Universidad Politécnica de Cartagena

ESCUELA INTERNACIONAL DE DOCTORADO

PROGRAMA DE DOCTORADO EN TECNOLOGÍAS DE LA INFORMACIÓN Y LAS COMUNICACIONES

TESIS DOCTORAL

CONTRIBUCIÓN AL AUMENTO DE LA CAPACIDAD COGNITIVA DE LOS SISTEMAS INTELIGENTES DE TRANSPORTE MEDIANTE INTELIGENCIA ARTIFICIAL

Presentada por D. ANTONIO GUILLÉN PÉREZ para optar al grado de Doctor
por la Universidad Politécnica de Cartagena

Dirigida por:
Dra. María Dolores Cano Baños

Cartagena, 2022

Universidad Politécnica de Cartagena

ESCUELA INTERNACIONAL DE DOCTORADO

DOCTORAL PROGRAMME IN INFORMATION AND COMMUNICATION TECHNOLOGIES

PhD THESIS

CONTRIBUTION TO ENHANCING THE COGNITIVE CAPABILITY OF INTELLIGENT TRANSPORTATION SYSTEMS USING ARTIFICIAL INTELLIGENCE

Presented by ANTONIO GUILLÉN PÉREZ
to the Technical University of Cartagena in fulfilment of
the thesis requirement for the award of PhD

Supervisor:
Dr. María Dolores Cano Baños

Cartagena, 2022

Declaration

I hereby declare that the work presented in this thesis has not been submitted for any other degree or professional qualification and that it is the result of my own independent work.

Antonio Guillén Pérez

May 2022

Resumen

Los Sistemas Inteligentes de Transporte (SIT) son un sector emergente dentro del área de investigación del transporte urbano. Se espera que la mayor capacidad cognitiva de los SIT permita crear ciudades y entornos futuros en los que las personas estén más seguras y tengan una mejor calidad de vida, además de ofrecer un uso más eficiente de los recursos, y una mayor seguridad.

La capacidad cognitiva de los SIT es la habilidad de estos sistemas para razonar, tomar decisiones, aprender y adaptarse, e interactuar con las personas, los vehículos y las infraestructuras en las que operan. Esta capacidad puede mejorarse mediante un uso adecuado de las tecnologías de la información y las comunicaciones, así como de algoritmos de Inteligencia Artificial (IA) centrados en la comprensión del mundo físico y social en el que actúan con el objetivo principal de optimizar los sistemas de transporte, reducir costes, mejorar la eficiencia y proporcionar una mayor calidad de vida a la sociedad.

Esta tesis se centra en un requisito clave para lograr un alto nivel de capacidad cognitiva: el desarrollo y despliegue de agentes cognitivos que puedan comunicarse, cooperar y comprender el comportamiento dinámico del conjunto urbano y así poder aprender, adaptarse y actuar en este entorno dinámico. De este modo, el objetivo que persigue este trabajo es proporcionar una clara comprensión de los retos del área de investigación de los SIT y, en particular, su relación con el campo de la IA, para proporcionar una mejor comprensión del estado actual de la técnica y su potencial para el futuro.

Para esto, en primer lugar, tras analizar las bondades que pueden ofrecer las redes ad-hoc voladoras formadas por vehículos aéreos no tripulados como drones, la tesis aborda la mejora de la capacidad cognitiva de los SIT centrada en las intersecciones de tráfico mediante técnicas de optimización basadas en IA. Es decir, cómo los SIT pueden contribuir a mejorar la comprensión de la dinámica física y social en intersecciones urbanas permitiendo optimizar la eficiencia de uso de estos sistemas. En segundo lugar, se discute la importancia de la interacción entre los SIT y los vehículos autónomos conectados (VACs), y cómo estos sistemas pueden beneficiarse mutuamente utilizando sistemas de comunicaciones móviles avanzados (5G - 6G) e inteligencia colectiva. Más concretamente, se explorará el uso de varios algoritmos de IA pertenecientes al campo del aprendizaje de refuerzo profundo multiagente (*Multi Agent Deep Reinforcement Learning*, MADRL). Todos estos avances se apoyan en el desarrollo de nuevos sistemas que aprovechan las bondades que ofrecen las redes ad-hoc voladoras, así como en técnicas de estimación de densidades de personas basadas en el estudio de los distintos canales de comunicación inalámbricos (Bluetooth, WiFi).

Los principales logros obtenidos durante el desarrollo de esta tesis son: *i)* la investigación sobre la interoperabilidad de los VACs y los SIT a través de sistemas de comunicación avanzados como el 5G y el 6G y los algoritmos MADRL para el desarrollo de varios algoritmos de control cooperativo descentralizado de vehículos autónomos. Estos algoritmos son capaces de encontrar una política de control cooperativa robusta que aprovecha las ventajas que ofrece la inteligencia colectiva y permite reducir el tiempo de espera en las intersecciones en más de un 90%, entre otras mejoras. Además, con la integración de estos sistemas dentro de las redes de comunicaciones 5G/6G, se ha desarrollado un marco de trabajo para facilitar el despliegue real de estos sistemas, así como un algoritmo de control que consideraba la latencia que pueden ofrecer estos sistemas y era capaz de adaptar la política de control a las fluctuaciones de rendimiento ofreciendo un control robusto y seguro; *ii)* el desarrollo de un algoritmo que permite acelerar el entrenamiento de los sistemas basados en MADRL gracias al entrenamiento por demostración que ofrece un agente llamado Oráculo, entrenado mediante aprendizaje por imitación, permitiendo reducir el tiempo de entrenamiento de los nuevos sistemas basados en MADRL hasta en ×6; y *iii)* la implementación de un sistema avanzado de control de intersecciones de tráfico mediante el uso de diferentes técnicas de optimización basadas en IA (como los algoritmos genéticos). Este sistema reduce el tiempo de espera de los vehículos en las intersecciones hasta un 80%, y un 20% de las emisiones de gases contaminantes, entre otras mejoras.

Los resultados de estos proyectos de investigación desarrollados durante la realización de esta tesis permitirán avanzar el camino de los desarrollos de SIT en el mundo real, permitiendo crear una movilidad urbana segura y sostenible con la integración de los VAC en su núcleo.

Palabras Clave:

Aprendizaje por refuerzo profundo multiagente; Gestión Cooperativa de Vehículos Autónomos; Inteligencia Artificial; Redes Ad-Hoc Voladoras; Sistemas de Comunicación Móvil; Sistemas Inteligentes de Transporte

Abstract

Intelligent Transportation Systems (ITS) are an emerging sector of the urban transportation research area. The increased cognitive capability of ITS is expected to create future cities and environments in which people are safer and have a better quality of life, as well as offering more efficient use of resources and improved safety.

The cognitive capability of ITS is the ability of these systems to reason, make decisions, learn and adapt, and interact with people, vehicles, and infrastructures in which they operate. This capability can be improved through better use of information and communication technologies, as well as Artificial Intelligence (AI) algorithms focused on understanding the physical and social world in which they interact with the main objective of optimizing transportation systems, reducing costs, improving efficiency, and providing a higher quality of life for society.

This thesis focuses on a key requirement for achieving a high level of cognitive capability, i.e., the development and deployment of cognitive agents that can communicate, cooperate, and understand the dynamic behavior of the urban traffic system and thus be able to learn, adapt and act in this dynamic environment. Thus, the objective pursued by this thesis is to provide a clear understanding of the challenges of the ITS research area and, in particular, its relation to the field of AI, to provide a better understanding of the current state of the art and its potential for the future.

To this end, firstly, after analyzing the benefits that ad-hoc aerial networks formed by unmanned aerial vehicles such as drones can offer, the thesis addresses the improvement of the cognitive capacity of ITS focused on traffic intersections by means of AI-based optimization techniques. That is, how ITS can contribute to improving the understanding of physical and social dynamics at urban intersections allowing to optimize the efficiency of use of these systems. Secondly, the importance of the interaction between ITS and Connected Autonomous Vehicles (CAVs) is discussed, and how these systems can benefit each other using advanced mobile communication systems (5G - 6G) and collective intelligence. More specifically, the use of various AI algorithms belonging to the field of Multi-Agent Deep Reinforcement Learning (MADRL) will be explored. All these advances are supported by the development of new systems that take advantage of the benefits offered by ad-hoc aerial networks, as well as people density estimation techniques based on the study of different wireless communication channels (Bluetooth, WiFi).

The main achievements obtained during the development of this thesis are: *i)* research on the interoperability of VACs and ITS through advanced communication systems such as 5G and 6G and MADRL algorithms for the

development of several decentralized cooperative control algorithms for autonomous vehicles. These algorithms are able to find a robust cooperative control policy that takes advantage of the benefits offered by collective intelligence and allows reducing waiting time at intersections by more than 90%, among other improvements. Furthermore, with the integration of these systems within 5G/6G communication networks, a framework has been developed to facilitate the actual deployment of these systems, as well as a control algorithm that considered the latency that these systems can offer and was able to adapt the control policy to performance fluctuations offering robust and secure control; *ii)* the development of an algorithm that allows to speed up the training of MADRL-based systems thanks to Learning-from-Demonstrations (LfD) offered by an agent called Oracle, trained by imitation learning, allowing to reduce the training time of new MADRL-based systems by up to ×6; and *iii)* the optimization of an advanced traffic intersection control system using different AI-based techniques (such as genetic algorithms). This system reduces vehicle waiting time at intersections by up to 80%, and 20% of pollutant gas emissions, among other improvements.

The results of these research projects developed during the realization of this thesis will advance the path of ITS developments in the real world, enabling the creation of safe and sustainable urban mobility with VAC integration at its core.

Keywords:

Artificial Intelligence; Autonomous Intersection Management; Communication Systems; Intelligent Transport Systems; Multi-Agent Deep Reinforcement Learning; Unmanned Aerial Vehicles; WiFi Crowd Counting; Flying Ad hoc Network; Reinforcement Learning

Scientific contributions derived from this research

Scientific contributions with impact factor:

Guillen-Perez, A.; Cano, M.-D., "Flying ad hoc networks: A new domain for network communications," *Sensors*, vol. 18, no. 10, p. 3571, Oct 2018, doi:10.3390/s18103571

2018 Journal Impact Factor (JIF): 3.031. (Q1), Rank: 15/61 in Instrument & Instrumentation.

Guillen-Perez, A.; Cano, M.-D., "Pedestrian Characterization in Urban Environments Combining WiFi and AI," *Int. J. Sens. Networks*, vol. 37, no. 1, p. 48, 2021, doi: 10.1504/IJSNET.2021.117964.

2020 Journal Impact Factor (JIF): 1.302. (Q4), Rank: 80/91 in Telecommunications.

Guillen-Perez, A.; Montoya, A-M; Sanchez-Aarnoutse, J. C.; Cano, M.-D., "A comparative performance evaluation of routing protocols for flying ad-hoc networks in real conditions," *Appl. Sci.*, vol. 11, no. 10, p. 4363, May 2021, doi: 10.3390/app11104363

2020 Journal Impact Factor (JIF): 2.679. (Q2), Rank: 38/90 in Engineering Multidisciplinary.

Guillen-Perez, A.; Cano, M.-D., "Intelligent IoT systems for traffic management: A practical application," *IET Intell. Transp. Syst.*, vol. 15, no. 2, pp. 273–285, Feb. 2021, doi: 10.1049/itr2.12021.

2020 Journal Impact Factor (JIF): 2.496. (Q2), Rank: 135/273 in Engineering, Electrical & Electronic.

Guillen-Perez, A.; Cano, M.-D., "AIM5LA: A Latency-Aware Deep Reinforcement Learning-Based Autonomous Intersection Management system for 5G Communication Networks," *Sensors*, vol. Accepted, pp. 1–20, 2022.

2020 Journal Impact Factor (JIF): 3.576. (Q2), Rank: 82/273 in Engineering, Electrical & Electronics.

Guillen-Perez, A.; Cano, M.-D., "Multi-Agent Deep Reinforcement Learning to Manage Connected Autonomous Vehicles at Tomorrow's Intersections," *IEEE Trans. Veh. Technol.*, vol. On review, pp. 1–12, 2022.

2020 Journal Impact Factor (JIF): 5.978. (Q1), Rank: 15/91 in Telecommunications; (Q1), Rank: 32/273 in Engineering, Electrical & Electronic.

Guillen-Perez, A.; Cano, M.-D., "Learning from Oracle Demonstrations – A new approach to develop Autonomous Intersection Management control algorithms based on Multi-Agent Deep Reinforcement Learning," *IEEE Access*, vol. On Review, pp. 1–12, 2022.

2020 Journal Impact Factor (JIF): 3.367. (Q2), Rank: 65/161 in Computer Science & Information Systems.

Guillen-Perez, A.; Cano, M.-D., "6G Communications Network Framework in the context of Edge-Decentralized Cooperative Autonomous Driving," *Appl. Sci.*, vol. On Review, pp. 1–12, 2022.

2020 Journal Impact Factor (JIF): 2.679. (Q2), Rank: 38/90 in Engineering Multidisciplinary.

Conferences:

Guillen-Perez, A.; Sanchez-Iborra, R.; Cano, M.-D., Sanchez-Aarnoutse, J. C.; and Garcia-Haro, J., "WiFi networks on drones," in *2016 ITU Kaleidoscope: ICTs for a Sustainable World (ITU WT)*, Nov. 2016, pp. 1–8, doi: 10.1109/ITU-WT.2016.7805730.

Guillen-Perez, A.; Cano, M.-D., "Comunicaciones Inalámbricas con Vehículos Aéreos no Tripulados", *I Jornadas Doctorales UPCT*, Universidad Politécnica de Cartagena. 2018. Oral communication.

Guillen-Perez, A.; Cano, M.-D., "A WiFi-based method to count and locate pedestrians in urban traffic scenarios," in *2018 14th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, Oct. 2018, vol. 2018-Octob, pp. 123–130, doi: 10.1109/WiMOB.2018.8589170.

Guillen-Perez, A.; Cano, M.-D., "Optimización de un Sistema Inteligente de Control y Gestión de Transporte en Intersecciones por medio de un Algoritmo Genético", *V Jornadas Doctorales UPCT*, Universidad Politécnica de Cartagena. 2019. Oral communication.

Guillen-Perez, A.; Cano, M.-D., "Counting and locating people in outdoor environments: a comparative experimental study using WiFi-based passive methods," *ITM Web Conf.*, vol. 24, pp. 1–10, Feb. 2019, doi: 10.1051/itmconf/20192401010.

Guillen-Perez, A.; Cano, M.-D., "Influencia del ciclo de trabajo de los semáforos en una intersección simple en múltiples parámetros ante una densidad de tráfico incremental," in *XIV Jornadas de Ingeniería Telemática (JITEL 2019)*, 2019, no. JITEL, pp. 22–24, [Online]. Available: http://jitel2019.i3a.es/.

Guillen-Perez, A.; Cano, M.-D., "Cómo la superresolución puede ayudar a los vehículos autónomos conectados", *VI Jornadas Doctorales UPCT*, Universidad de Murcia. 2020. Oral communication.

Guillen-Perez, A.; Cano, M.-D., "RAIM: Reinforced Autonomous Intersection Management - AIM based on MADRL," in *NeurIPS 2020 - Workshop Challenges of Real-World RL*, 2020, pp. 1–12, Accessed: Feb. 16, 2022. [Online]. Available: https://www.researchgate.net/publication/357957238_RAIM_Reinforced_Autonomous_Intersection_Management_-_AIM_based_on_MADRL.

