts and the theoretical basis of these machine learning approaches. Additionally, it reviews the latest advancements and research in reinforcement learning applied to traffic management to provide a comprehensive understanding of this crucial aspect of the study.

Chapter 4 is dedicated to the case study, the Low Emission Zone (LEZ) of Medellin. It conducts a detailed time series analysis of vehicular traffic within this specific zone, including temporal, roadwise and vehicle category-based analyses. It also shows the methods used for feature selection. This chapter forms the foundation for subsequent experiments and allows for gaining insights into the traffic dynamics of the targeted area.

Chapter 5 delves into the experimental phase of the research. It employs deep reinforcement learning techniques, specifically the use of Deep Q-Networks (DQN), to learn policies for traffic light control within a simulated environment. This section showcases the experiments, the methodology employed for policy learning, and the results obtained.

Chapter 6 brings the thesis to a close. Here, the key findings and insights from the previous chapters are summarized. The implications of the research on traffic management are discussed. Additionally, potential avenues for future work are outlined, identifying areas where further research can contribute to the field. This chapter provides a comprehensive conclusion to the thesis and sets the stage for future research efforts.

Chapter 2

Vehicle Traffic Management

Traffic management is a fundamental aspect of urban planning and development, particularly in densely populated areas where vehicular mobility is a constant concern. The smooth flow of vehicles is essential for maintaining efficient transportation networks. However, one of the most prevalent and challenging issues in urban traffic management is traffic congestion. This occurs when the demand for road and street usage surpasses their capacity, resulting in slow-moving or stationary vehicles, leading to standstills and traffic jams [9].

Traffic congestion is a multifaceted problem influenced by several factors, such as the ever-increasing number of vehicles on the road, traffic accidents, and adverse weather conditions [9]. These conditions often cause reduced traffic speeds and, in some cases, complete standstills, significantly extending travel times and adversely affecting the overall quality of life for urban residents.

2.1 Vehicle Traffic and Air Quality

Beyond the inconvenience of longer commute times, traffic congestion also has serious environmental consequences. When vehicles are forced to operate inefficiently due to congestion, they release higher volumes of pollutants into the atmosphere. This increased emission of pollutants, including carbon dioxide (CO2), carbon monoxide (CO), nitrogen oxides (NOx), and particulate matter (PM), has a substantial and detrimental impact on air quality [10]. These emissions not only contribute significantly to air pollution but also have direct health implications for urban inhabitants. Prolonged exposure to such pollutants can lead to respiratory and cardiovascular problems, exacerbating the already pressing issue of public health in cities and contributing to global warming.

In view of these environmental and health concerns, efficient traffic management in urban and suburban areas has become an ongoing challenge for traffic authorities and urban planners. As cities continue to expand, and the volume of vehicles on the roads gradually increases, the need to formulate effective strategies and harness innovative technologies becomes increasingly vital. These approaches are essential for optimizing traffic flow, mitigating congestion, and ultimately minimizing travel times. Not only do these strategies enhance transportation efficiency, but they also play a central role in addressing the serious environmental consequences of traffic congestion, making them integral to safeguarding the well-being of urban populations and the environment [10].

2.2 Traffic Management Strategies

In the pursuit of comprehensive solutions to address urban traffic challenges and improve air quality, various strategies have emerged. One key approach is the promotion of public transportation systems. These systems not only alleviate congestion but also offer more sustainable mobility alternatives to private vehicle use [11]. Additionally, cities are encouraging active transportation modes such as cycling and walking, which not only reduce congestion but also contribute to better public health. However, the lack of necessary infrastructure in many cities often leads citizens to opt for private vehicles due to limited transportation choices.

Furthermore, the implementation of urban tolls and traffic restrictions, as previously discussed in studies [12][13], encourages environmentally conscious transportation choices and the adoption of cleaner vehicles, including electric ones. These measures serve a dual purpose: they mitigate pollution and alleviate traffic congestion. Urban tolls and pricing zones are designed to incentive carpooling and boost public transportation usage by imposing fees on drivers entering specific areas during peak demand times [14]. The generated revenue can be reinvested in infrastructure enhancements and the development of sustainable public transportation systems, underscoring the importance of policies that support such modes of transportation. Additionally, the promotion of cycling, improvements in pedestrian infrastructure, and incentives for public transportation all play significant roles in reducing car dependency, alleviating congestion, and curbing emissions. To further complement these efforts, urban planning can facilitate the development of residential and commercial areas near public transportation hubs, which ultimately reduces reliance on private vehicles [15].

In addition to these strategies the field of traffic management encompasses classical, data-driven, and machine learning methodologies: classical methods, which are mainly based on traffic engineering principles, allow to understand traffic dynamics and signal optimization strategies. These incorporate traffic signal optimization, where historical traffic patterns inform the timing of signals to enhance traffic flow. Additionally, classical traffic flow theory, and models like the Greenshields' [16], provides a theoretical foundation for understanding traffic dynamics, helping to predict congestion and formulating optimal control strategies. Some traffic simulation tools such as VISSIM [17], AIMSUN [18], and SUMO [19] offer virtual environments to simulate various traffic scenarios, facilitating the evaluation of management strategies before practical implementation.

On the other hand, data-driven methods use big data and analysis to understand traffic better. These methods gather data from elements like traffic sensors [20], GPS devices, and social media. By studying this information, traffic controllers learn about how traffic moves, helping them make quick decisions. Real-time traffic control involves continuous monitoring and making changes to respond to incidents and optimize traffic flow. In recent years, the growth of ML has revolutionized traffic management, introducing adaptive and predictive capabilities. These techniques, which are very aligned with data-driven methods, consider, for example, predictive analytics. These analyses use historical traffic data to forecast future traffic patterns and identify congestion hotspots. Within this domain, reinforcement learning (RL) techniques enable traffic control systems to autonomously learn optimal control policies through interaction with the environment, dynamically adjusting traffic signals or routing strategies to minimize congestion. Moreover, ML algorithms excel in traffic pattern recognition, discerning complex patterns from sensor data or video, enhancing incident detection, accident prediction, and traffic flow.

These methodologies highlight the importance of an adaptive approach to traffic management to addresses congestion, but like urban tolls and pricing zones, they can be focused on environmental concerns such as air quality, which is highly related to vehicles pollutant emissions.

Some cities have started to implement smarter mobility plans to reduce traffic congestion, accidents, and environmental impact. For example, in Singapore, innovative traffic management systems have become very important to reduce environmental impact and enhancing overall traffic flow. The Junction Electronic Eyes (J-Eyes) system [21], comprising approximately 400 surveillance cameras strategically positioned at major traffic junctions, continuously monitors traffic conditions in real-time. These data enable the Land Transport Authority (LTA) [22] to promptly respond to incidents and implement effective action plans. Additionally, TrafficScan [21] utilizes GPS data from taxis to provide drivers with real-time information on road conditions, allowing for route planning and better journeys. The Expressway Monitoring Advisory System (EMAS) [21] improves traffic management by detecting accidents and coordinating traffic along expressways. Electronic signboards along expressways and major roads provide drivers with up-to-date traffic information, while LTA Traffic Marshals [21] ensure efficient management of incidents. These comprehensive strategies underscore Singapore's commitment to sustainable and efficient traffic management practices, ultimately contributing to reduced emissions and improved air quality.

Another example is the city of Amsterdam [23], where the Smart Flow platform stands out as an alternative to alleviate traffic congestion and optimize parking. This IoT cloud-based system utilizes a network of sensors spread across the city to monitor real-time traffic flow and parking availability. By providing drivers with up-to-date information on parking spots, Smart Flow significantly reduces the time spent searching for parking, leading to less congestion, lower fuel consumption, and decreased pollution levels.

Following a similar path, the Main Roads Western Australia (MRWA) [24] has developed applications for Australia's Traffic Management System. One of these applications is the Traffic Management System Network (TMS), which gathers information to "monitor and manage traffic congestion, incidents and planned events" [25]. A traffic control system to manage dynamic timing of traffic lights, an Intelligent Transport System (ITS), composed by message boards, lane use management signs, vehicle detection stations, and a travel time system which collects vehicle movement data though Bluetooth device are the components of the TMS. By sharing important details like traffic accidents, daily traffic reports, road closures, accident summaries, and travel maps, the TMS promotes teamwork with other entities while helping people make informed choices for managing traffic and planning. MRWA is also working with the gathering of live traffic information (like travel time) obtained from vehicles through monitoring devices, by using MAC addresses [25], but this strategy still has some restrictions due to privacy and regulations, so its use is restricted.

Another city that has been open to the use of technology and information to support traffic decisions is Moscow. One of the approaches is the creation of a "digital twin", which is a virtual replica or model of the city, where the traffic authorities can evaluate the situation of traffic, and apply changes to the city, that can be evaluated after. The second approach is a dynamic transport model, that allows to gather data and evaluate traffic in real time. This system makes forecasting and the inhabitants of the city may receive messages with information about the state of the roads [26]. This city is also trying to research and expand to V2I-based implementations with sensors and other devices, to get closer to the development of robust systems for autonomous vehicles.

Medellin city, on the other hand, is still new to these implementations, however, its information systems have improved and now the authorities have different types of CCTVs, sensors, and data gathering stations [27][28][29] that have been helping the city to build better information systems to monitor, understand and improve vehicular traffic and its environmental impact.

Now, in relation to the topic of air quality, which is our main focus for this work, we can also find another effective strategy to control traffic and mitigate emissions, which is the establishment of Low Emission Zones (LEZs), are described next.

2.2.1 Low Emission Zones

Low Emission Zones are designated urban areas with the purpose of addressing air pollution challenges and promoting sustainability in urban mobility [30]. These zones have clear objectives, such as improving air quality and reducing pollution in congested urban areas. To achieve this, regulations and emissions standards that consider factors such as the vehicle type, age, and fuel type are defined to rule the entry of vehicles into the protected zone [5]. Consequently, these standards are not only determinant for entry to the LEZ, but also influence urban planning by encouraging the use of public transportation and active modes of transportation, which leads to the transformation of the urban landscape, and improvement of the quality of public space. The implementation of LEZs results in varied effects on air quality and public health.

In addition to their environmental impact, LEZs can have economic implications by incentivizing the upgrading of vehicles to cleaner and more energy-efficient models, leading to a renewal of a city's vehicle fleet. The successful implementation of an LEZ often requires active participation from the local community and continuous monitoring to assess its effectiveness in terms of emission reduction and air quality improvement [6]. This periodic evaluation allows for adjustments to the regulations based on the results obtained, contributing to an adaptable and effective approach [31].

To achieve this effectively, traffic and road infrastructure have to be characterized, such that two important components for this task are traffic monitoring and measurement. For this purpose, strategies such as vehicle counting, identification of vehicle types; and variables such as circulation speed [32], lane occupancy, and traffic volume, combined with temporal information such as the day of the week [8], have been observed.

2.2.1.1 State of the Art

As we examine the state of the art in LEZs, it is worth noting recent research findings that underscore their effectiveness. For instance, a study conducted in the German cities of Berlin and Munich [33] provides valuable insights into the impact of LEZs. This research reveals that LEZs, particularly in their advanced stages, have demonstrated remarkable efficacy in reducing PM10[1] concentrations, with substantial reductions observed in both traffic and urban sites. Furthermore, the study highlights the efficiency of LEZs in lowering levels of elemental carbon (EC), a component considered more toxic than PM10. However, it is important to note that the effects on NO2[2] levels were inconsistent, with no significant impact observed in Berlin and limited reductions in Munich.

On the other hand, in the city of Brussels, researchers utilized a remote sensing system to collect data on pollutant emissions, the number of vehicles,

[1] Complex mixture with diverse chemical components, containing aerodynamic particles with diameters of less than or equal to 10μm [2]. These particles come mainly from agriculture, wildfires, waste burning, industrial sources, among others [34].

[2] "Gas commonly released from the combustion of fuels in the transportation and industrial sectors" - WHO [35].

and average speeds within the vehicle fleet [6]. These data allowed for the identification of the most common vehicle types and the estimation of pollutants emitted. Such detailed information played a crucial role in the initial implementation of the LEZ. After a period of operation, the city conducted an assessment to gauge the impact of the LEZ on regional air quality, providing valuable insights into the effectiveness of this environmental strategy.

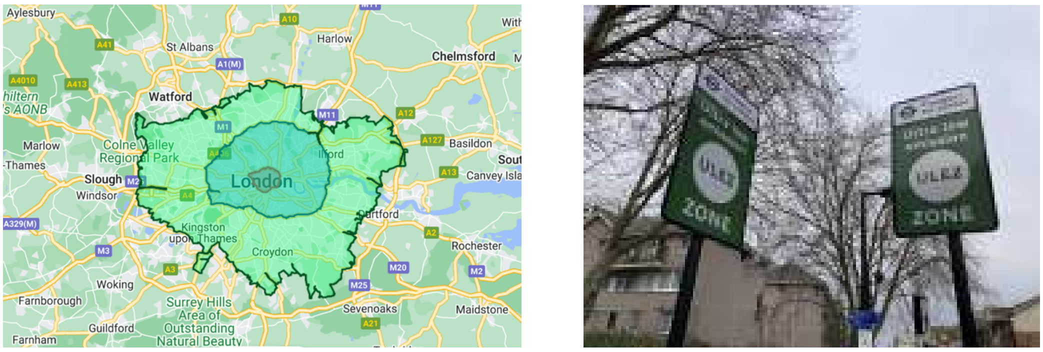
Similarly, the city of Lisbon adopted a multifaceted approach to assess the potential effects of introducing an LEZ [5]. They characterized the vehicle fleet by gathering data on vehicle counts, vehicle age, and conducting interviews with drivers to estimate the impact. This comprehensive approach allowed them to evaluate daily traffic patterns in three distinct areas of the city. Their findings indicated that the LEZ had the potential to be particularly effective in reducing PM10 emissions, though its impact on NOx reduction was somewhat less pronounced. Moreover, the study underscored that emission reduction is influenced not only by the number of vehicles but also by factors such as vehicle types, speed distribution, and travel distances.

Likewise, a study focused on the LEZ in Greater London and the stricter Ultra Low Emission Zone (ULEZ) [36] shed light on their significant impact. Both zones led to substantial reductions in NO2 and PM10 levels, primarily attributed to changes in the composition of the vehicle fleet. Beyond air quality improvements, both the LEZ and ULEZ demonstrated positive effects on public health, resulting in fewer health problems, reduced instances of long-term illnesses, decreased sick leave, and enhanced overall well-being. Notably, the ULEZ (Figure 2.1) exhibited even more pronounced impacts in these aspects. These findings underscore that LEZs possess the capacity to effectively enhance urban air quality, particularly by reducing PM concentrations, and deliver associated health benefits. However, their effectiveness may vary contingent on the rigor of the policy and localized factors.

Thanks to the success of these zones in various cities, their replicability and profound contributions to the creation of healthier and more sustainable urban environments have been unmistakably demonstrated. Beyond addressing air pollution concerns, these zones have a transformative effect on the quality of life for urban inhabitants. By promoting cleaner modes of transportation and alleviating traffic congestion, they foster improved mobil-

[1] Image source: `https://www.mandata.co.uk/insights/low-emission-zones/`

[2] Image source: `https://www.lse.ac.uk/granthaminstitute/wp-content/uploads/2023/08/ULEZ-sign_Matt-Brown-Flickr.jpg`



(a) ULEZ delimitation[3]

(b) ULEZ signs[4]

Figure 2.1. London's Ultra Low Emission Zone

ity and air quality simultaneously. For instance, drawing from the successful Greater London example [37], a report delivered in 2019 highlighted a significant reduction in the number of vehicles circulating in the ULEZ area, amounting to approximately 13,500 fewer vehicles. This accomplishment vividly illustrates how LEZs play an important role in mitigating traffic congestion within urban areas, leading to enhanced mobility and superior air quality for residents and commuters.

Although LEZs in urban areas have mainly been implemented in European cities [38], there is a relevant case study in the local context of interest. This is the case of the city of Medellin in Colombia, which has implemented its own Low Emission Zone (known as *ZUAP - Zona Urbana de Aire Protegido* in Spanish) [39] with the purpose of addressing pollution and congestion challenges in the urban environment. This Zone was established downtown in 2018 and it has an area of 2km^2. Its main objective is to reduce atmospheric pollutant concentrations by decreasing emissions from transportation (mobile sources) in the center, thus improving air quality and health for everyone in the city and the Aburrá Valley. The Zone was chosen by the AMVA [1] because it presented high levels of pollutants, resulting in a poor Air Quality Index (AQI)[5]. In the chapter 4 of this thesis, a deeper analysis of the traffic dynamics of the LEZ in Medellin will be undertaken with the aim of obtaining valuable information that can subsequently be used to address traffic management challenges.

[5] Standardized measurement for the level of air pollution in a specific location. Provides a value that represents overall air quality and its potential health effects [40]

In addition to these strategies, technology plays a crucial role in revolutionizing traffic management, with the aim of addressing its challenges and reducing its impact on air quality. Some of these approaches are described in the next chapter.

Chapter 3

Deep Reinforcement Learning for Vehicle Traffic Management

In this chapter, we introduce the theoretical foundation of reinforcement learning and the necessary concepts for understanding it. Additionally, we explore the state of the art in its application to vehicular traffic management, with a primary focus on the approach of deep reinforcement learning for traffic control.

3.1 Reinforcement Learning

Reinforcement learning is one of the fundamental paradigms of machine learning [41]. Along with supervised and unsupervised learning, it forms a set of algorithms applicable to a wide variety of problems, depending on the available observations and the specific application.

Reinforcement learning is based on the concept of learning through experience [42], which sets it apart from the other two paradigms. In supervised learning, algorithms are trained using labeled examples, while in unsupervised learning, the goal is to find patterns and structures in data without using labels. In contrast, in reinforcement learning, an *agent* learns to make optimal decisions by interacting with a dynamic environment.

The agent in reinforcement learning is the entity that learns and acts within the *environment*. It perceives the environment through observations and can take actions based on those perceptions. Each *action* the agent takes causes a *state* transition, meaning that the environment changes, and the agent faces a new situation [43]. The ongoing interaction between the agent and the environment allows the agent to learn how to make decisions that lead to maximizing a feedback signal called *reward*, taking its state into account.

Figure 3.1 illustrates the reinforcement learning cycle. In this figure, the state S_t is a representation of the current situation or configuration of the environment at a given time, and its value is based on data collected by the agent through its observations. The state should contain relevant information that enables the agent to make decisions that lead to obtaining a higher reward in the future [44].

The *reward* R_t, on the other hand, is a feedback signal that the environment provides to the agent after it has taken an action. This reward can be a numerical value, positive, negative, or zero, and it represents a measure of how good or bad the execution of action A_t was in relation to the system's ultimate goal. In reinforcement learning problems, the agent's purpose is to maximize the cumulative reward value over time.

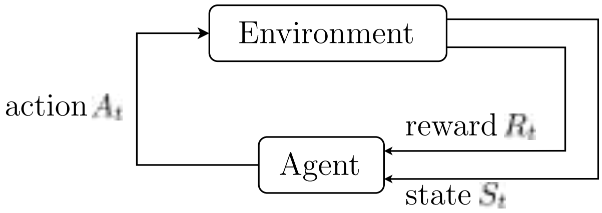


Figure 3.1. Reinforcement learning cycle

To manage decision-making in reinforcement learning, two fundamental concepts are used. First, the *discount factor* (denoted as γ) is employed to weigh the importance of rewards over time. A value of γ close to 1 gives greater importance to long-term rewards, implying that the agent will consider more the future consequences of its actions. Conversely, a value of γ close to 0 makes the agent focus on immediate rewards [45].

Second, the balance between exploration and exploitation, controlled by the parameter ϵ, is crucial in reinforcement learning. Since the agent is interacting with the environment and learning from its experiences, it must find an optimal way to act [46]. Exploration involves experimenting with previously untried actions to discover new opportunities for higher rewards, while exploitation entails leveraging known actions that have resulted in favorable rewards in the past. Striking the right balance between exploration and exploitation is a key challenge in reinforcement learning because excessive exploration can lead to inefficient resource usage, while excessive exploitation might limit the discovery of better strategies [46].

It's important to note that reinforcement learning uses a theoretical framework called Markov Decision Processes (MDPs) to model the interaction between the agent and the environment. An MDP is defined as a tuple $\langle S, A, P, R, \gamma \rangle$, where [44]:

✓ S is the set of system states. Each state $s \in S$ represents an environment configuration at a given moment.

✓ A is the set of actions available to the agent. Each action $a \in A$ represents a decision that the agent can make.

✓ $P : S \times A \times S \to [0, 1]$ is the state transition function. For each pair of states s and s' and action a, $P(s'|s, a)$ represents the probability of the system transitioning to state s' from state s when taking action a.

✓ $R : S \times A \times S \to \mathbb{R}$ is the reward function. For each state transition $s \to s'$ by action a, it is the immediate reward obtained by the agent.

✓ $\gamma \in [0, 1]$ is the discount factor. It represents the weight of future rewards relative to immediate rewards. A value of γ close to 1 indicates that the agent values long-term rewards, while a value close to 0 places greater importance on immediate rewards.

MDPs are a fundamental basis in reinforcement learning as they allow modeling problems in which actions can affect system state transitions and obtained rewards. By representing the problem in the form of an MDP, the agent can use learning algorithms to find an optimal *policy*.

Both in an MDP and in reinforcement learning, a policy is the strategy that guides the agent's decisions in an environment to maximize rewards over time [42], and it can be deterministic or stochastic depending on the situation and the problem at hand.

Having a deterministic policy means that for each state of the environment, a unique and specific action that the agent must take is specified [47]. For example, if there is a robot on a grid that must collect objects, it could have a policy that indicates that when it reaches a wall, it should always turn left until it has collected the objects in its path.

On the other hand, in a stochastic policy, instead of a single action for each state, a probability distribution over possible actions is assigned for each state. This means that in a given state, the agent can take different actions with certain probabilities [47]. This would be the case in a chess game, where different pieces can be assigned probabilities of movement, and in similar situations, the agent could make different decisions.

It is also important to consider, both in MDPs and in reinforcement learning, that a good use of the discount factor and the exploration-exploitation balance has proven to be especially useful for addressing complex problems where a complete set of labeled data is not available. By learning to maximize rewards over time, the agent acquires intelligent and adaptable behavior to solve complex tasks in various fields such as robotics [48], automatic control, games [49], recommendations [50], and many other areas where decision-making and adaptation to an uncertain environment are essential.

In attempting to address these and other challenges posed by reinforcement learning, two fundamental paradigms have emerged that address how agents make decisions and learn in dynamic environments. These paradigms are model-based reinforcement learning and model-free reinforcement learning.

3.1.1 Reinforcement Learning Paradigms

3.1.1.1 Model-Free Reinforcement Learning

This paradigm is based on the idea that the agent can learn through direct interaction with its environment without prior or explicit knowledge of it. In this approach, the agent explores different actions and observes the resulting rewards to improve its strategy over time [51]. Some key components within this paradigm include:

✓ The balance between exploration and exploitation (described in section 3.1).

✓ Policy evaluation methods, where the agent directly modifies its decision-making policy based on the rewards obtained during exploration.

✓ Temporal difference methods, where value estimates are adjusted as the agent interacts with the environment.

3.1.1.2 Model-Based Reinforcement Learning

The model-based paradigm involves establishing a model of the agent's operating environment, which is used for planning and decision-making. This model captures how states and rewards evolve based on the agent's actions [52].

As stated by [53], "model-based reinforcement learning attempts to overcome the problem of lack of prior knowledge by allowing the agent—whether it's a real-world robot, an avatar in a virtual world, or just a computer program carrying out actions—to construct a functional representation of its environment." Within this process, two additional components are important [54]:

- ✓ Planning, as the agent can simulate sequences of actions and predict the resulting rewards before choosing the action sequence that maximizes the anticipated reward.
- ✓ Learning from execution, where information about different sequences of states and actions is collected through interaction with the environment, which is then used to learn a policy through optimization or supervised learning methods.

In practice, the line separating the described paradigms can be quite blurry, as some algorithms can incorporate elements from both types [52]. Depending on the representation of their states and actions and how their values are updated, these algorithms can be classified as either tabular or non-tabular methods.

3.1.2 Tabular Methods

In the field of reinforcement learning, tabular methods represent an essential foundation upon which more advanced techniques are built. These methods are based on the representation of *Q-functions* (also known as Q-value functions) and policies in discrete tables that contain information about states and possible actions in an environment. These tables, known as Q-tables, are matrices where rows represent possible states of the environment, and columns represent possible actions an agent can take. Agents use them to learn explicitly through iterations based on the Bellman equation [55], as they interact with the environment and learn from the rewards received.

The structure and size of these tables are defined in the code implementing the algorithms and depend on the complexity of the problem the agent is trying to solve.

The following subsections delve into these concepts, as well as three fundamental tabular methods and their most relevant applications.

3.1.2.1 Q-Learning

The Q-Learning algorithm is one of the most relevant in reinforcement learning and is based on the Bellman equation to find the optimal Q-function. A Q-function (or action-value function) is a representation of the estimate of the expected value for the reward an agent can earn from taking an action a in a state s. The iterative update of Q is expressed as [42]:

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha \cdot [R_{t+1} + \gamma \cdot \max_{A} Q(S_{t+1}, A) - Q(S_t, A_t)] \quad (3.1)$$

In this equation, $Q(S_t, A_t)$ represents the value of action A_t in state S_t. α is the learning rate, and γ is the discount factor. The agent interacts with the environment, takes actions, and updates the Q-values based on the obtained rewards and future estimations of Q-values. Q-Learning is particularly effective in applications such as game control, robotics, autonomous navigation [56], among others, where the agent learns to navigate and pick up passengers by optimizing its sequential decisions. The general structure of Q-Learning is illustrated in Figure 3.2.

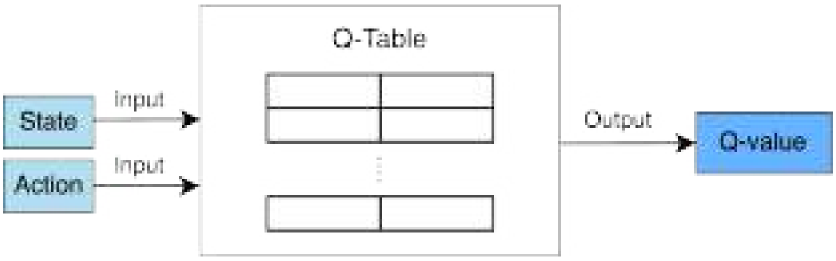


Figure 3.2. Q-Learning diagram

3.1.2.2 SARSA (State-Action-Reward-State-Action)

SARSA is another widely used tabular algorithm in reinforcement learning. Its process is similar to that of Q-Learning but with an on-policy focus, meaning it updates the policy based on the action taken in the current state [57]. The update equation is as follows:

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha \cdot [(R_{t+1} + \gamma \cdot Q(S_{t+1}, A_{t+1}) - Q(S_t, A_t)] \quad (3.2)$$

Here, A_{t+1} represents the next action taken by the agent in the state. SARSA is especially suitable for problems that require sequential control and planning, such as robot navigation in unfamiliar environments. Unlike Q-Learning, which selects the action with the highest possible reward in the next state, SARSA takes into account the agent's current policy.

3.1.2.3 Temporal-Difference Learning (TD)

The TD(0) method[1] is a generalized approach that encompasses both Q-Learning and SARSA as special cases. The update equation is similar to those of the two previous methods:

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha \cdot [R_{t+1} + \gamma \cdot Q(S_{t+1}, A_{t+1}) - Q(S_t, A_t)] \quad (3.3)$$

TD(0) seeks to estimate Q-values from immediate rewards and future Q-value estimates. Through this combination, the method aims to balance immediate reward and long-term expectations in decision-making [59]. Therefore, it is applicable in environments where a balance between immediate gain and long-term planning is needed, such as financial decision-making problems.

The main difference between SARSA and TD lies in their specific update rules and their focus on different types of value functions. SARSA is a variant of TD that specifically updates Q-values for state-action pairs based on agent interactions and policy, while TD is a broader category of methods that update value estimates using temporal difference updates.

3.1.2.4 Limitations of Tabular Methods in a Traffic Scenario

Although tabular methods in reinforcement learning provide a solid foundation for understanding fundamental concepts, they have certain limitations that restrict their applicability in more complex environments.

[1] Most basic type of TD learning [58], considering only one-step lookahead in decision making.

One of the primary limitations is their inability to handle high-dimensional state and action spaces. In real-world problems, state and action spaces can be vast and continuous, making their representation and storage in discrete tables challenging. Additionally, tabular methods are susceptible to the curse of dimensionality [44], which means that the amount of data and resources required to address complex problems increases exponentially with the dimensionality of the space [44].

Another significant limitation is the lack of generalization. Tabular methods tend to memorize specific situations rather than capture general patterns in the data. This makes them less effective in scenarios where it is essential to make inferences from previously unseen examples. Furthermore, tabular methods can be sensitive to noise in reward data, leading to erratic fluctuations in learning and suboptimal policies.

In the context of traffic management in a large-scale scenario, a tabular approach faces several critical limitations that render it inadequate for effective application. First and primarily, the inherent inability of tabular methods to handle high-dimensional state and action spaces is a substantial barrier. In real-world traffic scenarios, the number of potential states and actions can be vast and continuous, making it practically impossible to represent and store this information in discrete tables. This limitation is further compounded by the curse of dimensionality, where the volume of data and computational resources required increases exponentially with the complexity of the state and action spaces, rendering a tabular approach impractical [60].

Additionally, tabular methods lack the capacity for generalization, as they tend to memorize specific situations rather than capturing overarching patterns in the data. In the dynamic and constantly evolving domain of traffic management, the inability to make inferences from previously unseen situations becomes a significant drawback. Furthermore, tabular methods are sensitive to noise in reward data, which can lead to erratic fluctuations in learning and ultimately result in suboptimal traffic management policies. Given these limitations, it is evident that a tabular approach is not well-suited for the complexities of traffic management in a large-scale scenario, where adaptability, efficiency, and the ability to handle high-dimensional, dynamic environments are of paramount importance [61].

Despite these limitations, tabular methods remain an essential foundation for understanding key concepts in reinforcement learning. As we enter the era of non-tabular methods, these limitations drive the exploration of more sophisticated and adaptable approaches that can address the complexity and uncertainty of real-world problems. In the next section, we address how non-tabular methods have largely overcome these difficulties.

3.1.3 Non-Tabular Methods

In contrast to tabular methods, non-tabular approaches in reinforcement learning have managed to overcome many of the limitations of dimensionality and generalization, allowing for more effective adaptation to complex problems. These methods are based on the use of deep neural networks to approximate Q-functions and policies to map environmental states to the actions the agent should take in those states. This gives them the ability to learn more abstract and generalizable representations [62].

In the upcoming sections, we'll dive into the theoretical foundations of DRL, focusing in the most relevant concepts and components for the use and application of a Deep Q-Learning algorithm that incorporates a Deep Q-Network (DQN) to manage traffic in simulated environment. It is important to note that our approach takes a different path compared to traditional tabular methods, which we have just discussed due to their limitations. We have chosen to work with non-tabular methods because our research focuses on managing a large area with dynamic traffic (chapter 4), and conventional tabular methods struggle in such a complex environment. These traditional methods can't handle the complexity of vast, ever-changing traffic scenarios. In contrast, non-tabular approaches use deep neural networks with multiple layers to create more advanced and meaningful representations of data. In this case, the DQN extends the Deep Q-learning capability of being able to work within high-dimensional state spaces. This is particularly valuable in reinforcement learning because it allows the agent to make informed decisions based on meaningful data knowledge, resulting in more effective actions within the specific context of our dynamic traffic management scenario. As we look into the specifics of DRL and DQNs, we will explore how these non-tabular methods harness deep learning to address the challenges of traffic management [63].

3.1.3.1 Deep Reinforcement Learning and DQN

Deep learning relies on the use of neural networks with multiple hidden layers to model relationships within data. Through an iterative parameter tuning process, these networks can gradually capture increasingly abstract and meaningful representations of input data. This feature is particularly beneficial in reinforcement learning, where obtaining high-level representations can be essential for making informed decisions, i.e., decisions backed by meaningful data knowledge and representations that allow the agent to take more effective actions in the specific context. In other words, deep reinforcement learning combines the capacity of deep neural networks to model complex relationships with reward-based decision making in reinforcement learning.

Within the spectrum of non-tabular methods in deep reinforcement learning, the Deep Q-Network approach stands out as a concrete example of how deep learning can transform decision-making in complex environments.

3.1.3.1.1 Deep Q-Learning (DQN)

DQN is a type of algorithm that represents a convergence between reinforcement learning and deep learning. By combining the core concept of Q-learning with the capabilities of deep neural networks, it tackles challenging tasks in decision-making environments, even those characterized by high dimensionality and complexity. Its general structure, as illustrated in Figure 3.3, involves taking the current state as input and passing it through a deep neural network, resulting in Q-values for various available actions. These Q-values are essential for decision-making, as they estimate the cumulative expected value of future rewards for each action in a given state.

The algorithm's update equation, depicted in Equation 3.4, demonstrates how it refines its Q-value estimates based on the observed rewards and the predicted future rewards, effectively guiding the agent's decision-making process in pursuit of optimal outcomes.

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha \cdot [R_{t+1} + \gamma \cdot \max_A Q(S_{t+1}, A; \theta) - Q(S_t, A_t; \theta)] \quad (3.4)$$

In this equation, $Q(S_t, A_t; \theta)$ represents the estimation of the Q-function parameterized by the weights θ in a deep neural network architecture, and the term:

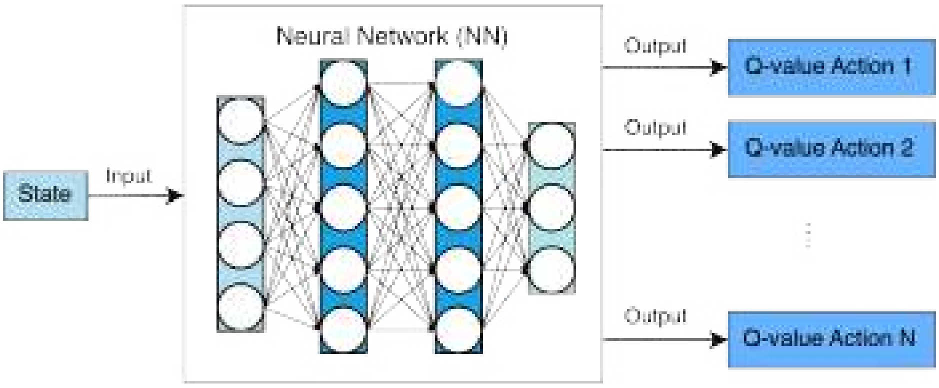


Figure 3.3. DQN Diagram

$$R_{t+1} + \gamma \cdot \max_{A} Q(S_{t+1}, A; \theta) - Q(S_t, A_t; \theta) \tag{3.5}$$

represents the difference between the expected reward and the current estimation of the Q-value, which is used to update the Q-value. The term α corresponds to the learning rate, and γ to the discount factor.

Neural Networks in Deep Reinforcement Learning

The chosen neural network architecture for the DQN model is crucial for its performance. In general, this network consists of several hidden layers, each composed of a set of interconnected neurons, as shown in Figure 3.3. These layers transform the input state representation into a more abstract representation and, ultimately, into Q-values for each possible action [64]. The choice of architecture and the number of layers directly affect the network's ability to accurately and efficiently approximate the Q-function.

Not only does the network architecture play a crucial role in DQN, but the choice of the type of neural network is also a crucial aspect that can influence the agent's ability to effectively and efficiently approximate the Q-function [65]. Some of the most commonly used types of neural networks in these models are:

✓ **Convolutional Neural Networks (CNN):** This type of network has proven to be effective in processing data with spatial structure, such as images and videos. In the context of DQN, CNNs are especially useful when working with environments that have a visual representation of the state. The convolutional layers in a CNN can capture relevant patterns and features in images, allowing the agent to learn more compact and meaningful state representations [63]. Figure 3.4 shows an example of a CNN.