Acknowledgments

I would like to thank everyone who has supported me in my doctoral studies, especially to Fundación Séneca, Professor Juan Sebastian, Professor Juan Carlos Sanchez-Aarnoutse, my thesis director Professor Maria Dolores Cano, my entire family, and my life partner Mari Carmen.

Without all of them, this work would not be possible and I could not have become the person I am today.

I want to thank you for your generosity, a generosity that I will never be able to repay.

Table of contents

List of figures

List of tables

List of abbreviations

3D-GM	3D-Gauss-Markov
ADAS	Advanced Driver-Assistance Systems
adv.RAIM	advanced.RAIM
AeroRP	Aeronautical Routing Protocol
AI	Artificial Intelligence
AODV	Ad Hoc On-Demand Distance Vector
AODVSEC	AODV Security
AP	Access Point
APAR	Ant Colony Optimization-Based Polymorphism Aware Routing Algorithm
AR	Augmented Reality
ARPAM	Ad Hoc Routing Protocol for Aeronautical Mobile Ad Hoc Networks
AUAV	Autonomous Unmanned Aerial Vehicles
AV	Autonomous Vehicles
BATMAN	Better Approach to Mobile Ad Hoc Network
BATMAN-ADV	BATMAN-Advanced
BC	Behavioral Cloning
BSA	Boundless Simulation Area
C-ADAS	Cooperative Driver-Assistance Systems
CAUN	Clustering Algorithm of UAV Networking
CAV	Connected Autonomous Vehicles
CBLADSR	Cluster-Based Location-Aided DSR
CE-OLSR	Cartography-Enhanced OLSR
CFR	Channel Frequency Response
CIR	Channel Impulse Response
CLMN	Column
COLSR	Contention-Based OLSR
CQI	Channel Quality Indicator
CRUV	Connectivity-Based Traffic-Density Aware Routing Using UAVs for VANETs
CSI	Channel State Indicator

C-V2X	Cellular Vehicle-to-Everything
CVC	Central Vehicle Cloud
DAGGer	Data AGGregation
DAU	Data Acquisition Unit
DCR	Data-Centric Routing
DDPG	Deep Deterministic Policy Gradient
DDPGfD	DDPG from Demonstrations
DFT	Discrete Fourier Transformation
DL	Deep Learning
DOLSR	Directional OLSR
DP	Dynamic Programming
DPL	Direct Policy Learning
DQfD	Deep Q-Learning from Demonstrations
DRL	Deep Reinforcement Learning
DSDV	Destination-Sequenced Distance Vector
DSR	Dynamic Source Routing
DSRC	Direct Short-Range Communication
DTM	Disruption-Tolerant Mechanism
DTN	Delay-Tolerant Network
EA	Evolutionary Algorithm
ECR	Exponential Correlated Random
EGM	Enhanced Gauss-Markov
EI	Edge Intelligence
eMBB	enhanced Mobile Broad Band
ETA	Estimated Time of Arrival
FANET	Flying Ad Hoc Network
FCFS	First Come First Served
FFS	Fast First Service
FSR	Fisheye-State Routing
FW-UAV	Fixed Wing - UAV
GA	Genetic Algorithm
GCS-R	Ground Control System-Routing
GDSTR-3D	Greedy Distributed Spanning Tree Routing 3D

GGF	Greedy Geographic Forwarding
GHG	Greedy-Hull-Greedy
GLSR	Geographic Load-Share Routing
GM	Gauss-Markov
GNB	Gaussian Naïve Bayes
GPMOR	Geographic Position Mobility-Oriented Routing
GPSR	Geographic Greedy Perimeter Stateless Routing
GRAA	Geographic Routing Protocol for Aircraft Ad Hoc Network
GRG	Greedy-Random-Greedy
H3MP	Hybrid Markov Mobility Model with Pheromones
HRP	Hybrid-Routing Protocol
HWMP	Hybrid Wireless Mesh Protocol
iEN	Intelligent Edge Nodes
IL	Imitation Learning
IM	Intersection Manager
iNB	intelligent-NodeB
IoT	Internet of Thing
iPCW	intelligent Pedestrian Characterization using WiFi
ISTN	Integrated Terrestrial-Space Network
ITS	Intelligent Transport System
*k*NN	*k*-Nearest Neighbor
LANMAR	Landmark-Routing Protocol
LAROD	Location-Aware Routing for Opportunistic Delay Tolerant
LCAD	Load Carry and Deliver Routing
LfD	Learning from Demonstrations
LfOD	Learning from Oracle Demonstrations
LOS	Line Of Sight
LQF	Long Queue First
LR	Logistic Regressor
LSTM	Long Short-Term Memory
MADRL	Multi-Agent Deep Reinforcement Learning
MANET	Mobile Ad Hoc Network.
MAODV	Multicast AODV

MILP	Mixed Integer Linear Programming
ML	Machine Learning
MLHR	Multi-Level Hierarchical Routing
ML-OLSR	Mobility and Load-Aware OLSR
MMT	Multi-Meshed Tree Protocol
mMTC	Machine Type Communications
M-OLSR	Modified-OLSR
MPCA	Mobility Prediction Clustering Algorithm
MPGR	Mobility Prediction-Based Geographic Routing.
MUDOR	Multipath Doppler Routing
mURLLC	Massive-URLLC
NC	Nomadic Community
NN	Neural Network
OLSR	Optimized Link-State Routing
PASER	Position-Aware Secure and Efficient Routing Approach
P-OLSR	Predictive-OLSR
PPRZM	Paparazzi Model
PRS	Purse
PSMM	Particle Swarm Mobility Model
PSO	Particle Swarm Optimization
QB	Query-Based
QoE	Quality of Experience
QoS	Quality of Service
RAIM	Reinforced Autonomous Intersection Management
RAIMfOD	RAIM from Oracle Demonstrations
RD	Random Direction
RF	Random Forest
RGR	Reactive-Greedy-Reactive
RL	Reinforcement Learning
RPGM	Reference Point Group Mobility
RSI	Reconfigurable Intelligent Surface
RSS	Received Signal Strength
RSSI	Received Signal Strength Indicator

RTD	Round Trip Delay
RTORA	Rapid-Reestablish TORA
RWP	Random Waypoint
RW-UAV	Rotary Wing – UAV
SDPC	Self-Deployable Point Coverage
SEARN	Search-based Structured Prediction
SHARP	Sharp Hybrid Adaptive Routing Protocol
SL	Supervised Learning
SMILe	Stochastic Mixing Iterative Learning
SNR	Signal-to-Noise Ratio
SRCM	Semi-Random Circular Movement
ST	Smooth Turn
SUAP	Secure UAV Ad Hoc Routing Protocol
SVC	Support Vector Classification
SVM	Support Vector Machine
TBRPF	Topology Broadcast Based on Reverse-Path Forwarding
TCP	Transport Control Protocol
TD3	Twin Delayed Deep Deterministic Policy Gradient
TD3fOD	TD3 from Oracle Demonstrations
TORA	Temporarily Ordered Routing Algorithm
TraCI	Traffic Control Interface
TS-AODV	Time-Slotted AODV
U2G	UAV-to-Ground
U2U	UAV-to-UAV
UAV	Unmanned Aerial Vehicle
URLLC	Ultra-Reliable Low-Latency Communications
USMP	UAV Search Mission Protocol
UVAR	UAV-Assisted VANET Routing Protocol
UWB	Ultra-Wide Band
UX	User eXperience
V2I	Vehicle-to-Infrastructure
V2N	Vehicle-to-Network
V2P	Vehicle-to-Pedestrian

V2V	Vehicle-to-Vehicle
V2X	Vehicle-to-Everything
VANET	Vehicular Ad Hoc Network
VEC	Vehicle Edge Computing
VEN	Vehicle Edge Node
VLC	Visible Light Communication
VLEC	Vehicle Local Edge Cloud
XLinGo	Cross-Layer Link Quality and Geographical-Aware Beaconless
ZRP	Zone-Routing Protocol

Chapter 1: Introduction

1.1 Thesis motivation

Urban mobility is progressively becoming one of the main concerns of today's society, with traffic jams and accidents in urban areas as well as the pollution they generate increasingly common, having a major impact both socially and economically [1]. The United Nations estimates that more than two-thirds of the world's population will live in crowded cities by 2050 [2]. It is also evident that the rate of global urbanization is growing at an increasing rate, with 55% of the world's population residing in urban areas in 2018, increasing urbanization is expected to have a significant impact on the world's population, economy, and environment.

Intelligent Transportation Systems (ITS) [3] have the potential to play a key role in helping to solve both the mobility problems already facing urban areas as well as future problems ahead.

The goal of ITS is to create efficient and intelligent solutions that improve the transportation of people, goods, and services through the use of advanced technologies such as wireless communication systems, unmanned aerial vehicles, or artificial intelligence algorithms. Therefore, the use of ITS is expected to increase the speed, efficiency, and comfort of urban transportation, while reducing accidents caused by human error at the same time improving the quality of life of the urban population [3], [4]. Examples of ITS applications can be found in advanced traffic light control systems, applications for intelligent parking search, route optimization of public transport services, improving the efficiency of traffic flow through flow prediction algorithms that allow adapting urban routes, etc.

The current state of the art shows a large gap between the theoretical potential and the reality of ITS. This is mainly because these ITS need breakthrough technology for proper and efficient operation, such as advanced artificial intelligence algorithms, ultra-reliable wireless communications, and sensors capable of making accurate and real-time data available to ITS applications on a wide spectrum of data sources such as traffic and environment, among others.

Driven by recent advances in the field of Artificial Intelligence (AI), wireless communications systems such as 5G, low-cost embedded devices, and the development of technologies such as Unmanned Aerial Vehicles (UAVs), autonomous vehicles, and Big Data, most of the stringent requirements that ITS must meet are now met. Thanks to these advances, new opportunities are opening up for the development of a plethora of innovative applications within the ITS field, such as autonomous intersection management and the deployment of Multi-Agent Deep Reinforcement Learning (DRL) algorithms for urban traffic planning and control.

The main purpose of this thesis is to develop advanced ITS with high cognitive capabilities by using cutting-edge techniques and technologies such as 5G, DRL, AI algorithms, and UAVs. Thus, the aim is to obtain ITS capable of adapting to a changing and unpredictable environment, learning and adapting in real-time, acting autonomously and proactively, and being able to communicate with other agents and humans. ITS will optimize the capabilities of the cities of the future, making them a more efficient, comfortable, and safe environment, as well as enabling a better quality of life for citizens through reduced congestion, better use of public resources, and less pollution.

1.2 Thesis objectives

The main objective of the thesis is to contribute to the improvement of ITS cognitive capabilities using cutting-edge technologies such as 5G/6G communication networks, UAVs, and connected autonomous vehicles, as well as through the use of breakthrough AI/DRL algorithms to achieve more efficient and simpler solutions that can be implemented using standards that allow for simple and scalable development.

To this end, the thesis is divided into three main sub-objectives:

1. Review of research in the field of AI, mobile communication and wireless technologies, UAVs, Internet of Things (IoT), and IoT sensors in the context of urban mobility and ITS applications.
2. Provide an overview of the state-of-the-art research in the field of traffic light control, intelligent intersection control, and traffic management, as well as the current challenges and limitations of the proposed systems, focusing on the main challenges of real-time performance, robustness, scalability, and communication capabilities.
3. Explore the use of AI, particularly Multi-Agent DRL algorithms, for traffic planning and control in urban areas such as intersections, taking into account different aspects to be considered, such as integration in mobile communication networks like 5G or 6G, interaction with UAVs, multi-agent communication, and the use of connected autonomous vehicles.

To achieve the proposed objectives, the following tasks were proposed:

1. Review research in the field of AI, including recent advances in the area of Deep Learning (DL), as well as DRL and advanced techniques used to improve the training of these algorithms.
2. Exploration of the use of the latest technologies such as 5G/6G, wireless networking technologies, IoT, and UAVs for integration in the context of urban mobility and ITS applications.

3. Discussion of the state of the art of research in the area of intelligent intersection control, traffic signal control, and traffic management, focusing on the advantages and limitations of current systems and their main challenges.
4. Investigation of the use of Multi-Agent DRL algorithms to perform real-time traffic orchestration in urban areas such as intersections, with a focus on the main features to be considered, such as integration into 5G or 6G mobile communication networks, ultra-low latency communication, distributed intelligence, and the use of connected autonomous vehicles.

1.3 Ph. D. dissertation structure

This dissertation is organized as follows:

Chapter 2: Theoretical background:

Explores the theoretical background of the concepts and technologies used during the development of this dissertation, as well as the basic background of the main aspects discussed, such as ITS, AI, UAVs, and wireless communications technologies. The concepts and technologies presented in this chapter are detailed in-depth in each of the chapters of the thesis where they have been employed.

Chapter 3: Unmanned Aerial Vehicles:

Focuses on the field of UAVs, exploring their applications, advantages, and how they could be used to complement current technologies used in the development of new advanced ITS.

Chapter 4: Smart Cities and Pedestrians:

Addresses ongoing research related to the integration of IoT sensor networks and WiFi devices in urban areas, including AI-based pedestrian detection and identification for traffic optimization in urban areas.

Chapter 5: Smart Traffic Light Control – AI approach:

Discusses the basic approaches to the operation of the main intelligent algorithms for traffic control at intersections using traffic lights, as well as the use of AI algorithms to improve their performance.

Chapter 6: Interoperability of Connected Autonomous Vehicles and Intelligent Transportation Systems:

Analyzes the advantages and disadvantages offered by connected autonomous vehicles and advanced communication networks such as 5G and 6G in the development of new advanced intersection control systems that eliminate the need for traditional traffic lights. Thus, by using cutting-edge multi-agent DRL algorithms, this chapter proposes several autonomous vehicle control algorithms

for intersections called Autonomous Intersection Management (AIM) that enable advanced centralized coordination of vehicles at intersections by leveraging collective intelligence offered by multi-agent systems. Thanks to this, the proposed AIMs allow minimizing the time lost at intersections by vehicles, as well as eliminating accidents at intersections due to human errors due to centralized coordination. Furthermore, this chapter studies a fully operational AIM over 5G considering the physical latency characteristics presented by this network architecture, maximizing the security offered, as well as analyzes the integration of these AIMs in the architectures of the future 6G communications network.

Chapter 7: General Conclusions and Future Research:

Summarizes the main conclusions and contributions obtained in the development of this dissertation and the possible lines of future research.

1.4 Thesis contributions

The main contributions of this dissertation are presented below:

Chapter 2: Theoretical background:

- Review the state of the art of the topics covered in this dissertation: ITS, AI, UAVs, and wireless communications technologies.

Chapter 3: Unmanned Aerial Vehicles:

- In-depth review of the state of the art of flying ad hoc networks (FANETs) composed of UAVs, focusing on mobility models, positioning protocols, propagation models, and routing protocols.
- Detailed study of the impact the UAV had on the radiation pattern of the integrated WiFi communication module using a controlled environment such as an anechoic chamber.
- Comparative study of the performance provided by different FANET routing protocols, in terms of throughput and packet loss, in a real deployment consisting of several UAV nodes using WiFi on 2.4 GHz and 5 GHz bands.

The articles related to this chapter that have been published in technical journals or international conferences are:

Guillen-Perez, A.; Cano, M.-D., "Flying ad hoc networks: A new domain for network communications," *Sensors,* vol. 18, no. 10, p. 3571, Oct 2018, doi:10.3390/s18103571

2018 Journal Impact Factor (JIF): 3.031. (Q1), Rank: 15/61 in Instrument & Instrumentation.

Guillen-Perez, A.; Montoya, A-M; Sanchez-Aarnoutse, J. C.; Cano, M.-D., "A comparative performance evaluation of routing protocols for flying ad-hoc networks in real conditions," *Appl. Sci.*, vol. 11, no. 10, p. 4363, May 2021, doi: 10.3390/app11104363

2020 Journal Impact Factor (JIF): 2.679. (Q2), Rank: 38/90 in Engineering Multidisciplinary.

Guillen-Perez, A.; Sanchez-Iborra, R.; Cano, M.-D., Sanchez-Aarnoutse, J. C.; and Garcia-Haro, J., "WiFi networks on drones," in *2016 ITU Kaleidoscope: ICTs for a Sustainable World (ITU WT)*, Nov. 2016, pp. 1–8, doi: 10.1109/ITU-WT.2016.7805730.

Guillen-Perez, A.; Cano, M.-D., "Comunicaciones Inalámbricas con Vehículos Aéreos no Tripulados", *I Jornadas Doctorales UPCT*, Universidad Politécnica de Cartagena. 2018. Oral communication.

Chapter 4: Smart Cities and Pedestrians:

- Development of a novel passive WiFi-based method to estimate the number of pedestrians at an intersection to improve the performance of traffic light ITS.
- Real scenario deployment, to show the benefits that the previously proposed algorithm could offer, as well as the possible improvements that could be made.
- By using AI algorithms, an algorithm based on the previous one was obtained, which allowed obtaining a superior performance in terms of accuracy and recall. Thus, by analyzing the WiFi messages sent passively by devices carried by pedestrians, the system was able to estimate the number of pedestrians at an intersection with high accuracy.

The articles related to this chapter that have been published in technical journals or international conferences are:

Guillen-Perez, A.; Cano, M.-D., "Pedestrian Characterization in Urban Environments Combining WiFi and AI," *Int. J. Sens. Networks*, vol. 37, no. 1, p. 48, 2021, doi: 10.1504/IJSNET.2021.117964.

2020 Journal Impact Factor (JIF): 1.302. (Q4), Rank: 80/91 in Telecommunications.

Guillen-Perez, A.; Cano, M.-D., "A WiFi-based method to count and locate pedestrians in urban traffic scenarios," in *2018 14th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, Oct. 2018, vol. 2018-Octob, pp. 123–130, doi: 10.1109/WiMOB.2018.8589170.

Guillen-Perez, A.; Cano, M.-D., "Counting and locating people in outdoor environments: a comparative experimental study using WiFi-based passive methods," *ITM Web Conf.*, vol. 24, pp. 1–10, Feb. 2019, doi: 10.1051/itmconf/20192401010.

Chapter 5: Smart Traffic Light Control – AI approach:

- Detailed study of the principle of operation of the main adaptive traffic light control systems.
- Development of an advanced adaptive traffic light control algorithm based on queuing theory. This system was optimized using genetic algorithms.

The articles related to this chapter that have been published in technical journals or international conferences are:

Guillen-Perez, A.; Cano, M.-D., "Intelligent IoT systems for traffic management: A practical application," *IET Intell. Transp. Syst.*, vol. 15, no. 2, pp. 273–285, Feb. 2021, doi: 10.1049/itr2.12021.

2020 Journal Impact Factor (JIF): 2.496. (Q2), Rank: 135/273 in Engineering, Electrical & Electronic.

Guillen-Perez, A.; Cano, M.-D., "Optimización de un Sistema Inteligente de Control y Gestión de Transporte en Intersecciones por medio de un Algoritmo Genético", *V Jornadas Doctorales UPCT*, Universidad Politécnica de Cartagena. 2019. Oral communication.

Guillen-Perez, A.; Cano, M.-D., "Influencia del ciclo de trabajo de los semáforos en una intersección simple en múltiples parámetros ante una densidad de tráfico incremental," in *XIV Jornadas de Ingeniería Telemática (JITEL 2019)*, 2019, no. JITEL, pp. 22–24, [Online]. Available: http://jitel2019.i3a.es/.

Chapter 6: Interoperability of Connected Autonomous Vehicles and Intelligent Transportation Systems:

- Novel development of an Autonomous Intersection Management (AIM) system using Multi-Agent Deep Reinforcement Learning (MADRL). The proposed AIM was named RAIM.
- RAIM enhancement through recurrent neural networks, as well as through other training acceleration methods such as curriculum learning for RL, Prioritized Experience Replay (PER). Thus, the proposed system called *adv*.RAIM, enabled much smarter vehicular control at intersections, reducing waiting time to a minimum.
- Study of the current state-of-the-art in the field of imitation learning, learning from observation and learning from demonstration, proposing a new Learning from Demonstration (LfD) algorithm for environments where there is no (or there is a hidden expert agent) from which to extract new demonstrations. The proposed system is able to train an agent by imitation that mimics the behavior of the hidden expert agent. This trained agent is called an Oracle. This Oracle is the one used by the proposed algorithm to train an agent by demonstration, speeding up vastly the training of new agents by MADRL.
- Development of a latency-aware AIM for 5G communication network. Thus, this proposed AIM was able to guarantee maximum security due to a latency forecaster module based on Transformers and to incorporate the temporal behavior of latency in the *adv*.RAIM control module based on MADRL.
- Proposal of the necessary modules so that the new AIM systems can be natively integrated into the future 6G communications network. In this way, vehicle control via AIM and 6G will reduce development costs and improve the performance of both.

The articles related to this chapter that have been published in technical journals (or currently under review) or international conferences are:

Guillen-Perez, A.; Cano, M.-D., "AIM5LA: A Latency-Aware Deep Reinforcement Learning-Based Autonomous Intersection Management system for 5G Communication Networks," *Sensors*, vol. Accepted, pp. 1–20, 2022.

2020 Journal Impact Factor (JIF): 3.576. (Q2), Rank: 82/273 in Engineering, Electrical & Electronics.

Guillen-Perez, A.; Cano, M.-D., "Multi-Agent Deep Reinforcement Learning to Manage Connected Autonomous Vehicles at Tomorrow's Intersections," *IEEE Trans. Veh. Technol.*, vol. On review, pp. 1–12, 2022.

2020 Journal Impact Factor (JIF): 5.978. (Q1), Rank: 15/91 in Telecommunications; (Q1), Rank: 32/273 in Engineering, Electrical & Electronic.

Guillen-Perez, A.; Cano, M.-D., "Learning from Oracle Demonstrations – A new approach to develop Autonomous Intersection Management control algorithms based on Multi-Agent Deep Reinforcement Learning," *IEEE Access*, vol. On Review, pp. 1–12, 2022.

2020 Journal Impact Factor (JIF): 3.367. (Q2), Rank: 65/161 in Computer Science & Information Systems.

Guillen-Perez, A.; Cano, M.-D., "6G Communications Network Framework in the context of Edge-Decentralized Cooperative Autonomous Driving," *Appl. Sci.*, vol. On Review, pp. 1–12, 2022.

2020 Journal Impact Factor (JIF): 2.679. (Q2), Rank: 38/90 in Engineering Multidisciplinary.

Guillen-Perez, A.; Cano, M.-D., "Cómo la superresolución puede ayudar a los vehículos autónomos conectados", *VI Jornadas Doctorales UPCT*, Universidad de Murcia. 2020. Oral communication.

Guillen-Perez, A.; Cano, M.-D., "RAIM: Reinforced Autonomous Intersection Management - AIM based on MADRL," in *NeurIPS 2020 - Workshop Challenges of Real-World RL*, 2020, pp. 1–12, Accessed: Feb. 16, 2022. [Online]. Available: https://www.researchgate.net/publication/357957238_RAIM_Reinforced_Autonomous_Intersection_Management_-_AIM_based_on_MADRL.

Chapter 2: Conceptual background

2.1 Introduction

This chapter reviews the main concepts and technologies that have been used during the development of this thesis, such as Intelligent Transportation Systems (ITS), Artificial Intelligence (AI), Unmanned Aerial Vehicles (UAV), and wireless communication systems.

2.2 Intelligent Transportation Systems

ITS are advanced applications capable of recognizing, analyzing, and responding intelligently to dynamic, uncertain, and ambiguous conditions in order to ensure the efficient, safe and comfortable operation of all actors involved in transportation networks: cars, pedestrians, public transport, cyclists, etc. Its capabilities may include: monitoring and evaluation of road conditions, monitoring of road user behavior, accident detection and early warning to drivers, definition of road geometry and traffic rules based on traffic patterns, traffic light control, etc.

The main benefits of ITS are: increased efficiency of the road network, through early detection of traffic jams, accidents, road conditions, or changes in road geometry; higher utilization of road space and early detection of roads in poor condition and therefore higher quality perceived by users and less pollution due to lower fuel consumption, as well as reduced noise and air pollution, as fewer vehicles drive on roads in good condition; improved road user behavior by providing drivers with safety messages and road conditions in real-time.

Therefore, ITS is a transport-focused technology that tries to solve some of the problems that traditional methods, such as traffic signals or traffic lights, cannot. In this sense, ITS seeks to improve the efficiency and reliability of transportation by making it smarter.

To achieve this, ITS typically have three main components, all of which contribute to achieving ITS functionality: (1) hardware; (2) software; and (3) data.

1. Hardware refers to the physical aspects of the system, such as sensors, actuators, embedded devices, and other elements, such as intelligent traffic signals or wireless communication devices (WiFi, cellular, Bluetooth, etc.).

2. Software refers to the logic, algorithms, and other instructions used to operate the system intelligently and proactively.

3. Data refers to the information gathered from the environment that is processed and used to make decisions or adapt the system.

As a result, the main challenges faced by the ITS field are focused on:

1. Integrating information from the environment, road users, and traffic to make the right decisions. Integrating information from multiple sensors, of different nature and different update periods is by no means trivial.

2. Adapt the systems to the environment to avoid accidents, act proactively, as well as optimize travel times, improve traffic flow, etc.

3. Ensure that these systems are able to integrate, communicate and act autonomously, and be able to adapt to the environment in real-time, and be proactive with users to optimize the use of resources.

4. Develop ITS that are robust, scalable, and capable of making autonomous decisions, and provide intelligent and efficient solutions using different technologies in a scalable manner, enabling easy and cost-effective deployment.

5. Develop ITS capable of working in real-time with information from all sensors, to provide top-quality services to road users.

6. Ensure that ITS also operates in highly dynamic environments with constantly changing conditions, e.g., urban, suburban, rural, or industrial areas.

The main communications technologies used in ITS are:

WiFi

WiFi is a Wireless Local Area Network (WLAN) protocol developed by the IEEE working group and is based on the IEEE 802.11 standard. WiFi uses a wide range of radio frequencies to communicate and can reach speeds of up to hundreds of megabits per second (Mbps). It operates in the 2.4 GHz and 5 GHz frequency bands and uses OFDM (Orthogonal Frequency Division Multiplexing) modulation.

The protocol defines a shared medium and a Distributed Coordination Function (DCF) for multiple stations to share a common radio resource, with Carrier Sense Multiple Access and Collision Avoidance (CSMA/CA). This feature allows multiple wireless devices to communicate on a shared wireless medium in random order and at random time intervals. In addition, WiFi allows two modes of operation: Access Point and Ad hoc. The former mode is used for point-to-point communications, where one access point acts as a central control for all connected stations, while the latter can be used for ad hoc communications, where two or more stations communicate directly with each other without the intervention of a separate access point.

The main advantages of WiFi are: its low cost, its easy deployment, and its ability to support high data throughput, thanks to its radio technology. The main challenge of this technology is that WiFi signals are sensitive to obstacles, so coverage areas must be strategically planned to ensure service coverage. In

addition, there is a high probability of interference when there is a high density of nodes, as well as some security flaws.

Bluetooth

Bluetooth is a wireless communication protocol defined by the IEEE 802.15.1 standard, operating in the 2.4 GHz radio frequency band. Bluetooth uses a low-power Frequency Hopping Spread Spectrum (FHSS) modulation with a CSMA/CA protocol to minimize interference between devices.

Its main features are that it can be used as a Personal Area Network (PAN) connecting multiple devices to a single device, but it can also be used in peer-to-peer ad hoc networks (piconets) that can be formed between a group of Bluetooth-enabled devices.

Its low power consumption, communication range of up to tens of meters, and low bit error rate are key features of Bluetooth technology.

The main use of Bluetooth in ITS is to enable low-power, long-distance communication between ITS and its users. It can be used in ITS to connect user devices, such as a cell phone or tablet, to an ITS, providing real-time traffic and user status information in a bidirectional and interactive manner. Bluetooth also provides an ideal means of interconnecting devices, for example, cars and traffic lights, and enables the communication between two different ITS.

Bluetooth Low Energy (BLE)

BLE is a wireless personal area network technology that was standardized by the Bluetooth Special Interest Group (SIG) to provide more efficient communication in Bluetooth systems.

BLE is designed for battery-powered devices with low power consumption. BLE is based on the classic Bluetooth protocol, but there are fundamental differences between BLE and the classic specification.

The main changes are the reduction in packet size, which reduces traffic and power consumption, and the introduction of the connection interval, which allows for longer periods of inactivity. The result is that BLE devices can operate for years without the need to change the battery.

The main use of BLE in ITS is to provide low power consumption and low deployment cost communication for sensors used in ITS, providing real-time information on numerous environmental parameters, such as the number of vehicles, vehicle direction, road conditions, pollution, noise, etc.

ZigBee

ZigBee is a wireless technology based on the IEEE 802.15.4 wireless standard focused on creating Wireless Personal Area Networks (WPANs) with small, low-power, low-cost devices. ZigBee operates in the 2.4 GHz and sub-GHz bands (784 MHz in China, 868 MHz in Europe, and 915 MHz in the United States and Australia). Data rates range from 20 kbps (868 MHz band) to 250 kbps (2.4 GHz band).

The main features of ZigBee are its low data rate and wide communication range, reaching up to 1KM in the sub-GHz band and about 10-100 meters in the 2.4GHz band.

ZigBee is therefore ideal for applications such as remote sensor networks, where many devices must communicate with each other in a low-power environment. However, due to the low data rates that ZigBee can achieve, its use may be limited for specific applications in ITS.

Low Range (LoRa)

LoRa is a proprietary modulation technique used for long-distance wireless communication with low power consumption and low bandwidth, creating Low Power Wide Area (LPWA) networks. It operates in the sub-GHz frequency band (433MHz to 868 MHz) and uses a spread spectrum technique, which means that the signal is spread over a wide frequency spectrum. It can achieve transmission speeds of between 0.3kbps and 27kbps, depending on the spreading factor used, and a range of up to 10km in urban areas and 15km in rural areas.

Since LoRa only defines the physical layer of wireless communication, LoRaWAN implements the rest of the network capabilities. LoRaWAN defines two types of nodes in the network architecture: end nodes and gateways. Data from the end nodes must be sent to the gateway, acting as a centralized network.

As such, it is a suitable technology for a diverse range of sensors and ITS applications. However, LoRaWAN also has serious limitations, such as the need to deploy gateways to enable the connection between the end nodes and the ITS, securing the connection between each end node and the gateway, and its low transmission speed.