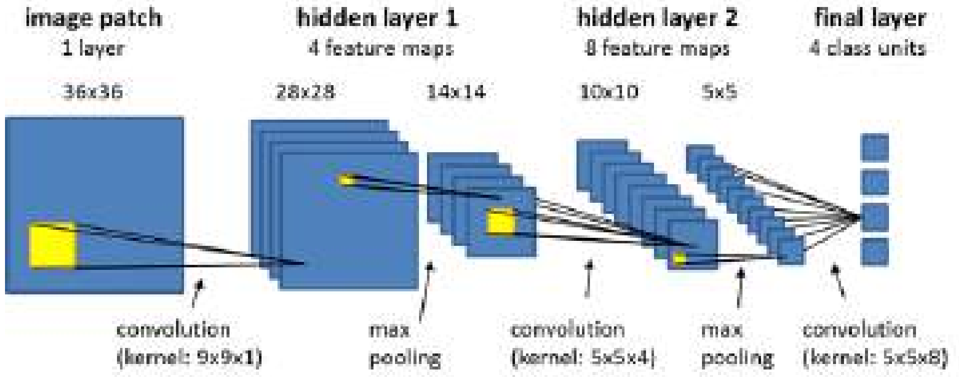


Figure 3.4. Convolutional Neural Network[2]

✓ **Recurrent Neural Networks (RNN):** These are a type of neural network architecture designed specifically for modeling and processing sequential or temporal data (Figure 3.5). Unlike traditional neural networks, RNNs incorporate recurrent connections, allowing them to maintain and utilize a kind of internal memory to capture patterns over time [66].

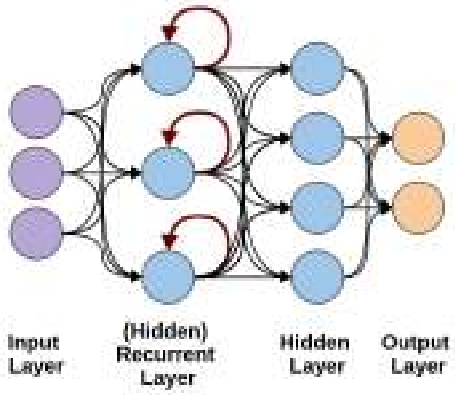


Figure 3.5. Recurrent Neural Network (RNN)[3]

[2] Image source: https://docs.ecognition.com/v9.5.0/Resources/Images/ECogUsr/UG_CNN_scheme.png

In an RNN, information flows through the hidden layers while being updated at each time step, allowing the network to learn to model complex contexts and temporal dependencies in the input data.

✓ **Fully Connected Neural Networks (Fully connected NN):** These consist of layers of interconnected neurons, where each neuron in one layer is connected to all neurons in the next layer [67], as illustrated in Figure 3.6. Multi-Layer Perceptrons (MLPs) are a prime example of fully connected neural networks. These networks offer the advantage of versatility, as they can handle input and output spaces of different sizes. This flexibility makes them suitable for a wide range of applications, particularly when dealing with structured data or problems where intricate relationships need to be learned.

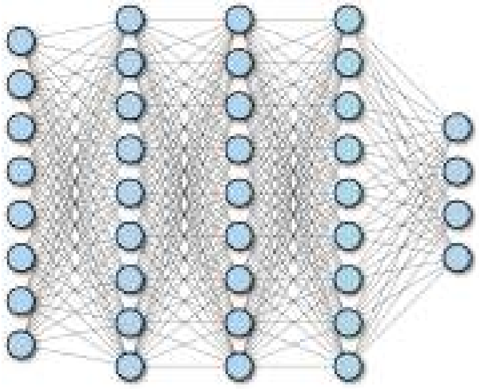


Figure 3.6. Fully Connected Neural Network[4]

Once the architecture is defined, the network is trained with the goal of minimizing the loss function. This loss function quantifies the disparity between the Q-value estimates derived from the network and the actual rewards obtained during interactions with the environment. Optimization algorithms like stochastic gradient descent [68] and the Adam algorithm [69] guide the weight updates during training. The choice of the optimization algorithm plays a crucial role as it can enhance the effectiveness, and stability of the

[3] Image source: `https://miro.medium.com/v2/resize:fit:553/0*xs3Dya3qQBx6IU7C.png`

[4] Image source: `https://www.oreilly.com/api/v2/epubs/9781491980446/files/assets/tfdl_0402.png`

training process, influencing the performance of the trained model. This choice can also influence the convergence speed and ability for generalization of the model and perform better with new data.

With training, the neural network becomes proficient in mapping states to Q-values without relying on a Q-table, as typically seen in traditional Q-learning. This approach effectively avoids the impracticality of storing and updating values for every conceivable combination of states and actions, particularly in high-dimensional problems, as discussed in Section 3.1.2.4.

Instead, the neural network, specifically the Q-Network in DQN, uses a continuous (it doesn't store discrete values for each state-action pair) and differentiable approach to estimate Q-values. The Q-Network is trained to learn a function that takes a state as input and generates Q-value estimates for all possible actions in that state [70]. This is accomplished by minimizing a loss function that quantifies the difference between the current Q-Network estimates and the targets provided by the target network [71]. The target network, a slower duplicate of the Q-Network, serves a crucial purpose in generating stable training targets during the learning process. While the Q-Network receives updates at each training step, the target network is updated more gradually by periodically copying the weights from the Q-Network.

The presence of both the Q-Network and the target network is essential to address a common challenge in training deep reinforcement learning models. The use of a target network helps stabilize the training process and prevents the network from oscillating or diverging during training [71]. By having a consistent set of target values, it enables more reliable and effective learning. Throughout the training process, the neural network gradually adapts so that its Q-value estimates become increasingly closer to the optimal values. As the agent interacts with the environment and accumulates training data, the Q-Network refines its estimates. This, in turn, enables the agent to make more informed decisions that maximize future rewards (Figure 3.7).

To perform this training process and minimize the difference between estimated Q-values and target Q-values, the agent's experiences are collected and stored in a memory known as the *replay buffer*.

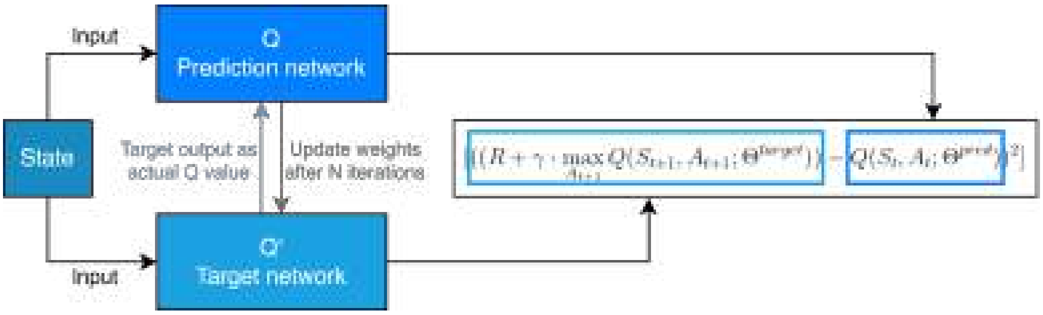

Input
Q
Prediction network
State
Target output as actual Q value
Update weights after N iterations
$[(R + \gamma \cdot \max_{A_{t+1}} Q(S_{t+1}, A_{t+1}; \Theta^{target})) - Q(S_t, A_t; \Theta^{pred}))^2]$
Q'
Target network
Input

Figure 3.7. Prediction and target network in DQN

Replay Buffer

The replay buffer is a fundamental component of DQN; it stores past experiences in memory and reuses them for learning. Instead of using each experience immediately to train the neural network, they are stored in memory (see Figure 3.8). During the agent's interaction with the environment, each time it takes an action, experiences a state change, and receives a reward. From this information, a tuple is generated containing the current state, the action taken, the received reward, and the resulting state after the action [20]. The tuple is stored in memory and then randomly sampled.

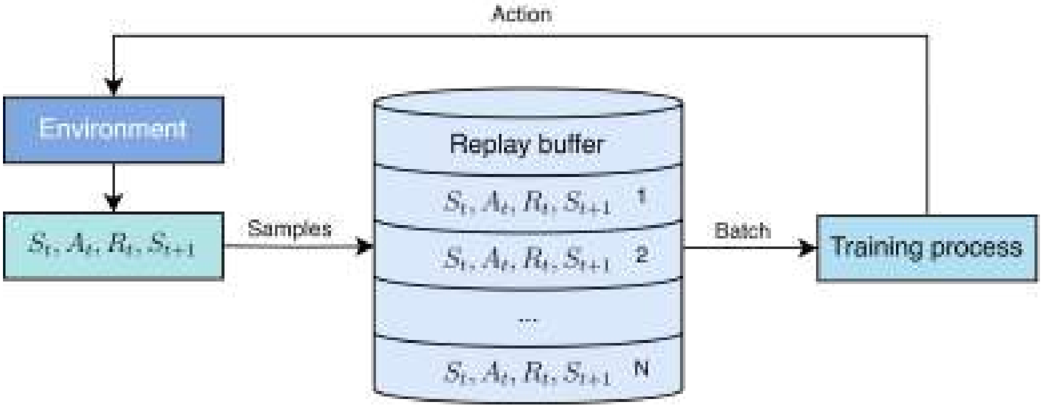


Figure 3.8. Replay buffer

Random sampling from the replay buffer provides several advantages. First, it helps prevent the neural network from being biased toward specific experience patterns that could arise if it were trained only with the most recent experiences [72]. Second, by shuffling experiences in the buffer, they are randomly selected in groups or batches. This helps reduce the temporal correlation between state-action transitions, contributing to more stable and faster convergence.

Furthermore, the use of the replay buffer is a key feature that allows DQN to be an off-policy algorithm, meaning it can learn from past experiences without directly relying on the agent's current policy. This also allows DQN to perform less frequent updates to the neural network, which is beneficial for training stability [63]. That's why, in addition to the learning rate and discount factor, the memory size is an important parameter in the replay.

Remarks on DQN

In addition to its solid theoretical foundation, DQN has demonstrated its effectiveness in a variety of practical applications, from games like Atari [73] to autonomous driving [74] and recommendation systems [75]. In these contexts, deep neural networks enable the algorithm to learn more sophisticated representations of states and actions, making it an attractive choice for addressing real-world problems. Although there are other notable methods such as A3C (Asynchronous Advantage Actor-Critic) [76] and PPO (Proximal Policy Optimization) [77], DQN presents particular advantages that make it especially suitable in certain scenarios.

One of the main advantages of DQN lies in its ability to address high-dimensional and complex environments, such as those found in Atari games [78] used as benchmarks in early research. These spaces can be large and continuous, and DQN can handle them more effectively. This feature is essential for practical applications where state spaces are rich in information and cannot be easily represented in a tabular manner.

While it's true that methods like A3C and PPO also possess strengths in handling complex environments and learning from past experiences, DQN offers a significant advantage with its ability to learn from past observations, allowing for the creation of more stable and generalized policies. Since DQN stores and reuses past experiences in its memory, it can mitigate issues of correlated sampling and noise in reward data. This property gives DQN greater stability in learning and the ability to obtain more robust policies. Additionally, A3C and PPO might require more computational resources due to their synchronous or asynchronous nature, which could be a limiting factor in real-world applications where efficiency is crucial. Also, they might need more samples; if data is limited, they could be more constraining.

Furthermore, while A3C and PPO excel in handling continuous action spaces, DQN's strength lies in its ability to effectively address both discrete and continuous action spaces. This versatility makes DQN well-suited for traffic management scenarios, where a combination of discrete decisions (such as lane changes or traffic light controls) and continuous actions (such as vehicle speed adjustments, if needed) may be required.

Based on the advantages of DQN highlighted above, the decision was made to use this algorithm to address the case study in this project, considering its ability to tackle high-dimensional and complex environments, essential features in our research, and the ability to take discrete decisions (changes of traffic lights phases). Next, we will go deeper into the research scenario we are addressing.

3.1.4 State of the Art of RL in Traffic Application

In the context of traffic, reinforcement learning systems have emerged as a dynamic and adaptable solution for optimizing various aspects of traffic flow. This technique empowers the real-time adjustment of traffic lights timings, routing strategies, and other variables to navigate complex and ever-changing urban environments effectively. The utilization of RL in traffic management is particularly promising for densely populated urban areas as it addresses critical challenges in traffic optimization.

These type of approaches enable smart traffic management through different technologies that involve real-time data collection via sensors and cameras, intelligent traffic lights control implementations [20], recommendation systems [79], and the integration of artificial intelligence models [80]. These data serve as inputs for advanced systems that employ various technologies. For instance, adaptive traffic light control systems, leveraging AI algorithms, have been instrumental in optimizing traffic flow by collecting real-time data, identifying patterns, and considering current traffic conditions to minimize congestion. This integration of data-driven technologies not only enhances the performance of traffic management systems but also contributes to more efficient and adaptive urban traffic flow.

An important application of RL in the context of traffic management revolves around autonomous vehicles, where it is employed to make driving decisions, taking into account factors such as vehicle speed, following distance, traffic lights, and lane changes. The decision-making process is influenced by real-time traffic conditions and individual objectives, illustrating the adaptability and intelligence of RL algorithms in ensuring safe and efficient autonomous vehicle navigation [81].

Furthermore, information and communication technologies, integral to autonomous vehicle applications, play a crucial role in enhancing the driving experience. Real-time navigation systems and mobile applications provide drivers with the necessary information to make informed route decisions and avoid traffic congestion. These technologies, in conjunction with autonomous vehicle systems, offer a seamless and safer driving experience. Vehicle detection systems further facilitate the collection of precise traffic data, aiding in swift incident identification. However, while some autonomous features are available to consumers (for example, those of Tesla vehicles), fully autonomous vehicles that can navigate complex urban environments are not yet widely accessible. The development and deployment of autonomous vehicles still face regulatory, safety, and technological challenges [82] before they become a commonplace mode of transportation for the average citizen.

In recent years, vehicle traffic related applications have been increasingly exploring the capabilities of deep reinforcement learning (explained in the previous subsection 3.1.3.1.1), thereby demonstrating a growing trend toward the utilization of more advanced and sophisticated techniques in the field.

3.1.4.1 Traffic Light Control Systems

Among the applications of deep RL, traffic light control systems stand out as a prime example, since algorithms like DQN excel in handling the intricate aspects of traffic control and signal optimization, surpassing the capabilities of traditional RL methods.

In the domain of traffic control systems, it's evident that traditional fixed-time traffic light phases, although widely employed, often fall short of optimizing traffic flow in real-world scenarios. A light phase refers to a specific state of a traffic light, which determines the allocation of right-of-way to various directions of traffic at an intersection. It includes the combination of lights (e.g., red, green, yellow) and their respective timing, indicating when vehicles, pedestrians, or other road users should stop, yield, or proceed [83].

The static approaches for traffic lights, like those currently in use in Medellin (our Case Study, chapter 4), typically rely on pre-determined timings that fail to adapt to changing traffic conditions, resulting in inefficiencies and congestion during peak hours and unexpected events. To address these limitations and improve traffic management, many researchers have turned to RL and DQN techniques, which have demonstrated significant promise in crafting dynamic, adaptive traffic control solutions.

For example, in one notable contribution, Vidali *et al.* [20] introduced an adaptive traffic lights management system that leverages an occupancy-based state representation and a fixed set of predefined actions. They employed a DQN agent to optimize traffic efficiency at a 4-way intersection though traffic phases control. To enhance the learning process, the authors incorporated a MLP and experience replay, while adopting a cumulative waiting time-based reward system. This approach demonstrated promise in traffic management. However, it's important to note that the experimental setting primarily utilized a synthetic grid scenario, lacking real-world data and a broader range of environmental complexities.

In pursuit of a similar goal, Gao *et al.* adopted an approach that aims to use the power of CNNs to extract features such as vehicle positions, speeds, and traffic light states from real-time traffic data. Their algorithm, built upon this feature-rich foundation, effectively determines the optimal traffic lights control strategy. To enhance the robustness of their model, they integrated critical reinforcement learning components, including experience replay and a target network. Much like Vidali's work, these authors tested their approach in a 4-way intersection scenario. However, it's worth noting that, in this case, they relied on synthetic data to facilitate their experiments.

Similarly, Van der Pol's research emphasizes the significance of hyperparameter tuning in deep RL [84]. The study systematically explored the influence of factors like learning rate, normalization, and network architecture on performance stability. Additionally, she incorporated prioritized experience replay in her methodology. The author extended her investigation to encompass multi-agent coordination within a 2x2 grid structure, involving up to four agents situated at different intersections. Despite the comprehensive exploration of hyperparameters and multi-agent coordination, the research relied on synthetic data and grid-based simulations, which may not fully capture the intricacies of real-world traffic dynamics.

On the other hand, Kővári *et al.* [85] present an interesting approach aimed at promoting sustainability within traffic management. Their work extends beyond conventional metrics, such as waiting time, travel time, and queue length, to include a comprehensive evaluation of various sustainability aspects, including CO2 emissions, NOx levels, CO emissions, and more. In their research, the authors also compare the performance of two distinct algorithms (DQN and Policy Gradient (PG)). It's worth noting that their study, like the previously described, is constrained by the use of an isolated intersection scenario, thereby suggesting the need for further research that accounts for more complex traffic environments and interconnections.

Now, regarding more realistic scenarios, we find examples such as the work developed by Fuad *et al.* [86], where a traffic network of the city of Jakarta was used as the simulation scenario, considering real-world data. This work presents a promising approach with its adaptive reward mechanism and real-world data usage, resulting in improved traffic throughput. However, there are potential limitations, including the pressure calculation method that assumes a direct proportionality between lane capacity and length, which may not always hold true.

The state of the art in traffic management through traffic lights control showcases several innovative approaches to optimize urban traffic flow. Researchers like the ones presented (and others mentioned in Table 3.1) have made significant strides in leveraging deep RL techniques, CNNs, and advanced RL components to enhance traffic lights control strategies. Their contributions have demonstrated promise in improving traffic efficiency and addressing various key performance metrics. Notably, Kővári's focus on sustainability metrics offers a broader perspective on traffic management, while Van der Pol's emphasis on hyperparameter tuning enhances our understanding of the fine points of deep RL. Fuad's work, which incorporates real-world data in the dynamic context of Jakarta's traffic network, provides valuable insights into adaptive reward mechanisms. These studies collectively inform and inspire our own research project, as we consider incorporating elements and insights from these diverse approaches to tackle the challenges of traffic management in a real-world urban environment, while considering the impact of traffic on air quality.

Yet, considering the practical application of AI-based vehicle management in real cities, it is crucial to acknowledge that most developments have been primarily tested and refined in simulated environments. Although the potential for real-world implementation is evident, a significant gap remains between research and practical application. This gap is primarily due to the profound implications that could result from system failures in real-world

Table 3.1. Previous DRL-based works for traffic light control

Name of the resource	Autor	State rep.	Type of NN
Traffic Signal Control via Reinforcement Learning for Reducing Global Vehicle Emission [85]	B. Kővári, L. Szőke, T. Bécsi, S. Aradi, and P. Gáspár	Occupancy	MLP
Adaptive Deep Q-Network Algorithm with Exponential Reward Mechanism for Traffic Control in Urban Intersection Networks [86]	M. R. T. Fuad, E. O. Fernandez, F. Mukhlish, A. Putri, H.Y. Sutarto	Green phase, density of incoming vehicles, queue length	MLP
A Deep Reinforcement Learning Approach to Adaptive Traffic Lights Management [20]	A. Vidali, L. Crociani, G. Vizzari, S. Bandini	Cell occupancy	MLP
IntelliLight: A Reinforcement Learning Approach for Intelligent Traffic Light Control [80]	H. Wei, G. Zheng, H. Yao, and Z. Li	Queue lenght, waiting time, image representation	CNN
Deep Reinforcement Learning for Coordination in Traffic Light Control [84]	E. van der Pol	Position matrix, speed, acceleration	NIPS, Nature, DDQN
Adaptive traffic signal control: Deep reinforcement learning algorithm with experience replay and target network [72]	J. Gao, Y. Shen, J. Liu, M. Ito, and N. Shiratori	Position, speed	CNN
Heterogeneous Multi-Agent Deep Reinforcement Learning for Traffic Lights Control [87]	Calvo, J. Dusparic, I	Position, speed	DDDQNs

traffic scenarios, given the critical nature of traffic management and its direct impact on public safety. Thus, simulation serves as a crucial tool for experimenting and refining algorithms in controlled environments, mitigating the risks associated with incorrect operation. Achieving the right balance between innovation, safety, and responsibility is essential in the development of RL-based vehicle management systems, making simulation an indispensable component of the process [88].

Chapter 4

Case Study

4.1 Low Emission Zone in Medellin

In 2018, the Metropolitan Area of the Aburrá Valley (AMVA) [1] defined a Low Emission Zone (LEZ) based on data collected by the traffic monitoring station called *Tráfico Centro* [39] between 2014 and 2017. This LEZ is bounded from south to north by *Calle San Juan* and *Calle Echeverri*, and from west to east by the *Avenida Ferrocarril* and *Carrera Girardot*, covering the downtown area of Medellin, as shown in Figure 4.1.

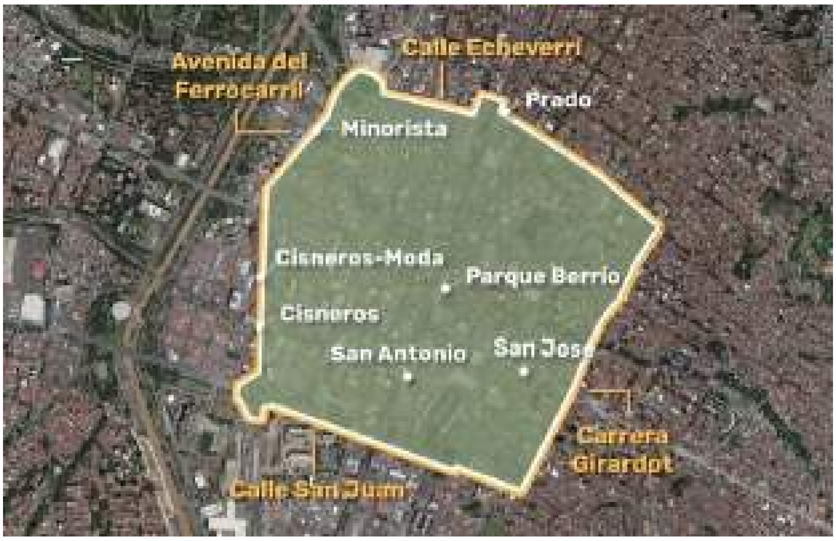


Figure 4.1. Medellin's Low Emission Zone polygon[1]

[1] Image taken from [39]

As part of the planned initiatives for this area, several entities in the environmental and mobility sectors are seeking strategies to reduce emissions from mobile sources (vehicles). The Integrated Traffic and Transportation Center (*CITRA - Centro Integrado de Tráfico y Transporte*) [28] is an agency that is part of the Medellin Mobility Secretariat and its role is to monitor and process traffic data to make better decisions that enable mobility management through technological infrastructure and information systems. This organization provided a dataset used to characterize vehicular traffic in Medellin's LEZ. This dataset is described in the following section.

4.2 Dataset

The vehicular traffic dataset provided by CITRA contains 134,824,218 observations, resulting from information captured from cameras located in the city of Medellin, corresponding to the Support for the Traffic Light Network (*ARS - Apoyo a la Red Semafórica* [89]), and CCTV systems of the city's Intelligent Mobility System (*SIMM - Sistema Inteligente de Movilidad de Medellin*) [90].

The information was collected between the years 2020 and 2023, and the available fields are shown in Table 4.1.

Table 4.1. Description of the dataset fields.

Variable	**Description**
Class_1	No. of vehicles with a length between 0 and 3 meters
Class_2	No. of vehicles with a length between 3 and 6 meters
Class_3	No. of vehicles with a length higher than 6 meters
Class_4	No. of motorcycles
Speed	Speed in km/h
Location	Latitude and longitude
Direction of travel	Direction in which data is captured (NS, SN, WE, EW)
Date_time	Timestamp in YYYY-MM-DD format
Road	Name of street or avenue
Records	No. of captured records per observation
Occupancy	Lane occupancy percentage[2]
Intensity	Total number of vehicles

In this traffic dataset, each observation represents a specific timestamp, so the *Records* column indicates the number of vehicles reported by CITRA for each specific observation. Consequently, the different data presented correspond to the average values of each variable for that number of vehicles recorded by the cameras in each specific time interval.

Previous to the data preprocessing presented in subsection 4.2.1, and given that the study area for the development of this project corresponds to the LEZ, a demarcation of this area was carried out using Python [92] and Google Earth [93], considering the geographical coordinates (*Location* field) and the road name (*Road* field) in the dataset. This was made to ensure that the resulting data came exclusively from cameras located within the LEZ. Figure 4.2 shows the data capture points. Finally, the number of resulting observations for the area of interest was 5,290,071.

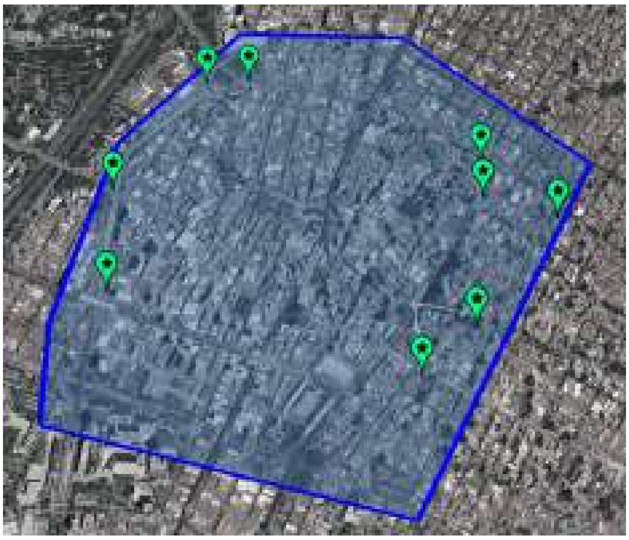


Figure 4.2. Data capture points within the LEZ

4.2.1 Data Preprocessing

Real-world data often contain missing fields, irrelevant data, and errors. Therefore, a proper preprocessing is necessary [94] to obtain reliable results and make more accurate decisions. This is an essential stage to ensure the

[2] Proportion of time a lane is occupied by vehicles, expressed as a percentage of the total time interval under consideration [91].

Occupancy (%) = Total time lane is occupied/Total time interval × 100 integrity of the results. After an initial analysis, observations from the year 2020 were removed due to their atypicality attributed to the mobility restrictions imposed during Covid-19 quarantines. Afterwards, observations with *null* data were also removed, and outliers were filtered. Additionally, some columns were added to the dataset (*hour*, *day*, *month*, *year*, *holiday* indicator) for subsequent visualization and analysis. Furthermore, considering that the received information is a product of grouped data (*Records*), the average was calculated for variables *Class_1*, *Class_2*, *Class_3*, *Class_4* and *Intensity*, dividing the number of vehicles by the number of records of the observation (e.g. $Class\_1[i]/Records[i]$ where i is the index of the observation) through the entire dataset. The resulting units are vehicles per observation (called *veh/obs* from now on).

4.3 Time Series Analysis for Vehicular Traffic Data in Medellin's LEZ

Data analysis, and in this case time series analysis, is an essential tool for identifying patterns and trends inherent in the collected data. It also allows understanding the changing dynamics of the vehicular flow in the LEZ. Through the exploration of observations over time, it is possible to identify seasonal and cyclical trends that can be crucial for traffic management strategies. In addition to the temporal component, variables such as traffic flow, speed, travel time, and density can be evaluated.

In the context of time series analysis, a wide range of statistical methods can be employed to extract such information. Among these methods, tools such as correlation [95] and Principal Component Analysis (PCA) [96] are highlighted, often supported by different types of graphs to represent and understand the data. Among the most used are line plots, which show the evolution of the series over time, allowing the identification of trends and fluctuations; and boxplots [97], which are useful for visualizing the dispersion and presence of outliers in specific intervals.

The visualization and feature selection of the LEZ will be detailed below.

4.3.1 Visualization and Analysis

In this section, the visualization and analysis of patterns in LEZ's vehicular traffic data are explored. First, the identification of periodicity in the time series is performed, highlighting trends and repetitions over different time intervals. Identifying seasonal and cyclical patterns is crucial for a better understanding of traffic dynamics in the LEZ, which in turn can inform strategic decisions in urban transportation planning.

Additionally, the analysis of vehicular circulation categories is addressed, examining how the flow of different types of vehicles varies on each road corridor. This characterization provides insights into the composition of traffic in the area of interest. Furthermore, a review of traffic by day of the week is conducted to discover differences between weekdays, weekends, and holidays.

The exploration of specific road corridors is also an essential component of this analysis. By segmenting the time series based on road corridors, specific patterns in the variables describing vehicular traffic can be identified and evaluated. This segmentation provides relevant information for the individualized management of different road areas within the LEZ.

Finally, the use of boxplots enhances visualization by providing a representation of the dispersion and distribution of traffic values at different times and road corridors. Boxplots allow the identification of outliers, providing information about data variability and distribution.

4.3.1.1 Periodicity

Temporal decomposition is an essential tool in vehicular traffic analysis, as it allows breaking down observed data into three fundamental components: trend, seasonality, and the observed part [98]. In this context, we have examined three key variables: *Occupancy*, *Intensity*, and *Speed* between January 2021 and January 2023.

The figures that will be described in next subsections 4.3.1.1.1, 4.3.1.1.2, and 4.3.1.1.3 have three components, each of them with its corresponding subplot[3]:

✓ The first subplot corresponds to the *observed* component, which represents the values or measurements over time and includes all underlying patterns, encompassing both the trend and seasonality.

✓ The second subplot represents the *trend* component showing the underlying direction in the data, which may be increasing or decreasing over time.

[3] This decomposition was made with Python's *statsmodel* library [99].

✓ The third subplot shows the *seasonal* component and represents the regular and repeating patterns that occur at specific intervals within the time series. These patterns often correspond to calendar-related cycles, such as daily, weekly, monthly, or yearly patterns [98]. In this component, the y axis shows values (e.g. -0.5 to 0.5, and -1 to 1, as seen in the next subsections), such that the maximum positive values indicate peaks where data reached its highest points during the observed period, and the most negatives indicate the lowest data points. These values indicate how much the data deviates from its long-term average at each point in the seasonal cycle [100].

4.3.1.1.1 Occupancy

Observed occupancy data (illustrated in Figure 4.3), experienced a peak between May and June 2021, suggesting a high traffic season during that period. After this peak, occupancy values began to rapidly decline and reached a stabilization point with a slight increase between November and December, 2021. Additionally, a pronounced pattern in the seasonal component was identified, with repetitive shapes in occupancy values from January 2021 to 2023.

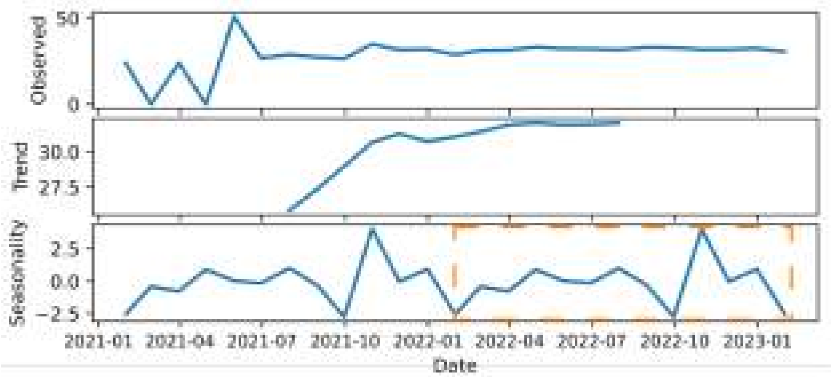


Figure 4.3. Temporal decomposition for Occupancy (%)

4.3.1.1.2 Intensity

Intensity data shows a similar behavior to *Occupancy* (Figure 4.4). It reached its peak for observed values (upper subplot) in May and June 2021 and then decreased, but with a steeper decline than *Occupancy*, reaching a minimum in July of the same year, where values stabilized until January 2022 when a decrease in flow was observed, followed by a slight but steady increase until the end of the observation period. The trend graph reflects this variation. Similar to *Occupancy*, a seasonal pattern is detected, repeating from January 2021 to January 2023.

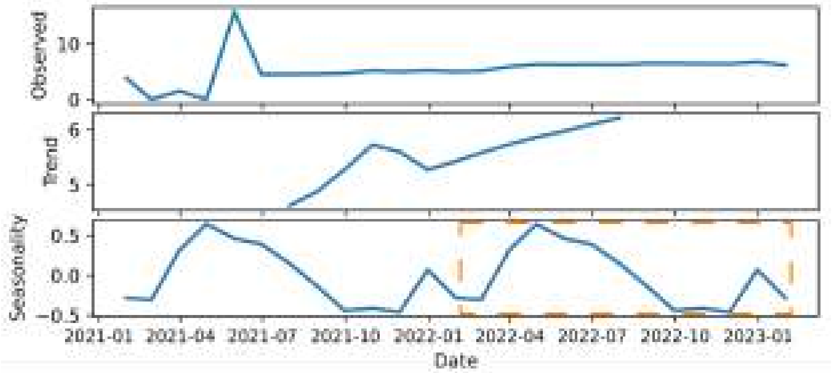


Figure 4.4. Temporal decomposition for Intensity (veh/obs)

4.3.1.1.3 Speed

In the case of *Speed*, plotted in Figure 4.5, an inverse dynamic is observed compared to *Occupancy* and *Intensity*. When *Occupancy* and *Intensity* decreased between June and July 2021, *Speed* experienced a peak. Subsequently, speed values decreased and stabilized, which is more evident around April 2022 (these observations can be seen both in the graph of observed values and in the trend). However, a slight but steady decrease is observed towards the end of the observation period. The trend graph reflects this evolution. Like the other two variables, there is a seasonal pattern that repeats from January 2021 to 2023. In this case, where *Intensity* presented curves with lower values, the highest values for speed were evident. This can be observed, for example, between the months of October and January and between March and April considering the observable behavior in the seasonality subplot.

It is important to note that, in this case, the focus is on the temporal description and decomposition of *Occupancy*, *Intensity*, and *Speed* variables in the LEZ. Although these temporal trends are notable and provide valu-

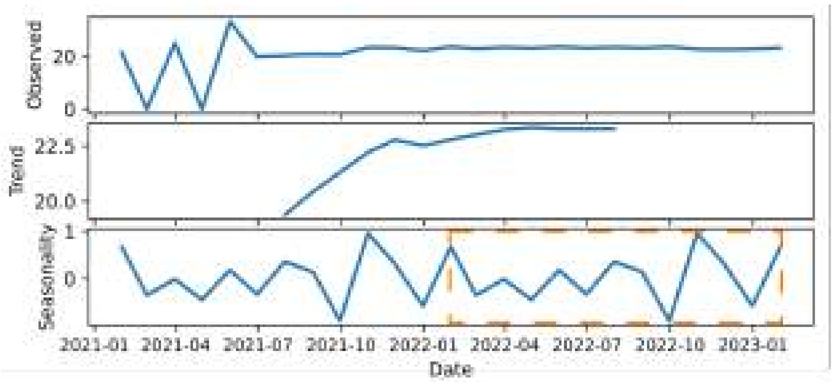


Figure 4.5. Temporal decomposition for Speed (km/h)

able information about vehicular traffic dynamics, we will not delve into the analysis of the underlying causes of these patterns in this particular context. The main objective is to provide an overview of how these variables change over time in the LEZ, which can serve as a basis for decision-making related to traffic management and mobility in the area.

4.3.1.2 Analysis by Month and Day of the Week

Since the data in the dataset exhibited cyclic behavior over time, and vehicular traffic has temporal variations, monthly averages were calculated for *Occupancy*, *Intensity*, and *Speed* for the years 2021 and 2022[4]. This was conducted to identify the most critical months for mobility in the LEZ and to evaluate how traffic varies between days of the week.

4.3.1.2.1 Monthly Traffic

In the case of *Occupancy* (see Figure 4.6), there was a general increase between 2021 and 2022. More specifically, the highest occupancy percentage occurred in May (51.0%) of 2021, followed by October (35.08%), while the lowest was in January (23.97%) of the same year. In 2022, the highest average occupancy percentages were in August (33.17%) and September (32.88%). In contrast, as observed in the previous year, the lowest occupancy was in January (28.9%).

On the other hand, *Intensity* (see Figure 4.7) reached its highest averages

[4] No observations were obtained for February and April 2021.