Cellular Communication Networks (4G, 5G & 6G)

4G is the fourth generation of mobile wireless communications, developed by the 3GPP (Third Generation Partnership Project) working group. LTE (Long Term Evolution) is one of the key components of 4G developed to provide users with enhanced services. It offers high data throughput, low latency, and high peak speeds compared to previous generation 3G.

It is a mobile wireless technology that has a wide coverage range (hundreds of kilometers) and fast data rates, with an average speed of about 100Mbps. It is designed for high-capacity, high-speed multimedia applications, such as video streaming, and is used in many applications, such as machine-to-machine (M2M) communication.

To achieve this, 4G makes use of multiple technologies. For example, to achieve greater coverage, it uses orthogonal frequency division multiplexing (OFDM) in the downlink and single-carrier frequency division multiple access (SC-FDMA) in the uplink. In addition, to achieve higher speeds, 4G technology employs multiple-input multiple-output (MIMO) technology to improve channel capacity and reduce the impact of interference.

It is an ideal wireless technology for ITS and is capable of handling massive amounts of data, up to 100Gbps, with high reliability. Its biggest advantage is that it uses more stable and powerful signal frequencies, making it easier to deploy and less susceptible to interference. However, its biggest disadvantage is that it is an expensive deployment.

5G is an evolutionary step in the development of 4G and will enable even higher performance and more services by enhancing the current LTE technology. 5G offers even faster connections (up to 5Gbps) and higher capacity, a wider range of coverage, and greater reliability. 5G will support a wide range of applications. These include mobile broadband services, new applications in machine-type communications, augmented reality, vehicular communication, tactile-internet, and vehicular-assisted transportation. 5G enables the integration of various services, such as massive Machine Type Communications (mMTC), enhanced Mobile BroadBand (eMBB), and Ultra-Reliable Low-Latency Communications (URLLC). These services specialize in offering different performances, such as very low communication latency, high bandwidth, low power consumption, and high node density. A comparison of these services can be seen in Table 2-1.

5G will be key to the expansion of ITS, as it offers users a better service with reduced cost and high-reliability thanks to the eMBB service, enabling augmented reality (AR) applications. It will also enable the integration of more IoT devices and sensors by making use of the mMTC service. Finally, the URLLC will enable applications requiring ultra-reliable, low-latency control, such as cooperative centralized driving, connected autonomous vehicles, or the tactile Internet.

6G is the next big thing and will be the foundation for the arrival of artificial intelligence (AI) at the core of communication systems. It will be able to support a future world in which a massive number of devices with different requirements and machines from different domains are connected on the same network, enabling a wide variety of applications and ultra-reliable, low-latency, high-

bandwidth, real-time communications for a variety of devices. It will also support a wide range of human-oriented services, such as cooperative autonomous robots, the metaverse, and real-time services based on cloud and virtualization. In addition, 6G will be fully integrated with other technologies, enabling ultra-reactive and ultra-responsive communication capabilities, bringing the sci-fi future we have all dreamed of within reach.

6G will be a revolutionary leap for all technologies, enabling the vision of 5G to become a reality, increasing spectral efficiency, improving network flexibility, energy efficiency, and enabling a wide variety of services. The new network paradigms will enable new services, research, and innovation in a wide variety of fields, such as decentralized collective intelligence for robots and vehicles, ITS, smart buildings, or real-time e-Health, to name a few. The main idea of the 6G paradigm is the support of a massive amount of new services through low complexity network topologies.

TABLE 2-1. Comparison of different 5G scenarios with different important characteristics.

	eMBB	URLLC	mMTC
Data Rate	High	Medium	Low
Scalability	High	High	Low
User Experience Rate	High	Medium	Low
Mobility	Medium	High	Low
Reliability	Medium	High	Low
Latency	Medium	Low	High
Density node	Medium	Low	High
Energy Efficiency	Medium	Low	High

2.3 Artificial intelligence

The concept of AI [5] refers to the set of computer and computational techniques focused on the creation of computer systems and devices capable of performing tasks that normally require human intelligence, such as reasoning, problem-solving, decision making, prediction, and perception.

Although there is no formal definition of AI [5], for a better understanding of what IA means for this dissertation the above definition is proposed.

Even though AI has been around for many years (1956) [5] and has been capable of remarkable achievements, AI has experienced an explosion in recent years due to the growth of computing power, big data, and IoT sensors.

In this section, we will briefly explore concepts and fundamentals of several areas into which AI can be divided, such as Machine Learning (ML) [6], [7], Deep Learning (DL) [8], [9], and Reinforcement Learning (RL) [10], [11]. For a better understanding of each of the topics discussed, we refer readers to the references

above. Each of the areas discussed and their relationship to each other can be seen in Figure 2-1.

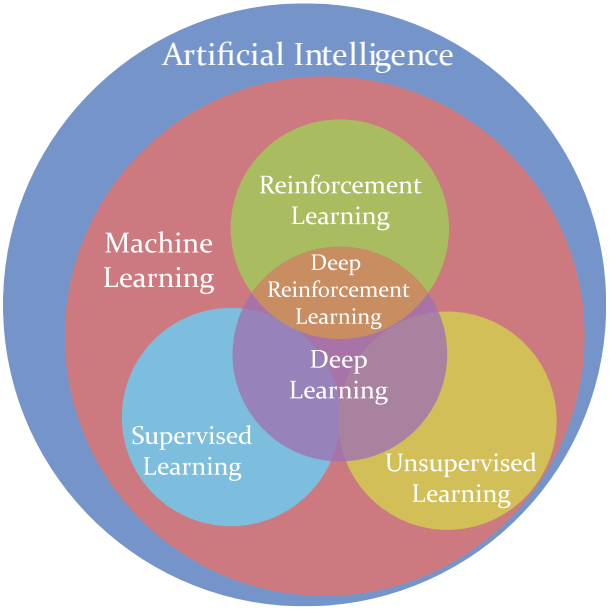


Figure 2-1. Relationship between areas of Artificial Intelligence such as Machine Learning, Supervised Learning, Unsupervised Learning, Reinforcement Learning, Deep Learning, and Deep Reinforcement Learning. Deep learning and Deep Reinforcement Learning are the areas that are receiving the most attention from the scientific community, as they allow solving a vast majority of problems.

2.3.1 Machine Learning

Machine Learning (ML) [6], [7] is the subfield of AI that applies statistical methods to obtain insights from a dataset. A dataset is a large collection of data structured and organized in a matrix in rows and columns. Traditionally, each sample is placed in each row and the columns correspond to the features of the data in each sample. This means that our dataset will have as many rows as samples and as many columns as features in the dataset. Samples can be of any type: numerical, categorical, time series, graphs, and sequences, audio, images, etc. ML methods learn by exploiting the features of the dataset used during training. After successfully training an ML model, another separate but similar data set is used to evaluate the actual performance of the system on unseen data. In this way, it is possible to check whether the ML model is overfitted.

The result of ML methods is an algorithm that has learned to answer specific questions, for example, predicting the price of a house based on characteristics such as its surface area, number of bedrooms, number of bathrooms, whether it has a garage, the distance to a large city, etc.

ML can be grouped into several subfields, such as classification, regression, clustering, anomaly detection, recommender systems, and forecasting systems. Another way to classify ML algorithms into three broad categories: supervised, unsupervised, and reinforcement learning.

Supervised learning is a set of ML techniques that uses labeled data to train a model that predicts or classifies new observations. The most commonly used learning algorithms are decision trees [12], Artificial Neural Networks (ANNs) [13], Support Vector Machines (SVMs) [14], and ensemble models [15], such as random forest [16] or XGBoost [17].

In unsupervised learning, the input data are not labeled, however, there is some structure inherently present in the data. For example, clustering [18] can group similar data or assign each input observation to a group or cluster, or dimensionality reduction [19], where an algorithm is responsible for decomposing a multivariate data set into a set of fewer dimensions with the least loss of information, thus being able to perform exploratory data analysis, information compression, data de-noising, etc.

Reinforcement Learning (RL) [10], [11] is an area of AI that involves an intelligent agent learning to perform actions by trial and error through interaction with its environment. In this case, a reward signal is used to indicate to the agent how good its actions are. Like human learning, RL agents try to find the best actions that maximize the total reward. The main RL algorithms used are Q-Learning [20], SARSA [21], and Temporal difference (TD) learning [22].

2.3.2 Deep learning

Deep Learning (DL) [8], [9] is a subset of ML techniques that differentiates it from "shallow" learning in that the latter requires a process of feature extraction from a dataset, whereas DL focuses on learning features directly from the dataset using large models that allow them to extract these features. A comparative example of ML and DL in a classification task can be seen in Figure 2-2. In addition, another key feature of DL involves using layers of ANNs for model building. These ANNs have a hierarchical structure and several layers of nonlinear neurons and activation functions (or simply, layers) and it is what is known as a feedforward deep neural network or Multi-Layer Perceptron (MLP). An example of MLP architecture can be seen in Figure 2-3.

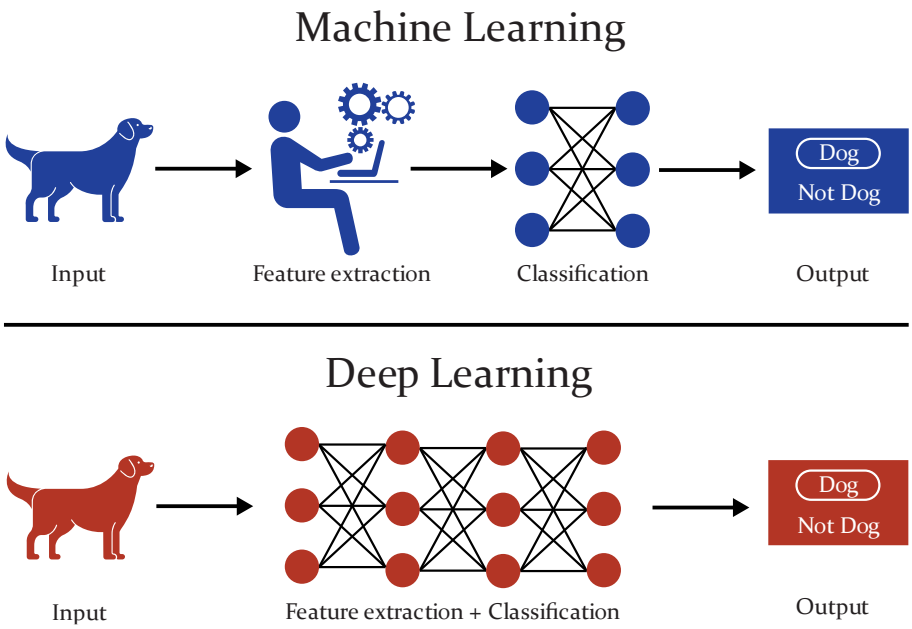


Figure 2-2. Illustrative example for the comparison between ML and DL in a classification task.

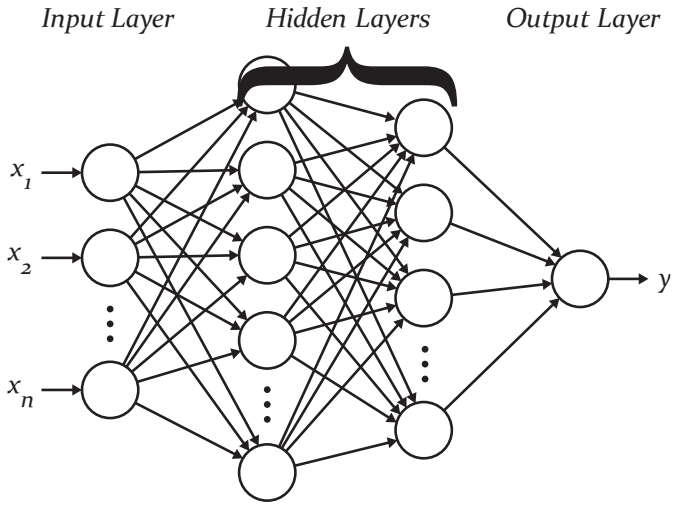


Figure 2-3. Architecture of an MLP composed of n input features, 2 hidden layers, and 1 output neuron.

In ML algorithms, the models used have an input layer and an output layer; however, DL models have multiple intermediate layers (hidden layers). These hidden layers are in charge of learning which features are crucial for the task to be solved, as well as solving it. In each layer (except the input layer), the output of each neuron is calculated by weighting the output of the neurons in the previous layer. A nonlinear activation function such as a sigmoid, tanh, Rectified Linear Unit (ReLU), or its variants is applied to this value. This activation function is a key component for the correct operation of the model since it allows to create functions of approximation with a high degree of freedom. A representation of the output of a neuron can be seen in Figure 2-4. The output equation of a neuron can be seen in Equation 2-1. Where φ is the activation function, w_i is the weight of neuron i, x_i is the value of the input to neuron i and n is the number of neurons.

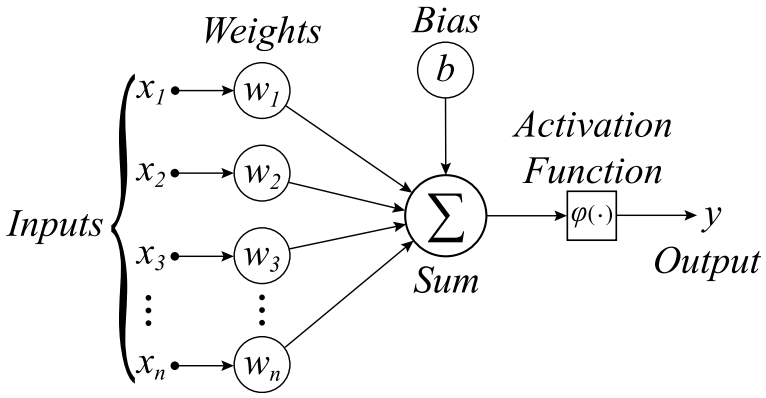


Figure 2-4. The structure of an artificial neuron.

$$y = \varphi\left(b + \sum_{i=1}^{n} w_i x_i\right) \tag{2-1}$$

Once the information flows from the input layer to the output layer, the backpropagation algorithm is applied to gradually adjust the neuron link weights and optimize the loss function. This backpropagation algorithm [23] consists of calculating the partial derivative of the error in each neuron of the previous layer and using the stochastic gradient descent algorithm [24] to reduce the error produced by each neuron.

In addition to MLP, there are other interesting types of deep feedforward neural networks, such as Convolutional Neural Networks (CNNs) [8], and Recurrent Neural Networks (RNNs) [23].

CNNs [8] are deep feed-forward neural networks that employ filters (or kernels) to apply different types of feature extraction. CNNs are composed of convolutional layers, pooling layers, and fully connected layers. CNNs are capable of handling input data of any size, such as images, audio, or time series. In fact, CNNs show great performance in various imaging tasks.

An example of a simple CNN is shown in Figure 2-5, which consists of two convolutional layers, two pooling layers, a fully connected layer, and an output layer with 10 neurons. The first convolutional layer (Conv 1) applies a kernel of 3x3. The second convolutional layer (Conv 2) applies a kernel of 5x5. The pooling layers apply a filter 2x2. The input data to this layer is the activation of the previous convolutional layer. The pooling layer decreases the size of the input data by extracting the maximum value of the data when applying the 2x2 mask. The next layer is a fully connected layer, which connects the last layer with the output layer using a weight vector. This layer has the same number of neurons as the last max-pooling layer. The final layer is the output layer, which predicts the output class of the data in this example.

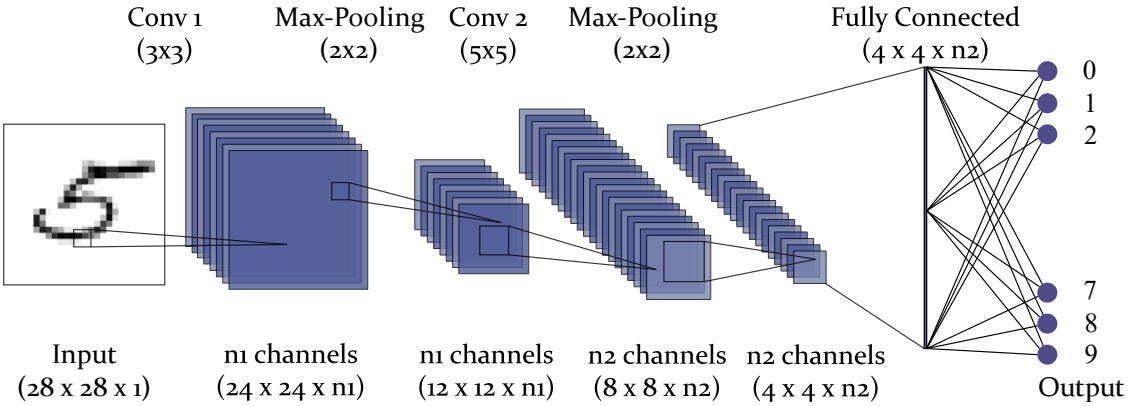


Figure 2-5. Architecture of a convolutional neural network (CNN) composed of *2* convolutional layers, *2* max-pooling layers, and *1* fully-connected layer. It can notice the reduction in dimensionality due to the application of the filter and, especially, the max-pooling layers.

The main advantage of CNNs over MLPs is that CNNs have the ability to understand the spatial correlation between image pixels, which allows us to extract edge and shape features from images. Furthermore, CNNs also have a small number of trainable parameters, which allows for fast training. The main drawback of CNNs is the computational overhead in very deep networks.

On the other hand, RNNs [23] are a type of neural network specialized in temporal data analysis. For this purpose, RNNs integrate short- and long-term memory units that make them capable of learning complex dependencies between data sequences. RNNs are used in a wide range of tasks, such as Natural Language Processing (NLP), speech recognition, time series analysis and prediction, stock market forecasting, early detection of faults in continuous monitoring systems, etc. In fact, Long-Short Term Memory Networks (LSTMs) have shown excellent results in the NLP field. The main advantages of RNNs are their ability to perform memory-based computations, the ability to handle inputs of any size, and the ability to capture long-term dependencies. The main drawback of RNNs is their higher computational complexity and that they tend to suffer from evanescent gradient and exploding gradient problems as well as forgetting long-term dependencies.

Some of the most important RNNs extensions are: LSTMs [25], Gated Recurrent Units (GRUs) [26], and their variants (Bidirectional RNNs [27], etc.).

In order to increase DL performance and handle complex inputs such as images, video, text, translation, or audio, researchers are developing ways to help models focus on the most important features of the data, e.g., in image classification problems, help identify those parts of images that are most representative.

To achieve this, researchers have developed a technique known as attention [28], [29], which is an internal mechanism that the DL model uses to learn how important each feature is and achieve higher performance.

The attention mechanism is as follows. First, the input is encoded by layers (they can be recurrent or convolutional) until the information is compressed into a fixed-length vector. Then, the model learns to decode the fixed-length vector into the target output using different mechanisms. The attention mechanism allows focusing only on the parts that provide the most information and incorporate and extract them from the fixed-length vector (of a much smaller size than the input data). For a better understanding, see Figure 2-6, where it is possible to see the procedure of a transformer-based algorithm for the task of image caption [30].

Figure 2-6. Example of an image caption task using an attention-based algorithm (the white indicates the attended regions, the predicted word is indicated in the upper left corner of each image, the input image is the first image). The complete predicted sentence is: "A woman is throwing a frisbee in a park.". (Figure 6-b, p. 11, [30]).

In recent years the main model of attention is the Transformers model [31], which uses self-attention to eliminate the recurrent and convolutional layers that have traditionally been used by the various attention methods proposed. The transformer encodes the input as a set of key-value pairs. At the decoder, the previous output is compressed into a query and the next output is produced by mapping this query and the key-value pair using scaled dot-product attention. Furthermore, Transformer uses a multi-head self-attention mechanism to execute the scaled dot-product attention multiple times in parallel. The results are then concatenated and transformed into the expected shape.

Comparing RNNs with Transformers, the advantages offered by Transformers are numerous, as they completely eliminate recursion and process sentences as a whole. This gives a huge efficiency during training by being able to parallelize the data input. However, it should be noted that the original Transformers could only deal with fixed input size data. To solve this, new variants of Transformers are being proposed, such as Transformer-XL [32] (2019), Compressive Transformers [33] (2019), Reformer [34] (2020), or Longformer [35] (2021).

2.3.3 Reinforcement learning

Reinforcement Learning (RL) [11] is an area of artificial intelligence dedicated to training an agent that is able to perform actions that interact with its environment it is in and receives a reward signal that tells it how good/bad the action performed was based on the state of its environment it was in. The agent's objective is to find out the actions that maximize the total cumulative reward obtained from the task to be performed through interaction with its environment, using a trial-and-error approach.

The general RL problem is formulated as a discrete-time stochastic control process in which an agent interacts with its environment. Every time interval t, the agent receives the state $s_t \in S$ and performs an action $a_t \in A$ following its policy $\pi(a_t|s_t)$, which allows the agent to map each possible state with the action to be performed. After interacting with its environment through the action a_t, the agent receives a reward value $r_t \in R$ and the next state $s_{t+1} \in S$ according to the dynamics of the environment or model. In an episodic task, the procedure is repeated until a terminal state is reached. The sequence of states, actions, and rewards in an episode constitutes a trajectory or rollout of the policy. This perception-action-reward loop is illustrated in Figure 2-7.

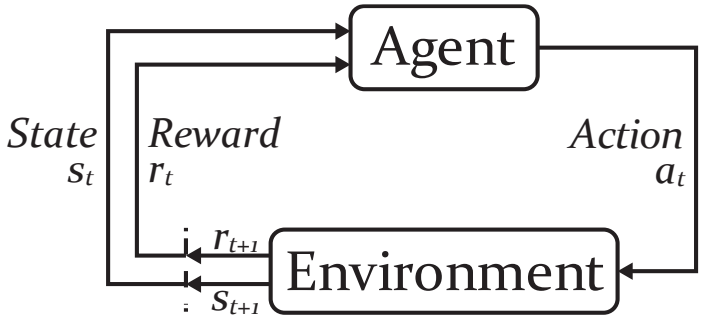


Figure 2-7. Schematic representation of the perception-action-reward loop of a generic RL model. At each time step (t), the agent receives the state (s_t) and the reward (r_t), and accordingly performs an action (a_t) that affects its environment. The environment then generates the new state (s_{t+1}) and reward signal (r_{t+1}) and forwards them to the agent to continue the loop.

The return is the discounted, accumulated reward with the discount factor $\gamma \in (0, 1]$, shown in Equation 2-2.

$$R_t = \sum_{k=0}^{\infty} \gamma^k r_{t+k} \qquad (2\text{-}2)$$

When an RL problem meets the Markov property [36], i.e., the future state depends only on the current state and current actions, but not on the past, it can be formulated as a Markov Decision Process (MDP) and can be defined by the 5-tuple $\langle S, A, R, T, \gamma \rangle$, where S represents a set of states of an environment, A represents the set of actions that the agent can take, T is the transition function $T: S \times A \times S \rightarrow [0,1]$ that determines the transition probability from any state $s \in S$ to any state $s' \in S$ when the action $a \in A$ is taken. R is the reward function $R: S \times A \times S \rightarrow \mathbb{R}$ and $\gamma \in (0,1]$ represents the discount factor that adjusts the trade-off between immediate and future rewards.

Resolving an MDP generates a policy $\pi : S \rightarrow A$, which maps the states $s \in S$ to the actions $a \in A$. An optimal policy π^* maximizes the expected discounted total reward for all states.

To solve an MDP, multiple methods can be used, but within the field of RL, there are two main approaches: *value function* methods and *policy search* methods. In addition, there are hybrid methods that are able to leverage the advantages of both methods, such as the *actor-critic* approach, showing remarkable performance compared to simple methods.

Value Functions

A *value function* is a prediction of cumulative, discounted, expected future reward and estimates how good each state, or each state-action pair, is.

The *state value function* ($V^{\pi}(s)$) estimates the expected return when the agent starts in state s and follows the policy π to perform actions. The *state value function* ($V^{\pi}(s)$) is as shown in Equation 2-3.

$$V^{\pi}(s) = \mathbb{E}[R_t | s_t = s], \text{where}, R_t = \sum_{k=0}^{\infty} \gamma^k r_{t+k} \qquad (2\text{-}3)$$

Another interesting value function is the *state-action value function*. The *state-action value function* (also denoted as quality function, Q-function, $Q^{\pi}(s, a)$) estimates the expected return when the agent starts in state s, perform the action a (which may not have come from the policy π), and then follows the policy π to perform actions in the next actions. The *state-action value function* ($Q^{\pi}(s, a)$) is as shown in Equation 2-4.

$$Q^{\pi}(s,a) = \mathbb{E}[R_t|s_t = s|a_t = a] \quad (2\text{-}4)$$

Finally, the best policy (optimal policy, π^*) can be obtained simply by greedily selecting in each state the optimal Q-value, $Q^*(s,a)$, following the rule shown in Equation 2-5:

$$\pi^* = \max_a Q^*(s,a), \forall\, s \in S \quad (2\text{-}5)$$

Where the optimal *state-action value function* ($Q^*(s,a)$) is defined as shown in Equation 2-6:

$$Q^*(s,a) = \max_{\pi} Q^{\pi}(s,a) \quad (2\text{-}6)$$

To obtain Q^{π}, the Markov property is used to obtain it as a Bellman equation [37], which has the recursive form shown in Equation 2-7.

$$Q^{\pi}(s_t,a_t) = \mathbb{E}_{s_{t+1}}[r_{t+1} + \gamma\, Q^{\pi}(s_{t+1}, \pi(s_{t+1}))] \quad (2\text{-}7)$$

This means that Q^{π} can be improved by *bootstrapping*, i.e., the current values of the estimate Q^{π} can be used to improve the estimate Q^{π}. This is the basis of *Q*-learning [38] and the state-action-reward-state-action (s-a-r-s-a, SARSA) algorithm [21], and can be seen summarized in Equation 2-8:

$$Q^{\pi}(s_t,a_t) \leftarrow Q^{\pi}(s_t,a_t) + \alpha\delta \quad (2\text{-}8)$$

Where α is the learning rate (LR), and $\delta = y_t - Q^{\pi}(s_t,a_t)$ is the temporal difference (TD) error. Also y_t is a value used as a TD target to approximate the real Q^{π} function and this is where RL methods differ.

Q-learning is an *off-policy model-free* method that learns the Q^{π} function, following the rule shown in Equation 2-9. Another representation of this update can be seen in Equation 2-10.

$$Q^{\pi}(s_t,a_t) \leftarrow Q^{\pi}(s_t,a_t) + \alpha\left(\left(r_t + \gamma \max_a Q^{\pi}(s_{t+1},a)\right) - Q^{\pi}(s_t,a_t)\right) \quad (2\text{-}9)$$

$$Q^{\pi}(s_t,a_t) \leftarrow (1-\alpha)\underbrace{Q^{\pi}(s_t,a_t)}_{old\ value} + \underbrace{\alpha}_{LR}\left(\underbrace{r_t}_{reward} + \gamma\ \underbrace{\max_a Q^{\pi}(s_{t+1},a)}_{optimal\ future\ value}\right) \quad (2\text{-}10)$$

SARSA is an *on-policy model-free* method that learns the Q^{π} function, following the rule shown in Equation 2-11.

$$Q^{\pi}(s_t,a_t) \leftarrow Q^{\pi}(s_t,a_t) + \alpha\left(\left(r_t + \gamma Q^{\pi}(s_{t+1},a_{t+1})\right) - Q^{\pi}(s_t,a_t)\right) \quad (2\text{-}11)$$

Algorithm 2-1 and Algorithm 2-2 present the pseudocode for *Q*-learning (0) and for SARSA(0), where "0" indicates that it is based on one-step returns.

Algorithm 2-1: *Q-learning, adapted from* [11]

Output: *state-action* value function Q^{π} .
Initialize Q^{π} arbitrarily, e.g., to 0 for all *state-actions*.

for *each episode* **do**:
  Initialize state s
  for *each step* **do**:
    if *not terminal_state* **then**:
      Choose a using policy derived from Q(e.g., ε-greedy)
      Perform action a, observe r, s'
      $Q(s,a) \leftarrow Q(s,a) + \alpha\left(\left(r + \gamma \max_{a'} Q(s',a')\right) - Q(s,a)\right)$
      $s = s'$
    end if
  end for
end for

Algorithm 2-2: *SARSA, adapted from* [11]

Output: *state-action* value function Q^{π} .
Initialize Q^{π} arbitrarily, e.g., to 0 for all *state-actions*.

for *each episode* **do**:
  Initialize state s
  Choose a using policy derived from Q(e.g., ε-greedy)
  for *each step* **do**:
    if *not terminal_state* **then**:
      Perform action a, observe r, s'
      Choose a' using policy derived from Q(e.g., ε-greedy)
      $a_t = \pi_\theta(s_t)$
      $Q(s,a) \leftarrow Q(s,a) + \alpha\left(\left(r + \gamma Q(s',a')\right) - Q(s,a)\right)$
      $s = s', a = a'$
    end if
  end for
end for

Policy Search

Another approach to obtain an optimal policy (π^*) is using policy search methods. These methods use parameterized policies $\pi(a|s;\theta)$ to which optimization techniques (such as gradient ascent) are applied to parameters θ to maximize the expected return $\mathbb{E}[R|\theta]$. Note that gradient ascent is applied in these methods since the aim is to maximize the expected returns. This optimization is almost always performed in *on-policy* methods, which means that each update only takes into account experiences gathered when acting according to the most recent version of the policy.

The main advantages of policy search methods over value-based methods are that they are more efficient in continuous action spaces or problems with high

dimensionality, as well as being able to learn stochastic policies. However, these methods tend to converge to local optima, are inefficient to evaluate, and suffer from high variance [39]. Within this group, the actor-critic methods stand out.

Actor-critic methods combine the value functions with an explicit representation of the policy. To achieve this, these algorithms use two architectures, one to learn the policy (actor) and one to learn the value function (critic). The "critic" updates action-value function parameters, and the actor updates policy parameters, in the direction suggested by the critic. In doing so, these methods trade-off the reduction of the variance of the policy gradients against the introduction of the bias of the value function methods [40], [41].

One of the most relevant developments within actor-critical algorithms is Deterministic Policy Gradient (DPG) [42]. DPG extends the fundamentals of policy gradients for use with deterministic policies. The main advantage of DPG is that they only need to integrate into state space, requiring fewer samples in problems with large action spaces, unlike stochastic policy gradients that integrate into both state and action space.