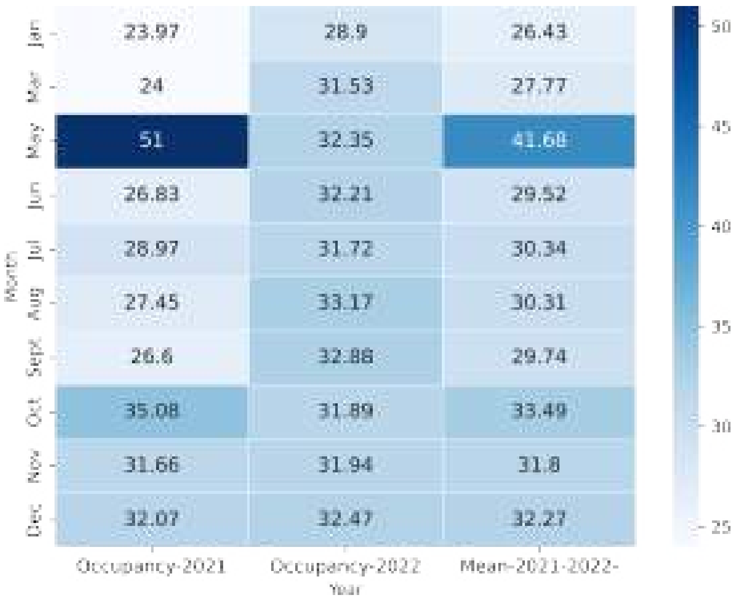


Figure 4.6. Average occupancy (%) by month for 2021 and 2022

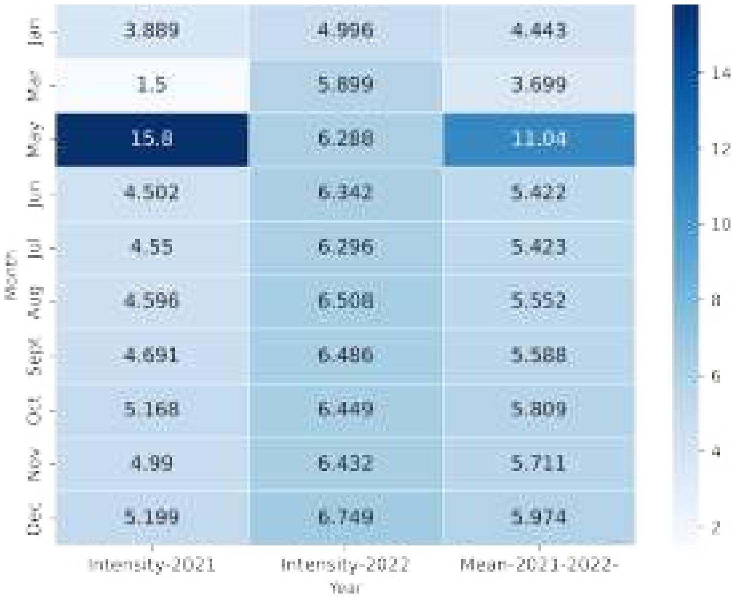


Figure 4.7. Average intensity (*veh/obs*) by month for 2021 and 2022

for 2021 in May (15.8 *veh/obs*) and December (5.19 *veh/obs*); the lowest were observed in March (1.5 *veh/obs*) and January (3.889 *veh/obs*). In 2022, December had the highest *Intensity* value (6.749 *veh/obs*) compared to the other months, and January had the lowest (4.996 *veh/obs*). Generally, there was an increase between 2021 and 2022, resulting in a behavior similar to that obtained for *Occupancy*.

Using the same procedure, monthly values for *Speed* were calculated (see Figure 4.8). When analyzing the monthly behavior for the same period, it was found that for 2021, the highest speed was evident in May (33.40 km/h) and March (25.25 km/h), while the lowest was in June (19.99 km/h). In 2022, the lowest average speed occurred in November (22.67 km/h), and the highest was in May (23.79 km/h). Comparing the years 2021 and 2022, it cannot be conclusively determined whether there was a general decrease or increase.

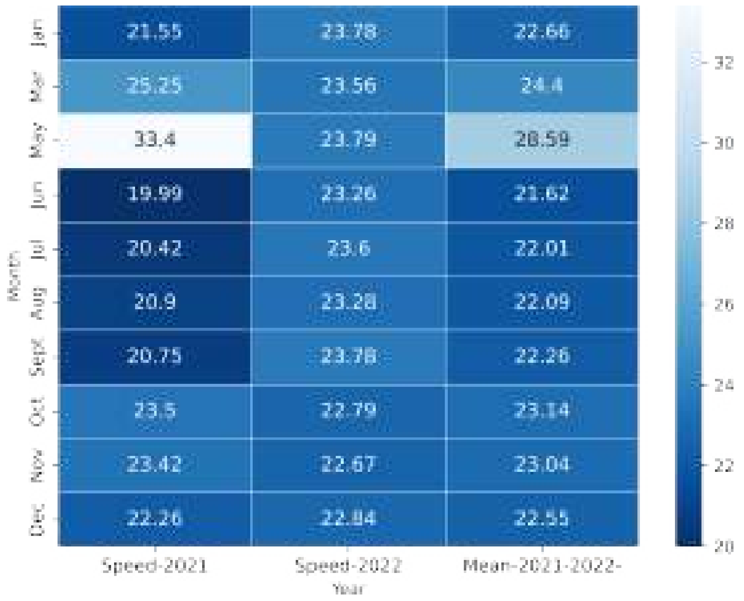


Figure 4.8. Average speed (km/h) by month for 2021 and 2022

4.3.1.2.2 Weekly Traffic

The analysis of *Occupancy* data (Figure 4.9) on different days of the week shows that Fridays are the busiest day of the week, with an average occupancy of approximately 32.60%, suggesting higher mobility activity. In contrast, Sundays show the lowest average occupancy, at approximately 27.0%, indicating lower road utilization, possibly due to reduced vehicle circulation on Sundays when work activities are reduced.

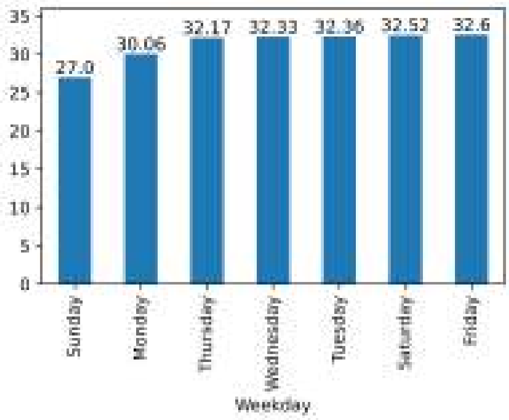


Figure 4.9. Average occupancy (%) for weekdays of 2021 and 2022

On the other hand, Saturdays stand out with the highest average intensity (Figure 4.10), recording approximately 6.24 *veh/obs*, followed by Fridays with 5.97 *veh/obs*, indicating higher vehicle activity and flow during these weekend days. Sundays have the lowest *Intensity*, at approximately 5.26 *veh/obs*, suggesting a similar pattern to occupancy on this day.

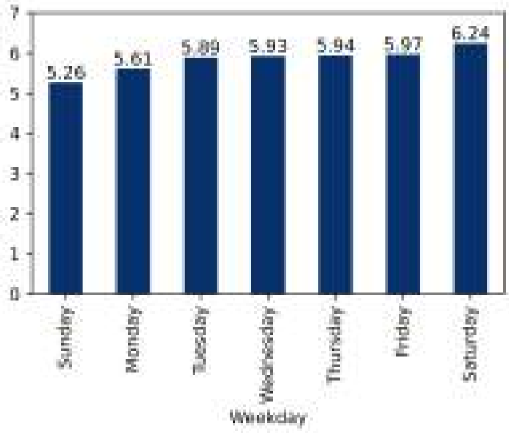


Figure 4.10. Average intensity (*veh/obs*) for weekdays of 2021 and 2022

Regarding *Speed*, the results shown in Figure 4.11 indicate that Sundays have the highest average speed, registering approximately 25.09 *km/h*, suggesting smoother traffic conditions and higher speeds compared to other days of the week. In contrast, Fridays have the lowest average speed, at around 22.13 *km/h*.

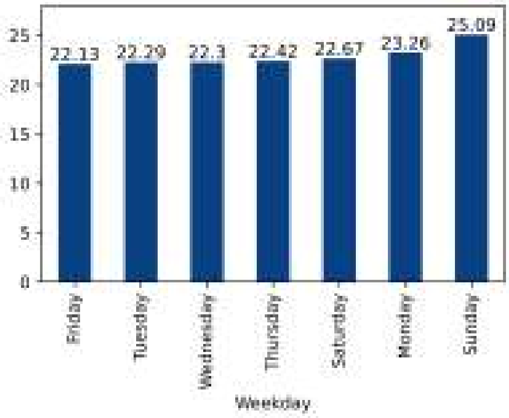


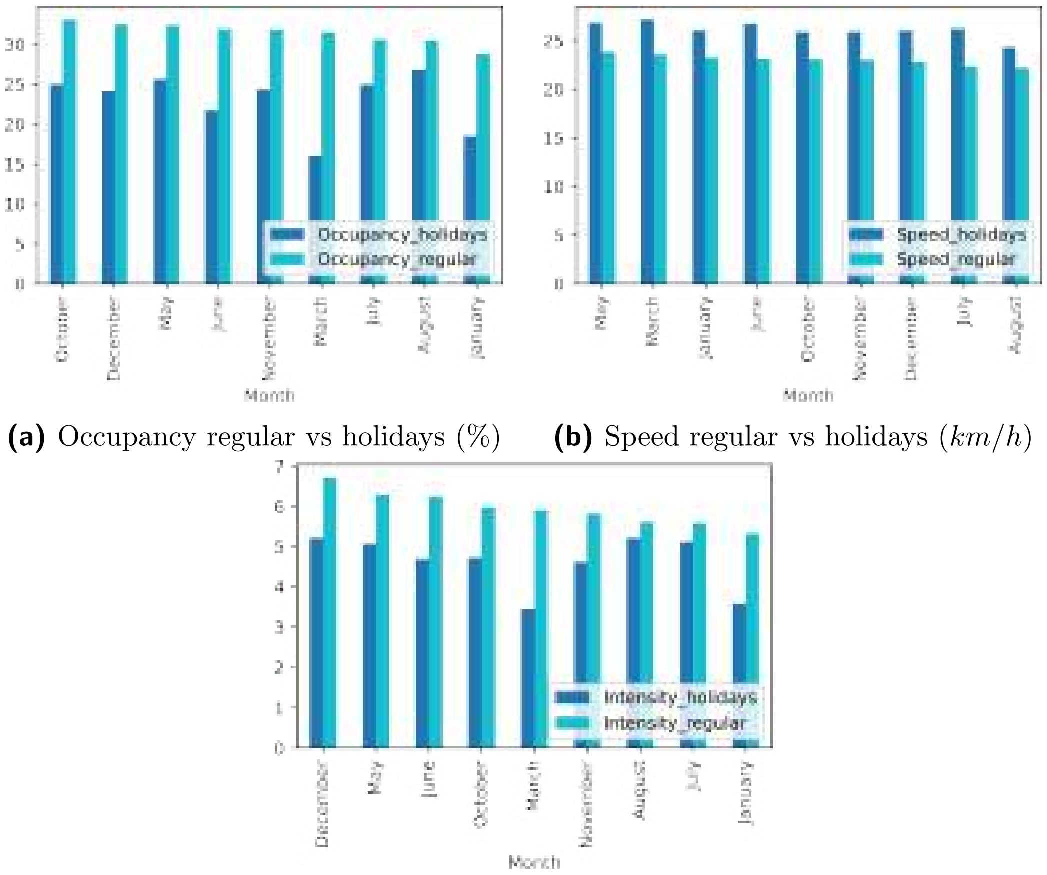
Figure 4.11. Average speed (km/h) for weekdays of 2021 and 2022

4.3.1.2.3 Traffic on Holidays

Colombia is a country with 18 holidays, one of the highest amounts in the world [101], which impact traffic as many people take days off or reduce their working hours. For this reason, to identify if there is a significant difference in mobility between regular days and holidays, monthly and daily averages were evaluated for the variables of *Occupancy*, *Intensity*, and *Speed* for 2021 and 2022.

The results obtained (see Figure 4.12) reflect lower occupancy (Figure 4.12a) and intensity (Figure 4.12c) during holidays for all months of the year[5], while higher average speed (Figure 4.12b) was observed on all recorded holidays.

[5] No observations were obtained for February and September (no holidays) and April (no observations available for holidays in either year).



(a) Occupancy regular vs holidays (%) **(b)** Speed regular vs holidays (km/h)

(c) Intensity regular vs holidays (veh/obs)

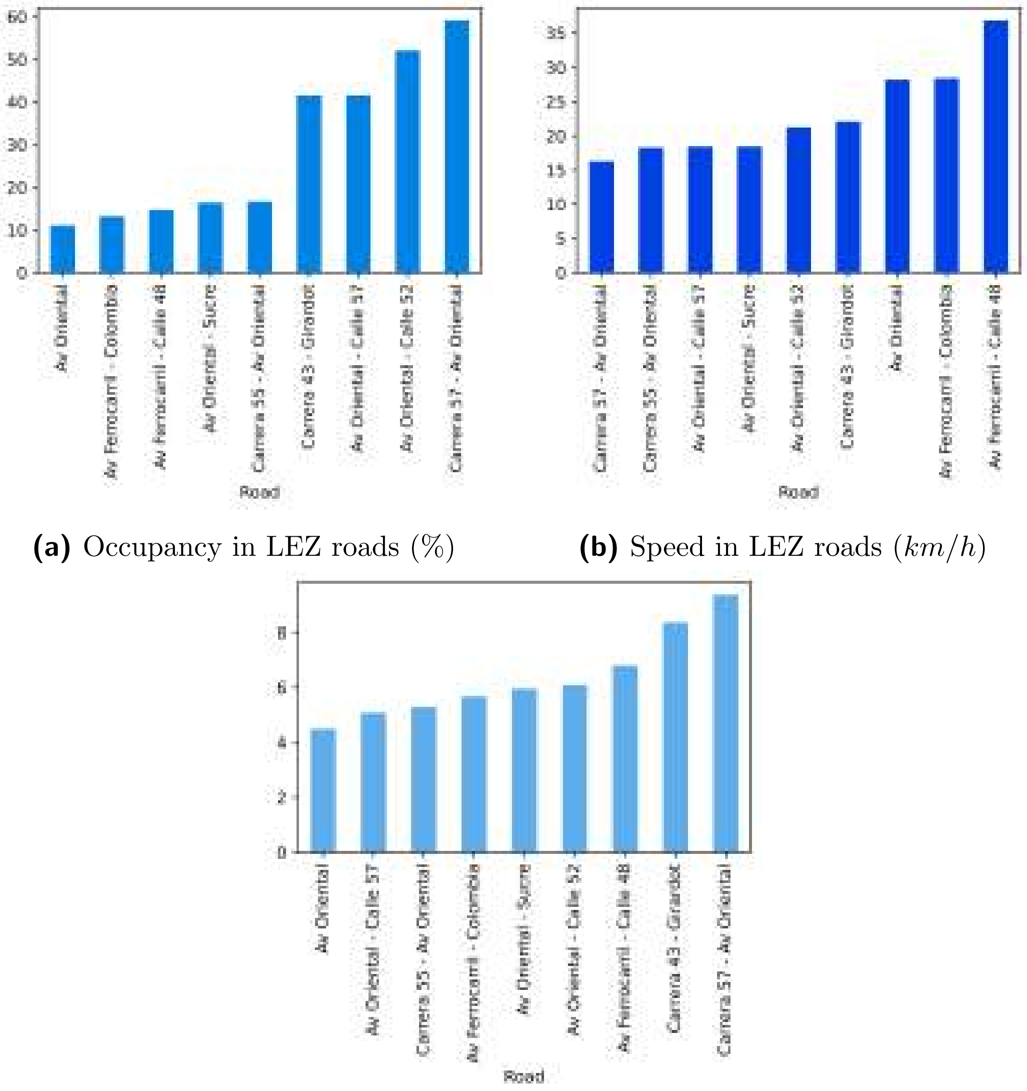
Figure 4.12. Comparison between regular and holidays

4.3.1.3 Analysis by Road Corridor and Vehicle Category

Although all data capture points are located in the LEZ, each road may exhibit different behavior, reflected in traffic descriptive variables (*Occupancy*, *Intensity*, and *Speed*), as well as in the type of vehicles that circulate. For this reason, the following analyses were performed.

4.3.1.3.1 Analysis by Road

When evaluating the average occupancy values (Figure 4.13a), *Carrera 57 - Avenida Oriental*, and *Avenida Oriental - Calle 52* presented the highest occupancy values, while *Av. Oriental* and *Av. Ferrocarril - Colombia* showed



(a) Occupancy in LEZ roads (%)

(b) Speed in LEZ roads (km/h)

(c) Intensity in LEZ roads (veh/obs)

Figure 4.13. Variables results in LEZ's roads

the lowest values.

On the other hand, the *Av. Ferrocarril - Calle 48* had the highest speeds, and the *Carrera 57 - Avenida Oriental* recorded the lowest speeds (Figure 4.13b). In this case, on the road where the lowest average speed was observed, the highest occupancy was obtained. This may provide clues to an inverse relationship between these variables. This relationship was not as clear when comparing *Intensity* and *Speed* variables.

Vehicle intensity (Figure 4.13c) showed its highest values on *Carrera 57 - Avenida Oriental* and *Carrera 43 - Girardot*, and the lowest values on the *Av. Oriental*.

4.3.1.3.2 Analysis by Vehicle Category

Considering that traffic varies constantly due to several factors, the averages of vehicle counts per record were graphed for different categories on each of the road corridors, as shown in Figure 4.14.

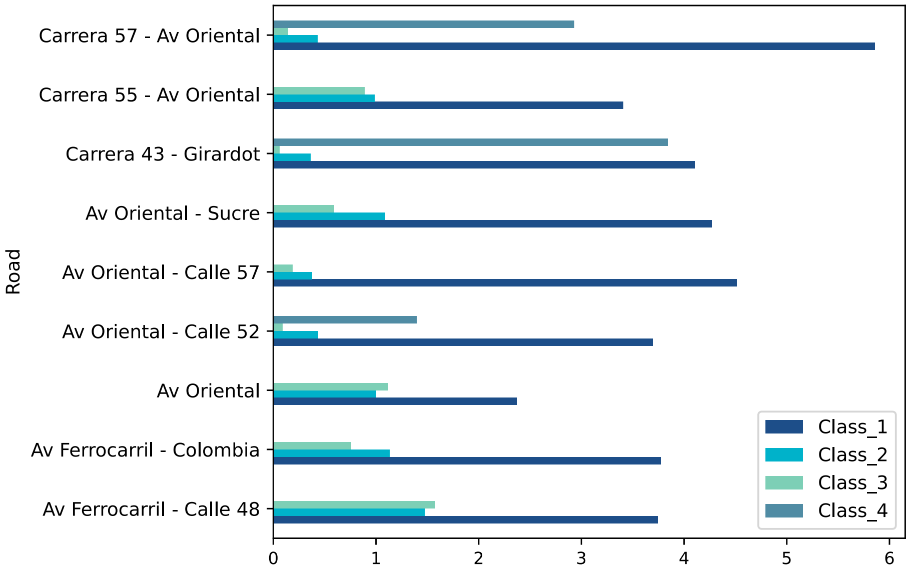


Figure 4.14. Average vehicle count circulating in LEZ according to Class_X variables (veh/obs)

From these, it was obtained that *Class_1*, which mainly represents small private cars, dominates in all the corridors, indicating its influence on the traffic in the LEZ. In general, there is less presence of motorcycles compared to the other types of vehicles. A significant presence is only noticeable on the corridors *Carrera 57 - Avenida Oriental*, *Carrera 43 - Girardot*, and *Avenida Oriental - Calle 52*. On the other hand, *Class_2* vehicles, which include those with lengths between 3 and 6 meters, are more noticeable on the corridors *Avenida Oriental - Sucre*, *Avenida Ferrocarril - Colombia*, and *Avenida Ferrocarril - Calle 48*, indicating a preference for larger private cars on these specific routes. The circulation of *Class_3* vehicles, which are longer than 6 meters, is is primarily noticeable on the corridors *Avenida Ferrocarril - Calle 48*, and *Avenida Oriental*. There is not a very significant presence on *Avenida Oriental - Calle 52*, and *Carrera 43 - Girardot*. This information may reflect the better suitability of certain roads for the circulation of longer and heavier vehicles.

Considering the above, the distribution of vehicles in the LEZ through different roads reveals clear traffic flow patterns. The constant predominance of small private cars (*Class_1*) suggests their central role in the daily mobility of the area. The limited presence of motorcycles on some roads could be attributed to specific traffic and infrastructure conditions, and the variation in the presence of *Class_2* and *Class_3* vehicles on certain roads points to the possible suitability of those routes for different types of vehicles based on their sizes and characteristics. These results are not only crucial for road network planning and optimization but also highlight the need for specific measures for each road, with a view to improving safety, efficiency, and (in relation to the interest of this project) traffic management. The analysis conducted demonstrates the importance of management that is sensitive to the presence of different types of vehicles in a dynamic area like the Medellin's LEZ.

4.3.2 Feature Selection

Feature selection is a process that involves reducing the number of variables in such a way that the most consistent and relevant ones are identified [23]. It can reduce the dimensionality of the input space, which can lead to faster convergence and improved computational efficiency while focusing the learning process on the most informative features. For the LEZ's data, this selection was carried out through different methods, described next.

4.3.2.1 Correlation

Correlation is a statistical measure that indicates the relationship between two variables through a coefficient that varies from -1 to 1, where a magnitude close to 1 indicates that the variables are highly related either directly (if positive) or inversely (if negative) [102].

In the context of data analysis, correlation allows exploring the underlying connections between the collected data, revealing patterns and dependencies that can provide a better insight into the data of interest. The use of correlation coefficients allows quantifying and qualifying the strength and direction of relationships between variables.

The correlation analysis between the different variables related to vehicular traffic in the LEZ reveals patterns that allow exploring its dynamics. A heatmap with correlation coefficients was used to visualize the relationships between the variables: *Occupancy*, *Class_1*, *Class_2*, *Class_3*, *Class_4*, Vehicle *Intensity*, and *Speed*. This map is shown in Figure 4.15, where significant coefficients are highlighted with red rectangles.

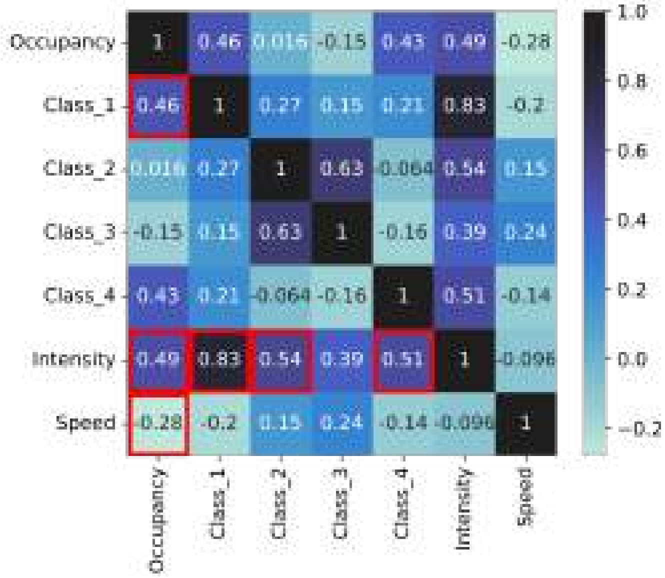


Figure 4.15. Correlation heatmap for LEZ variables

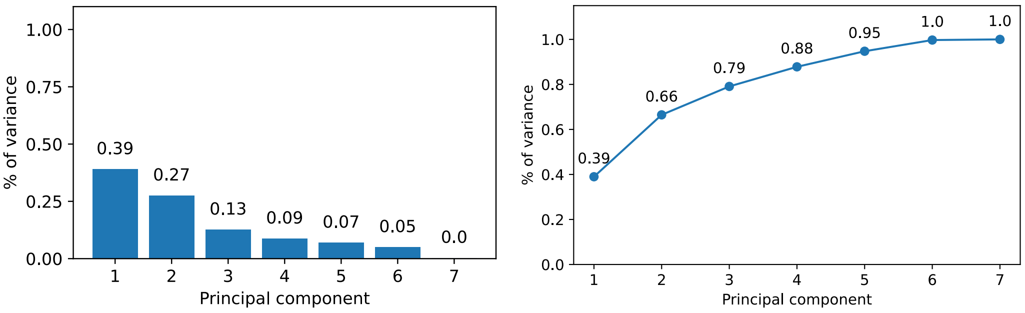
Firstly, a moderate negative correlation between *Speed* and road *Occupancy* (correlation coefficient: -0.28) was observed. This suggests that as road occupancy increases, speed tends to decrease, which could be related to congestion and reduced traffic flow. Furthermore, significant positive correlations were found between vehicle *Intensity* and *Occupancy* (correlation coefficient: 0.49), as well as with vehicle classes of type *Class_1*, corresponding to small private vehicles (correlation coefficient: 0.83), followed by *Class_2* vehicles, corresponding to vehicles with lengths between 3m and 6m (correlation coefficient: 0.54), and motorcycles, i.e., *Class_4* vehicles (correlation coefficient: 0.51). These relationships may indicate the general traffic flow trend on the road, highlighting the impact of *Class_1* vehicles on road occupancy and traffic density.

Additionally, albeit not very strong, a negative correlation was found between *Speed* and vehicle *Intensity* (correlation coefficient: -0.096), suggesting that higher traffic intensity tends to lead to lower speed. This could indicate the presence of congestion on the road and the need for traffic management measures to improve flow.

4.3.2.2 Principal Component Analysis (PCA)

Considering that there are significant correlations between some of the variables, it is feasible to apply PCA (Principal Component Analysis). This technique is used to reduce dimensionality in large datasets by creating new variables or Principal Components (PCs) that are linear functions of the original variables while preserving as much information as possible. Generally, components that can explain between 70% and 90% of the total variance are used [103]. PCA can also be used to identify the importance of variables in a dataset in terms of their contribution to the principal components. This contribution can be evaluated with variable weights: weights close to 1 or -1 indicate that the variable significantly influences a component, while weights close to zero represent a moderate or low contribution of a variable to a specific component.

The Python library called Scikit-learn [104] was used to apply PCA to the vehicular traffic-related variables in the LEZ dataset, obtaining the following results. Firstly, a total of 79% of the variance was captured with the first three components (Figure 4.16b).



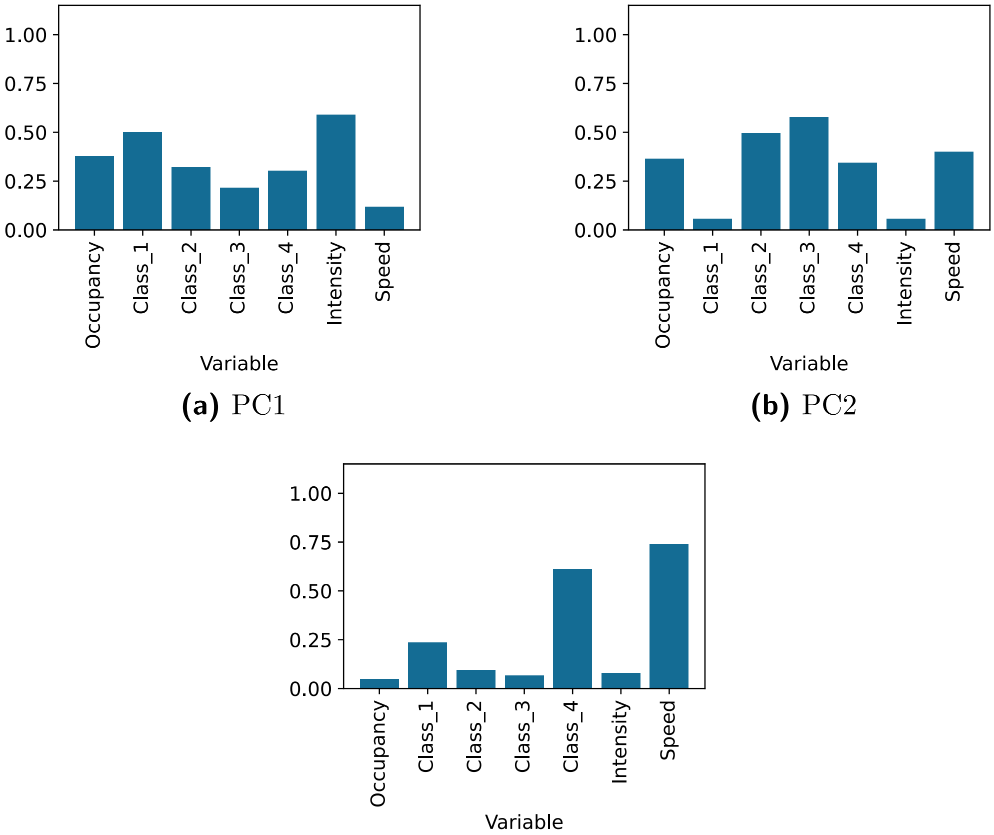
(a) Percentage of variance for each component

(b) Cumulative variance in principal components

Figure 4.16. PCA for LEZ dataset variables

The first principal component (Figure 4.17a) emerged as a cluster of *Intensity*, *Class_1*, and *Occupancy* primarily, revealing the high correlation between the first two. This suggests that these variables share an underlying relationship that may indicate the general traffic trend on the road. The close relationship between Vehicle *Intensity* and *Class_1* can be attributed to the influence of vehicles in this category on occupancy and traffic density.

In the second principal component (Figure 4.17b), it was observed that *Speed* emerged as the most prominent variable, followed by *Occupancy*, implying its importance in explaining variance. Regarding the variables related to vehicle categories, *Class_3* contributes the most information. The appearance of *Speed* in this component suggests its significant role in traffic flow, possibly related to traffic smoothness in the absence of vehicles from other categories.



(a) PC1

(b) PC2

(c) PC3

Figure 4.17. Contribution of variables to PCs

In the third principal component (Figure 4.17c), *Speed* stood out as the main variable, followed by *Class_4*. This configuration highlights the differential influence of *Class_4* on the variability not captured by the previous components. The presence of *Class_4* in this component may indicate a specific association of this vehicle category with different traffic patterns, possibly related to higher speeds or specific traffic flow behaviors for motorcycles.

After reviewing these components, it can be affirmed that variables related to vehicle categories emerge as key factors in explaining the observed variability in traffic patterns. This underscores the importance of considering the specific composition of vehicles on the road when understanding and managing traffic flow.

4.3.2.3 Boxplots

Box and whisker plots are visual tools that provide an understanding of data distribution and dispersion. A boxplot represents data distribution through a box covering the interquartile range (IQR), a segment or line indicating the median, and "whiskers," which are lines extending to the minimum and maximum values or to certain limits that indicate the presence of outliers. In addition to being useful for identifying factors as mentioned, they can also be used to compare distributions between different variables or groups [97]. For example, if the median line of one box extends beyond the limits of another box, this may indicate notable differences between the variables.

In this case, boxplots for the numerical variables (previously scaled) are plotted using the Python library Matplotlib [105].

It can be observed that the boxes and whiskers tend to vary in length and presence of outliers, except for the case of *Occupancy*. Regarding these outliers, it is important to mention that they were not removed during the initial data cleaning, as they were validated by CITRA [28] as values that are still within a range considered normal during specific hours. For example, in the case of speed, during late-night and early morning hours, drivers can reach speeds of up to approximately 120 km/h.

On the other hand, when comparing the numerical variables plotted in the diagrams, it is possible to observe that the distributions of the vehicle categories *Class_2* and *Class_3* are similar, while *Class_4* has a long tail, but its median is very close to 0.

Now, the variable *Class_1*, in line with what was observed in correlation

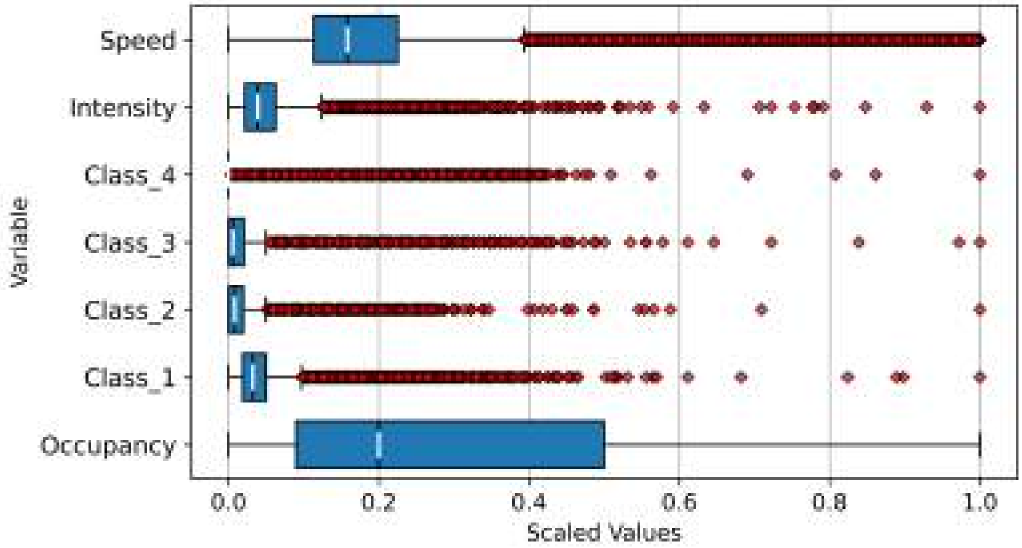


Figure 4.18. Box and whisker plot for LEZ variables

and PCA, has a distribution similar to *Intensity*, both in the width of its box and in its median. Taking into account this last measure, it could be stated that variables like *Class_2* and *Class_3* have similar characteristics. Below, in Table 4.2, the information obtained from the box and whisker plot is presented, where "X" indicates that two variables being compared may show notable differences between them, while the "*" represent that no significant differences between the variables were observed. For example, when comparing *Speed* and *Intensity*, it is clear that their distributions are different and their median lines are also very distant from the limits of the other box; in this case, an "X" was assigned. On the other hand, considering the previous statement about *Class_2* and *Class_3*, a "*" was set in the table.

4.3.3 Key Takeaways

The analysis of vehicular traffic data in Medellin's Low Emission Zone revealed important insights. Notably, the temporal component played a significant role, observing variations between months and weekdays, also showing repetitive patterns in the seasonal decomposition. Likewise, several corridors exhibited variations in the vehicle category, as well as changes in average values for *Speed*, *Occupancy*, and *Intensity*.

During feature selection process, the correlation analysis provided signif-

Table 4.2. Summary of the boxplot visualization.

	Occupancy	**Intensity**	**Speed**	**Class_1**	**Class_2**	**Class_3**
Occupancy	-	-	-	-	-	-
Intensity	X	-	-	-	-	-
Speed	*	X	-	-	-	-
Class_1	X	*	X	-	-	-
Class_2	X	X	X	X	-	-
Class_3	X	X	X	X	*	-
Class_4	X	X	X	X	*	*

icant insights into the relationships among various traffic-related variables. For example, positive and significant correlations were found between *Intensity* and *Occupancy*, as well as with specific vehicle classes, highlighting their influence on road occupancy and traffic density.

PCA showed that the first three components captured 79% of the variance, emphasizing their importance in explaining traffic patterns. The analysis highlighted the interplay between variables such as *Intensity*, *Class_1* vehicles, and *Occupancy* underlining their shared influence on traffic trends.

Finally, the boxplots validated the findings, and helped understanding some aspects of traffic behavior. For instance, the presence of outliers in the *Speed* variable can be attributed to higher speeds during late-night and early morning hours. These plots also indicated variations in the length of boxes emphasizing differences between variables.

These findings provide valuable insights for understanding and managing traffic patterns in Medellin's Low Emission Zone [6].

In the upcoming section, we will define the key elements of our Deep Q-Network implementation, which is informed by the analyses and feature selection conducted in this section. This includes a detailed description of the state representation, reward structure, and the action set. Furthermore, we will explore how the findings from the time series analyses and the feature selection process, particularly with regard to important variables and vehicle, are incorporated into the DQN state representation.

[6] Pending publication of the conference paper *Analysis and Characterization of Vehicular Traffic in a Low Emission Zone* in Advances in Transdisciplinary Engineering book series by IOS Press. This work was presented at the 7th International Conference on Intelligent Traffic and Transportation (ICITT) in September 2023 in Madrid, Spain.

Chapter 5

Experiments and Results

5.1 Deep Q-Learning for the LEZ's Traffic Management

In this section, we transition from the theoretical foundation of reinforcement learning (as discussed in subsection 3.1.3.1) to the practical domain, where we explore the essential components of our project's agent architecture, with traffic lights as the central agents guided by the DQN framework. We will look into the specific details of the neural network structure used to estimate Q-values, the utilization of a replay buffer to store and reuse experiences, and training methods employed to improve the agent's performance. This comprehensive examination of the DQN architecture serves as a fundamental building block for the experiments and results we present, providing insight into how our agent learns, adapts, and ultimately achieves its goals through interactions with the environment.

5.1.1 DQN Agent Architecture

5.1.1.1 State Representation

Defining an effective state representation is crucial in a reinforcement learning algorithm. Our state is characterized by a set of continuous variables, including *Speed* and *Occupancy*. These two variables were chosen based on the analysis and feature selection previously described in subsection 4.3.1 and subsection 4.3.2, since they may capture essential aspects of the traffic conditions and provide a comprehensive snapshot of the traffic environment for our specific case study, allowing the agents to make context-aware decisions using a specific action set (subsection 5.1.1.2).

In addition, the current traffic light phase and current emissions[1] are also an integral part of the state representation, such that the current traffic light phase directly influences traffic flow at intersections, guiding the agent in making right-of-way and safe driving decisions. On the other hand, emissions data have been included to assess the impact of the DQN on this environmental variable, such that, by monitoring emissions, the reinforcement learning agent can make more environmentally "conscious" choices, prioritizing not only direct traffic variables but also seeking to reduce emissions. The capture of these variables is detailed in section 5.3.

5.1.1.2 Action Set

As stated in subsection 3.1.4, fixed-time traffic light phases are one of the most prevalent approaches in traffic management systems.These phases involve predetermined signal configurations that dictate when vehicles in different directions are allowed to proceed or stop, regardless of real-time traffic conditions. In the case of Medellin's Low Emission Zone, traffic lights are static, following a fixed-time pattern. To illustrate this, we show an example of a typical traffic light configuration (in XML notation). This static setting corresponds to the *Cra 55 - Avenida Oriental* road's traffic light:

```
<tlLogic id="360392656" type="static" programID="1">
   <phase duration="49" state="GGGrrr"/>
   <phase duration="6" state="yyyrrr"/>
   <phase duration="49" state="GrrGGG"/>
   <phase duration="6" state="yrryyy"/>
</tlLogic>
```

This traffic light, located in the intersection corresponding to the mentioned road, has 4 possible phases and a period (sum of all phases). In this context, the *state* is the configuration for every phase. For example `<phase duration="49" state="GGGrrr"/>` represents the first phase of the traffic light; the *duration* attribute is set to 49 seconds, indicating that this phase will last for 49 seconds. The *state* attribute is "GGGrrr", which means that during

[1] Both data captured at *run-time* from the simulation environment.

this phase, the traffic light allows vehicles in the "GGG" (Green) direction to proceed, while vehicles in the "rrr" (Red) direction are stopped.

The default *state* configurations were imported from the OpenStreetMap (OSM) [106] infrastructure network of the LEZ, and the period set for these traffic lights is 110 seconds, a decision reached after consultation with CITRA [28], entity that informed us that traffic lights across the city generally operate with periods ranging from 90 to 120 seconds. Based on this, we opted for a period of 110 seconds to strike a balance between various traffic management scenarios within the LEZ. The OSM and traffic network import process will be shown in subsection 5.2.1.2.

However, in contrast to Medellin's static traffic light approach, our system employs a dynamic approach. With an action set for traffic lights that adapts to real-time traffic conditions, our approach offers the flexibility to adjust signal configurations based on the specific intersection under consideration. This dynamic action set may encompass either four or eight distinct phases, according to Table 5.1.

Table 5.1. Number of phases by road

Road	**Number of Phases**
Avenida Oriental - Sucre	4 phases
Carrera 43 - Girardot	4 phases
Avenida Oriental - Calle 57	4 phases
Carrera 55 - Avenida Oriental	4 phases
Carrera 57 - Avenida Oriental	4 phases
Avenida Oriental	8 phases
Avenida Oriental - Calle 52	8 phases
Avenida Ferrocarril - Calle 48	8 phases
Avenida Ferrocarril - Colombia	8 phases

Considering this Table, the action sets are:

- ✓ For 4-phase traffic lights: A = [phase_1, phase_2, phase_3, phase_4]
- ✓ For 8-phase traffic lights: A = [phase_1, phase_2, phase_3, phase_4, phase_5, phase_6, phase_7, phase_8]

These actions, when chosen, will indicate the traffic light phase, only taking the *state* attribute from the corresponding traffic light settings, but the total duration and the order of the phases will depend on the decision process made by the algorithm.

5.1.1.3 Reward Function

The design of the reward function is a crucial component of a reinforcement learning framework since this choice should lead the agent to discern whether its chosen actions serve to optimize or degrade the intersection's efficiency.

In our research, we have opted to incorporate a specific reward function introduced by Vidali *et al.* [20]. This function is based on the *cumulative total waiting time* as the traffic metric. Within the framework of this reward function, the computation of the reward (R_t) occurs at each step. This computation relies on the difference between the cumulative total waiting time at the present step ($ctwt_t$) and that at the preceding step ($ctwt_{t-1}$). The cumulative waiting time is calculated as:

$$ctwt_t = \sum_{veh=1}^{n} cwt_{(veh,t)} \tag{5.1}$$

Equation 5.1 defines the cumulative total waiting time, which is calculated by adding up the amount of time (in seconds) that each vehicle in the corresponding intersection has been stopped. From this, the reward function is defined in Equation 5.2 as:

$$R_t = ctwt_{t-1} - ctwt_t \tag{5.2}$$

This formulation aids in reflecting the influence of the agent's actions on the overall waiting time, and, unlike reward functions that reset waiting time metrics, the cumulative waiting time continues to account for vehicles that have spent time waiting. This helps the agent recognize the persistence of congestion and take actions to mitigate it. Furthermore, this approach considers not only the immediate effect but also the sustained influence on traffic conditions. This can lead to more strategic decision-making that considers the long-term consequences of each action.

In addition to Vidali's approach (referred as the *baseline* reward function from now on), we propose a weighted reward function in our study. This *alternative reward function* still uses cumulative waiting time as a basis, following Vidali's proposal. However, in the context of evaluating the impact of the DQN on emissions, we have introduced an additional component that accounts for emissions by utilizing a weighted (multi-objective) reward approach [107][108]. Our weighted reward is defined as:

$$R_t = weight_{wt} * \Delta ctwt + weight_{em} * emissions_t \tag{5.3}$$

Where $emissions_t$ is the cumulative emissions at the present step, and

$$\Delta ctwt = ctwt_{t-1} - ctwt_t \tag{5.4}$$

By introducing $weight_{wt}$ and $weight_{em}$, this reward function offers a flexible mechanism to prioritize one factor over another. For example, if $weight_{wt}$ is greater than $weight_{em}$, the model will prioritize reducing congestion over emissions and vice-versa.

In this approach $weight_{wt} + weight_{em} = 1$. This is used to represent the relative importance of the components of the reward and it helps to avoid a reward overly influenced by any single term (waiting time term or emissions term).

The performance of these two reward functions, also considering different weight values for the *alternative reward*, will be compared in the results section (subsection 5.3.2.1) to assess their respective impacts on the intersection's efficiency and emissions.

5.1.1.4 DQN Agent

In the development of our DQN algorithm to handle data derived from time series (described in chapter 4), a thoughtful exploration of neural network architectures was conducted to determine the most suitable approach. While considering the advantages of RNNs, like their sequential data processing capabilities, we found that the results did not exhibit significant improvement in our specific case. This observation led us to opt for a fully connected neural network instead. Our decision is grounded in practicality and empirical results, as fully connected networks have demonstrated their effectiveness in various time series applications and are known for their capacity to model complex relationships within data. By leveraging fully connected networks, we can efficiently capture patterns and dependencies within the time series data, enabling the DQN to make informed decisions and adapt to dynamic changes in our traffic management scenario.

An evaluation of CNNs was also carried out for our time series data, considering their strengths in spatial data processing. However, as our state representations primarily consist of continuous variables such as occupancy, speed, and emissions, the grid-like structure that CNNs excel at processing was not as prevalent in our data. This, coupled with the high computational cost of using this kind of network in a complex and large simulation environment, reinforced the choice of a fully connected neural network.

Our choice also aligns with the approach adopted by previous researchers [20][85][86] and serves as evidence of the flexibility and versatility of fully connected networks in handling time series data, which ultimately facilitates the achievement of our project's objectives.

In the context of our research, we are using a simplified representation of Bellman's equation (see subsection 3.1.3.1.1) into the DQN approach:

$$Q(S_t, A_t) \leftarrow R_{t+1} + \gamma max_A Q(S_{t+1}, A) \tag{5.5}$$

Equation 5.5 encapsulates the fundamental mechanism by which we update the Q-values, while integrating the immediate reward R_{t+1}, with the discounted maximum Q-value of the subsequent state S_{t+1}, and the associated action, A. In this approach, the agent is trained by receiving the states as inputs. Once trained, it approximates the Q-function, returning the estimate of the Q-values for all potential actions as the output. The agent then selects the action with the highest Q-value to execute, aiming to maximize its expected cumulative reward.

This agent is presented in Figure 5.1, where the architecture of the neural network can also be observed, defined as a fully connected network with an input layer consisting of 4 neurons (1 neuron for each state variable), followed by 4 hidden layers, each with 128 neurons. These hidden layers use Rectified Linear Unit (ReLU) [109] as the activation function. With this function, we can approach the non-linearity and complexity of traffic patterns. The output layer, on the other hand, contains as many neurons as there are phases (see Table 5.1) and employs the Softmax activation function [109]. Softmax transforms the network's output into a probability distribution over the possible light phases (4 or 8 depending on the road). This probability distribution helps the DQN make informed decisions about which light phase to select based on the continuous input state, ensuring that the network selects the most appropriate phase.

As described in subsection 3.1.3.1, instead of immediately incorporating the most recent experience into Q-function updates, the DQN stores each transition as a tuple in a memory called the *replay buffer*. Subsequently, collections of data samples from the buffer, called batches (or mini-batches), are used to stochastically and uniformly sample these experiences. These batches consist of a predetermined number of experiences determined by the *batch size*, that indicates how many experiences are processed in each training step. Data batches are then employed for training the neural network's parameters using the *Adam* optimizer, a variant of stochastic gradient descent that adapts learning rates for each parameter during optimization [69]. The use of the replay buffer has been highly used and recommended in DQN implementations [110][111][112], since it does not only aids in constraining the network's learning from correlated experiences but also empowers the DQN to revisit and learn from prior experiences.

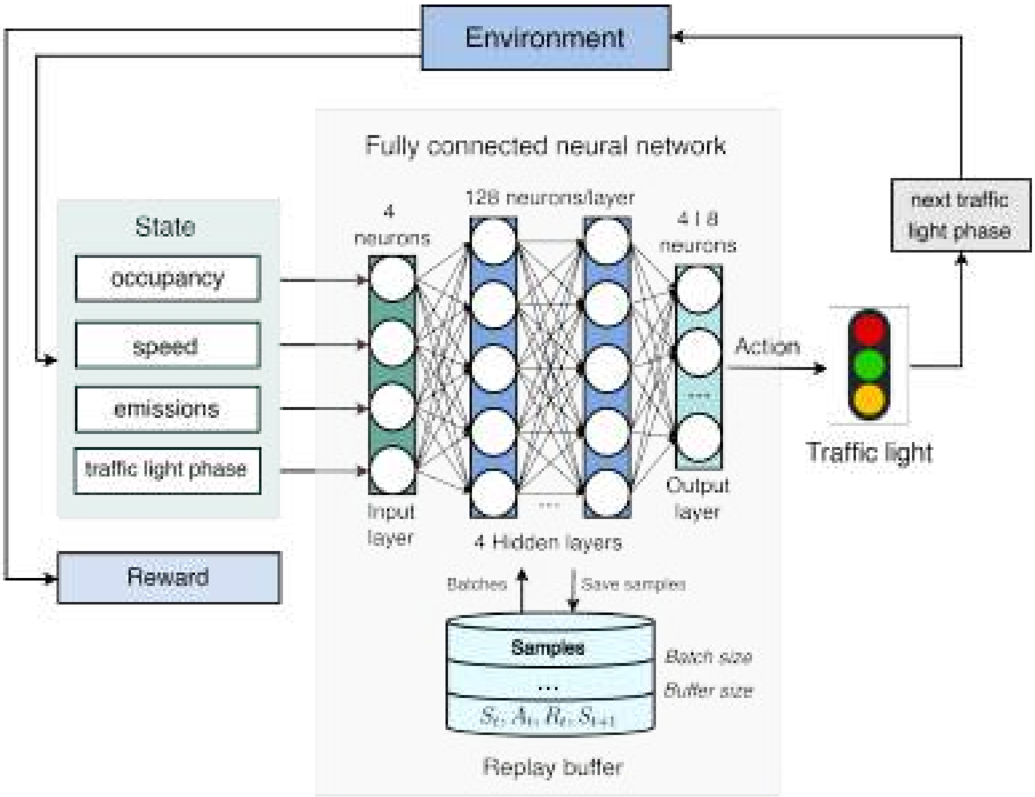


Figure 5.1. DQN agent

Having covered the key components of the DQN agent depicted in Figure 5.1, the next section will describe the experimental setup established for simulating the designed agent in the traffic scenario corresponding to the representation of Medellin's Low Emission Zone.

5.2 Experimental Framework

In this section, we detail the experimental framework used in the work conducted, as well as the results obtained during the course of the research. It focuses on the software used, simulation parameters, and simulated scenarios. This section provides an insight into the planning and execution of the experiments, establishing the necessary context for understanding and evaluating the results that will be presented later in the chapter.

5.2.1 Traffic Simulation

Simulation is a highly valuable tool as it enables the representation of scenarios and phenomena for evaluating their behavior and feasibility before real-world implementation. To bridge the gap between the simulated scenario and the real world, information from the LEZ dataset characterized in the previous sections, along with the road infrastructure in that area, were used to transform observations into data that were suitable for use in a traffic simulation. In this case, to achieve this objective, an experimental environment was established using the SUMO (Simulation of Urban Mobility) traffic simulator [19] and its Traffic Control Interface (TraCI) [113]. These components are described below.

5.2.1.1 Software

5.2.1.1.1 SUMO

SUMO is an open-source traffic simulation platform that allows modeling and analyzing the flow of vehicles, pedestrians, public transport, and other elements in transportation systems [19]. SUMO includes a variety of supporting tools that automate tasks for the creation, execution, and evaluation of traffic simulations, such as network import, route calculations, visualization, emissions and fuel consumption calculations. To interact with SUMO and perform more advanced experiments, the TraCI tool is used.

5.2.1.1.2 TraCI

TraCI is a communication protocol that enables the user to control and monitor SUMO traffic simulation in *run-time*. The operation of this protocol is illustrated in Figure 5.2.

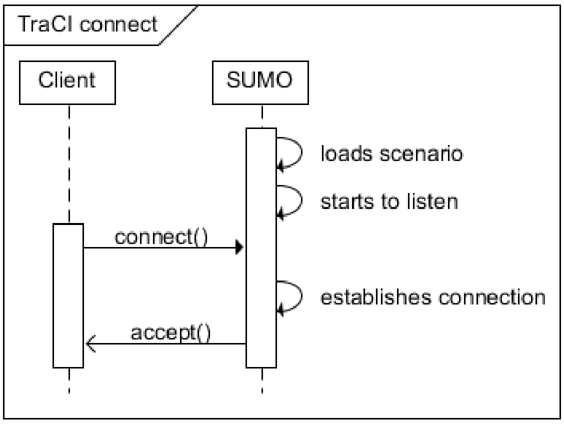


Figure 5.2. Connection between SUMO and TraCI[2]

TraCI uses a client-server architecture based on TCP (*Transmission Control Protocol*[3]) to provide access to SUMO. In this approach, SUMO acts as the server, and once started, its primary function is to set up the simulation and wait for all external applications to connect and take control. TraCI acts as an intermediary between these external applications (or clients) and SUMO [113]. Some of the functionalities of TraCI include:

✓ **Dynamic Control:** Allows dynamically adjusting simulation conditions, such as vehicle speed, traffic light signals, and vehicle routes.

✓ **Run-time Data Collection:** Users can collect detailed *run-time* simulation data, facilitating data analysis and data-driven decision-making.

✓ **Integration with External Tools:** TraCI can integrate with other tools and programming languages such as Python, C++, and others, making it easier to implement custom experiments and analyses. This allows running control algorithms in external applications, enabling them to retrieve information and/or perform actions on the ongoing simulation.

[2] Image taken from `https://sumo.dlr.de/docs/TraCI/Protocol.html`

[3] A protocol that provides bidirectional, connection-oriented communication between two devices on a network [114]

Some of the works that have used the previously described simulation tools are shown in Table 5.2.

Table 5.2. References of projects that use SUMO

Name of the resource	Autor	TraCI
Traffic Signal Control via Reinforcement Learning for Reducing Global Vehicle Emission [85]	B. Kővári, L. Szőke, T. Bécsi, S. Aradi, and P. Gáspár	Yes
Adaptive Deep Q-Network Algorithm with Exponential Reward Mechanism for Traffic Control in Urban Intersection Networks [86]	M. R. T. Fuad, E. O. Fernandez, F. Mukhlish, A. Putri, H.Y. Sutarto	Yes
Microscopic Simulation of Parking Violations in Curbside With-Flow Bus Priority Lanes Using Sumo Traffic Control Interface (TraCI) [115]	G. S. Samarakoon, T. Sivakumar2	Yes
A Deep Reinforcement Learning Approach to Adaptive Traffic Lights Management [20]	A. Vidali, L. Crociani, G. Vizzari, S. Bandini	Yes
IntelliLight: A Reinforcement Learning Approach for Intelligent Traffic Light Control [80]	H. Wei, G. Zheng, H. Yao, and Z. Li	Not specified
Deep Reinforcement Learning for Coordination in Traffic Light Control [84]	E. van der Pol	Not specified
Adaptive traffic signal control: Deep reinforcement learning algorithm with experience replay and target network [72]	J. Gao, Y. Shen, J. Liu, M. Ito, and N. Shiratori	No

5.2.1.2 Data Transformation and Generation

In the methodology for the traffic simulation required in this project, the starting point was the time series data, cleansed and analysed as previously detailed in chapter 4.

Subsequently, daily trip data (for 535 days), for each of the nine roads present in our data set, were stored in an Origin-Destination (OD) matrix. These origins and destinations were assigned unique IDs.

To create the underlying map for our simulation, we utilized SUMO tools [19]. Specifically, we employed the osmWebWizard.py package to generate an OpenStreetMap (OSM) [106] file that encapsulated the road infrastructure of our study area, including roads, traffic lights, edges, and other relevant elements. Some default settings included in the OSM original file were also deleted to customize factors such as edges configuration and maximum speed. This OSM file was then transformed into a traffic network (shown in Figure 5.3), establishing the framework for our simulation environment.

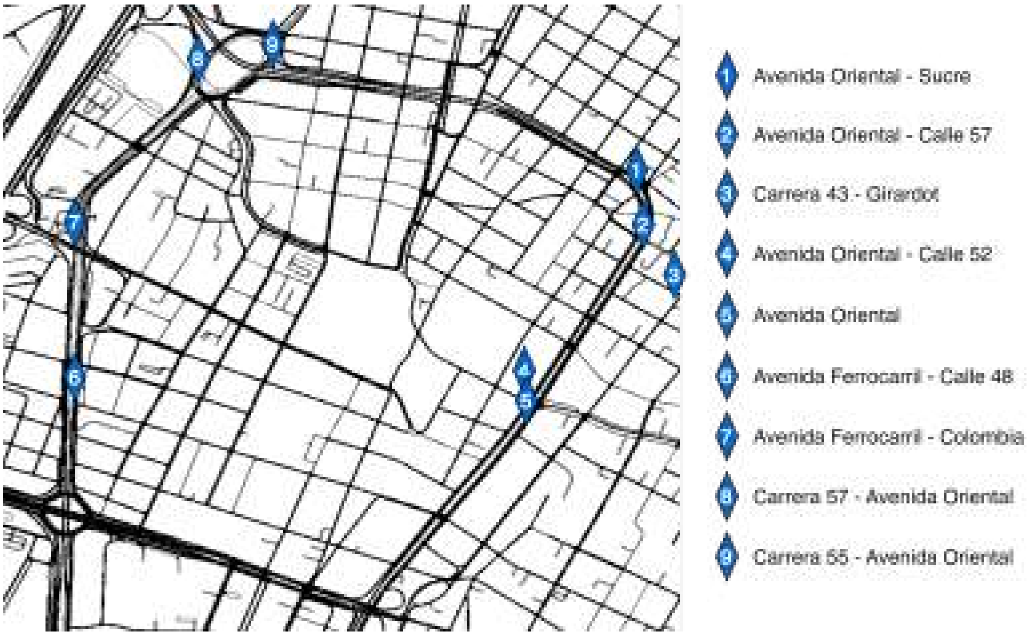

1 Avenida Oriental - Sucre
2 Avenida Oriental - Calle 57
3 Carrera 43 - Girardot
4 Avenida Oriental - Calle 52
5 Avenida Oriental
6 Avenida Ferrocarril - Calle 48
7 Avenida Ferrocarril - Colombia
8 Carrera 57 - Avenida Oriental
9 Carrera 55 - Avenida Oriental

Figure 5.3. Traffic network of Medellin's Low Emission Zone

Following the creation of this map, Traffic Analysis Zones (TAZ)[4] [116] were indicated though TAZ_IDs to replace the previously assigned IDs. Using SUMO libraries such as *OD2trips* and *duarouters*, and leveraging Python scripts to manage XML files, the OD data was converted into specific vehicle routes.

These routes were then integrated into the simulation environment, with vehicle types assigned based on a random weighted generator.

This approach ensures that the assignment of vehicle types is not entirely arbitrary but takes into account specific probabilities or weights associated with each category. In this particular case, the assigned weights for vehicle types are 0.65 for passenger cars (Class_1), 0.17 for buses and trucks (Class_3), and 0.18 for motorcycles (Class_4), chosen based on the results seen in subsection 4.3.1.3.2. This weighted system intends to get closer to the real-world scenario where certain vehicle types are more common than others.

By using this method, the generation of vehicles in the simulation is purposefully not in a strict sequential order, creating a more realistic and diverse representation of traffic within the simulated environment.

[4] Source and destination edges

5.2.1.3 Emission Classes

Since each type of vehicle generates different amounts of pollutants, emission classes were set with the SUMO HBEFA-based[5] emission model in its 4th version (HBEFA4). This model provides emission factors for all current vehicle categories. Table 5.3 shows the chosen model emission classes (*Emission Class*) for the simulated vehicle categories (*Vehicle type*).

Table 5.3. Emission classes for SUMO vehicle classes

Vehicle type	Emission Class
motorcycle	HBEFA4/MC_2S_le250cc_Euro-4
passenger	HBEFA4/PC_petrol_Euro-4
bus	HBEFA4/UBus_Midi_le15t_Euro-IV_EGR
truck	HBEFA4/RigidTruck_BEV_le7.5t

Considering the context of Colombia, where the vehicle fleet may not yet conform to the latest European-based emission classes[6], Euro 4 emission standard was chosen for the vehicles used in the simulation. This choice aims to more closely represent conditions of the real-world vehicle scenario of the city in terms of pollutant emissions, since Euro 6 just started its application in the country in 2023 [119], and most vehicles do not meet this standard [120].

The previous steps, as illustrated in the process depicted in Figure 5.4, facilitated the creation of a comprehensive traffic simulation environment that closely mirrors real-world traffic conditions. Consequently, the integration of the input data and the DQN agent presented in Figure 5.1, along with the environment configured in SUMO, the TraCI interface and the routing files, is presented in Figure 5.5.

Taking into account the DQN architecture and the experimental framework shown in the previous subsections, the following one explores the practical implementation of the accumulated knowledge and data, focusing on the experiments. Thus, in the next section, insights into the parameters employed to conduct these trials are presented, considering both single-agent and multi-agent cases.

[5] HBEFA: Handbook Emission Factors for Road Transport [117]

[6] Euro 6 is the latest Euro standard applied. Euro 7 is planned to be introduced in 2025 [118]

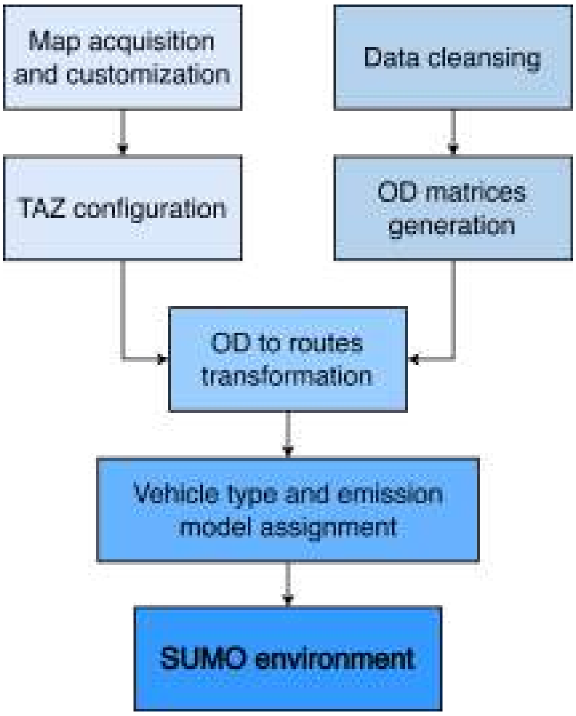

Map acquisition and customization
Data cleansing
TAZ configuration
OD matrices generation
OD to routes transformation
Vehicle type and emission model assignment
SUMO environment

Figure 5.4. Process of data preparation for simulation

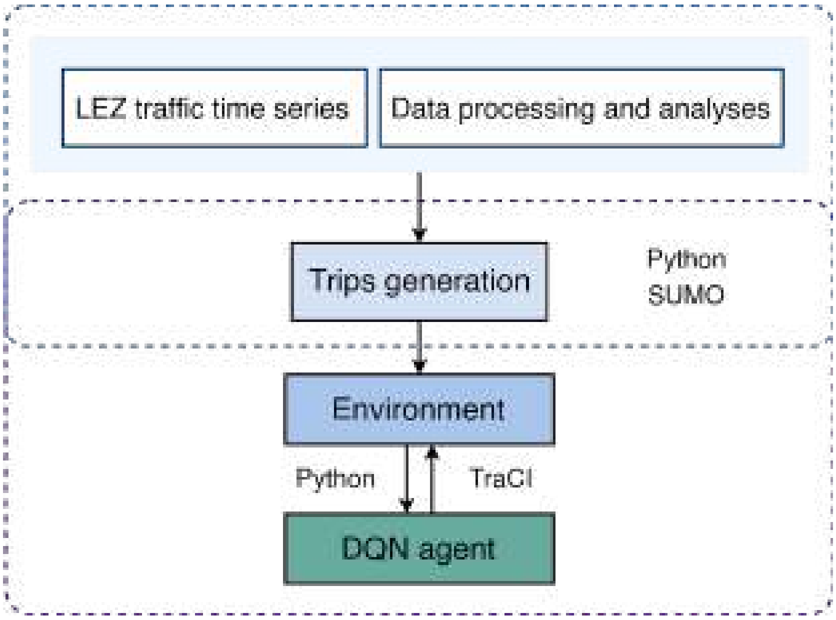


Figure 5.5. Integration of traffic data with SUMO and the DQN agent

5.3 Experiment 1: Single Agent

In the single-agent scenario, we describe the specific experiments conducted to assess the performance of our approach. Here, we examine the results of training one agent within our designed traffic simulation environment.

To provide a comprehensive overview, Figure 5.6 illustrates the complete architecture employed in this single-agent setup, with the intricate components, described in the previous subsections (Figure 5.1 and Figure 5.5), and the interactions that contribute to the agent's decision-making process.

Once established, this setup was integrated into the already described simulation scenario (in SUMO), where one agent was utilized for a parameter tuning process [84]. The goal of this process was to identify the most effective parameters for facilitating a more efficient learning process among the agents, considering the objective of the implementation of the model (improving traffic and reducing pollutant emissions). Parameter tuning is a critical step in optimizing the performance of AI models, ensuring that they adapt better to complex real-world scenarios. After the parameter tuning phase, the resulting optimized parameters, constituting the *tuned agent*, were applied to simulate all agents during the execution of the multi-agent simulation scenarios.

5.3.1 Simulation Parameters

With this scenario, we aimed to establish a baseline for simulating traffic across several roads and evaluate relevant hyperparameters. To achieve this goal, the agent was configured with the initial parameters shown in Table 5.4.

Regarding the training process, it was organized into multiple episodes, starting with an initial set of 200 episodes, equivalent to 200 simulation days, corresponding to the generation of trips in daily files. A matching number of route files was employed to align with the dynamic nature of daily traffic variations, which encompass differences in traffic intensity. This approach effectively captures a diverse range of traffic flows and patterns each day.

Also, in the context of optimizing traffic lights control using reinforcement learning, it is essential to establish a minimum duration for traffic light phases. Without this constraint, the model's actions could lead to impractically short signal times, such as 1 or 2 seconds. In a real-world scenario, traffic lights timings need to adhere to safety standards and operational feasibility. This includes providing minimum durations for the possible phases

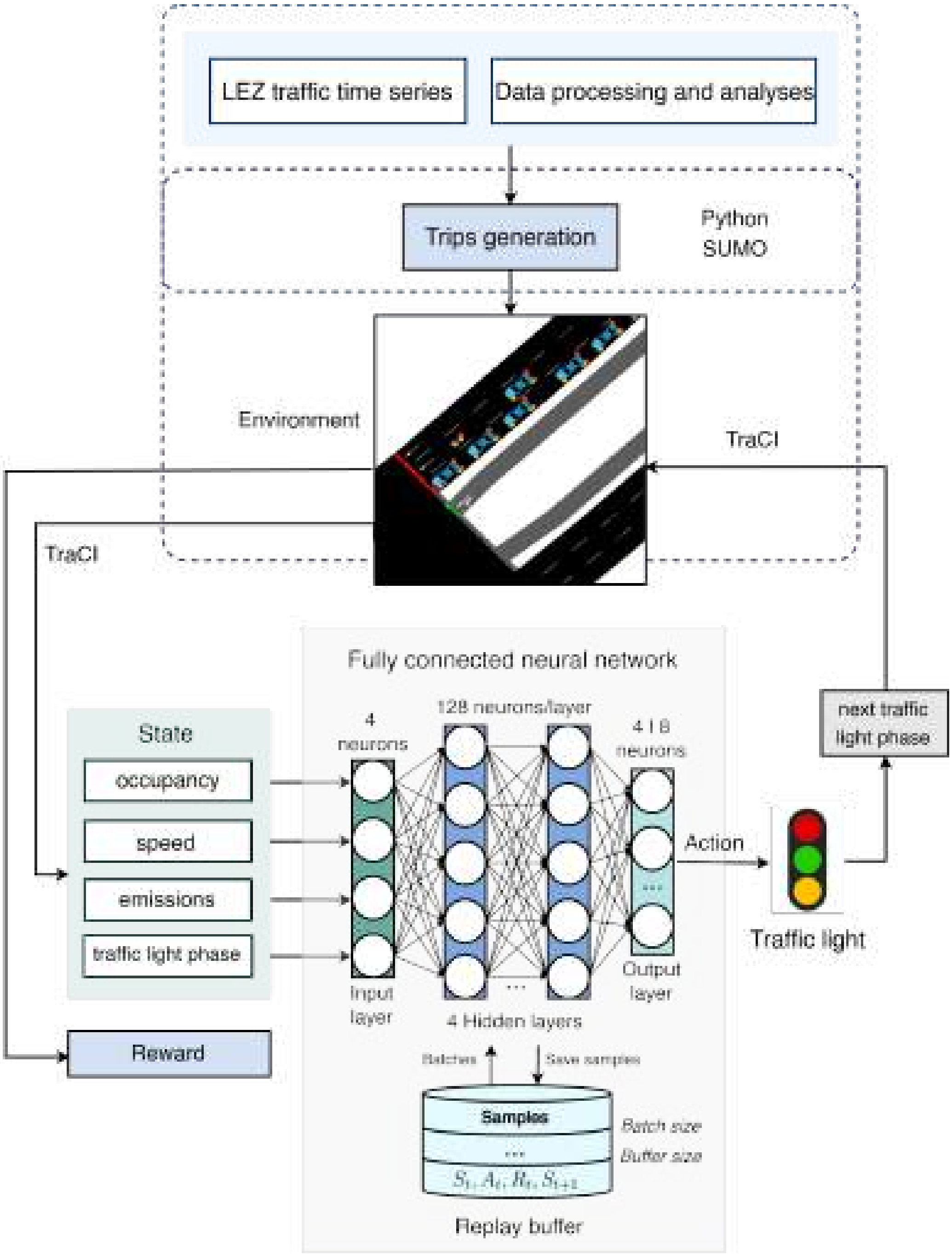


Figure 5.6. DQN for the LEZ traffic light control framework

Table 5.4. Initial parameters for the single DQN simulation

Component	Parameter	Value
Fully connected network	Batch size	64
Fully connected network	Learning rate (α)	0.001
Fully connected network	Training epochs	600
Fully connected network	Width of layers	128
Fully connected network	Number of layers	4
Fully connected network	Replay buffer size	Min: 600, Max: 30000
Traffic light	Green light duration	15 seconds
Traffic light	Yellow light duration	5 seconds
DQN	Reward	Total cumulative waiting time (baseline)
DQN	State representation	[occupancy, speed, emissions, traffic_light_phase]
Bellman's update rule	Discount factor (γ)	0.8
Bellman's update rule	Epsilon (ϵ)	1 - (current_episode/total_episodes)*decay_factor (factor = 1)

to ensure smooth traffic flow and safety for all road users. Thus, incorporating these minimum phase duration is crucial in maintaining a realistic and effective simulation environment for traffic management and control strategies. That is why minimum green and yellow times were set in initial values of 15 seconds and 5 seconds, respectively. The yellow face is activated if the previous and new action are different, to make a transition. These values were set considering the average duration of a complete traffic light cycle in Medellin (110 seconds)[7] and also considering previous works like Vidali's [20].

In addition, as stated in section 3.1, the exploration policy is really important in Reinforcement Learning, since it allows the agent to try new actions, but also to exploit the knowledge it has acquired. One of the most known strategies is the ϵ-greedy policy [121][84]. In ϵ-greedy, the agent makes a choice between exploration and exploitation by randomly selecting an action (exploration) with probability ϵ, or selecting the action with the highest estimated value (exploitation) with probability 1-ϵ.

Apart from this, in relation to the simulation tasks, the provided Graphical User Interface (GUI) of SUMO was used to identify the TAZ ID points,

[7] Information provided by CITRA

and to check weather simulations were running as expected, but for practical purposes (execution time and computational resources), this GUI was deactivated when running the DQN agent since TraCI facilitates the connection between the Python environment and SUMO, allowing to retrieve (e.g. waiting time, speed, emissions) and send information (execute actions) in during the simulation without needing a graphical interface.

Given the elements discussed, the pseudo-code of the algorithm is presented in algorithm 1, outlining the core methodology for optimizing traffic lights control.

Algorithm 1: Pseudocode for the DQN agent simulation

```
Data: Input routes and training parameters
Function main()
    Initialize simulation parameters;
    Initialize simulation memory;
    Initialize simulation NN model;
    for each episode do
        while Simulation is active do
            Simulate time step;
            Collect state and environment data;
            Choose action for traffic light ($\epsilon$-greedy);
            Execute traffic light actions;
            Calculate rewards;
            Add sample to memory;
        Train models using memory replay;
        Save episode statistics;
    Save data and visualize results;
    Clear episode variables (accumulators and episodic data collectors);
Function replay(model)
    Retrieve a group of samples from buffer;
    Update Q-values using the learning equation;
    Train the neural network model;
```

Algorithm 1 implements the DQN framework designed for traffic light control with the objective of training a neural network to choose the optimal traffic light action in specific states.

At the beginning, before the actual simulation commences, three crucial initializations are performed. First, the simulation parameters are set up. These encompass traffic light minimum durations and the SUMO environment components, such as the route files, TraCI, GUI (if needed), simulation delay, vehicle characteristics, the number of simulation episodes, and traffic lights default configuration (traffic light states, subesction 5.1.1.2). Additionally, specific parameters intrinsic to the DQN approach are set, including learning rate, batch size, number of layers, width of the layers, and training epochs. In relation to the agent, other important aspects are defined, such as the replay buffer size, the size of the input state, and the discount factor.