Other basic concepts

On-policy vs off-policy:

On-policy learning, the optimal value functions/policy is learned from the actions performed using the current policy $\pi(a|s)$ For example, SARSA evaluates the policy based on samples obtained following the same policy and then refines the policy with respect to the action values.

Off-policy, the optimal value functions/policy, is learned from different actions (e.g., random actions). For example, *Q*-learning finds the state-action values of the optimal policy, without necessarily fitting the policy that generates the data.

The notion of *on-policy* and *off-policy* can be understood as *same-policy* and *different-policy*.

Model-based vs model-free:

In *model-based* algorithms, the RL method makes use of a model of the environment (dynamics), i.e., the transition function (and the reward function), to estimate the optimal policy. The models may be given (e.g., the game of chess, where the valid moves and the probability that the opponent executes a move are known) or learned during training. *Model-based* algorithms have great advantages in domains where interaction is costly, however, an error in the learned model may generate an incorrect control policy.

Alternately, there are *model-free* algorithms, in which RL agents learn an optimal policy from interactions with their environments through trial and error.

Exploration-Exploitation dilemma:

The exploration-exploitation dilemma is a trade-off widely analyzed in game theory and in the field of RL (e.g. [43]). Exploration consists of gaining information about the environment by executing non-optimal actions, while exploitation consists of optimizing the expected return given current information.

When an agent has gained some knowledge about its environment, it has to make a trade-off between learning more about its environment (exploration of new actions) or doing more investigation (exploitation) on the strategy that seems to be the most promising in order to secure the experience accumulated so far.

The most common strategies for dealing with this dilemma are ε-greedy [11], adaptive ε-greedy [44], Upper Confidence Bound [45], and Thompson Sampling [46].

2.3.4 Deep Reinforcement learning

Due to the problems presented by RL algorithms to deal with problems with high dimensionality, and the advancement of artificial intelligence, advanced methods and algorithms were developed to leverage the advantages offered by neural networks when acting as function approximators. In general, Deep Reinforcement Learning (DRL) is based on training deep neural networks to approximate the optimal policy π^*, and/or the optimal value functions V^*, Q^* to solve a task. In this case, the generic perception-action-reward loop for a DRL agent would be as shown in Figure 2-8.

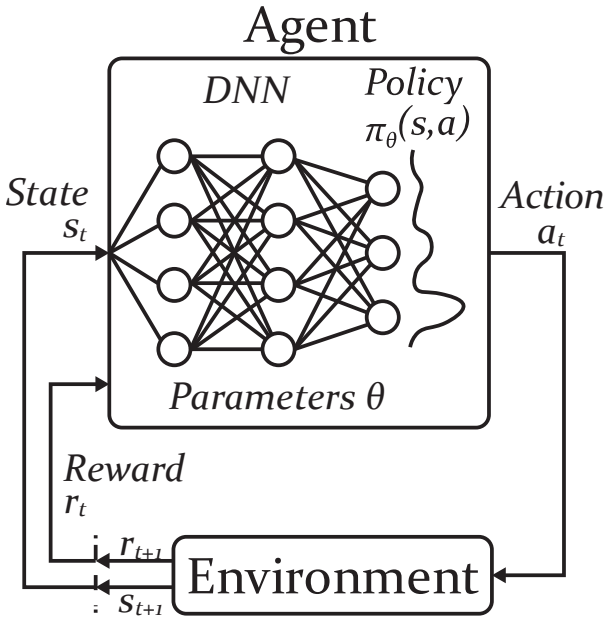


Figure 2-8. Schematic representation of the perception-action-reward loop of a generic DRL model.

One of the main methods of DRL was Deep Q-Network (DQN) [47]. DQN uses several techniques to stabilize training, such as experience replay [48] and target networks. DQN employs a deep CNN to approximate the optimal *state-action* value function (Q^*).

Experience replay stores in a replay buffer the transitions tuples (s_t, a_t, r_t, s_{t+1}) and randomly sampling transitions to remove data correlations and maximize the i.i.d. (independent and identically distributed) in the dataset.

On the other hand, the use of target networks (Q') makes it possible to reduce the correlation between the *state-action* value function Q and the objective ($r + \gamma \max_a Q(s', a')$). This is achieved by slowly updating this target network Q', keeping the parameters of this network (θ^-) in a separate network and updating it periodically only (each $C \in \mathbb{N}$ iterations) Equation 2-12 is the one that uses Q-learning as a loss function to update the network parameters at iteration i.

$$\left(r + \gamma \max_{a'} Q(s', a'; \theta_i^-) \; - \; Q(s, a; \theta_i)\right)^2 \qquad (2\text{-}12)$$

Where θ_i are the Q-network parameters at iteration i, θ_i^- are the targe Q-network parameters at iteration i

In addition to the target Q-network and the replay memory, DQN uses another interesting tool to stabilize training. In order to keep the target values in a reasonable range and guarantee smoothed training, the rewards are clipped between –1 and +1. DQN has a preprocessing step to reduce the 216x160x3 input image to 84×84×1 gray-scale image.

The DQN pseudo code is presented in Algorithm 2-4.

Many DQN-based architectures have emerged to try to solve the problems presented by DQN such as the overestimation problem in Q-values [49], including Double DQN [50] Dueling DQN [51], or Rainbow [52], which employ different techniques such as the use of dueling networks [51], prioritized experience replay [53], multi-step learning [54], and noisy nets [55].

The main limitation of DQN is that it can only handle low-dimensional, discrete action spaces. As an alternative to DQN, more advanced algorithms capable of adapting to high-dimensional continuous action spaces were suggested, such as Trust Region Policy Optimization (TRPO) [56], Proximal Policy Optimization (PPO) [57], and Deep Deterministic Policy Gradient (DDPG) [39].

Algorithm 2-4: *Deep Q-Network (DQN), adapted from* [47]

Initialize replay buffer $\mathcal{B}$ to capacity N
Initialize *state-action value* function Q with random weights θ.
Initialize target *state-action value* function Q' with weights $\theta^- = \theta$

for $episode = 1, M$ **do**:
  Initialize sequence $s_t = \{x_1\}$ and preprocessed sequenced $\phi_1 = \phi(s_1)$
  for $t = 1, T$ **do**:
    With probability ε select a random action a_t
    otherwise select $a_t = \max_a Q(\phi(s_t), a; \theta)$
    Perform action a_t and obtain reward r_t and image x_{t+1}
    Set $s_{t+1} = s_t, a_t, x_{t+1}$ and preprocess $\phi_{t+1} = \phi(s_{t+1})$
    Store transition $(\phi_t, a_t, r_t, \phi_{t+1})$ in $\mathcal{D}$
    Sample random minibatch of transitions $(\phi_j, a_j, r_j, \phi_{j+1})$ from $\mathcal{B}$
    Set $y_j = \begin{cases} r_j & for\ terminal\ \phi_{j+1} \\ r_j + \gamma \max_{a'} Q(\phi_{j+1}, a'; \theta) & for\ non\ terminal\ \phi_{j+1} \end{cases}$
    Perform a gradient descent step on $\left(y_j - Q(\phi_j, a_j; \theta)\right)^2$
    Every C iterations, set $Q' = Q$, i.e. $\theta^- = \theta$
  end for
end for

DDPG is a model-free, off-policy, actor-critical algorithm that addresses the problem of continuous, high-dimensional action spaces extending DQN and DPG [42]. DDPG employs a similar idea to DQN with target networks. In this case, DDPG employs a *soft-copy* technique so that, instead of copying the weights directly, it updates the weights of the target network (θ') in a smoothed fashion following the equation shown in Equation 2-13. In addition, DDPG adds a small amount of noise to the exploration actions.

$$\theta' \leftarrow \tau\theta + (1-\tau)\theta' \ \ with\ \ 0 < \tau < 1 \tag{2-13}$$

Where τ is the parameter responsible for controlling the smoothness of the *soft-copy*. The pseudocode of the DDPG can be seen in Algorithm 2-5.

2.3.5 The emergence of AI in ITS

One of the most exciting and promising application areas of AI is Intelligent Transportation Systems (ITS). This field has attracted a great deal of research interest as a source of innovation and inspiration for the design and development of innovative and cognitive transportation systems.

This challenging interdisciplinary research area integrates the interaction of transportation systems with computer systems such as sensors, telecommunication networks, database systems, big data analysis techniques, and other technologies, like AI.

Algorithm 2-5: *Deep Deterministic Policy Gradient (DDPG)*

Initialize critic network Q_θ and actor-network π_ϕ with random parameters; θ, ϕ.
Initialize target network $\theta' \leftarrow \theta, \phi' \leftarrow \phi$
Initialize replay buffer $\mathcal{B}$
for timestep $t \in \{1, \dots, T\}$ **do**:
  Observe state s and select action $a = clip\big(\pi_\phi(s) + \varepsilon, a_{Low}, a_{high}\big), where\ \varepsilon \sim \mathcal{N}$
  Perform a in the environment
  Observe reward r, new state s' and done signal d that indicates if s' is terminal
  Store transition tuple (s, a, r, s', d) in $\mathcal{B}$
  Sample mini-batch of N transitions (s, a, r, s', d) from $\mathcal{B}$
  Compute targets

$$y_i = r_i + \gamma(1 - d)Q'\left(s_{i+1}, \pi_{\phi'}(s_{i+1})\right)$$

  Update Q-function by one step of gradient descent using

$$\nabla_\phi \frac{1}{|N|} \sum \left(y_i - (Q_\phi(s, a)\right)^2$$

  Update policy by one step of gradient ascent using

$$\nabla_\theta \frac{1}{|N|} \sum Q_\phi\left(s, \pi_\phi(s)\right)$$

  Update target networks using

$$\theta' \leftarrow \tau\theta + (1 - \tau)\theta'$$
$$\phi' \leftarrow \tau\phi + (1 - \tau)\phi'$$

end for

In the mid-2000s, innovative developments in ML, DL, RL, and other AI techniques such as multi-agent systems (MAS) and DRL began to significantly improve the cognitive capabilities of ITS [58]–[63].

AI and ITS research are helping to create more adaptive and intelligent systems. For example, autonomous vehicle driving technologies such as adaptive cruise control, intelligent lane control, and traffic jam assist, as well as full intelligent autonomous vehicle control, are already commonly used in intelligent vehicles [64]–[69].

In addition, in recent years, the use of DL and RL algorithms have been employed for the planning of intelligent transportation systems, such as advanced traffic light control algorithms [70]–[72], traffic signal coordination [73]–[76], trajectory prediction [77]–[81], traffic demand prediction algorithms [82]–[85] to optimize public transportation routes or car-sharing [86]–[91], congestion monitoring and incipient detection [92]–[95], incident detection, forecasting, and severity prediction [96]–[105], predictive road maintenance [106]–[110], etc.

The continuous development of these algorithms has posed new challenges and limitations in the areas of real-time performance, robustness, scalability, security, privacy, and data optimization [111]–[116]. Nevertheless, we can say that AI research has played an important role in improving the cognitive capabilities of ITS [117]–[121].

The challenges of the future are now the combination of ongoing advances in AI with the use of next-generation 5G/6G communication networks, low-cost embedded devices, and IoT sensors. Here, artificial intelligence must be combined with wireless communications, sensors, computing, and processing power to achieve a system capable of adapting to a changing and unpredictable environment, learning, and adapting in real-time, acting autonomously and proactively, and being able to communicate with other actors, in order to achieve deep collaborative and cognitive intelligence.

2.4 Conclusion to this chapter

Communication technologies and artificial intelligence are playing an increasingly important role in the creation of new systems and products. Communication systems and new artificial intelligence algorithms could play a key role in improving the quality of life, as well as in the development of new, safer, smarter, and more efficient transportation systems and services, improving urban mobility and bringing us closer to a truly ITS.

In this section, we have presented a brief introduction to ITS, as well as the fundamentals of AI and a series of terms of great importance for the development of this dissertation on ITS, communication networks, and AI. In the following chapters, we will see the advances proposed in this dissertation based on this theoretical background.

Chapter 3: Unmanned Aerial Vehicles

3.1 Introduction

In this chapter, we will focus our attention on the study of Unmanned Aerial Vehicles (UAVs).

UAVs are flying devices designed to perform tasks autonomously and remotely. In both military and civilian domains, UAVs have been adopted in many tasks in sectors such as agriculture, forest fire control, border surveillance, photography, surveying, mapping, inspection, or telecommunications, to name a few [122], [123]. Even more, UAVs are considered to be one of the most promising technological developments of the 21st century mainly due to their autonomy, flexibility, and mobility.

Thanks to the great versatility that these UAVs can offer, there are different simple taxonomies to classify UAVs, for example, depending on the way they can fly (autonomously or remotely), size (large or light), wings type, or communication capabilities. In terms of wing types, two principal categories exist: Fixed-Wing UAVs (FW-UAVs) and Rotary-Wing UAVs (RW-UAVs). FW-UAVs are characterized by higher flight time, speed, and more aerodynamic design, while RW-UAVs can conduct Vertical TakeOffs and Landings (VTOL), present higher dynamics and control (they can control yaw, pitch, roll, and throttle), and can glide over static points.

Table 3-1 provides a quick comparative overview of FW-UAVs and RW-UAVs by size.

TABLE 3-1. Classification of UAVs by size. (Table 1, p. 2, [124]).

Type of UAV		Speed	Energy Autonomy	Mobility Degree	Static Hover	Altitude
Large-UAV	FW	High	High	Low	No	High
	RW	Low-Med	Low-Med	Med-High	Yes	Med-High
Small-UAV	FW	Med-High	Med-High	Low-Med	No	Low-Med
	RW	Low	Low	High	Yes	Low

In terms of communication capabilities, a distinction is made between individual UAVs, generally deploying star topologies to establish communication with a base station (BS) or satellite [125], and multi-UAV systems [126]. Nowadays, the majority of deployments are carried out using multi-UAV systems, commonly referred to as Flying Ad-hoc NETwork (FANET), which are a swarm or formation of multiple small UAVs connected through an ad-hoc network. Using this approach yields multiple benefits, such as reduced task execution times, reduced costs, increased scalability, and higher reliability, among others [127]–[129].

FANETs belong to the group of Mobile Ad-hoc NETworks (MANETs). MANETs are a special type of ad hoc network created by mobile devices such as notebooks, smartphones, sensors, etc., which have a degree of mobility that gives them a series of characteristics that the network must adapt for their correct operation. MANET nodes have several characteristics that make them unique (mobility, energy consumption, etc.), which imposes several conditions and restrictions on the network that must be met for it to function properly.

Traditionally, Vehicular Ad hoc Networks (VANETs), composed of cars, bicycles, buses,, emergency vehicles, etc., have been placed within MANETs. These nodes present a series of characteristics that differentiate them from MANETs, such as a higher travel speed and a mobility pattern strongly marked by traffic routes.

Following the nomenclature used by MANETs and VANETs, ad-hoc networks created by UAVs can be classified as a special type of ad-hoc network. This type of network is called FANETs, and while FANETs share some properties with MANETs and VANETs, they also have several distinguishing traits, such as: mobility of their nodes, network topology changes, radio propagation, and computational power and energy limitations. Thus, FANETs have high requirements, especially when they are intended for different applications. FANETs can be used for a variety of tasks, such as photography, surveying, tracking, and surveillance, among others, providing numerous advantages in terms of cost, security, mobility, durability, etc., in contrast to MANETs and VANETs. The key distinctions between MANETs, VANETs, and FANETs are shown in Table 3-2.

TABLE 3-2. Comparison of MANETs, VANETs, and FANETs. (Table 2, p. 3, [124]).

Characteristics	MANETs	VANETs	FANETs
Node Mobility	Lower (2D)	Low (2D)	RW-UAV: High (3D) FW-UAV: Medium (3D)
Node Speed	Lower (6 Km/h)	Medium - High (20-120Km/h)	RW-UAV: Medium (40 Km/h) FW-UAV: High (150 Km/h)
Mobility Model	Random	Manhattan models	RW-UAV: RWP [130] FW-UAV: PPRZM [131] / ST [132]
Topology Change	Low	Medium	High
Energy Constraints	Medium	Low	RW-UAV: High (15-60 min) FW-UAV: Medium (to 5 hours)

Due to the characteristics that make FANETs unique, in the state of the art, we find different works that obtain models and protocols that adapt to the unique characteristics of the nodes. Mobility models, as well as positioning protocols, propagation models, and proposed routing protocols are shown below.

3.1.1 Mobility models

The mobility of a FANET node is much higher than that of a MANET and a VANET, mainly due to its 3 degrees of freedom corresponding to the 3 spatial dimensions. In addition, the motion of these nodes obeys a series of rules that consider inertia, wind friction, etc. Mobility models describe the physical behavior of the nodes of a network in a real environment. In addition, they allow characterizing networks allowing them to be anticipated, offering high dynamism and increased performance. By choosing the right mobility model, researchers can simulate FANETs with significantly greater realism and accuracy, and performance results can be provided in advance, guaranteeing a baseline performance [133]. This subsection shows the different mobility models proposed and classifies them into five different groups: pure randomized, time-dependent, path planned, swarm, and hybrid.

Pure Randomized mobility models

Pure randomized mobility models are based on randomness. Thus, using the lowest computational load, they are able to perform tasks where the time required to complete them is not important. Within this group, the mobility models Random Walk (RW) [134], Random Waypoint (RWP) [130], Random Direction (RD) [135], and Manhattan Grid (MG) [136] stand out. Figure 3-1 shows the behavior of a node using these mobility models. It is worth mentioning the MG, specially designed for cities and large metropolitan areas. In Figure 3-1 d it can be seen how the node adapts to the characteristics of the situated environment, however, the physics (velocity, mass, uncertainty, thrust, etc.) of the node is not modeled, so the turns are performed abruptly. This neglect of physics is intrinsic in all pure random mobility models.

Time-Dependent mobility models

To model the environment and the physics of the nodes, time-dependent mobility models were developed. For this purpose, the movement of the nodes is based on mathematical expressions that take into account the time instant of the simulation as well as the state of each node (position, velocity, distance to the edges, mass, etc.). In this way, the mobility model obtained avoids abrupt changes in velocity and direction. Within this category, the following stand out: Boundless Simulation Area (BSA) [137], Gauss-Markov (GM) [138], Enhanced Gauss-Markov (EGM) [139], 3D-Gauss-Markov (3D-GM) [140] and Smooth Turn (ST) [130]. The GM mobility model (see Figure 3-2b) allows for much more realistic modeling of a RW-UAV node's behavior than the RW, RWP, and RD models, but it does not provide high accuracy when modeling the behavior of FW-UAVs, particularly turns. The EGM mobility model for FW-UAVs permits generating results that are highly comparable to the behavior of an FW-UAV in flight, avoiding sudden pauses and

quick turns from the simulation. Moreover, one of the features of the EGM is a progressive boundary avoidance mechanism, resulting in smooth boundary curves. Furthermore, the 3D-GM is a variation of the GM model created for FANETs, with the major characteristic of including mobility in all three dimensions. The behavior of some of these models can be seen in Figure 3-2.

Path-Planned mobility models

As can be seen, in previous mobility models the movement of nodes was completely random within a delimited region. However, for situations where one wants to focus a task on a specific location, they are very inefficient, for example in traffic and city surveillance. This is the reason for the emergence of planned path MM. These models are distinguished by the fact that nodes select from a set of pre-calculated routes and only make decisions around the different routes and movement patterns that each model owns. This group includes the Semi-Random Circular Motion (SRCM) [141] and Paparazzi (PPRZM) models [131] stand out. SRCM is perfectly suited to tasks where a UAV must fly over a point, with different radii, to collect information. PPRZM, on the other hand, is a stochastic mobility model based on a state machine, with each state representing one of the possible movements of a UAV [142]. The PPRZM models are: Stay-At, Eight, Idle, Way-Point, Scan and Oval (depicted in Figure 3-3). A representation of the described models can be seen in Figure 3-3.

Swarm mobility models

Swarm mobility models are based on the coordinated movement of network nodes around a common element. Such characteristic is highly desired for a set of tasks within FANETs, such as search and rescue. Within this group, we highlight: Exponential Correlated Random (ECR) [143], Particle Swarm Mobility Model (PSMM) [144], and Reference Point Group Mobility (RPGM) [145]. PSMM is based on the Particle Swarm Optimization approach (PSO) [146]. This model determines the velocity and direction of each node based on the previous velocity/direction and the position of the nodes with respect to the reference point. Figure 3-4 depicts the PSMM's operational principle. Nodes in RPGM are organized into groupings known as clusters. There is a center in each cluster, which can be either a logical center or a leader node. The center moves according to a RWP mobility model, while the nodes move around it (Figure 3-4). There are numerous variants of this mobility model, among which the following stand out: Column (CLMN) [134] (Figure 3-5), Nomadic Community (NC) [134] (Figure 3-6), and Purse (PRS) [134] (Figure 3-7).

Hybrid mobility models

Finally, models created from the union of two or more mobility models in the previous classes are called hybrid models. This set of hybrid models combines the advantages offered by the separate models and reinforces the disadvantages by using other models. For instance, the hybrid pheromone Markov mobility model (H3MP) [147] relies on Markov chains and Distributed Pheromone Repel models [148]. While Markov chains promote better UAV behavior, the pheromone technique enables for knowledge sharing across UAVs.

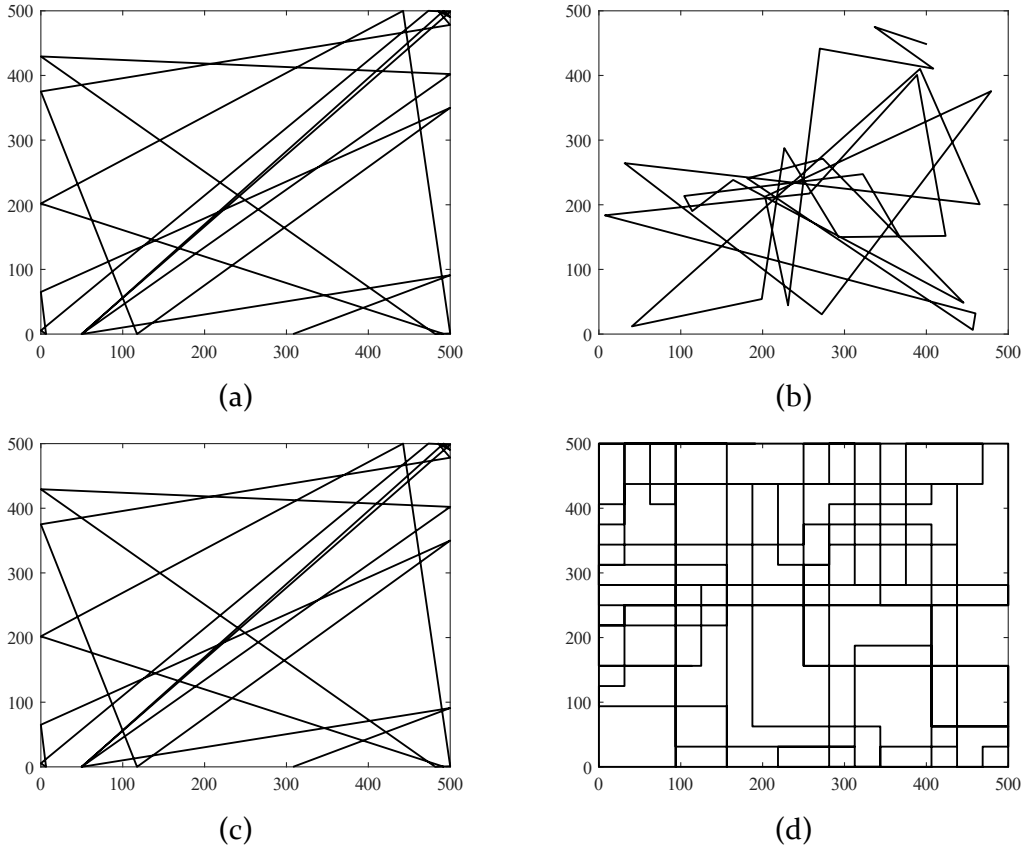


Figure 3-1. Pure Randomized mobility model:. (a) Random Walk (RW); (b) Random Waypoint (RWP); (c) Random Direction (RD); (d) Manhattan Grid (MG). (Figure 3, p. 6, [124]).

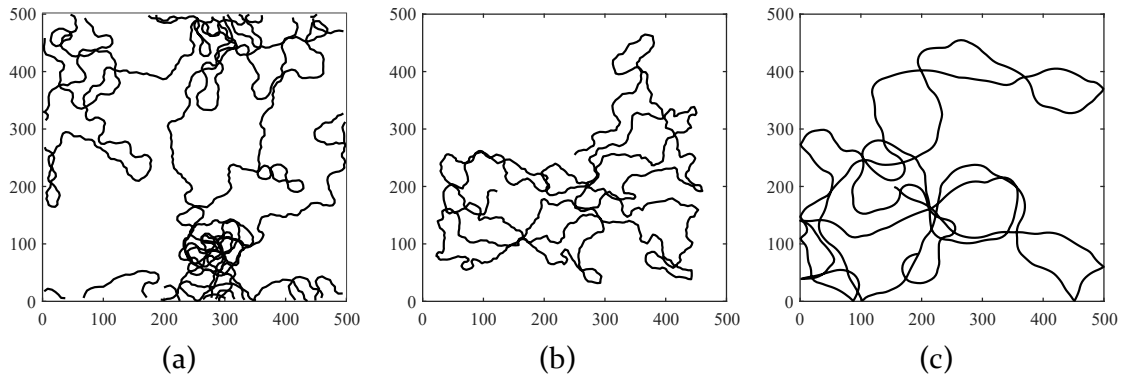
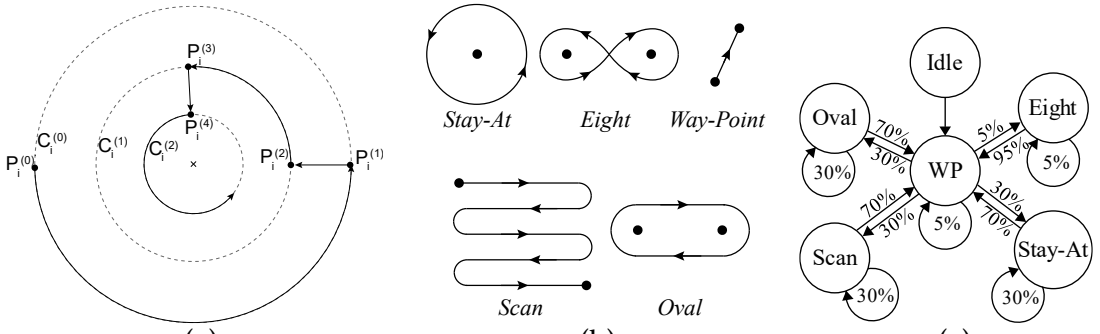
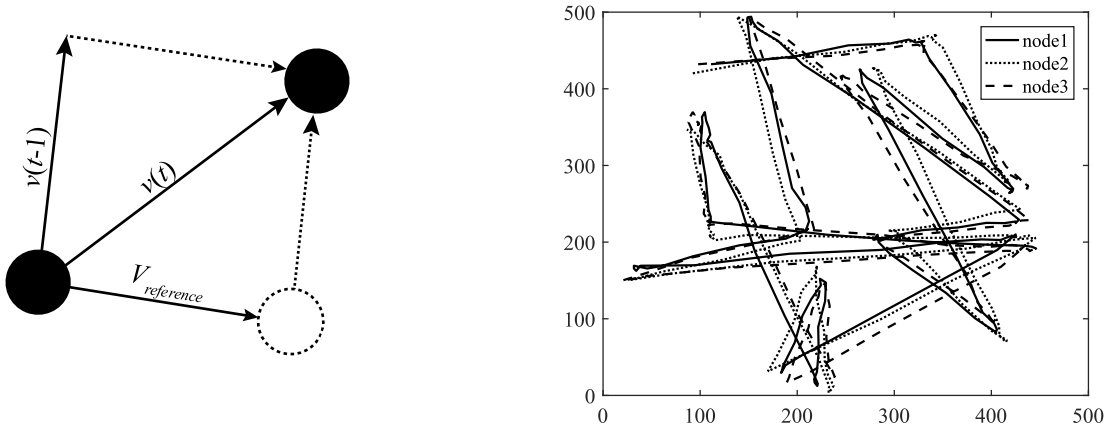


Figure 3-2. Time-Dependent mobility trajectories: (a) Boundless Simulation Area (BSA); (b) Gauss-Markov (GM); (c) Smooth Turn (ST). (Figure 5, p. 7, [124]).



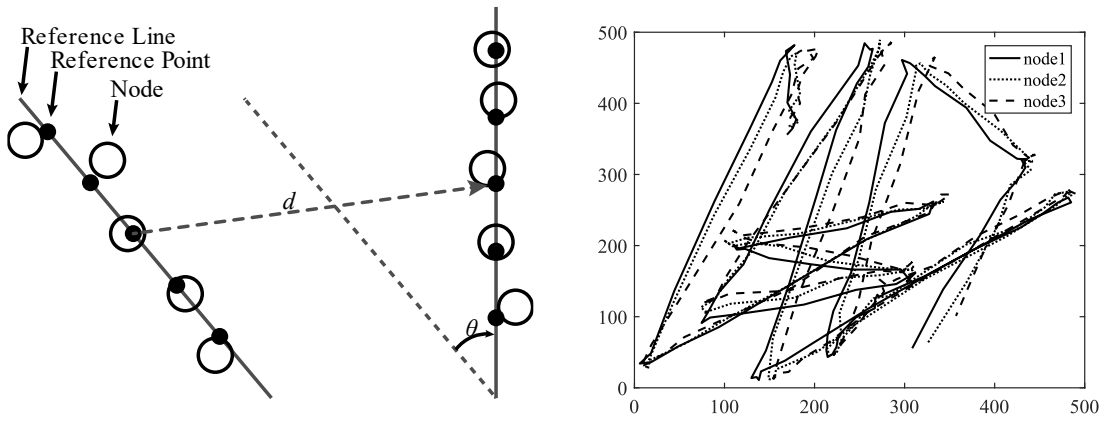
(a) (b) (c)

Figure 3-3. Path-Planned mobility trajectories: (a) Semi-Random circular mobility model (SRCM); (b) Paparazzi (PPRZM) autopilot UAV Movements (Stay-At, Eight, Idle, Way-Point, Scan, and Oval); (c) PPRZM state machine. (Figure 2, p. 6, [124]).