Once the simulation environment and the parameters are initialized, a bash script is run to call the main simulation file which is responsible for importing the configurations and running the episodes. Each episode in the algorithm corresponds symbolically to a day of traffic, and a distinct SUMO road file is imported, setting the stage for that day's simulation. Within the execution of each episode, the simulation flows through a series of steps: the environment is advanced by a time increment, followed by the collection of state information and environment data, which encompass metrics such as average speed, emissions, occupancy, waiting time, vehicle count, and the current phase of the traffic light. Actions for the traffic light are then determined using the ϵ-greedy strategy. Once the action is determined, it is executed, leading to the computation of the baseline reward, which is based on the cumulative waiting time. This entire experience is then stored as a sample in the replay buffer.

As the day's simulation ends, the neural network model starts training using a batch of experiences drawn from the replay buffer. This training leads to the update based on Bellman's rule, using the configured discount factor. After the epochs of training that were indicated for the training process, the episode concludes and its statistics are stored in .txt files for subsequent analysis. Finally, some of the accumulator variables and other elements used exclusively for the episode are cleared to be empty for the next episode's data and free up memory resources.

5.3.2 Results

In the domain of traffic management using DQN, waiting time and rewards have traditionally been the primary metrics for performance evaluation (subsection 3.1.4). Waiting time (subsection 5.1.1.3) directly indicates traffic flow efficiency, while rewards reflect the model's decision-making process. However, since this study intends to evaluate the DQN model's impact on environmental emission, it also incorporates emissions as a key metric.

In this section, we will examine the results of a hyperparameter analysis for the DQN algorithm, our primary aim was to fine-tune[8] and optimize the performance of our DQN model by varying crucial hyperparameters and settings, considering the baseline agent. Specifically, we investigated the impact of different rewards, variations in the state representation, replay buffer sizes, neural network's batch size, discount factors, and, ϵ values for ϵ-greedy exploration.

Our goal was to gain a comprehensive understanding of how these hyperparameters affect the learning process and decision-making capabilities of our DQN agent before going through a multi-agent architecture simulation.

It's important to consider that the same data and scenario (*Av. Ferrocarril - Calle 48*) was used to perform these evaluations, ensuring that the results are directly comparable, and that the variables measured and stored during simulation run-time were, besides the negative rewards, cumulative waiting time and emissions (explained in subsection 5.1.1.3). For visualization purposes, moving averages were used in the plots, with a $windowsize = 5$.

5.3.2.1 Reward Function

In relation to the reward function, a comparison between three strategies (subsection 5.1.1.3) was conducted: the *baseline reward* (Equation 5.1), which primarily focuses on reducing waiting times; the *alternative reward* (Equation 5.3), which introduces a direct consideration of emissions with a weighted approach, and a reward that only considers emissions (*(*emissions-based). We explored the impact of these reward structures on waiting time (as indicator of congestion) and emissions.

The *alternative reward strategy* was evaluated using two different weight configurations:

- ✓ First alternative reward: $weight_{wt} = 0.6$ and $weight_{em} = 0.4$
  $R_t = 0.6 * \Delta ctwt + 0.4 * emissions_t$

- ✓ Second alternative reward with normalized values: $weight_{wt} = 0.6$ and $weight_{em} = 0.4$
  $R_t = 0.6 * norm(\Delta ctwt) + 0.4 * norm(emissions_t)$

- ✓ Third alternative reward: $weight_{wt} = 0.2$ and $weight_{em} = 0.8$
  $R_t = 0.2 * \Delta ctwt + 0.8 * emissions_t$

[8] Find the value of a parameter that may lead to an improvement in the performance of a model

✓ Fourth alternative reward: $emissions_t$
$R_t = emissions_t$

Figure 5.7 presents the results for the different reward approaches, in this case, with the moving averages (MA).

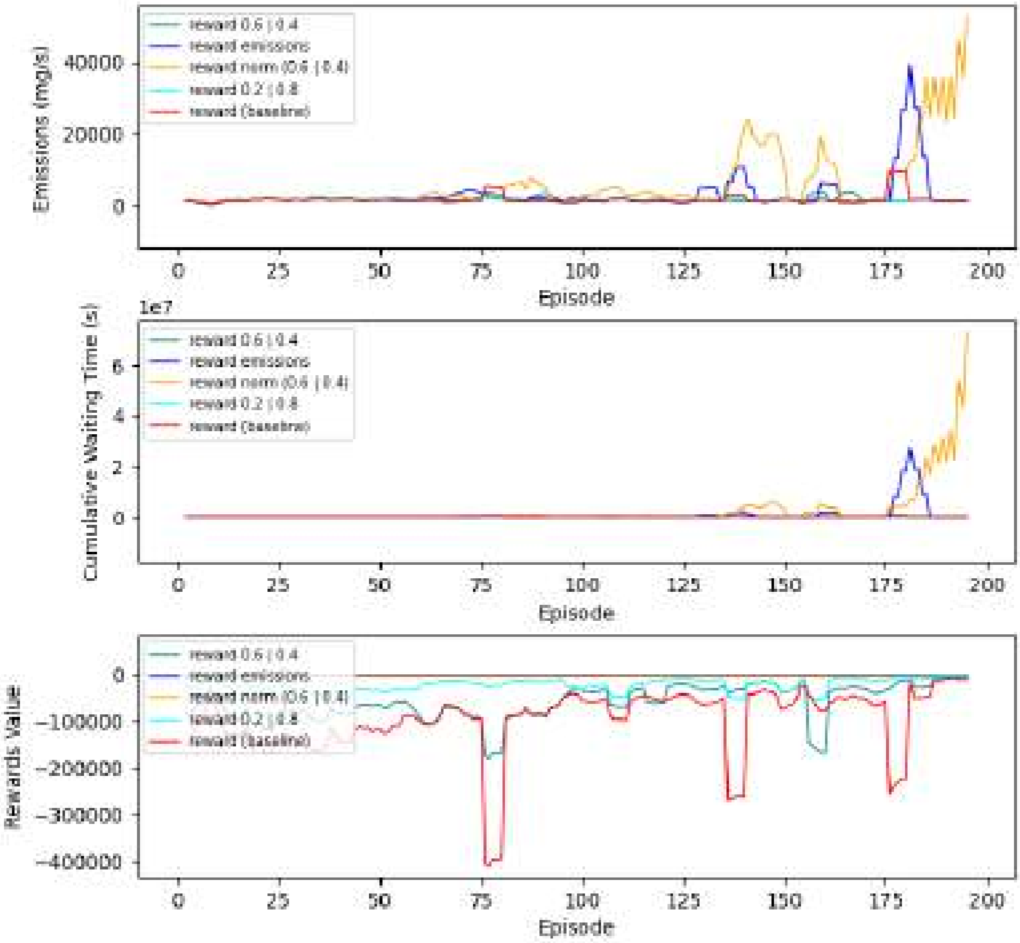


Figure 5.7. Reward-based comparison for the baseline and alternative approaches

As seen in the figure, the emissions only-based reward, and the normalized weighted reward presented the highest values for emissions and waiting time, showing a bad performance for these two metric of interest, for which the agent doesn't seem to learn. For observation purposes, a similar plot is included next, excluding the results of the simulation carried out with these two reward functions.

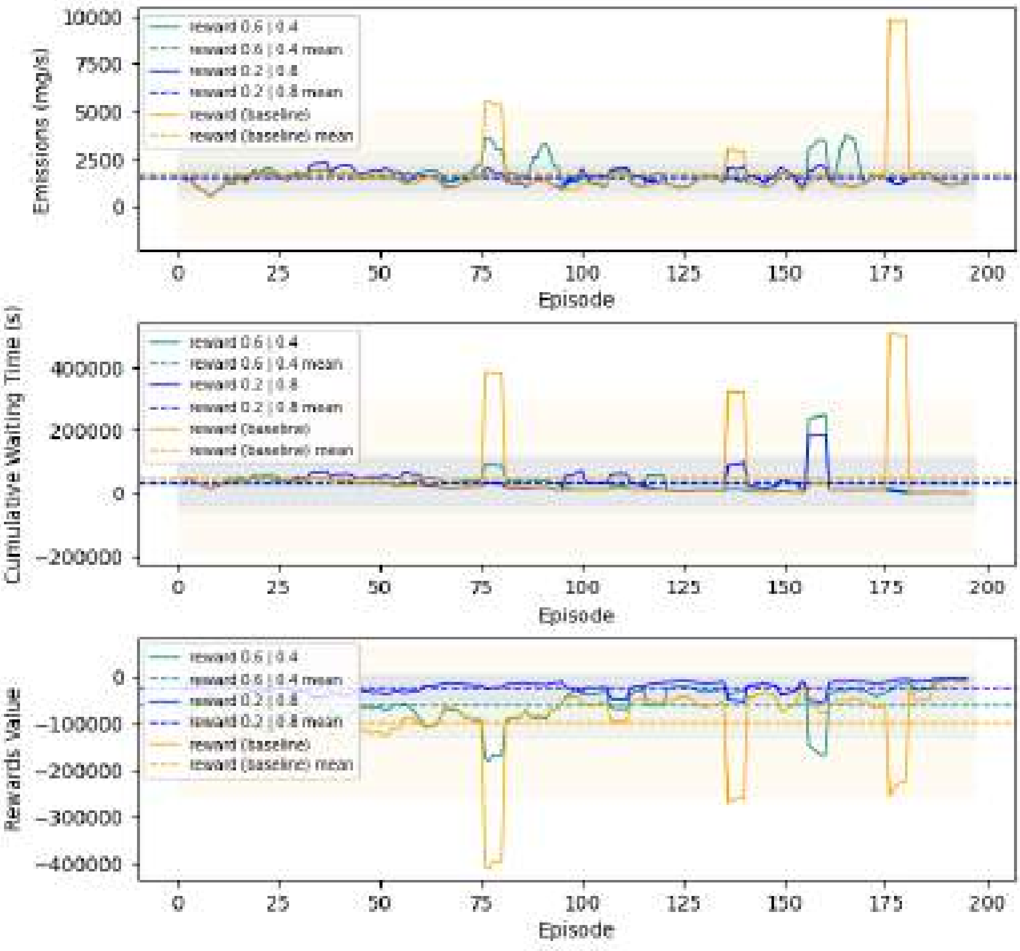


Figure 5.8. Reward-based comparison without emission only-based and normalized reward function results

The graphic presented in Figure 5.8, reveals interesting insights. The *baseline strategy*, which solely targets reducing waiting times, does achieve reduction in both congestion and emissions. However, when the reward structure is modified to incorporate emissions directly, even with a intermediate weight in the first *alternative reward strategy* ($weight_{em} = 0.4$), there's a discernible improvement, showing a more effective management of both congestion and emissions in terms of mean values and stability. However, considering the importance of emission reduction, the third approach was tested (third *alternative reward strategy*), assigning a higher weight to cumulative emissions: ($weight_{wt} = 0.8$ and $weight_{em} = 0.2$). This strategy showcases

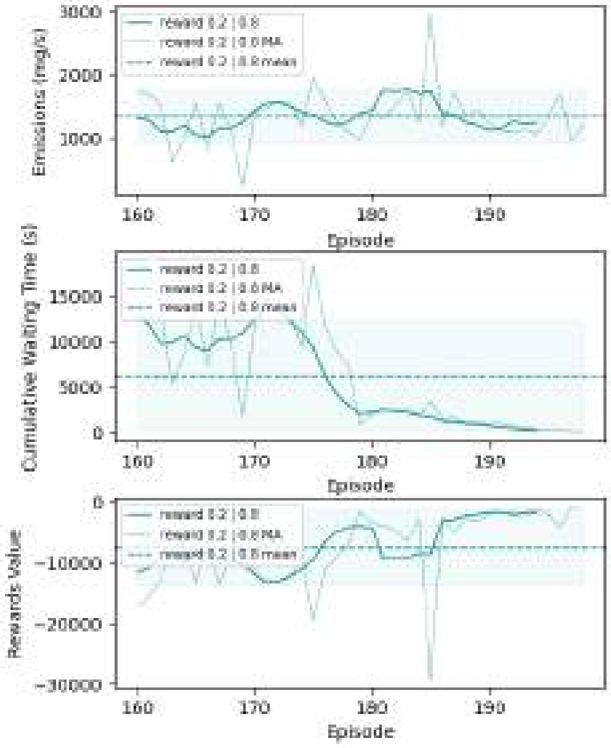


Figure 5.9. Zoom for the last 40 episodes of the reward-based comparison

the most promising results, since it not only reduces the waiting time more efficiently but also manages to keep emissions in check, with a lower standard deviation and mean, translating in a more stable result during the training process for emissions and cumulative waiting time while converging as episodes progress. The zoom of the comparison between the chosen reward function and the baseline is shown in Figure 5.9.

This analysis suggests that a concerted effort to address congestion does lead to a reduction in emissions, but a more comprehensive approach, where emissions are also directly considered in the decision-making process, can amplify these benefits. It's important to note that while reducing congestion is the primary goal, the interdependent relationship between congestion and emissions implies that strategies targeting both can achieve better outcomes for urban traffic management.

5.3.2.2 State Representation

In our exploration of state representations, we compared four distinct configurations:

- ✓ Baseline state: [speed, emissions, occupancy, traffic_light_phase]
- ✓ Alternative state 1: [vehicle_count[9], emissions, occupancy, traffic_light_phase]
- ✓ Alternative state 2: [speed, vehicle_count, emissions, occupancy, traffic_light_phase]
- ✓ Alternative state 3: [waiting_time, emissions, occupancy, traffic_light_phase]

In this case, the possibility of having waiting time values was a starting point to explore variables that were not initially considered in chapter 4, but they are not far from what we got as our initial data, and its possible they may be obtained [10]. In this case, waiting times are given by our simulator.

After examining the visualizations, shown in Figure 5.10, for the four state representations in our DQN model, some interesting insights were observed regarding the behavior of rewards, waiting time and emissions.

Regarding reward, this metric provided further information into the effectiveness of the learning process of each state representation. The *third space representation* consistently showed lower mean rewards and emissions, while waiting times values were only equal to the baseline's. The trend of rewards stabilizing as episodes advance was observed, further supporting the notion of the DQN model's learning capability.

Besides these remarks, the response of the *third space representation* in high traffic scenarios was better than the others.

In relation to waiting time, data revealed that vehicles experienced the longest average waiting times in certain episodes with the *alternative state 1* and the *alternative state 2*, even in latter episodes, where those presented

[9] In the context of our work, *Intensity*

[10] Medellin city, in its plan for gathering data in different domains is building a public repository called MEDATA. A dataset called "Velocidad y tiempo de viaje" (speed and travel time) was found there. In the end, it was not used since CITRA suggested that the quality of the data is not as good as the data we have now, but it shows the possiblility of obtaining other traffic variables [122].

higher standard deviations. In contrast, *state representation 3* along with the *baseline space representation* presented the the lowest waiting times in average along with lower standard deviations.

In relation to emissions, the *state representation 3*, presented the lowest average value and standard deviation, when compared to the other approaches.

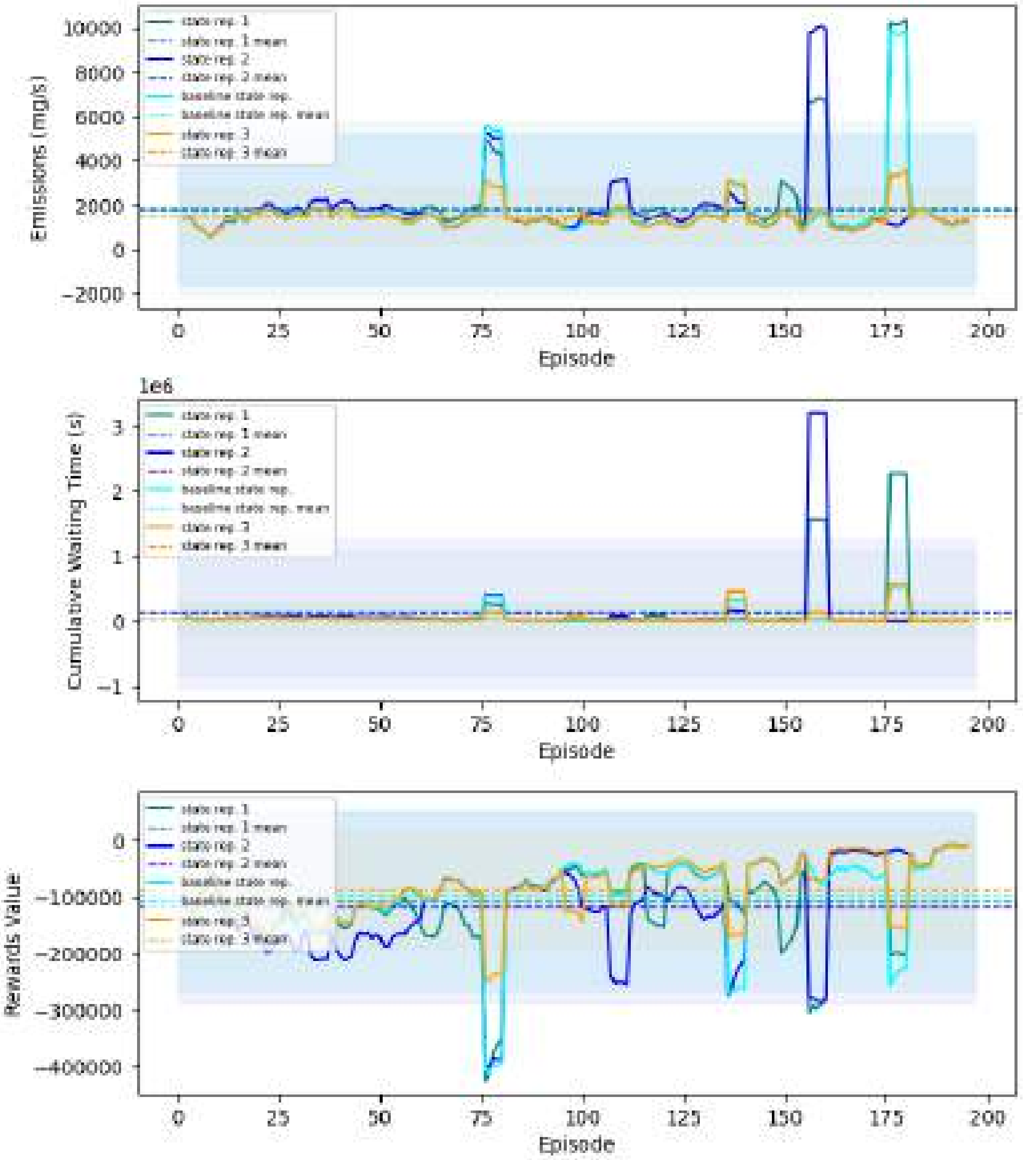


Figure 5.10. State-based comparison

Considering the previous information, *the alternative state 3* appears as the best of the evaluated state representations, since it was the most stable, and presented the lowest emissions and waiting time values, achieving a better performance. The comparison between the baseline approach and the chosen state representation is shown in Figure 5.11.

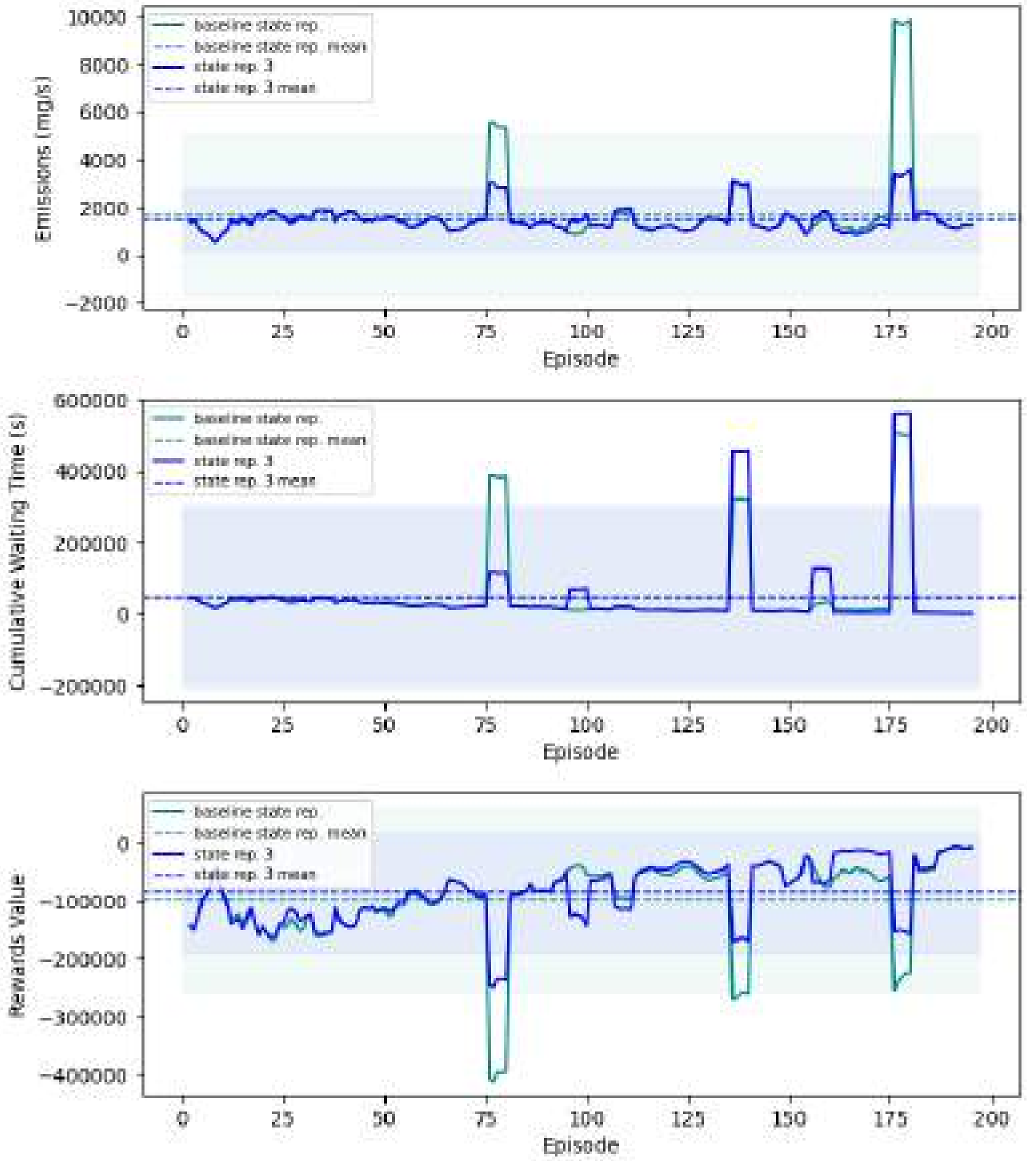


Figure 5.11. State-based comparison. Baseline vs State representation 3

5.3.2.3 Learning Rate

In Figure 5.12, we present the behavior of reward, emissions, and waiting time, allowing for a visualization of how these key metrics evolve. During our analysis, we performed a comparative evaluation of three learning rates, namely, $\alpha = 0.001$ (baseline), $\alpha = 0.0005$ (second LR), and $\alpha = 0.1$ (third RL) This enabled us to understand how varying learning rates impact the learning and decision-making capabilities of our DQN agent.

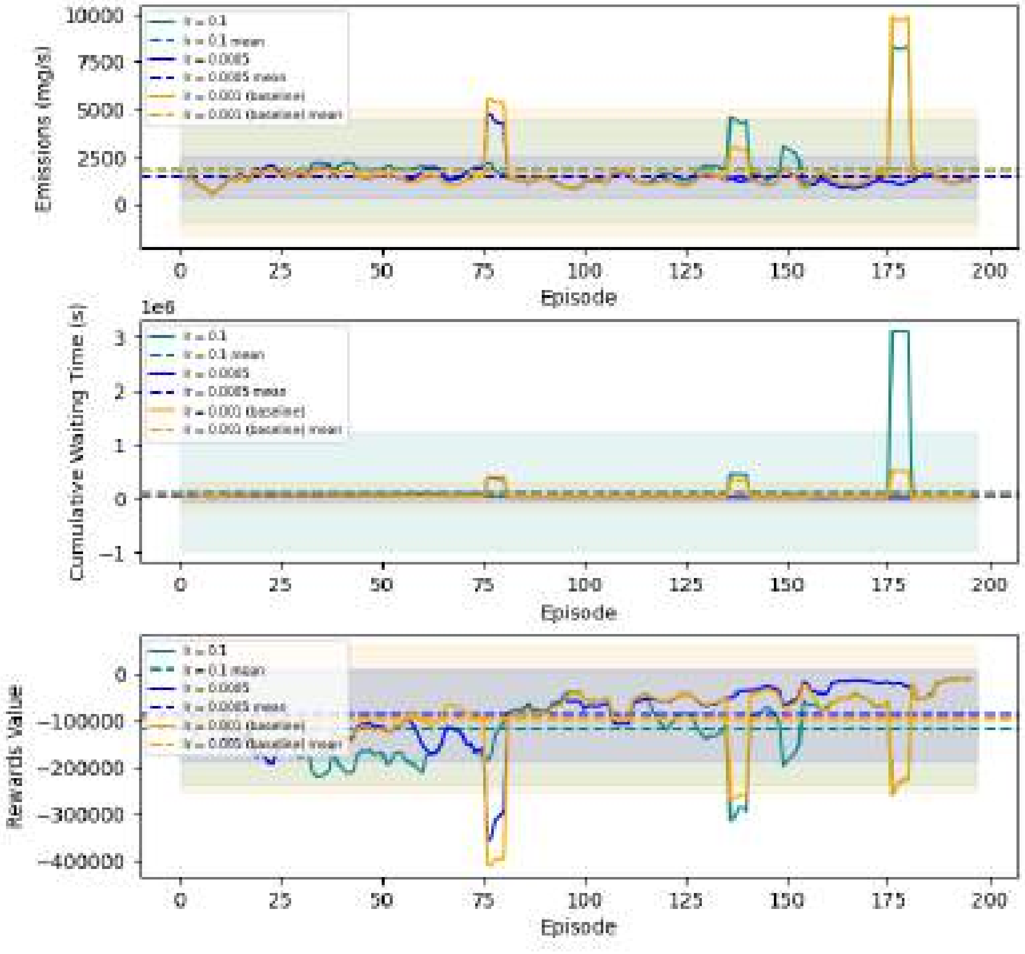


Figure 5.12. Learning rate-based comparison

As seen in the figure, the $\alpha = 0.1$, presented the highest values for emissions and waiting time. Although peaks may happen when an exploration decision is taken, when compared to the baseline, we found that ($\alpha = 0.1$), reduces waiting time and emissions at first, while leading rewards towards

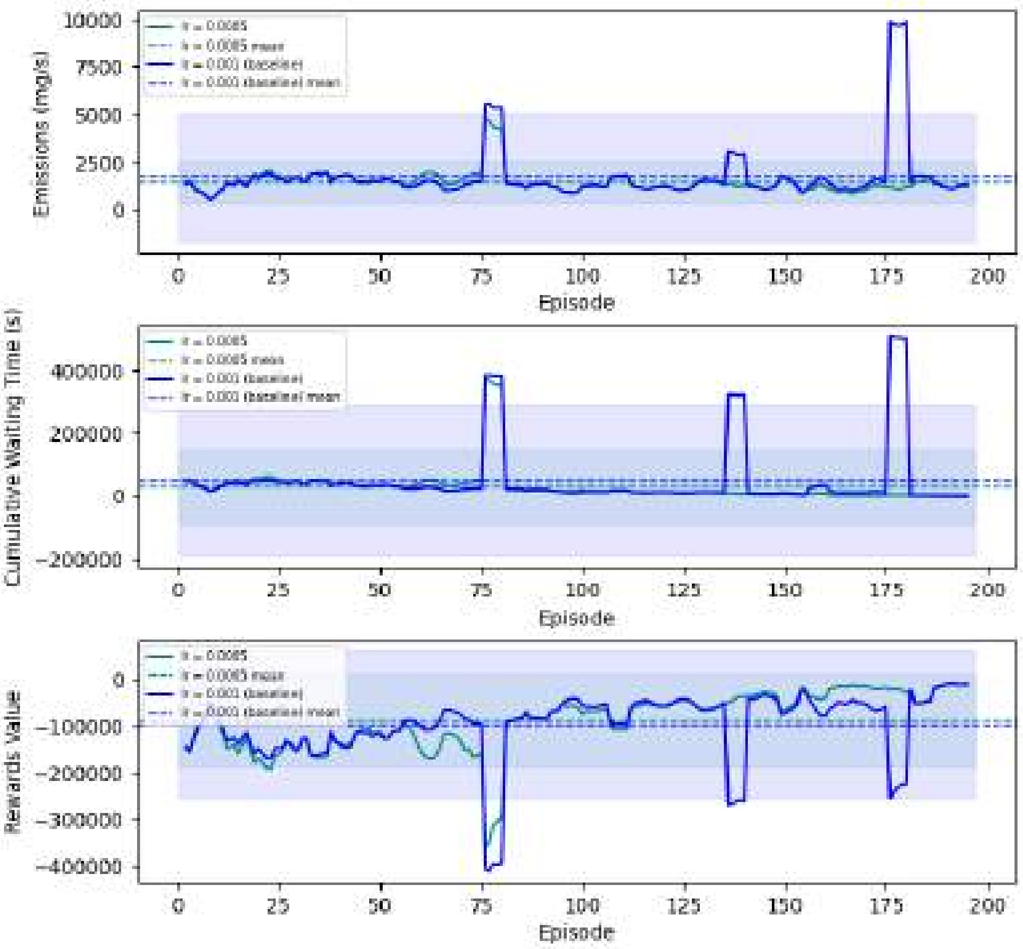


Figure 5.13. Learning rate-based comparison

more positive values, but encountered setbacks in the learning process, where it exhibited drops in the rewards in latter episodes and peaks in emissions and waiting times. $\alpha = 0.001$ (baseline) and $\alpha = 0.0005$, displayed better reward curves. In this case $\alpha = 0.0005$ presented a better performance for emissions and waiting times, with lower average values and standard deviation. This may indicate that, for this application, a lower learning rate could be beneficial during the learning process. For our specific case it lead to more stable training and prevented large oscillations or a significantly divergent behavior during training. A plot with $\alpha = 0.0005$ and $\alpha = 0.001$ is shown in Figure 5.13.

According to this, the *alternative 1* with an ($\alpha = 0.0005$), seems to present the best performance for emissions and waiting times.

5.3.2.4 Replay Buffer Size

The varying of the replay buffer sizes was also revised, by comparing buffer sizes of s = 10,000, s = 30,000, and s = 50,000.

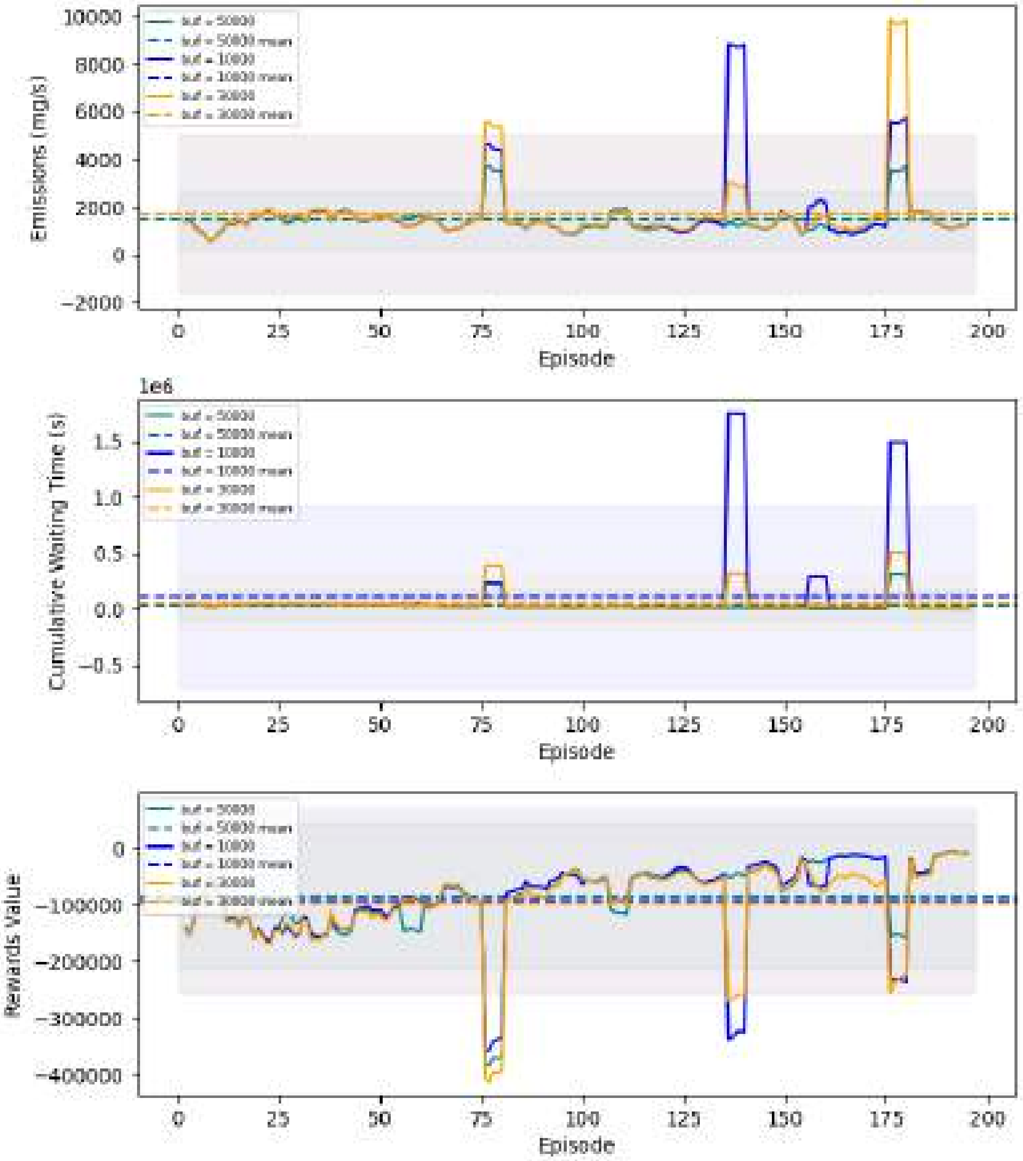


Figure 5.14. Buffer size-based comparison

In terms of rewards, all buffer sizes display an upward trend as episodes progress, indicating the agent's ability to learn over time. The s = 10,000 buffer size exhibits exhibits a reward behavior similar to the one presented by the s = 30,000 in terms of mean and standard deviation, but performed worst in waiting time. The rewards obtained with s = 50,000 didn't present as much fluctuations, and showed lower mean and standard deviation compared to the other two.

By the other hand, in relation to emissions over the course of episodes, the 50,000 buffer size shows more stability (less fluctuations), particularly in the later episodes, in contrast to the other two sizes tested, which suggests that it might be offering a more consistent learning experience. A similar behavior was observed in the waiting time curve.

This led us to conclude that despite baseline buffer size also contributed to improvement and learning as episodes progress, the 50,000 buffer size stands out with an advantage, offering enhanced performance and stability when evaluating our metrics of interest.

5.3.2.5 Discount factor

During our analysis, we also conducted a comparative assessment of discount factors, specifically evaluating values of $\gamma = 0.99$, $\gamma = 0.8$, and $\gamma = 0.7$, to evaluate the influence of different discount rates on the learning and decision-making performance of our DQN agent.

In general, performance based on rewards was similar in terms of standard deviation and mean value for all three values.

Regarding waiting times across episodes, we observed that the discount factor in both alternatives $\gamma = 0.99$ and $\gamma = 0.7$, and very similar to the $\gamma = 0.8$.

In terms of emissions, all three discount factors exhibit a downward trend throughout the episodes, suggesting that the agent effectively learns to reduce emissions over time.

Considering overall stability and balanced performance across all metrics, none of the discounts factor showed a significant improvement when compared to the baseline $\gamma = 0.8$, since its values for the standard deviation were biased by a specific peak. Keeping in mind the importance of balancing long and short term response. This was the chosen value for this parameter, since its an intermediate choice among the tested ones.

5.3.2.6 Epsilon

Furthermore, we experimented with different ϵ values for ϵ-greedy exploration, examining two calculation methods with factors of 1 and 1.5 applied to the decay $1 - (current\_episode/total\_episodes) * decay\_factor$ shown in Table 5.4. This helped us determine the impact of exploration strategies on the DQN's learning process and overall performance.

When comparing the baseline decay factor ($decay\_factor = 1$) to a $decay\_factor = 1.5$, the second displayed more effective learning and performance at the beginning, getting faster to a higher reward. However, when analyzing waiting times and emissions, it presented more flucturations. Besides, $decay\_factor = 1.5$ presented really high emissions and waiting times from ep. 160 (aprox.), which highly increased the mean value and standard deviation. These results suggest that this model performs significantly worse than the baseline model in terms of waiting time and emissions. The rewards metric also indicates a lower performance for this epsilon model, with more negative rewards on average and greater variability.

The drastic differences, especially in waiting time and emissions, may suggest that the decay factor adjustment significantly impacts the model's ability to make efficient decisions. This could be due to the model exploring less optimal actions more frequently or not converging to a stable policy as effectively as the baseline model.

In the end, the baseline value was kept.

5.3.2.7 Batch size

Another comparison performed during the tuning process was for the batch size[11] parameter. In this case, in relation to the rewards, all batch sizes present fluctuations observed along the simulation episodes, being the $batch\_size = 64$ the one with a slighly higher mean, followed by $batch\_size = 128$. All three show an ascending curve for rewards, showing the tendency to convergence.

Regarding waiting times across episodes, the $batch\_size = 128$ showed the highest standard deviation, specifically due to the peaks observed aroung ep. 175, along the highest mean value. On the other hand, $batch\_size = 64$ and

[11] Refers to the number of samples that are passed to the neural network at once, before the model's internal parameters are updated [123]

batch_size = 256 showed similar behaviors, but the *batch_size* = 256 had bigger fluctuations, leading to a higher standard deviation.

In terms of emissions, *batch_size* = 128 presented high peaks and the highest mean, which derive in the worst performance, while the other two sizes had similar measurements, where the *batch_size* = 64 had slightly lower mean and standard deviation.

Although three discount factors exhibit a downward trend throughout the episodes. The *batch_size* = 64 was kept since the values presented were the best among the evaluated options, since the standard deviation and mean were the lowest for waiting times and emissions, and the highest for the reward.

After the tuning procedure that considered several parameters of the DQN, the chosen values assign to what we'll call the *tuned model*, are shown in Table 5.5:

Table 5.5. Fine-tune parameters for the DQN agent

Component	Parameter	Value
Fully connected network	Batch size	64
Fully connected network	Learning rate (α)	0.0005
Fully connected network	Training epochs	600
Fully connected network	Width of layers	128
Fully connected network	Number of layers	4
Fully connected network	Replay buffer size	Min: 600, Max: 50000
Traffic light	Green light duration	15 seconds
Traffic light	Yellow light duration	5 seconds
DQN	Reward	Alternative reward function ($weight_{wt} = 0.2$ and $weight_{em} = 0.8$)
DQN	State representation	[waiting_time, emissions, occupancy, traffic_light_phase]
Bellman's update rule	Discount factor (γ)	0.8
Bellman's update rule	Epsilon (ϵ)	1 - (current_episode/total_episodes)

5.3.2.8 Baseline vs Tuned Parameters Model

To compare the training curves between the baseline and the tuned model, moving averages were plotted in Figure 5.15, using the same data for the training process.

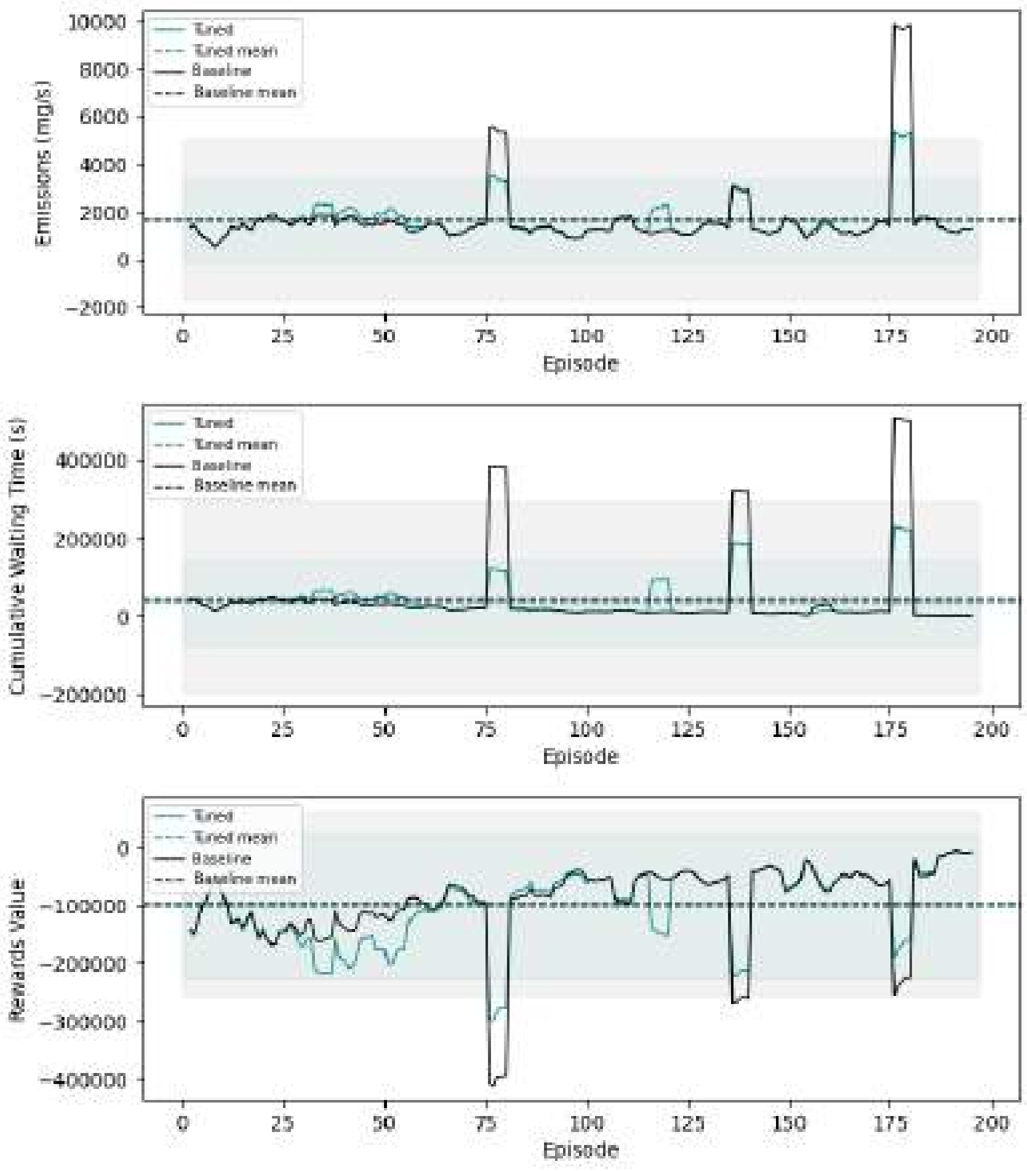


Figure 5.15. Training process of the models with tuned vs baseline parameters (moving averages)

In this case we can observe that the tuned model presents a better response in high traffic situations; the peaks reached lower values, leading to a higher mean and lower standard deviation for emissions and waiting times, that also present a descending value-behavior.

Regarding rewards, the tuned model presented lower values during the first episodes, but then they started to increase gradually, maintaining a growing tendency.

After performing the training process for both models, 20 episodes of testing were run with new simulation days. The results can be observed in Figure 5.16. According to them, the tuned model appears to perform better in terms of waiting times and emissions, indicating improved efficiency in these aspects.

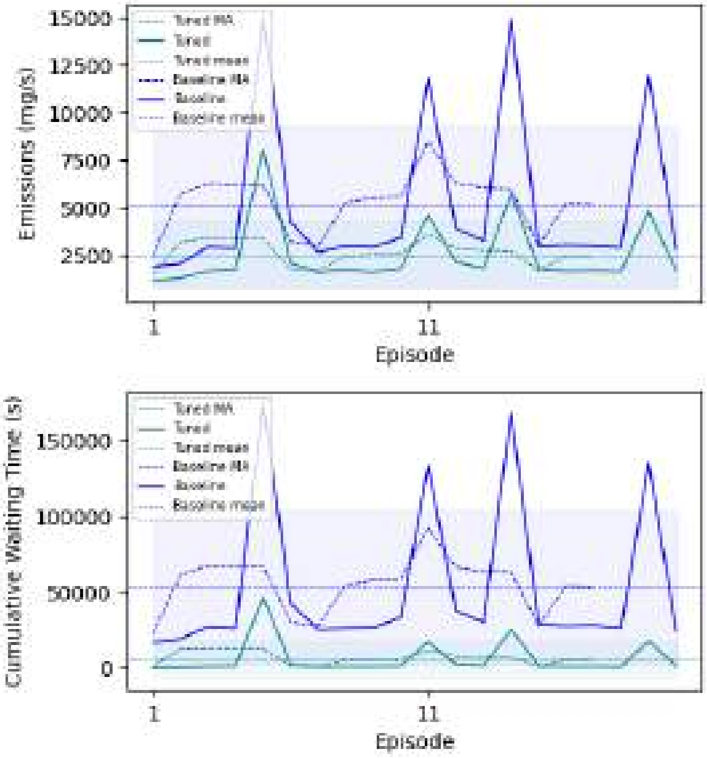


Figure 5.16. Testing episodes of the models with tuned vs baseline parameters

5.3.2.9 DQN vs no-DQN Simulations

In addition to comparing the tuned DQN model with the baseline, we also conducted a separate comparison with a fixed time approach (no-DQN). For this comparison, we ran simulations using the 110-second period-traffic light static settings of *Av. Ferrocarril - Calle 48*, the road that we used for the tuning, training, and testing procedures.

```
<tlLogic id="cluster_364144614_365473230_3663240788_3663240789"
type="static" programID="1" offset="0">
   <phase duration="26" state="rrrrrrrrrrrGGGGGGrrrrrrGrGGGG"/>
   <phase duration="6" state="rrrrrrrrrrryyyyyyrrrrrryryyyy"/>
   <phase duration="6" state="rrrrrrrrrrrGGGGGGGGGGrrrrrrrr"/>
   <phase duration="6" state="rrrrrrrrrrryyyyyyyyyyrrrrrrrr"/>
   <phase duration="27" state="GGGGGrrrrrrrrrrrrrrrrGgggrrrr"/>
   <phase duration="6" state="yyyyyrrrrrrrrrrrrrrrryyyyrrrr"/>
   <phase duration="27" state="rrrGGGGGGGGrrrrrrrrrrrrrrrrrr"/>
   <phase duration="6" state="rrryyyyyyyyrrrrrrrrrrrrrrrrrr"/>
</tlLogic>
```

In this case, the DQN approach significantly outperformed the fixed time traffic light metrics, as shown in Figure 5.17.

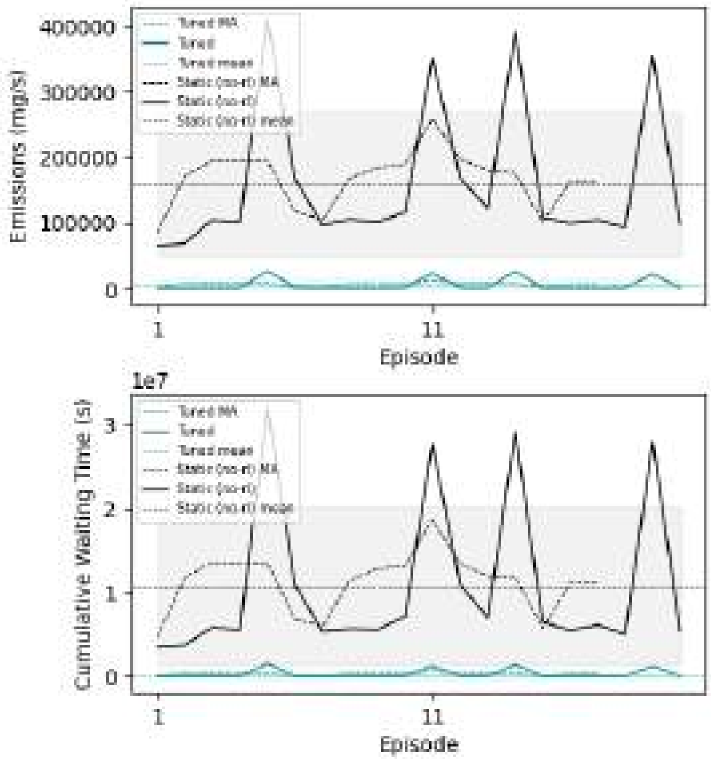


Figure 5.17. Testing episodes for the tuned model with proposed reward vs no-DQN

5.4 Experiment 2: Multi-Agent

Within the scope of our research, as described in chapter 4, the LEZ we are working with encompasses nine different roads. To address this, we have adopted a multi-agent architecture, with each agent being a unique entity developed based on the attributes of the baseline agent, leveraging the insights gained from the single-agent framework, described in section 5.3.

During the development of our multi-agent approach, a critical decision was made regarding the architecture that agents should use for their learning process. After an evaluation, we opted for a decentralized architecture for the individual agents. In contrast to a centralized model, where all agents share a single neural network, a decentralized approach minimizes the risk of computational inefficiency, particularly when scaling the system to accommodate more agents or adapting to changing environmental conditions. Moreover, introducing a new agent or modifying the network structure in a centralized model could involve substantial computational overhead.

One significant aspect that influenced our choice of a decentralized architecture is the real-world nature of our scenario. In our real-world scenario, traffic coming from nearby roads enters into the area of influence of a traffic light, introducing noise and dynamic factors that also have to be considered in the training process. The decentralized architecture provides each agent with its own neural network, enabling them to learn independently and adapt to their specific conditions more effectively, which is crucial when dealing with the unpredictable nature of traffic in a complex urban environment. This approach promotes a higher degree of autonomy among agents, allowing them to respond to the environment's dynamics without disrupting the entire system. Thus, our choice of a decentralized architecture, shown in Figure 5.18, reflects our goal of achieving a more adaptable, scalable, and computationally efficient multi-agent learning framework customized to the specific challenges of our real-world scenario-based environment.

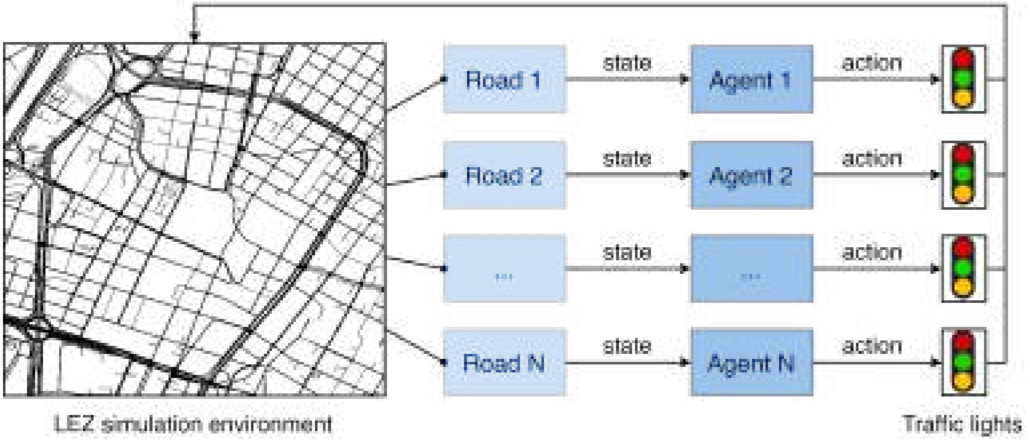


Figure 5.18. Multi-agent architecture

5.4.1 Simulation Parameters

After tuning our single-agent Deep Q-Network (DQN) algorithm in section 5.3, the fine-tuned parameters were taken as the setup for the agents in our multi-agent DQN architecture with the reward function $R_t = 0.2 * \Delta ctwt + 0.8 * emissions_t$. These parameters, shown in Table 5.5 have proven effective in enhancing the performance of a single agent, and they serve as a solid starting point for a multi-agent framework. In the following sections, results and discussion of how this framework behaves in terms of learning process, waiting times, and emissions across different roads are shown, going through different scenarios and loads of traffic (number of simulated routes, thus, the number of vehicles). Simulations for two (subsection 5.4.2.1), four (subsection 5.4.2.2), six (subsection 5.4.2.3), and nine roads (subsection 5.4.2.4) were run.

Considering the chosen architecture (Figure Figure 5.18) and the simulation parameters, the corresponding algorithm is designed and executed. The pseudocode for the multi-agent approach for the DQN is presented in Algorithm 2, implementing a multi-agent extension of the DQN framework for traffic light control. The primary objective remains consistent: training individual neural networks for each agent to select the optimal traffic light actions based on specific states.

By allowing each intersection to optimize its traffic light control, the system can adapt more effectively to local traffic conditions. Additionally, this decentralized approach enables intersections to explore different strategies independently, promoting a more diverse set of behaviors and decisions. Furthermore, in a decentralized system where each intersection operates autonomously, failures might remain localized, ensuring the overall system's operation. This is particularly crucial in dynamic environments where traffic circumstances may be unpredictable.

On the other hand, a decentralized approach offers scalability advantages, especially in real-world scenarios with large and complex traffic networks. As the number of intersections increases, scalability becomes more manageable. This scalability is essential for addressing the challenges posed by expanding urban environments and growing traffic volumes. Despite this being the chosen approach for the development of this work, multi-agent coordination is still considered important and encouraged to evaluate its performance under the case study scenario and test its response, and it is included as a future research line in section 6.2.

To implement the described proposal, in this case, an agent-specific initialization is needed before the simulation begins, where the simulation parameters shared across all agents are set. These might include general settings for the SUMO environment, traffic configurations, and the tuned simulation parameters. For this purpose, the algorithm has an adaptable structure where the number of roads to simulate in used as an input. With this parameter, the main simulation script identifies the number of route files,

Algorithm 2: Pseudocode for the DQN multi-agent simulation

```
Data: Input routes, number of roads/agents (same as edges), and tuned training parameters
Function main():
    Initialize simulation parameters;
    foreach agent do
        Initialize agent-specific simulation memory;
        Initialize agent-specific neural network model with corresponding output size (action
          state);
    for each episode do
        Initialize simulation;
        while multi-agent simulation is active do
            Simulate time step;
            foreach agent do
                Collect state information;
                Choose action for traffic light ($\epsilon$-greedy);
                Execute traffic light actions;
                Calculate rewards;
                Add sample to agent-specific memory;
        foreach agent do
            Train agent-specific model using memory replay;
        Save episode statistics;
    Save data and visualize results;
Function replay(model, agent-specific memory):
    Retrieve a group of samples from agent-specific memory;
    Update Q-values using the learning equation;
    Train the neural network model;
```

models, and memories that it should create. Following this, for each agent, specific simulation memory is initialized, as well as the agent-specific neural network models. The size of the neural network's output layer is configured to match the action state of that particular agent. This is achieved by using a *model_config* dictionary, with the specific number of phases of each traffic light.

After running the episodes and gathering data, each agent's neural network model is trained using experiences from its own replay buffer. The training process involves updating the Q-values based on Bellman's rule (Equation 5.5).

This multi-agent setup helps each agent to operate within its specific environment and learns from its own experiences, considering traffic that may come from other roads. It allows for a more decentralized approach where each agent can potentially tackle different traffic scenarios or intersections independently.

5.4.2 Results

In the analysis of multiple road scenarios, we will use the sum for metrics such as emissions, rewards, and waiting times, aggregating values across all roads for each episode. This approach enables us to estimate the total cumulative impact across all roads and episodes, offering a comprehensive view of the overall performance and environmental impact. In addition, test simulations are shown for the four-road and six-road scenarios, which represent intermediate traffic scenarios among the one (already presented in subsection 5.3.2.9 two, four, six, and nine roads.

5.4.2.1 Two-Road Scenario

Considering Figure 5.19, *Av. Ferrocarril - Colombia* tends to have higher emissions compared to *Av. Ferrocarril - Calle 48*. This indicates potentially heavier or more challenging traffic scenarios at the first one.

In relation to rewards, *Av. Ferrocarril - Colombia* shows more pronounced fluctuations, when compared to *Av. Ferrocarril - Calle 48*, which presented lower peaks.

Regarding waiting time, *Av. Ferrocarril - Colombia* presents two significant peaks, but the behavior is decreasing with the progression of episodes, where it finally meets values similar to those presented by the other road.

Emissions also show a decreasing tendency, with peaks along the process, which may be an indication of exploration actions carried out during the training episodes, accompanied in some cases by higher traffic situations.

5.4.2.1.1 DQN vs no-DQN in a two-road scenario

To observe the behavior of the trained model, 30 episodes of testing were run with emissions and waiting time data that had not been used during the training process (same data for both approaches). Test and no-DQN simulations were run for both roads simultaneously. Results are shown in Figure 5.20.

The DQN model outperforms the static model with a significant reduction in both emissions and waiting times in both roads. This suggests that the DQN model is better suited for dynamic traffic conditions.

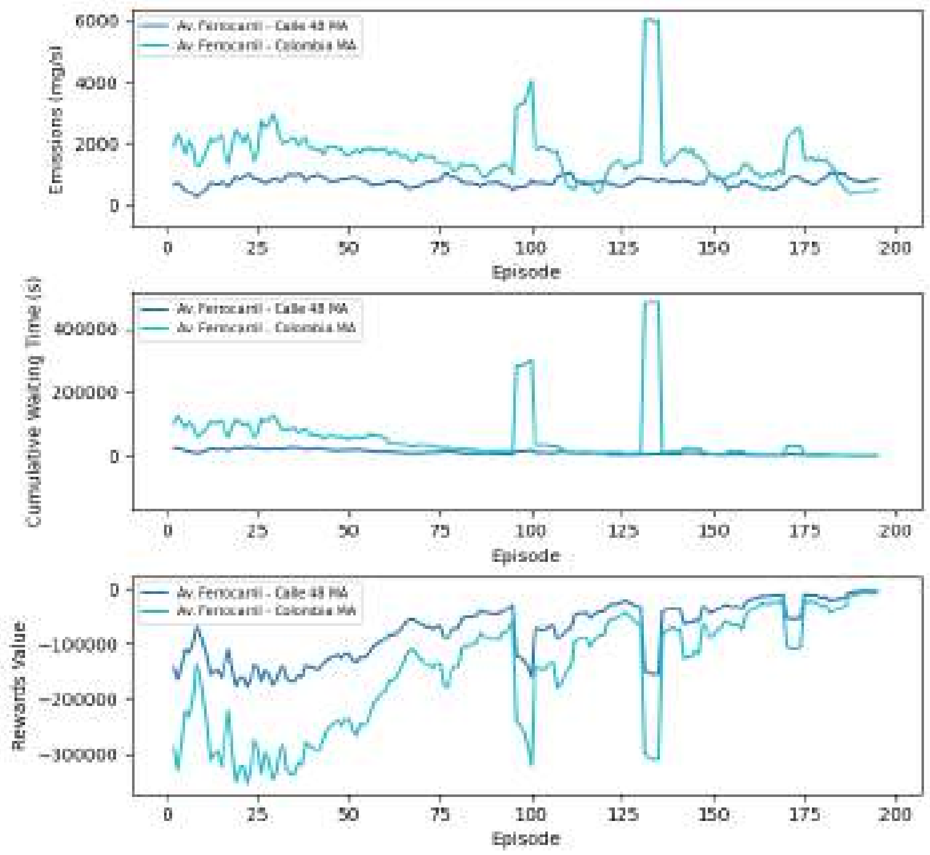


Figure 5.19. Multi-agent simulation training process for two roads

5.4.2.2 Four-Road Scenario

Considering the combined performance of the Deep Q-Network (DQN) across four roads (plotted in Figure 5.21), several insights emerge:

In terms of emissions, *Av. Ferrocarril - Calle 48* shows higher emissions compared to other roads, suggesting more challenging traffic conditions or heavier traffic flow. The curves for all plots suggest a decreasing behavior in emissions and waiting times, aligned with the rewards obtained during the process. However, this is consistent until episode 150 approximately (see sum lines), as shown in Figure 5.22.

Episodes around 150 and 175 show high peaks for *Av. Ferrocarril - Calle 48* and *Av. Ferrocarril - Colombia*. This reflects heavy traffic conditions that represented a challenging scenario where the response of the model might have been exceeded. However, it is noteworthy that these peaks were not

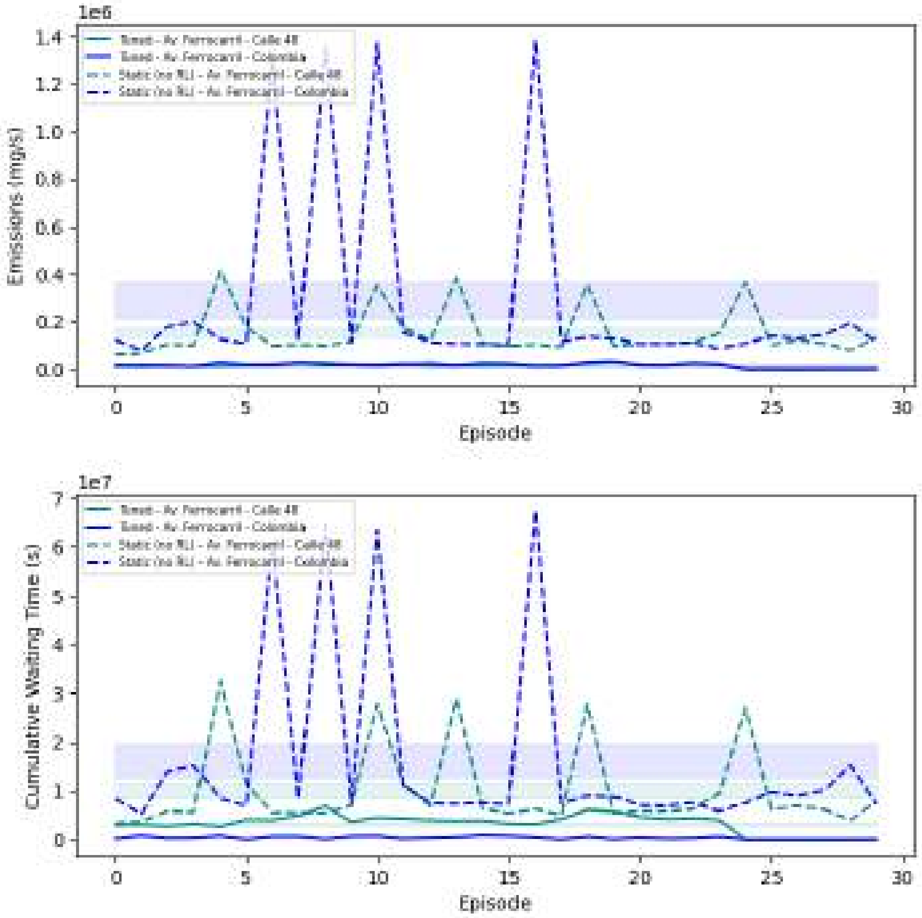


Figure 5.20. Testing episodesfor the DQN vs no-DQN (fixed time) approaches in the two-road scenario

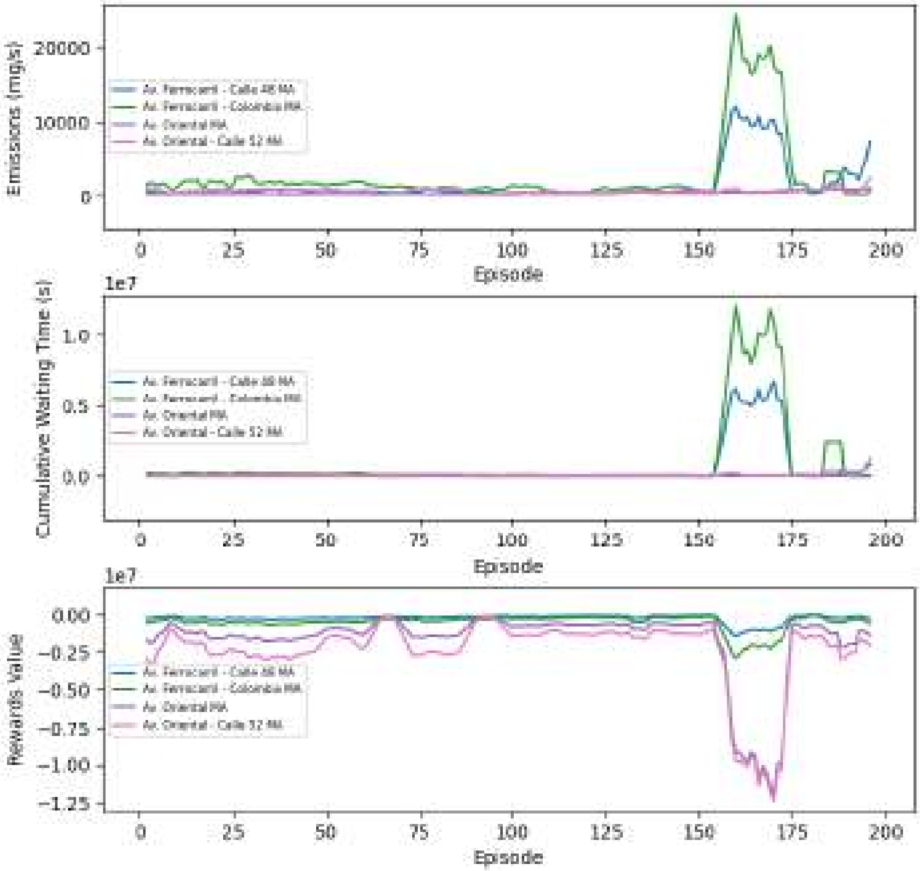


Figure 5.21. Multi-agent simulation training process for four of the Low Emission Zone's roads

persistent since subsequent episodes demonstrated a return to more manageable levels of traffic, indicating the adaptability and effectiveness of the model in navigating through fluctuating traffic scenarios.

When analyzing rewards, both *Av. Oriental* roads demonstrate significant fluctuations, with some episodes yielding lower rewards, that considering subsequent episodes, tended to increase

Regarding waiting times, *Av. Ferrocarril - Colombia* again presented higher values, suggesting longer waiting periods which could be contributing to its higher emissions. The summed waiting times across all roads provide an overview of the DQN's impact on reducing congestion and improving traffic flow.

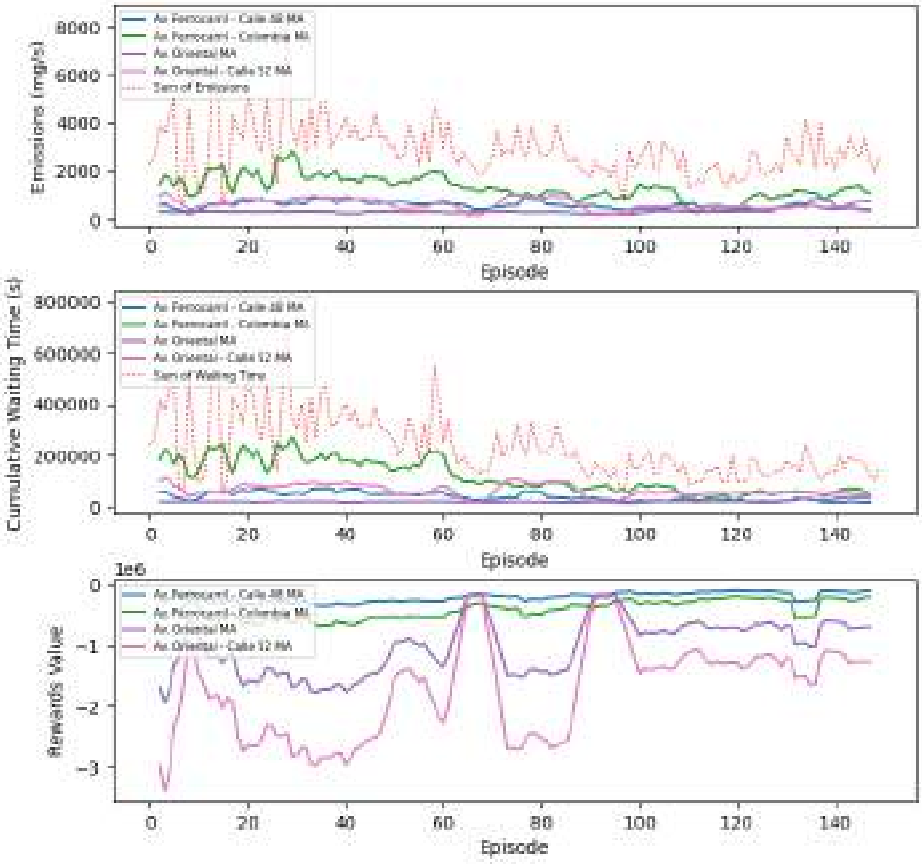


Figure 5.22. Multi-agent simulation training process for four of the Low Emission Zone's roads - Episodes 0 to 150

5.4.2.2.1 DQN vs no-DQN in a four road scenario

To observe the behavior of the trained model, 30 episodes of testing were run with emissions and waiting time data that had not been used during the training process (same data for both approaches). Test and no-DQN simulations were run for all four roads simultaneously. The results are shown in Figure 5.23.

The DQN simulation in this scenario demonstrated significantly lower emission values compared to the static approach, indicating its potential as a more eco-friendly solution. The DQN approach also led to shorter waiting times, contributing to improved traffic flow. Overall, for both variables, the DQN approach appears to offer benefits in terms of traffic flow and pollution

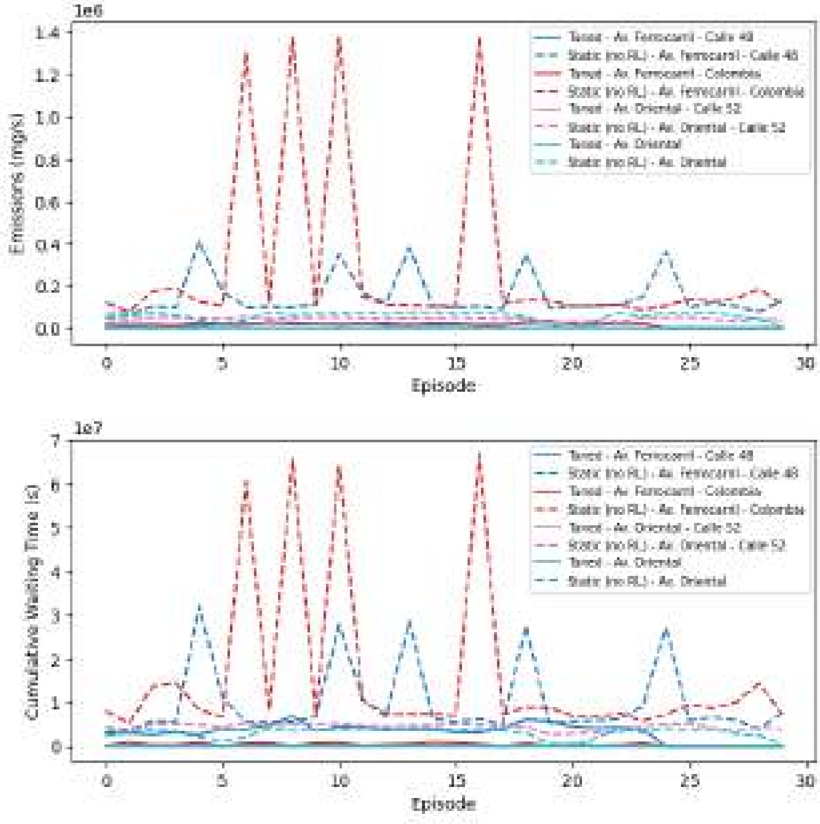


Figure 5.23. Testing episodes for the DQN vs no-DQN (fixed time) approaches in the four-road scenario

reduction over the static approach.

Figure 5.24 displays the comparison between DQN and the static approach during the same episode of simulation, at the same step, using as input the same set of traffic data.

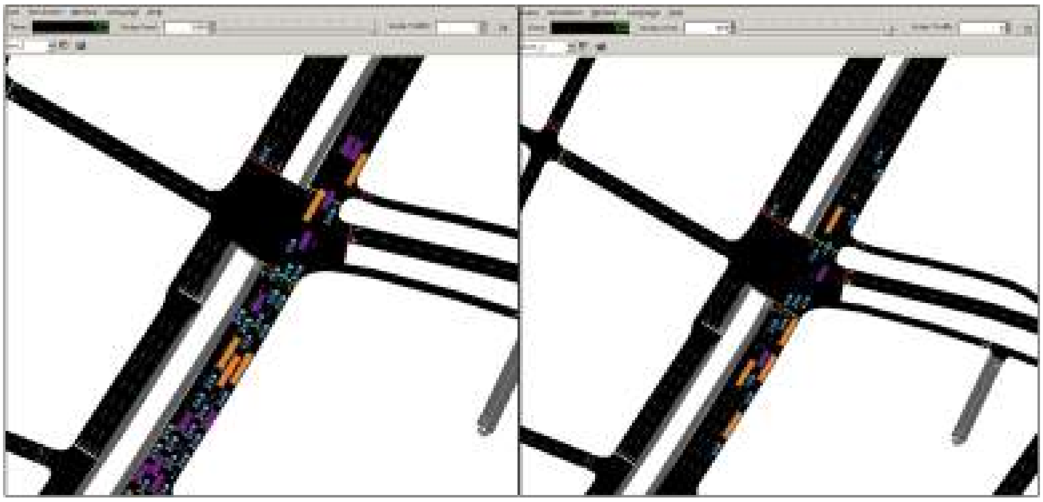


Figure 5.24. Comparison of the same intersection, DQN vs Static approach (4 roads - step 500 of simulation)

5.4.2.2.2 DQN vs no-DQN in a four-road scenario during weekdays

To test how well the model performs in an hourly basis, a simulation with this scenario (4 roads) was executed, extracting the information from the data analysis process. The four-road setup was chosen since its performance represents an intermediate point among the models (one, two, four, six, and nine).

The scenario was simulated between 6:00 am and 10:00 pm (22:00), and the available traffic data (trips) are shown in Figure 5.25. Next, tests were carried out using the static and the trained model, as shown in Figure 5.26.

These results showed that, despite having a daily granularity for training our models, the hourly performance is really good, and as expected, the DQN approach outperforms the static one.

We can also see that, as in the daily basis simulations, *Av. Ferrocarril - Colombia*, and *Av. Oriental - Calle 57* represent the highest contributions

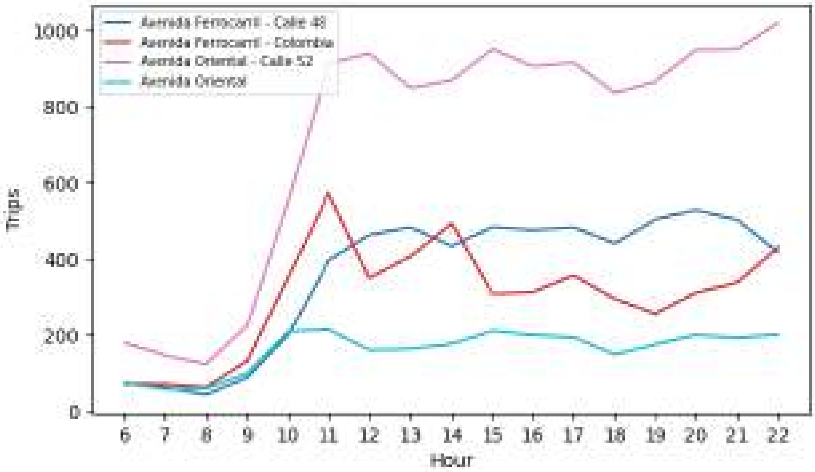


Figure 5.25. Data for the four-road scenario - Wednesday 06/04/2022

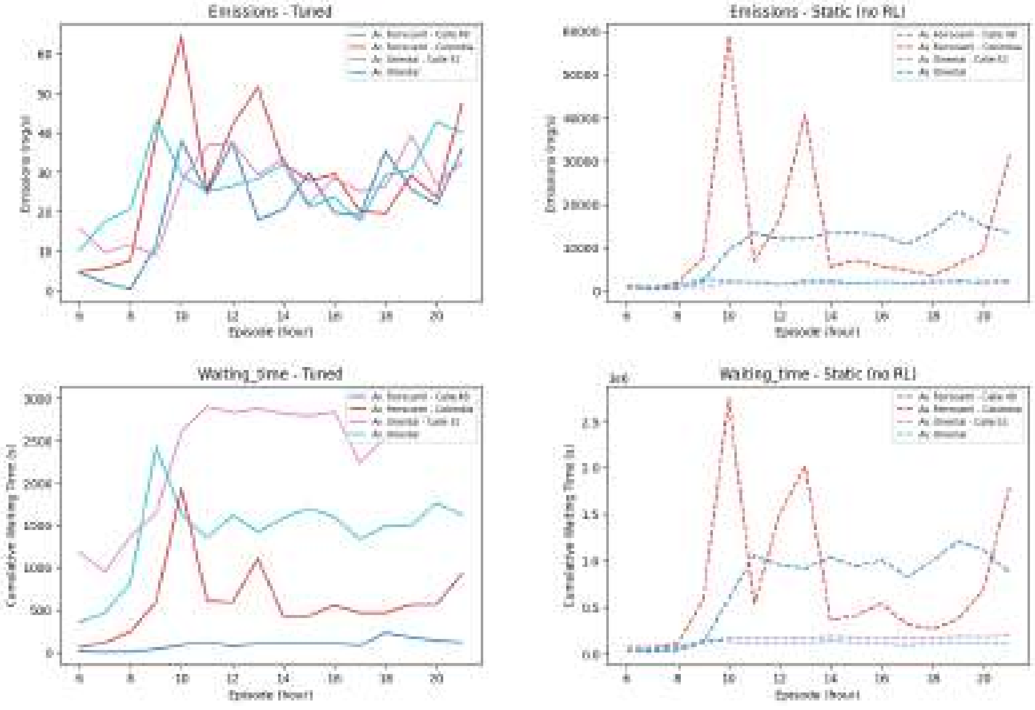


Figure 5.26. Grid plot - Test for the DQN vs no-DQN (fixed time) approach in the four-road scenario - Wednesday 06/04/2022

to waiting times and emissions.

5.4.2.3 Six-Road Scenario

An assessment of the DQN's operation across the six-road scenario, shown in Figure 5.27, gave us some insights:

The fluctuations in total emissions across the entire six-road network reflect the DQN's ongoing attempts to regulate traffic emissions. Yet, these fluctuations also point to a lack of consistent and effective control over emission levels.

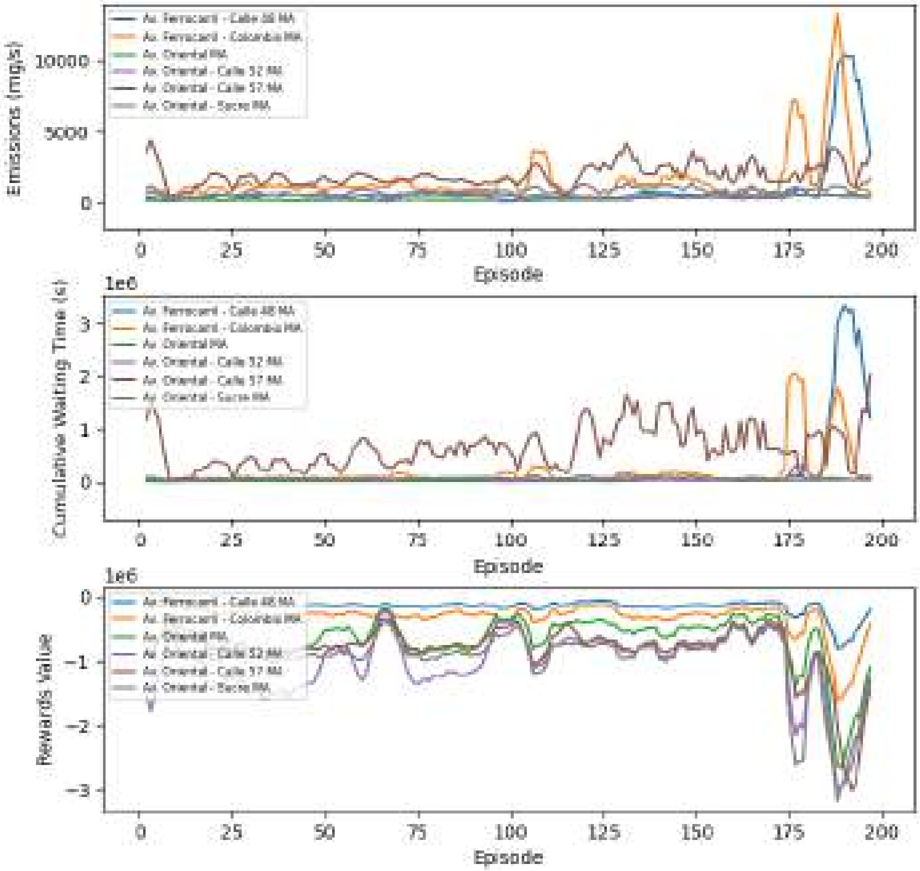


Figure 5.27. Multi-agent simulation training process for six of the Low Emission Zone's roads

Certain roads like *Av. Ferrocarril* - *Calle 57* and *Av. Ferrocarril* - *Colombia* stand out for their substantial contribution to overall emissions.

In terms of rewards, some roads show significantly higher negative rewards, signaling episodes of suboptimal traffic management. This is particularly evident on roads like *Av. Oriental - Sucre* and *Av. Oriental - Calle 52*, specially around episode 175 and subsequent.

When examining waiting times, the scenario is similar, with *Av. Oriental - Calle 57* experiences more elevated waiting times, indicative of serious congestion issues. The performance in waiting times is similar to the observed for emissions, in relation to the overall behavior along the simularion episodes. This contrast in waiting times across the roads further illustrates the diverse challenges encountered by the DQN in managing traffic flow effectively.

After performing these simulations, we can observe that roads like *Av. Ferrocarril - Calle Colombia*, and *Av. Oriental - Calle 57* represent focal points due to their significant contributions to emissions and waiting times.

Additionally, it's important to note that depending on the roads (thus agents) working within the DQN, the behavior of traffic across the road might present variations. For example, when comparing this scenario with the four-road and the nine-road scenario (shown next), the number of traffic lights operating with the DQN approach is different. This variation in the number of agents (traffic lights) can lead to differing traffic dynamics. In some cases, traffic might be alleviated in certain areas due to more efficient light coordination, but this could inadvertently lead to increased congestion in others if the network is not holistically balanced.

Also, the behavior observed around specific episodes like those between 175 and 200, where emissions and waiting times presented high values, could be due to the agent encountering and learning to respond to heavy traffic scenarios, and exploration-oriented decisions that could potentially lead to increasing waiting times.

The DQN's learning curve (rewards) and adaptability are shown progressively along the episodes up to episode 170 approximately (see Figure 5.28, since its tendency goes up and emissions and waiting time levels are "stable". However, the varying performance across different roads highlights the need for more road-specific strategies and potential areas for algorithmic improvement.

This phenomenon underscores the complexity of traffic management, where interventions in one part of the network can have ripple effects elsewhere, showing the need for a comprehensive approach to ensure overall traffic optimization.

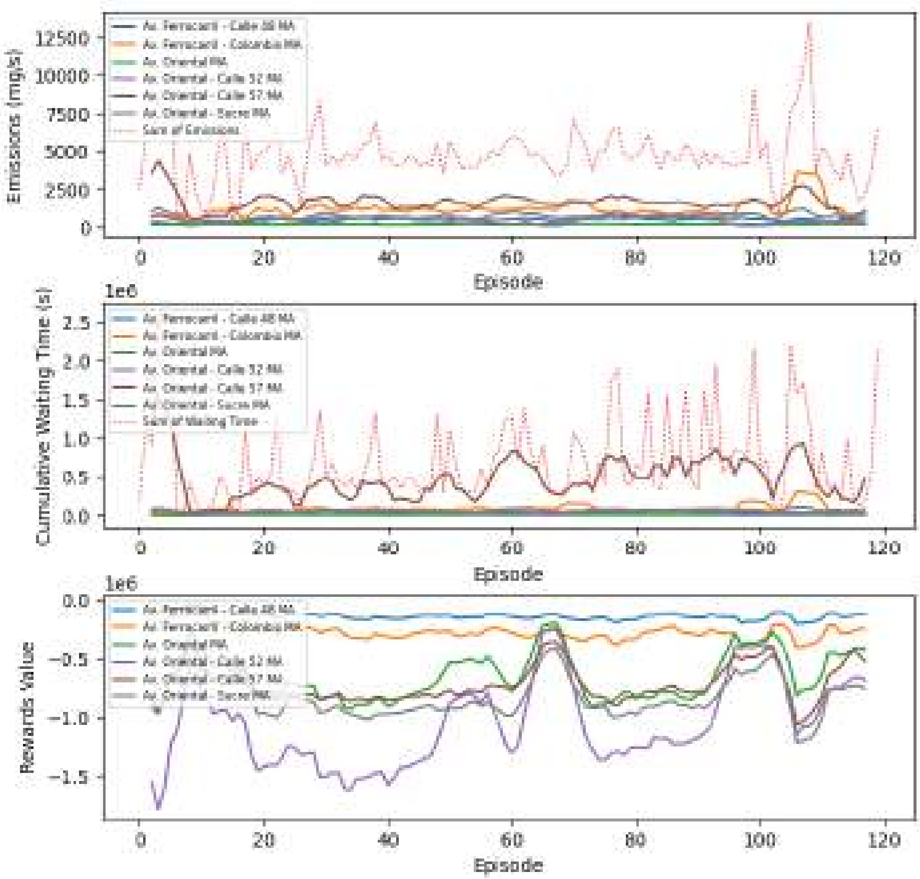


Figure 5.28. Multi-agent simulation training process for six of the Low Emission Zone's roads - Episodes 0 to 120

5.4.2.3.1 DQN vs no-DQN in a six-road scenario

To observe the behavior of the trained model in comparison to the fixed approach, as performed in the four-road scenario, 30 episodes of testing were run with emissions and waiting time data. The results for both approaches are shown in Figure 5.29.

According to the results, we observed significant differences between the static (no-DQN) approach and the DQN simulation. The static approach exhibited notably high emission values, specially in roads like *Av. Oriental - Calle 57* and *Av. Ferrocarril - Colombia*, which were also challenging during the training process. Additionally, *Av. Oriental - Calle 57* displayed longer waiting times. On the other hand, the DQN simulation demonstrated

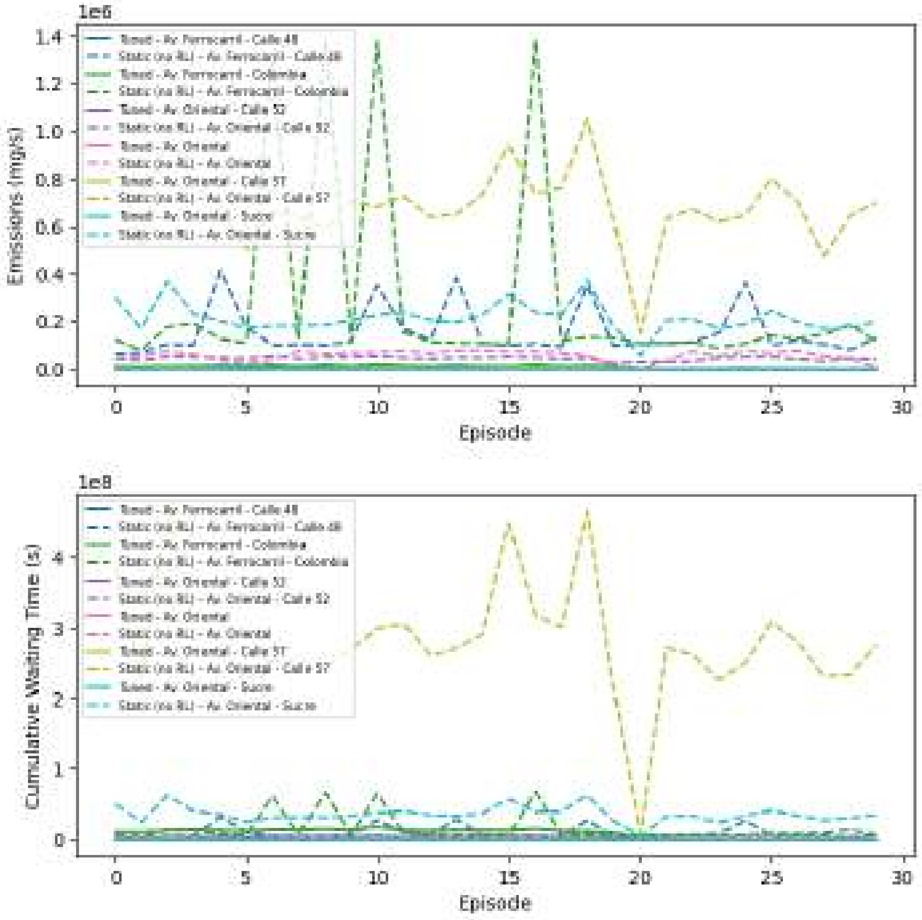


Figure 5.29. Testing episodes for the DQN vs no-DQN (fixed time) approaches in the six-road scenario

lower emissions and more efficient management of traffic during simulation episodes.

These results imply that the DQN approach might offer advantages in terms of traffic flow and pollutant levels over the no-DQN approach.

These findings are consistent with those in the four-road scenario, reinforcing the potential advantages of the DQN approach.

5.4.2.4 Nine-Road Scenario

Considering the performance of the DQN across the nine roads, a comprehensive analysis, based on Figure 5.30, gives diverse insights:

Emissions vary across the roads, but during the first 100 episodes, values range within an interval not exceeding 2000 mg/s, in the case of *Avenida Oriental - Calle 57* and *Av. Ferrocarril - Colombia*, that exhibit the highest levels (see Figure 5.31).

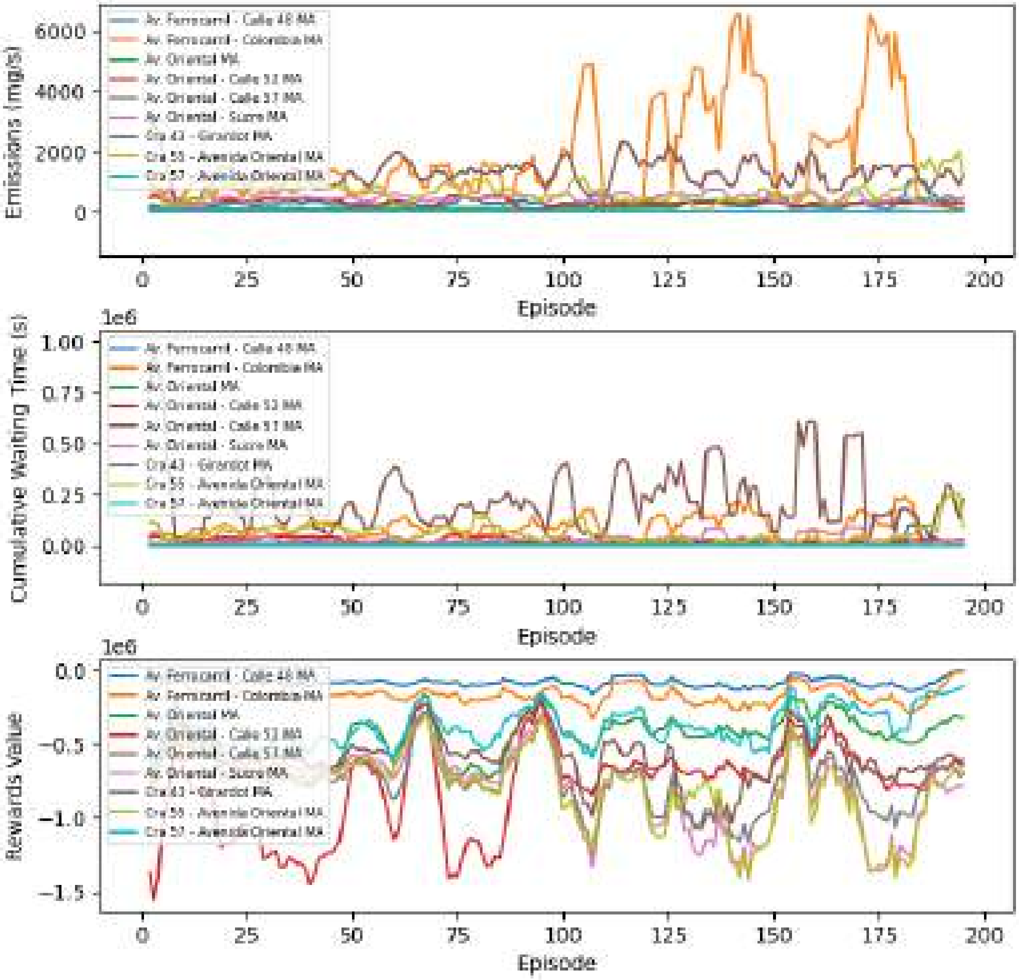


Figure 5.30. Multi-agent simulation training process for nine of the Low Emission Zone's roads

The summed emissions across all roads display fluctuations, indicative of the DQN's work in managing traffic emissions. However, this also indicates

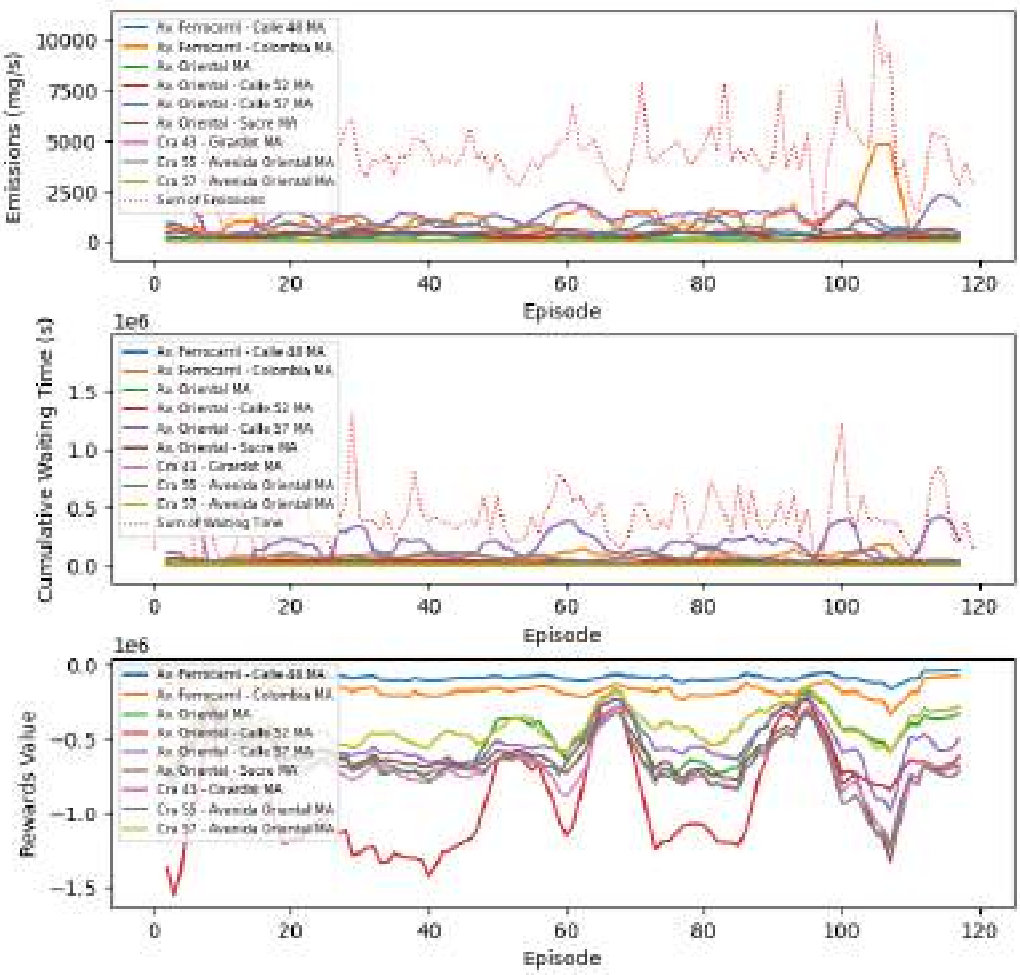


Figure 5.31. Multi-agent simulation training process for nine of the Low Emission Zone's roads - Episodes 0 to 120

that emissions are not being handled properly and consistenly. Notably, the pronounced contributions of certain roads to the total emissions highlight the variability in the DQN's impact across different locations.

Waiting times present a similar picture, with *Avenida Oriental* - *Calle 57* experiencing the highest values.

In terms of rewards, roads like *Av. Oriental* - *Calle 52* and *Cra 55* - *Avenida Oriental* demonstrate significantly higher negative rewards, indicating episodes of less successful traffic management.

The DQN's learning curve and adaptability are observed progressively across episodes with an upward treng, but the divergent performance on different roads, specially towards latter episodes (120¿) highlights that there is room for improvement and exploration of other techniques when approaching these kind of scenarios (complex and big).

5.4.2.4.1 DQN vs no-DQN in a nine-road scenario

To observe the behavior of the trained model in comparison to the fixed approach, as performed with the other scenarios, 30 episodes of testing were run with emissions and waiting time data. The moving averages for both simulations are shown in Figure 5.32. The utilized road was *Av. Ferrocarril - Calle 48*.

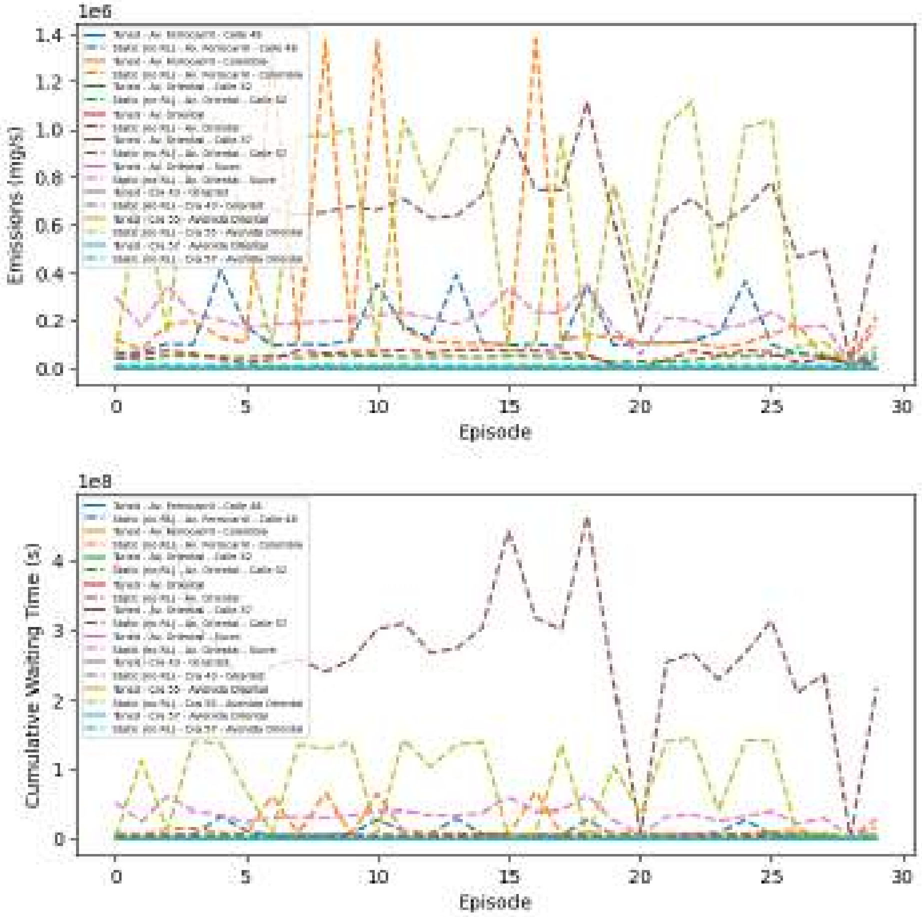


Figure 5.32. Testing episodes for the DQN vs no-DQN (fixed time) approaches in the nine-road scenario

According to the results, we observed significant differences between the static (no-DQN) approach and the DQN simulation. The static approach exhibited notably high emission values, contributing to a higher mean and standard deviation. Additionally, it showed longer waiting times on average. On the other hand, the DQN simulation demonstrated lower emissions and more efficient waiting times on the same days.

In terms of waiting times, the DQN approach initially (between 15th and 21th episodes) showed some values rise to levels similar to those of the static approach but then decreased again. This suggests the agent's adaptive capability. These results imply that the DQN approach might offer advantages in terms of traffic flow and pollutant levels over the no-DQN approach.

These findings are consistent with those from the four-road and six-road scenarios, reinforcing the potential advantages of the DQN approach.

After performing this simulations, it's clear that roads *Cra 55 - Avenida Oriental*, *Av. Ferrocarril - Colombia*, and *Av. Oriental - Calle 57* represent focal points due to their significant contributions to emissions and waiting times. This leads us to believe that all the road segments that compose *Avenida Oriental* (Figure 5.33) should be addressed with more intensive traffic management mechanisms, as, according to the simulation results, this road is one of the most critical in terms of congestion, (followed by *Av. Ferrocarril*) thus, emissions in the Low Emission Zone.

As a remark, it's important to consider that the DQN positively affects traffic flow, but so far it only focuses on the traffic lights for the specific road segments analyzed. It's plausible that some static lights within the simulation environment may contribute to additional delays, indicating a broader scope for optimization and coordination in traffic light management.

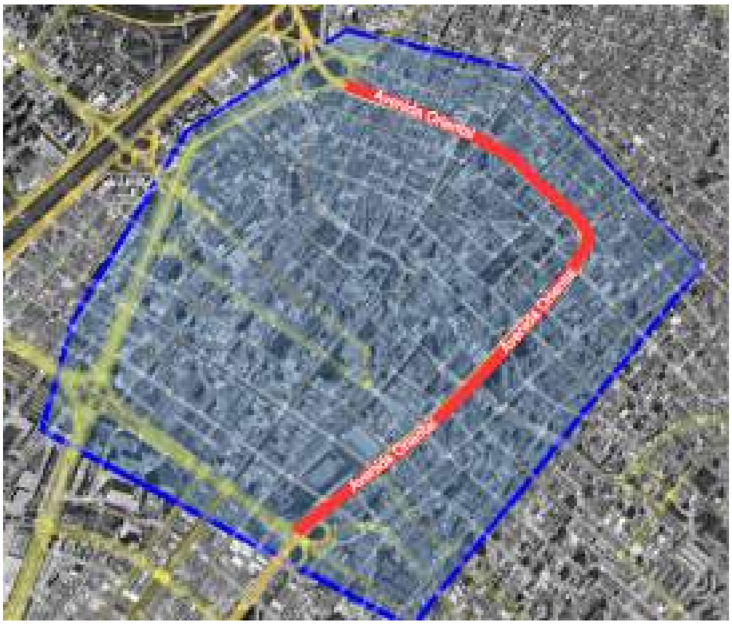


Figure 5.33. Avenida Oriental road in Medellin's Low Emission Zone

5.5 System Performance Analysis

In this section, we go through the hardware specifications of the computational platform utilized in our research, as well as the resulting execution times and memory management.

5.5.1 Hardware Resources

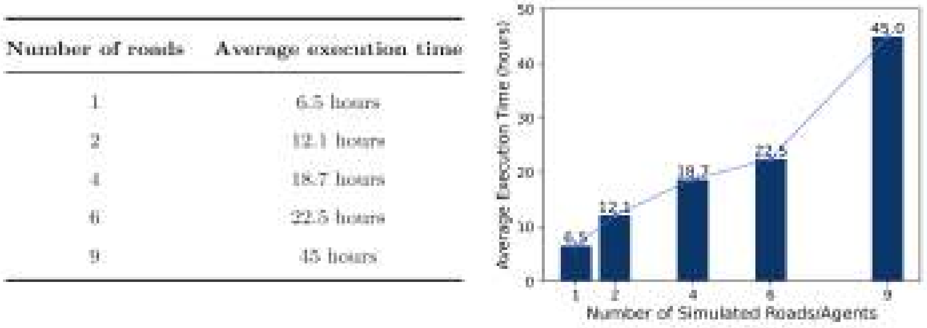
Our computational work was carried out on a DELL workstation. The system features a Core i9 12900 processor with an Intel architecture. With a substantial 128GB of DDR5 ECC RAM, this computer boasts significant memory capacity, complemented by a hybrid storage solution. A NVIDIA RTX A4000 graphics card with 16GB DDR6 memory enhances the system's processing capabilities and supports CUDA. This combination of hardware and software capabilities helped speed up training tasks in TensorFlow framework [124] along with the SUMO scenario simulation. Table 5.6 summarizes the workstation features.

Table 5.6. Workstation Features

Feature	Value
Format	Workstation
Processor	Intel Core i9 12900
RAM	128GB DDR5 ECC
Hard Drive	1 TB SSD PCIe + 2TB SATA 7.2K RPM
Video Card	NVIDIA RTX A4000 16GB DDR6
Operating System	Ubuntu 22.04

5.5.2 Simulation Execution Times

Since the implementation of the proposed DQN architecture required the training of several neural networks and higher-traffic scenarios, despite traffic being simulated simultaneously, the execution times were affected when increasing the number of simulated roads. Figure 5.34 shows the execution times for the simulation and training processes of the DQN framework with different numbers of roads.



Number of roads	Average execution time
1	6.5 hours
2	12.1 hours
4	18.7 hours
6	22.5 hours
9	45 hours

Figure 5.34. Number of roads vs execution times for the Low Emissions Zone's simulations

In Figure 5.34, the number of simulated roads and their respective execution times reveals a clear trend: as the complexity of the simulation, represented by the number of roads, increases, the average execution time also rises. This suggests a positive correlation between these two variables which appears to be linear, indicating that each additional road leads to a consistent increase in execution time. Consequently, resource-intensive computational requirements could pose challenges, especially when dealing with larger scenarios. For this, techniques like transfer learning [125] or shared experience replay [126] could be explored to be more data-efficient and learn policies for diverse road conditions.

Also, to address these computational challenges and optimize the system, strategies like multi-GPU systems for parallel execution of neural network training can be explored. With this, execution times for traffic scenario simulations could be reduced. Additionally, employing parallel processing techniques, like distributing neural network training across GPUs may lead to efficiency gains.

5.5.3 Checkpoints and Memory Usage

For the execution of the DQN algorithm, a checkpoint system was implemented, which involved saving and loading the neural networks model files and their weights during the training process for two key reasons: first, the checkpoints acted as a safety net against potential disruptions, such as energy interruptions or unexpected external factors. In the event of an interruption, the saved checkpoints enabled to resume the training process from the last point where it saved the model's weights. This approach was facilitated by the creation of a bash script that executed the training process, specifying initial and final episode ranges for each checkpoint. Depending on the episode reached, by using a parameter the script determined whether to initialize a new model or load an existing one, ensuring training continuity and robustness. Second, even though our workstation has high computational resources, the process of training and saving DQN models in memory was still a resource-intensive task, that got even more intense by increasing the number of simulated roads. By using checkpoints, it could effectively manage and lighten the computational load, allowing the training to continue without overburdening the system.

5.5.4 Applicability and Scenario Complexity

In the process of understanding system performance, we can discuss the applicability of the DQN model within real-world traffic scenarios. Unlike controlled environments (like SUMO's), the real-world often come with challenges and complexities that extend beyond hardware scalability. These challenges can significantly affect the model's effectiveness and scalability in deployment.

For example, in our simulation, we acknowledge that certain elements, such as traffic light IDs and origin-destination points (subsubsection 5.2.1.2), were established manually. This manual setup not only highlights the need for human intervention but also emphasizes the importance of realistic scenario design. In real-world applications, such manual setups can be both time-consuming and error-prone, especially when dealing with large urban areas or complex road networks. Scalability, therefore, depends on the ability to automate and generalize this scenario design process, ensuring that the model can adapt to various urban scenarios and traffic conditions.

In terms of roads, the scalability of our DQN model relies on its ability to generalize across different geographic regions and coverage areas. A model designed for a specific road network may not readily adapt to a completely different urban area, the model should be able to transfer knowledge and adapt its policies across various regions and traffic scenarios.

Data is another crucial factor in real-world applications, because the dynamic nature of traffic conditions and the availability of real-time data play an important role. Keeping this in mind, there should be availability and a processing capability of ingesting data and adapting to this dynamic information.

On the other hand, there are some ethical and regulatory considerations. As autonomous systems become more prevalent on the road, regulations and safety standards will play an important role in determining the scalability and deployment of such models.

Chapter 6

Conclusion and Future Work

6.1 Conclusion

In response to the research question, *"How can a Machine Learning-based model approach be effectively applied in urban traffic management to enhance traffic efficiency and reduce emissions within a Low Emission Zone?"*, this study has made notable efforts to address the intricacies and dynamics of urban traffic management. The findings presented throughout this research affirm that Machine Learning approaches, particularly Deep Reinforcement Learning, hold immense promise for the enhancement of traffic efficiency and emissions reduction within urban settings, particularly in the context of Low Emission Zones. The combination of SUMO, TraCI, and Python enabled us to conduct parameterized experiments and run-time evaluations of traffic control strategies, facilitating the optimization of traffic management policies and the improvement of efficiency in simulated road networks. An integral part of this journey was the discovery of the significance of model tuning, where our selection of state variables was adapted based on empirical data, with *Intensity* emerging as a significant contributor to improved model performance. This iterative methodology extends beyond the refinement of DQN elements, to encompass the optimization of specific components within the neural networks. This underscores the adaptive nature of model development, resulting in a more efficient DQN model.

The exploration of reward models played a significant role in achieving our project's goal: reducing congestion and evaluating their impact on emissions. This underscores the fundamental connection between congestion reduction and emissions control, while highlighting the ongoing need for adaptability and refinement of the DQN to optimize traffic management effectively.

As highlighted in subsection 5.4.2, the analyses showed a varied impact of the DQN across different locations, revealing disparities in effectiveness among road segments. This underlines the importance of continuous adaptation, and improvement of the DQN to enhance traffic management strategies.

While this research illustrates the potential of reinforcement learning in vehicular traffic management, it also presents some challenges that need to be addressed before these strategies can be effectively deployed in the real world. Despite reinforcement learning models have shown promise in optimizing traffic management, the transition from simulated environments to real-world scenarios presents complexities. The need for good quality real-world data and significant interaction of the agents with the environment to learn effective policies, coupled with ensuring citizen safety and data security, remains a critical concern. As these systems become integral to urban traffic management, safeguarding both the well-being of citizens and the integrity of data is essencial.

Furthermore, the successful implementation of these systems in real-time urban environments calls for a robust and reliable infrastructure. This includes addressing the challenges of integrating these technologies with existing traffic management systems, securing real-time data transmission, and ensuring that the systems can operate under unpredictable urban and weather conditions. Additionally, the computational capacity required to handle the complexities of real-world traffic scenarios, where countless variables are at play, presents a huge challenge, however, the potential contribution of machine learning techniques to improve traffic efficiency and reduce emissions in urban environments should not be ignored.

6.2 Future Work

In this section, we present potential paths for future research and development. These directions aim to enhance the effectiveness of traffic management and emissions reduction efforts, building on the findings of this study.

- ✓ Hourly analyses and data input enhancement: Consider conducting hourly or time-of-day-specific analyses to account for temporal variations in traffic patterns and emissions. Enhance the simulation environment by incorporating real-time data input, enabling the model to adapt to changing conditions throughout the day.

- ✓ Validation of the obtained test values for air quality (emissions) and waiting times, to check whether they are within accepted ranges for the Air Quality Index, and other national or international considerations regarding this matter.

- ✓ Customization for road-specific strategies: Implement a higher degree of customization for each road or intersection within the urban area; adapt traffic management strategies to the specific characteristics and demands of individual roads, thereby optimizing traffic flow and emissions control in a more granular manner.

- ✓ Exploration of multi-agent coordination techniques: Investigate advanced multi-agent coordination techniques to enhance the overall efficiency of traffic management in urban areas, considering the interactions between various agents within a Low Emission Zone. This must be implemented carefully, since challenges such as scenario complexity, computational demand might be present. besides, there could be biases towards local optimum values found by individual agents who are trying to maximize their own rewards [127]. Also, in strategies such as sequential decision making [84], scalability could present issues, due to the exponentially growing action space.

- ✓ Exploration of alternative state representations: Extend the research by exploring different state representations to capture a more comprehensive view of the urban traffic environment. This may involve incorporating additional variables or alternative data sources to improve the model's decision-making capabilities. Also, if the infrastructure of the traffic light network allows it, communication schemes among agents, wireless sensor networks, or V2I [7] could be involved to improve decision taking [128].

- ✓ Implementation of additional training episodes and supplemental data: Explore the potential benefits of incorporating additional training episodes supported by supplemental data provided by CITRA. This approach holds promise for enhancing the model's performance in real-world applications, providing an optional avenue for improvement.

Bibliography

[1] Area Metropolitana, *Páginas - ¿quiénes somos?* Retrieved 11 August 2023, from `https://www.metropol.gov.co/area/Paginas/somos/quienes-somos.aspx`, 2019.

[2] W. H. O. (WHO), *WHO Global Air Quality Guidelines. Particulate Matter (PM2.5 and PM10), Ozone, Nitrogen Dioxide, Sulfur Dioxide and Carbon Monoxide.* 2021. [Online]. Available: `https://iris.who.int/bitstream/handle/10665/345329/9789240034228-eng.pdf?sequence=1`.

[3] Medellín cómo vamos, *Informe de calidad de vida de medellín ICV 2020*, Retrieved 21 January 2022, from `https://www.medellincomovamos.org/system/files/2021-09/docuprivados/Documento%20Informe%20de%20Calidad%20de%20Vida%20de%20Medell%C3%ADn%202020.pdf`, 2021.

[4] I. Allegrini and F. Costabile, "An intelligent transport system based on traffic air pollution control," *WIT Transactions On Ecology And The Environment*, vol. 74, 2004. [Online]. Available: `https://www.witpress.com/elibrary/wit-transactions-on-ecology-and-the-environment/74/12478`.

[5] F. Ferreira, P. Gomes, A. C. Carvalho, *et al.*, "Evaluation of the implementation of a low emission zone in lisbon," *Journal of Environmental Protection*, vol. 3, pp. 1188–1205, Jan. 2012. DOI: `10.4236/jep.2012.329137`.

[6] Y. Bernard, T. Dallmann, K. Lee, I. Rintanen, and U. Tietge, *Evaluation of real-world vehicle emissions in brussels*, Retrieved 9 August 2023, 2021. [Online]. Available: `https://www.trueinitiative.org/media/792040/true-brussels-report.pdf`.

[7] D. Kanthavel, S. Sangeetha, and K. Keerthana, “An empirical study of vehicle to infrastructure communications - an intense learning of smart infrastructure for safety and mobility,” *International Journal of Intelligent Networks*, vol. 2, pp. 77–82, 2021, ISSN: 2666-6030. DOI: `https://doi.org/10.1016/j.ijin.2021.06.003`. [Online]. Available: `https://www.sciencedirect.com/science/article/pii/S2666603021000105`.

[8] J. Ye, S. Xue, and A. Jiang, “Attention-based spatio-temporal graph convolutional network considering external factors for multi-step traffic flow prediction,” en, *Digital Communications and Networks*, vol. 8, no. 3, pp. 343–350, Jun. 2022, ISSN: 23528648. DOI: `10.1016/j.dcan.2021.09.007`. [Online]. Available: `https://linkinghub.elsevier.com/retrieve/pii/S2352864821000675` (visited on 05/19/2023).

[9] Federal Highway Administration, *Traffic congestion and reliability: Trends and advanced strategies for congestion mitigation: Chapter 2*, Retrieved 31 July 2023, 2023. [Online]. Available: `https://ops.fhwa.dot.gov/congestion_report/chapter2.htm`.

[10] K. Zhang and S. Batterman, “Air pollution and health risks due to vehicle traffic,” *Science of The Total Environment*, vol. 450-451, pp. 307–316, 2013, ISSN: 0048-9697. DOI: `https://doi.org/10.1016/j.scitotenv.2013.01.074`. [Online]. Available: `https://www.sciencedirect.com/science/article/pii/S0048969713001290`.

[11] I. Thomson and A. Bull. “La congestión del tránsito urbano: Causas y consecuencias económicas y sociales.” (2001), [Online]. Available: `https://repositorio.cepal.org/bitstream/handle/11362/6381/1/S01060513_es.pdf`.

[12] D. E. B. Cardona, *Congestión vehicular y políticas públicas*, 2020. [Online]. Available: `http://hdl.handle.net/20.500.11912/6003`.

[13] J. J. García, C. E. Posada, and A. Corrales, “Congestión vehicular en medellín: Una posible solución desde la economía,” *Coyuntura Económica*, vol. XLVI, no. 16, p. 32, 2016. [Online]. Available: `http://dx.doi.org/10.2139/ssrn.2827118`.

[14] D. E. Betancur Cardona, *Congestión vehicular y políticas públicas*, 2020. [Online]. Available: `http://hdl.handle.net/20.500.11912/6003`.

[15] A. Ardila, "Control de la congestión vehicular en bogotá con herramientas microeconómicas," *Revista Desarrollo y Sociedad*, no. 35, pp. 7–26, 1995. [Online]. Available: `https://doi.org/10.13043/dys.35.1`.

[16] N. Taylor, *Evidence for speed flow relationships*, Retrieved 23 February 2024, 2014. [Online]. Available: `https://www.researchgate.net/publication/262004530_Evidence_for_speed-flow_relationships`.

[17] VISSIM. "Who we are." (2022), [Online]. Available: `https://www.vissim.no/about/`.

[18] Aimsun. "Acerca de aimsun." (2023), [Online]. Available: `https://www.vissim.no/about/`.

[19] P. A. Lopez, M. Behrisch, L. Bieker-Walz, *et al.*, "Microscopic traffic simulation using sumo," in *The 21st IEEE International Conference on Intelligent Transportation Systems*, IEEE, 2018. [Online]. Available: `https://elib.dlr.de/124092/`.

[20] A. Vidali, L. Crociani, G. Vizzari, and S. Bandini, "A deep reinforcement learning approach to adaptive traffic lights management," *Woa (pp. 42-50)*, p. 9, 2019.

[21] Land Transport Authority. "Intelligent transport systems." (2023), [Online]. Available: `https://www.lta.gov.sg/content/ltagov/en/getting_around/driving_in_singapore/intelligent_transport_systems.html#:~:text=Across%20the%20island%2C%20over%20a,how%20you%20get%20to%20places.`.

[22] Land Transport Authority. "Who we are." (2023), [Online]. Available: `https://www.lta.gov.sg/content/ltagov/en/who_we_are.html`.

[23] L. Smith. "Amsterdam smart city: A world leader in smart city development." (2022), [Online]. Available: `https://www.beesmart.city/en/smart-city-blog/smart-city-portrait-amsterdam`.

[24] Main Roads Western Australia (MRWA). "About us." (2020), [Online]. Available: `https://www.mainroads.wa.gov.au/about-main-roads/`.

[25] Office of the Auditor General. "Traffic management system." (2023), [Online]. Available: https://audit.wa.gov.au/reports-and-publications/reports/traffic-management-system/.

[26] Intertraffic. "Three smart cities in traffic management: Perth, moscow, mexico city." (2024), [Online]. Available: https://www.intertraffic.com/news/traffic-management/three-smart-cities-in-traffic-management-perth-mexico-city-moscow#Mexico.

[27] Área Metropolitana, SIATA. "Ciudadanos científicos." (2019), [Online]. Available: https://www.metropol.gov.co/ambiental/siata/Paginas/ciudadanos-cientificos.aspx.

[28] Secretaría de Movilidad. "Centro integrado de tráfico y transporte (citra)." (2022), [Online]. Available: https://www.medellin.gov.co/es/secretaria-de-movilidad/centro-integrado-de-trafico-y-transporte/.

[29] SIATA. "Quienes somos." (2022), [Online]. Available: https://siata.gov.co/sitio_web/index.php/nosotros.

[30] Energy Saving Trust, *Guide to low emission zones*, Retrieved 9 August 2023, 2022. [Online]. Available: https://energysavingtrust.org.uk/advice/guide-to-low-emission-zones.

[31] P. Panteliadis, M. Strak, G. Hoek, E. Weijers, S. van der Zee, and M. Dijkema, "Implementation of a low emission zone and evaluation of effects on air quality by long-term monitoring," *Atmospheric Environment*, vol. 86, pp. 113–119, 2014, ISSN: 1352-2310. DOI: https://doi.org/10.1016/j.atmosenv.2013.12.035. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1352231013009801.

[32] D. Allende, F. Castro, and S. Puliafito, "Air pollution characterization and modeling of an industrial intermediate city," *International Journal of Applied Environmental Sciences*, vol. 5, 2010.

[33] J. Gu, V. Deffner, H. Küchenhoff, *et al.*, "Low emission zones reduced pm10 but not no2 concentrations in berlin and munich, germany," *Journal of Environmental Management*, vol. 302, p. 114048, 2022, ISSN: 0301-4797. DOI: https://doi.org/10.1016/j.jenvman.

2021.114048. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0301479721021101.

[34] C. A. R. Board. “Inhalable particulate matter and health (pm2.5 and pm10).” (2023), [Online]. Available: https://ww2.arb.ca.gov/es/resources/inhalable-particulate-matter-and-health#:~:text=Particles%20are%20defined%20by%20their,diameter%20(PM2.5)..

[35] W. H. O. (WHO). “Ambient (outdoor) air pollution.” (2022), [Online]. Available: https://www.who.int/news-room/fact-sheets/detail/ambient-(outdoor)-air-quality-and-health.

[36] H. Beshir and E. Fichera, ““and breathe normally”: The low emission zone impacts on health and well-being in england,” English, Health Econometrics Data Group, University of York, WorkingPaper, May 2022.

[37] Greater London Authority, “CENTRAL LONDON ULTRA LOW EMISSION ZONE – SIX MONTH REPORT,” Greater London Authority, City Hall, The Queen’s Walk, More London, London SE1 2AA, 2019. [Online]. Available: https://www.london.gov.uk/programmes-and-strategies/environment-and-climate-change/environment-publications/expanded-ultra-low-emission-zone-six-month-report.

[38] D. Ku, M. Bencekri, J. Kim, S. Lee, and S. Lee, “Review of european low emission zone policy,” *Chemical Engineering Transactions*, vol. 78, pp. 241–246, 2020. DOI: 10.3303/CET2078041. [Online]. Available: https://www.cetjournal.it/index.php/cet/article/view/CET2078041.

[39] Alcaldía de Medellín. “Zonas urbanas de aire protegido.” (2021), [Online]. Available: https://www.medellin.gov.co/es/secretaria-de-movilidad/medellin-caminable-y-pedaleable/zuap/ (visited on 07/31/2023).

[40] “Air quality index (aqi) basics,” AirNow.Gov. (), [Online]. Available: https://www.airnow.gov/aqi/aqi-basics/.

[41] A. Plaat, “Deep reinforcement learning,” *CoRR*, vol. abs/2201.02135, 2022. arXiv: 2201.02135. [Online]. Available: https://arxiv.org/abs/2201.02135.

[42] R. S. Sutton and A. Barto, *Reinforcement learning: an introduction* (Adaptive computation and machine learning), Second edition. Cambridge, Massachusetts London, England: The MIT Press, 2020, 526 pp., ISBN: 978-0-262-03924-6.

[43] T. M. Moerland, J. Broekens, A. Plaat, and C. M. Jonker, *Model-based reinforcement learning: A survey. foundations and trends® in machine learning*, 2023. arXiv: 2006.16712 [cs.LG].

[44] V. François-Lavet, P. Henderson, R. Islam, M. G. Bellemare, and J. Pineau, "An introduction to deep reinforcement learning," *Foundations and Trends® in Machine Learning*, vol. 11, no. 3-4, pp. 219–354, 2018. DOI: 10.1561/2200000071. [Online]. Available: https://doi.org/10.1561%2F2200000071.

[45] G. Jörneskog and J. Kandelan, *Using deep reinforcement learning for adaptive traffic control in four-way intersections*, 2019. [Online]. Available: https://www.diva-portal.org/smash/get/diva2:1331101/FULLTEXT01.pdf.

[46] R. N. Boute, J. Gijsbrechts, W. van Jaarsveld, and N. Vanvuchelen, "Deep reinforcement learning for inventory control: A roadmap," *European Journal of Operational Research*, vol. 298, no. 2, pp. 401–412, 2022, ISSN: 0377-2217. DOI: https://doi.org/10.1016/j.ejor.2021.07.016. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0377221721006111.

[47] D. Precup. "Lecture 16: Markov decision processes. policies and value functions," McGill University. (2013), [Online]. Available: https://www.cs.mcgill.ca/~dprecup/courses/AI/Lectures/ai-lecture16.pdf.

[48] X. Zhu, F. Zhang, and H. Li, "Swarm deep reinforcement learning for robotic manipulation," *Procedia Computer Science*, vol. 198, pp. 472–479, 2022, 12th International Conference on Emerging Ubiquitous Systems and Pervasive Networks / 11th International Conference on Current and Future Trends of Information and Communication Technologies in Healthcare, ISSN: 1877-0509. DOI: https://doi.org/10.1016/j.procs.2021.12.272. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1877050921025114.

[49] H. Sethy, A. Patel, and V. Padmanabhan, "Real time strategy games: A reinforcement learning approach," *Procedia Computer Science*, vol. 54, pp. 257–264, 2015, Eleventh International Conference on Communication Networks, ICCN 2015, August 21-23, 2015, Bangalore, India Eleventh International Conference on Data Mining and Warehousing, ICDMW 2015, August 21-23, 2015, Bangalore, India Eleventh International Conference on Image and Signal Processing, ICISP 2015, August 21-23, 2015, Bangalore, India, ISSN: 1877-0509. DOI: https://doi.org/10.1016/j.procs.2015.06.030. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S187705091501354X.

[50] X. Chen, L. Yao, J. McAuley, G. Zhou, and X. Wang, "Deep reinforcement learning in recommender systems: A survey and new perspectives," *Knowledge-Based Systems*, vol. 264, p. 110 335, 2023, ISSN: 0950-7051. DOI: https://doi.org/10.1016/j.knosys.2023.110335. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0950705123000850.

[51] Y. Sato, *Model-free reinforcement learning for financial portfolios: A brief survey*, 2019. arXiv: 1904.04973 [q-fin.PM].

[52] M. Janner. "Model-based reinforcement learning: Theory and practice." (2019), [Online]. Available: https://bair.berkeley.edu/blog/2019/12/12/mbpo/ (visited on 08/13/2023).

[53] integrate.ai. "What is model-based reinforcement learning?" Medium. (2018), [Online]. Available: https://medium.com/the-official-integrate-ai-blog/understanding-reinforcement-learning-93d4e34e5698 (visited on 08/13/2023).

[54] J. B. Hamrick, A. L. Friesen, F. Behbahani, *et al.*, *On the role of planning in model-based deep reinforcement learning*, 2021. arXiv: 2011.04021 [cs.AI].

[55] N. Jiang. "Notes on tabular methods." (2020), [Online]. Available: https://nanjiang.cs.illinois.edu/files/cs598/note3.pdf (visited on 08/13/2023).

[56] N. Giraldo, *Sailboat navigation control system based on spiking neural networks*, 2023. [Online]. Available: https://bibliotecadigital.udea.edu.co/handle/10495/35164.

[57] D. Zhao, H. Wang, K. Shao, and Y. Zhu, "Deep reinforcement learning with experience replay based on sarsa," in *2016 IEEE Symposium Series on Computational Intelligence (SSCI)*, 2016, pp. 1–6. DOI: 10.1109/SSCI.2016.7849837.

[58] J. Zhang. "Reinforcement learning—td(λ) introduction." (2019), [Online]. Available: https://towardsdatascience.com/reinforcement-learning-td-%CE%BB-introduction-686a5e4f4e60.

[59] V. H. T. Duong. "Intro to reinforcement learning: Temporal difference learning, sarsa vs. q-learning." (2021), [Online]. Available: https://towardsdatascience.com/intro-to-reinforcement-learning-temporal-difference-learning-sarsa-vs-q-learning-8b4184bb4978?gi=a75cfd2a2719 (visited on 08/10/2023).

[60] Z. Saloum. "Summary of tabular methods in reinforcement learning." Published on Towards Data Science. (2018), [Online]. Available: https://towardsdatascience.com/summary-of-tabular-methods-in-reinforcement-learning-39d653e904af.

[61] A. K. Shakya, G. Pillai, and S. Chakrabarty, "Reinforcement learning algorithms: A brief survey," *Expert Systems with Applications*, vol. 231, p. 120 495, 2023, ISSN: 0957-4174. DOI: https://doi.org/10.1016/j.eswa.2023.120495. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417423009971.

[62] V. Mnih, K. Kavukcuoglu, D. Silver, *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015. DOI: 10.1038/nature14236. [Online]. Available: https://www.nature.com/articles/nature14236.

[63] X. Liang and X. Du, "A deep q learning network for traffic lights' cycle control in vehicular networks," *IEEE Transactions on Vehicular Technology*, p. 11, 2018.

[64] S. S. Mousavi, M. Schukat, and E. Howley, "Deep reinforcement learning: An overview," in *Proceedings of SAI Intelligent Systems Conference (IntelliSys) 2016*, Y. Bi, S. Kapoor, and R. Bhatia, Eds., Cham: Springer International Publishing, 2018, pp. 426–440, ISBN: 978-3-319-56991-8.

[65] Y. Jaafra, J. Luc Laurent, A. Deruyver, and M. Saber Naceur, "Reinforcement learning for neural architecture search: A review," *Image and Vision Computing*, vol. 89, pp. 57–66, 2019, ISSN: 0262-8856. DOI: `https://doi.org/10.1016/j.imavis.2019.06.005`. [Online]. Available: `https://www.sciencedirect.com/science/article/pii/S0262885619300885`.

[66] IBM. "¿qué son las redes neuronales recurrentes?" Consultado en el sitio web de IBM, sin fecha. (), [Online]. Available: `https://www.ibm.com/es-es/topics/recurrent-neural-networks`.

[67] P. Mahajan. "Fully connected vs convolutional neural networks." (2020), [Online]. Available: `https://medium.com/swlh/fully-connected-vs-convolutional-neural-networks-813ca7bc6ee5`.

[68] Dive into Deep Learning (D2L). "Stochastic gradient descent." (), [Online]. Available: `http://www.d2l.ai/chapter_optimization/sgd.html`.

[69] Dive into Deep Learning (D2L). "Adam." (), [Online]. Available: `http://www.d2l.ai/chapter_optimization/adam.html`.

[70] S. Lang, N. Lanzerath, T. Reggelin, M. Müller, and F. Behrendt, "Integration of deep reinforcement learning and discrete-event simulation for real-time scheduling of a flexible job shop production," Dec. 2020. DOI: `10.1109/WSC48552.2020.9383997`.

[71] D. Karunakaran. "Deep q network (dqn) - applying neural network as a functional approximation in q-learning." Retrieved from Medium. (2020), [Online]. Available: `https://medium.com/intro-to-artificial-intelligence/deep-q-network-dqn-applying-neural-network-as-a-functional-approximation-in-q-learning-6ffe3b0a9062`.

[72] J. Gao, Y. Shen, J. Liu, M. Ito, and N. Shiratori, *Adaptive Traffic Signal Control: Deep Reinforcement Learning Algorithm with Experience Replay and Target Network*, en, arXiv:1705.02755 [cs], May 2017. [Online]. Available: `http://arxiv.org/abs/1705.02755` (visited on 05/19/2023).

[73] M. G. Bellemare, Y. Naddaf, J. Veness, and M. Bowling, "The arcade learning environment: An evaluation platform for general agents," *Journal of Artificial Intelligence Research*, vol. 47, pp. 253–279, 2013.

[74] M. V. del Moral, “Algoritmo deep q-learning para el aprendizaje por refuerzo de una estrategia de conducción en 2d,” 2021.

[75] H. Chen, “A dqn-based recommender system for item-list recommendation,” in *2021 IEEE International Conference on Big Data (Big Data)*, 2021, pp. 5699–5702. DOI: 10.1109/BigData52589.2021.9671947.

[76] V. Mnih, A. P. Badia, M. Mirza, *et al.*, *Asynchronous methods for deep reinforcement learning*, 2016. arXiv: 1602.01783 [cs.LG].

[77] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, *Proximal policy optimization algorithms*, 2017. arXiv: 1707.06347 [cs.LG].

[78] J. Fan, *A review for deep reinforcement learning in atari:benchmarks, challenges, and solutions*, 2023. arXiv: 2112.04145 [cs.AI].

[79] R. Frank and M. Forster, “Demo: A recommendation based driver assistance system to mitigate vehicular traffic shock waves,” in *2014 IEEE Vehicular Networking Conference (VNC)*, 2014, pp. 125–126. DOI: 10.1109/VNC.2014.7013327.

[80] H. Wei, G. Zheng, H. Yao, and Z. Li, “IntelliLight: A Reinforcement Learning Approach for Intelligent Traffic Light Control,” in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, London United Kingdom: ACM, Jul. 2018, pp. 2496–2505, ISBN: 978-1-4503-5552-0. DOI: 10.1145/3219819.3220096. [Online]. Available: https://dl.acm.org/doi/10.1145/3219819.3220096 (visited on 09/01/2022).

[81] S. Aradi, “Survey of deep reinforcement learning for motion planning of autonomous vehicles,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 2, pp. 740–759, 2022. DOI: 10.1109/TITS.2020.3024655.

[82] R. Hussain and S. Zeadally, “Autonomous cars: Research results, issues, and future challenges,” *IEEE Communications Surveys & Tutorials*, vol. 21, no. 2, pp. 1275–1313, 2019. DOI: 10.1109/COMST.2018.2869360.

[83] "Traffic signal timing manual: Chapter 4." Office of Operations, Federal Highway Administration. (2021), [Online]. Available: https://ops.fhwa.dot.gov/publications/fhwahop08024/chapter4.htm#:~:text=A%20traffic%20phase%20is%20defined,all%20movements%20at%20an%20intersection.

[84] E. V. der Pol, *Deep reinforcement learning for coordination in traffic light control*, 2016.

[85] B. Kővári, L. Szőke, T. Bécsi, S. Aradi, and P. Gáspár, "Traffic Signal Control via Reinforcement Learning for Reducing Global Vehicle Emission," en, *Sustainability*, vol. 13, no. 20, p. 11 254, Oct. 2021, ISSN: 2071-1050. DOI: 10.3390/su132011254. [Online]. Available: https://www.mdpi.com/2071-1050/13/20/11254 (visited on 09/13/2022).

[86] M. R. T. Fuad, E. O. Fernandez, F. Mukhlish, *et al.*, "Adaptive Deep Q-Network Algorithm with Exponential Reward Mechanism for Traffic Control in Urban Intersection Networks," en, *Sustainability*, vol. 14, no. 21, p. 14 590, Nov. 2022, ISSN: 2071-1050. DOI: 10.3390/su142114590. [Online]. Available: https://www.mdpi.com/2071-1050/14/21/14590 (visited on 05/19/2023).

[87] J. A. Calvo and I. Dusparic, "Heterogeneous multi-agent deep reinforcement learning for traffic lights control," in *Irish Conference on Artificial Intelligence and Cognitive Science*, 2018. [Online]. Available: https://api.semanticscholar.org/CorpusID:57661298.

[88] S. Bird, S. Barocas, K. Crawford, F. Diaz, and H. Wallach, "Exploring or exploiting? social and ethical implications of autonomous experimentation in ai," in *Workshop on Fairness, Accountability, and Transparency in Machine Learning*, 2016. [Online]. Available: https://ssrn.com/abstract=2846909.

[89] Secretaría de Movilidad de Medellín. "Apoyo a la red semafórica." (), [Online]. Available: https://www.medellin.gov.co/SIMM/apoyo-a-la-red-semaf%C3%B3rica.

[90] "Sistema inteligente de movilidad de medellín (simm)." Retrieved 11 August 2023, Sistema Inteligente de Movilidad de Medellín (SIMM). (2023), [Online]. Available: https://www.medellin.gov.co/es/secretaria-de-movilidad/sistema-inteligente-de-movilidad-de-medellin/.

[91] F. L. Hall, *Traffic flow theory.* [Online]. Available: https://www.fhwa.dot.gov/publications/research/operations/tft/chap2.pdf.

[92] "Python programming language," Python Software Foundation. (2001), [Online]. Available: https://www.python.org/.

[93] "Google earth," Google LLC. (2005), [Online]. Available: https://www.google.com/earth/.

[94] K. K. Al-jabery, T. Obafemi-Ajayi, G. R. Olbricht, and D. C. Wunsch II, "2 - data preprocessing," in *Computational Learning Approaches to Data Analytics in Biomedical Applications*, K. K. Al-jabery, T. Obafemi-Ajayi, G. R. Olbricht, and D. C. Wunsch II, Eds., Academic Press, 2020, pp. 7–27, ISBN: 978-0-12-814482-4. DOI: https://doi.org/10.1016/B978-0-12-814482-4.00002-4. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9780128144824000024.

[95] "Jmp. correlation." (), [Online]. Available: https://www.jmp.com/en_ca/statistics-knowledge-portal/what-is-correlation.html.

[96] I. T. Jolliffe and J. Cadima, "Principal component analysis: A review and recent developments," *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 374, no. 2065, p. 20 150 202, 2016. DOI: 10.1098/rsta.2015.0202.

[97] Open Learning. "Interpreting data: Boxplots and tables." (2019), [Online]. Available: https://www.open.edu/openlearn/science-maths-technology/mathematics-statistics/interpreting-data-boxplots-and-tables/content-section-2.5.

[98] B. Bonaros. "Time series decomposition in python." (2022), [Online]. Available: https://towardsdatascience.com/time-series-decomposition-in-python-8acac385a5b2.

[99] "Statsmodels.tsa.seasonal.seasonal_decompose." (2023), [Online]. Available: https://www.statsmodels.org/stable/generated/statsmodels.tsa.seasonal.seasonal_decompose.html.

[100] R. J. Hyndman and G. Athanasopoulos. "Forecasting: Principles and practice." (), [Online]. Available: https://otexts.com/fpp2/classical-decomposition.html.

[101] World Population Review. "Countries with the most holidays 2023." (), [Online]. Available: https://worldpopulationreview.com/country-rankings/countries-with-the-most-holidays.

[102] "What is correlation?" (2023), [Online]. Available: https://www.simplilearn.com/tutorials/statistics-tutorial/what-is-correlation-in-statistics#:~:text=Correlation%20refers%20to%20the%20statistical,a%20variety%20of%20data%20sets..

[103] J. C. Ian and T. Jollife, "Principal component analysis: A review and recent developments," *Philosophical Transactions A: Mathematical, Physical and Engineering Sciences*, vol. 374, no. 2065, 2016. DOI: 10.1098/rsta.2015.0202.

[104] "Pca," Scikit-learn. (), [Online]. Available: https://scikit-learn.org/stable/modules/decomposition.html.

[105] "Matplotlib, boxplots." (), [Online]. Available: https://matplotlib.org/stable/gallery/statistics/.

[106] OpenStreetMap contributors, *Planet dump retrieved from https://planet.osm.org*, 2017. [Online]. Available: https://www.openstreetmap.org.

[107] A. Paul and S. Mitra, "Exploring reward efficacy in traffic management using deep reinforcement learning in intelligent transportation system," *ETRI Journal*, vol. 44, no. 2, pp. 194–207, 2022. DOI: https://doi.org/10.4218/etrij.2021-0404. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.4218/etrij.2021-0404. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.4218/etrij.2021-0404.

[108] P. Agand, A. Iskrov, and M. Chen, *Deep reinforcement learning-based intelligent traffic signal controls with optimized co2 emissions*, 2023. arXiv: 2310.13129 [eess.SY].

[109] S. Sharma. "Activation functions in neural networks," Towards Data Science. (2017), [Online]. Available: https://towardsdatascience.com/activation-functions-neural-networks-1cbd9f8d91d6.

[110] M. Weltevrede, M. T. J. Spaan, and W. Böhmer, *The role of diverse replay for generalisation in reinforcement learning*, 2023. arXiv: 2306.05727 [cs.LG].

[111] A. Rahimi-Kalahroudi, J. Rajendran, I. Momennejad, H. van Seijen, and S. Chandar, *Replay buffer with local forgetting for adapting to local environment changes in deep model-based reinforcement learning*, 2023. arXiv: 2303.08690 [cs.LG].

[112] B. Chen, T. Gao, and Q. Mi, "An approach to optimize replay buffer in value-based reinforcement learning," in *2023 18th Annual System of Systems Engineering Conference (SoSe)*, 2023, pp. 1–5. DOI: 10.1109/SoSE59841.2023.10178657.

[113] A. Wegener, M. Piórkowski, M. Raya, H. Hellbrück, S. Fischer, and J.-P. Hubaux, "Traci: An interface for coupling road traffic and network simulators," in *Proceedings of the 11th Communications and Networking Simulation Symposium*, ser. CNS '08, Ottawa, Canada: Association for Computing Machinery, 2008, pp. 155–163, ISBN: 1565553187. DOI: 10.1145/1400713.1400740. [Online]. Available: https://doi.org/10.1145/1400713.1400740.

[114] NIST. "Transmission control protocol (tcp)." (), [Online]. Available: https://csrc.nist.gov/glossary/term/transmission_control_protocol.

[115] G. S. Samarakoon and T. Sivakumar, "Microscopic simulation of parking violations in curbside with-flow bus priority lanes using sumo traffic control interface (traci)," in *Research for Transport and Logistics Industry Proceedings of the 7th International Conference, Colombo*, 2022.

[116] Eclipse. "Importing o/d matrices." (), [Online]. Available: https://sumo.dlr.de/docs/Demand/Importing_O/D_Matrices.html.

[117] INFRAS. "Handbook emission factors for road transport." (), [Online]. Available: https://www.hbefa.net/.

[118] "Euro 7: Meps back new rules to reduce road transport emissions," European Parliament. (2023), [Online]. Available: https://www.europarl.europa.eu/news/en/press-room/20231009IPR06746/euro-7-meps-back-new-rules-to-reduce-road-transport-emissions.

[119] Autos de Primera. "Euro vi, la norma ya se aplica en colombia." (2023), [Online]. Available: https://autosdeprimera.com/euro-vi-la-norma-ya-se-aplica-en-colombia/.

[120] "Emisiones vehiculares en colombia," Ministerio de Ambiente y Desarrollo Sostenible. (2017), [Online]. Available: `https://www.globalfueleconomy.org/media/418808/emisiones-vehiculares-en-colombia.pdf`.

[121] GeeksforGeeks. "Epsilon greedy algorithm in reinforcement learning." (2020), [Online]. Available: `https://www.geeksforgeeks.org/epsilon-greedy-algorithm-in-reinforcement-learning/`.

[122] Alcaldía de Medellín, MEDATA. "Velocidad y tiempo de viaje gt." (), [Online]. Available: `https://medata.gov.co/dataset/1-023-25-000287`.

[123] J. Brownlee. "Difference between a batch and an epoch in a neural network." (), [Online]. Available: `https://machinelearningmastery.com/difference-between-a-batch-and-an-epoch/`.

[124] "Compatibilidad con gpu," TensorFlow. (2021), [Online]. Available: `https://www.tensorflow.org/install/gpu?hl=es-419`.

[125] "Transfer learning." MATLAB & Simulink, MathWorks. (), [Online]. Available: `https://la.mathworks.com/discovery/transfer-learning.html#:~:text=Transfer%20learning%2C%20o%20transferencia%20del,que%20realiza%20otra%20tarea%20similar`.

[126] S. Schmitt, M. Hessel, and K. Simonyan, *Off-policy actor-critic with shared experience replay*, 2019. arXiv: `1909.11583 [cs.LG]`.

[127] E. Conti, V. Madhavan, F. P. Such, J. Lehman, K. O. Stanley, and J. Clune, "Improving exploration in evolution strategies for deep reinforcement learning via a population of novelty-seeking agents," *CoRR*, vol. abs/1712.06560, 2017. arXiv: `1712.06560`. [Online]. Available: `http://arxiv.org/abs/1712.06560`.

[128] C. Zhu, M. Dastani, and S. Wang, "A survey of multi-agent deep reinforcement learning with communication," *Autonomous Agents and Multi-Agent Systems*, vol. 38, no. 1, p. 4, Jan. 2024, ISSN: 1573-7454. DOI: `10.1007/s10458-023-09633-6`. [Online]. Available: `https://doi.org/10.1007/s10458-023-09633-6`.