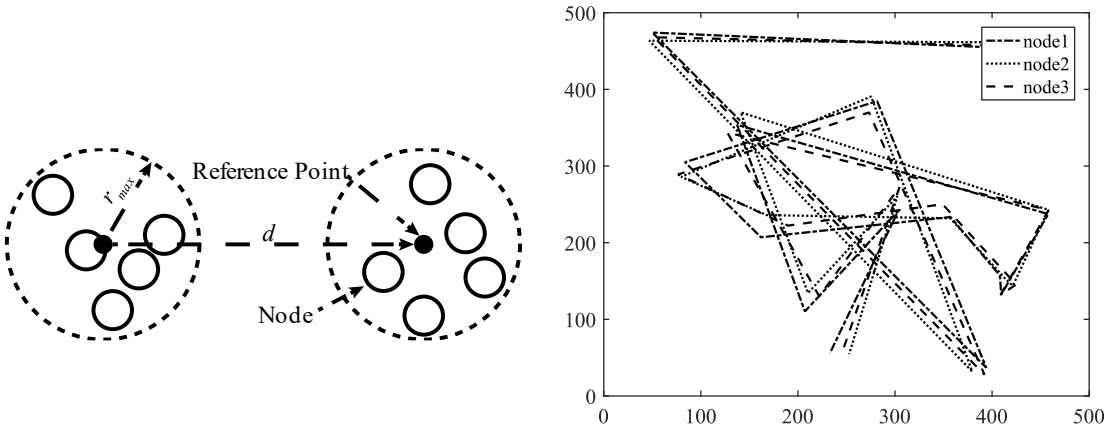
(a) (b)

Figure 3-4. Swarm mobility models: (a) Single-step example in PSMM; (b) RPGM trajectory simulation. (Figure 4, p. 7, [124]).



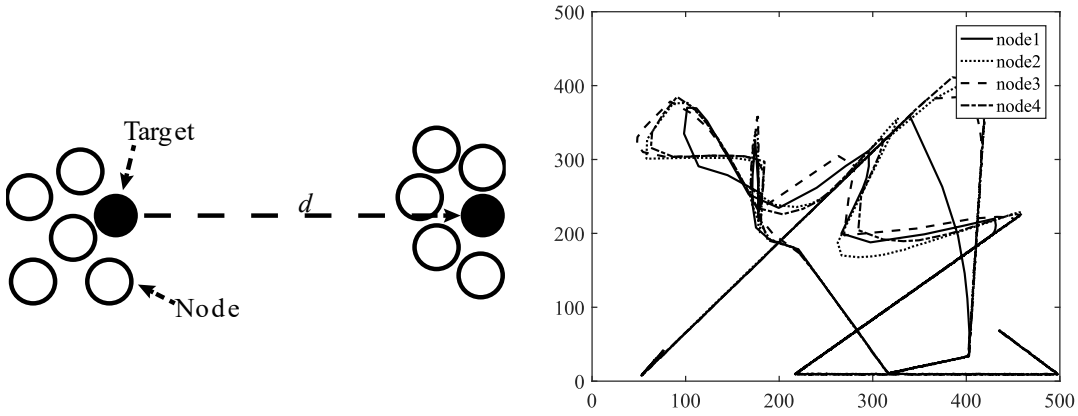
(a) (b)

Figure 3-5. Column mobility models: (a) Single-step example in CLMN, each node has its own reference point, and the reference points are placed on a reference line that moves a distance d and rotates an angle θ; (b) CLMN three-node trajectory simulation, the reference points move arbitrarily, and each node moves randomly around its own reference point. (Figure 6, p. 7, [124]).



(a) (b)

Figure 3-6. Column mobility models: (a) Single-step example in NC with five nodes, nodes have a maximum r_{max} to move away from the reference point, and the reference point moves through the simulation area a random distance *d* following a random pattern; (b) NC three-node trajectory simulation. (Figure 7, p. 8, [124]).



(a) (b)

Figure 3-7. Column mobility models: (a) One-step example in PRS with a target node (black) moving a distance *d* and being pursued by 5 pursuer nodes (white); (b) PRS of a target node and three pursuer nodes trajectory simulation. (Figure 8, p. 8, [124]).

Table 3-3 summarizes the mobility models discussed in this chapter, highlighting the most suitable type of UAV for each model and its typical use case.

TABLE 3-3. Taxonomy of mobility models, the most suitable type of UAV and application scenarios. (Table 3, p. 8, [124]).

Class	Ref.	Mobility Model	RW/FW UAV	Applications
Pure Randomized	[134]	RW	RW	Environmental sensing Traffic and urban Monitoring
	[130]	RWP	RW	
	[135]	RD	RW	
	[136]	MG	RW	
	[137]	BSA	RW	
Time-Dependent	[138]	GM	RW	Environmental sensing Search and rescue
	[139]	EGM	RW/FW	
	[140]	3D-GM	RW/FW	
	[132]	ST	RW/FW	
Path-Planned	[141]	SRCM	RW	Agricultural management Traffic and urban Monitoring
	[131]	PPRZM	RW/FW	
Group	[148]	ECR	RW	Environmental sensing Search and rescue
	[144]	PSMM	RW/FW	
	[145]	RPGM	RW	
	[134]	CLMN	RW	
	[134]	NC	RW	
	[134]	PRS	RW/FW	
Hybrid	[147]	H3MP	RW	Surveillance Search and rescue

3.1.2 Positioning protocols

In this section, we will see the different algorithms proposed for the positioning of UAV nodes in FANETs. Traditional positioning algorithms try to obtain the optimal position of base stations/relays in order to reduce the number of nodes used, maximizing the network deployment. The positioning problem has been widely studied in the telecommunications domain, either by evolutionary algorithms [149], mixed linear programming [150], or greedy algorithms [151]. However, due to the unique characteristics of FANETs (high UAV mobility, 3D, limited UAV lifetime, etc.), the node positioning problem causes the proposed traditional solutions to provide suboptimal results. Hence, novel methods for the optimal positioning of UAV nodes in FANETs have been presented. The objective of UAV nodes in FANETs is to provide ground-level coverage or to extend the capacity of traditional mobile networks [152], [153], or MANETs [154]. Consequently, we can divide the proposed taxonomy into two classes: height-based positioning and network-based positioning.

Height-based positioning

Height-based positioning consists of analyzing the impact of the height of the UAV nodes to obtain a spatial position that maximizes the full performance of the network. Several works analyze the coverage as a function of the height of the UAV

nodes. In the work [155] using a mathematical model that considers transmission losses, they obtained the optimal height for a UAV node to maximize coverage. Following this work, a later work [156] obtained the optimal height of a UAV to minimize the transmitted power and maximize the coverage area offered by the UAV. However, these works used a theoretical channel model, which has a negative impact on the reliability of the obtained models. Analogously, the authors of reference [157] investigated the best height of a UAV for coverage. Here, the channel was considered to have propagation and scattering losses. They demonstrated that there exists an ideal position that increases coverage area with the lowest transmit power.

Network-based positioning

Another approach to deploying FANETs is based on the capabilities/requirements of the network to be deployed. Many works focus on this approach, such as in [158], where the deployment of a FANET composed of FW-UAV is evaluated. Because these FW-UAVs cannot remain stationary over a place, the authors' algorithm constantly modifies the position of the nodes and the coverage radius of each FW-UAV based on the demand. The results obtained improved the coverage offered, compared to a traditional deployment of fixed nodes, as well as a reduction of the delay experienced and an increase in QoS. A different path within this approach was proposed by Gruber *et al.* in [159]. In this work they looked at the attraction and repulsion movements of bacteria against attractants and repellents (known as chemotaxis [160]) to obtain a positioning algorithm, fixing the height of each FANET. While this bacterial-based procedure has already been used for node deployment, this is one of the few works that employ it for UAV nodes [161]. The results demonstrated an increase in the offered performance of the deployed FANET. Based on the coverage probability offered by a UAV, the authors in [152] analyzed the optimal positioning of UAVs in a FANET. For this, based on the circle packing theory [162] and taking into account the number of available UAVs as well as the gain/beamwidth of their directional antennas, they obtained the optimal position for each UAV in order to obtain an best deployment. Adopting a divergent approach, Lyu *et al.* [163] analyzed the lowest number of UAVs required to cover an area by guaranteeing that each ground node was connected to at least one UAV. In [164] Self-Deployable Point Coverage (SDPC) algorithm for RW-UAV nodes was proposed. SDPC investigated the optimal location for disaster coverage extension duties by determining the best position for each UAV to cover the greatest number of people while keeping a link between each UAV. However, SDPC does not consider its implementation with FW-UAV since it does not contemplate smooth trajectories, restricting the application of this algorithm. Finally, other UAV positioning algorithms for FANETs can be found in [165]–[167].

3.1.3 Propagation models

Wireless channel characteristics over which electromagnetic waves are transmitted are critical for any communications system's planning and design. In particular, in order to use acceptable transmit power, obtaining a channel model (or propagation model) that predicts transmission losses is crucial to ensure proper reception at all nodes. There are many propagation models in the literature, however, the most widely used in communication network deployments due to its simplicity is the Friis free space propagation model [168]. This model calculates propagation losses only as a function of the distance between transmitter and receiver, and the frequency used for communication. Although this model adjusts propagation losses quite well, its use is very limited to situations where there is a line-of-sight (LoS) between transmitter and receiver.

The unique characteristics of FANETs require that the characteristics of the radio channel be known in advance, which allows for their correct planning and deployment. Traditionally, two different models have been used to model the radio channel in FANETs, one to model the channel for communications between UAVs (usually with direct vision, but with a high variation in the distance between UAVs) and another for communications between UAVs and ground nodes (with the effect of reflection on the ground, diffraction by objects, etc.). Thus, for the calculation of propagation losses between two UAVs or between a UAV and a ground node, the most appropriate model for each case will be used. Due to the imperative need to model transmission losses in order to plan the network and obtain adequate performance, many theoretical, empirical, and semi-empirical models have been developed by the scientific community to approximate a channel loss model for each usage scenario (UAV-to-UAV, U2U, or UAV-to-ground, U2G) through which electromagnetic waves are attenuated. In the specialized literature, most of the works on channel modeling in FANETs focus on the U2G channel (see [169]–[175]), nevertheless, a series of works focusing on the U2U channel can also be found (see [176], [177]).

Having seen this, in this section we will divide the propagation models that can be found in the literature focused on FANETs for both the U2U and U2G channels. Our categorization divides the models that can be found into four categories: theoretical models, empirical models, hybrid models, and well-known models. The final class investigates the tunability of well-known models, whereas the previous three classes focus on obtaining novel models. TABLE 3-4 provides a full assessment of the channel models covered.

Theoretical models

With respect to theoretical models, Pokkunuru *et al.* proposed in [175], the authors proposed a series of equations modeling the U2G communication channel in the

2.4GHz band. This model considers several important features that other theoretical models do not consider, such as gaseous absorption, Doppler spread, attitude-dependent shadowing, weather, different types of urban areas, and multipath fading. However, this model did not consider the influence that the UAV may cause on the radiation pattern of the embedded communication device, in addition to the fact that the model was limited to airborne small cells. In almost all similar studies we can observe that they avoid considering the impact that the UAV can have on the radiation pattern and, consequently, on the propagation losses. Another theoretical propagation loss model for the U2G channel can be found in the work of Holis *et al.* [169]. The proposed model considered different types of urban areas, the heights of UAVs, as well as the possibility of LoS, for the frequency band between 2-6 GHz. These frequency bands were specifically chosen as they are the most used in mobile communication systems. Finally, the proposed propagation model was tested against a set of real-world observations, proving its practicality.

Empirical models

Another way to obtain transmission loss models is to derive them based on measurements made in different urban/rural scenarios. Although these models will be limited to the set of measurements made, they can obtain adequate results depending on the network to be deployed. Specifically, there are several papers concerned with obtaining a propagation loss model empirically. An example of these models can be found in [170], [171]. The U2G channel was modeled in these studies using observations acquired in the 800MHz frequency. In addition, in [172], the authors also obtained the theoretical model of the U2G channel for the L (960-977 MHz) and C (5030-5091 MHz) bands. For this, they used a medium-sized aircraft, flying at a nearly constant altitude (~500 m - 2 km). Khawaja *et al.* [173] analyzed the 3.1–5.3 GHz Ultra-WideBand (UWB) propagation models for U2G channels using an RW-UAV. For this, they performed a set of measurements in both time and frequency domains. Based on the captured data, a stochastic propagation loss model was obtained, capable of characterizing the UWB propagation channels for the U2G band, considering large scale fading, multipath propagation, and small scale fading in various scenarios.

Hybrid models

Hybrid models start from a set of theoretical expressions modeling the most common propagation losses (free space attenuation losses, multipath fading, scattering, atmospheric absorption, etc.), and then perform a fine-tuning from a set of measurements taken in a real scenario. In this way, the models obtained fit very well to the scenarios analyzed and, beyond the scenarios used in the measurements, the results obtained are usually also quite close to the prediction.

Furthermore, using this hybrid approach, a set of models can be obtained from the same basic theoretical model, which simplifies the development of new models. Nevertheless, the set of models obtained based on this approach usually presents a high complexity, due to the fitting process of the base model to the set of samples taken. An example of these models can be found in [174], which performed a measurement campaign of the U2G channel in an urban area. After this, they chose the theoretical model that best matched the measurements. This model was the Rice + Log-Normal (Loo's model [178]). Finally, they made the necessary adjustments to the theoretical model to reduce the error of the model predictions with the measurements taken. Goddemeier *et al.* [177] obtained another hybrid model, but in this case, they focused their work on the U2U channel. They chose the Rice theoretical model, so they could consider the multipath effect of the signals produced by UAVs when they are located at low altitude. The results concluded with the hybrid model. However, the UAV antennas were placed vertically, making reflections on the ground less frequent and with less power than the direct beam. Finally, they did not consider the impact of the UAV on the diffractions of the radiated power, and only the horizontal radiation plane was studied, leaving the vertical plane unstudied.

Well-known models

Finally, in this section, we will look at works that focus on determining the most appropriate channel models to be used in FANETs depending on their application, scenario, requirements, etc. An example of this work can be found in the study done by Jung *et al.* in [61]. In that paper, they proposed an algorithm that allowed selecting the most appropriate U2G channel model depending on the type of environment in which the FANET would be deployed (urban, suburban, rural, mountainous, near the sea, etc.). In this way, the U2G channel model chosen allowed modeling propagation losses with high accuracy, as well as allowing to adapt to a multitude of environments and scenarios. Another analysis of the adaptation of traditional propagation loss models to FANETs was performed by Daniel *et al.* [179]. The authors compared different propagation models widely analyzed and used for traditional mobile communication systems (e.g., Cost 231-Hata, Walfish-Ikegami, Erceg, Har, WINNER II B1, C1, C2, and D2) with a ray-tracing model, which allowed almost perfect results. The results they obtained concluded that WINNER II C1 and Walfish-Ikegami models were the most suitable when the FANET was deployed below 30m. Above that height, losses due to reflections and multipath are negligible and the free space model could be used to predict propagation losses.

TABLE 3-4. Comparative of the literature on channel models. (Table 4, p. 11, [124]).

Ref.	Category	Channel	Frequency (GHz)
[175]	Theoretical	UAV–Ground	2.4
[169]	Theoretical	UAV–Ground	2–6
[170]	Empirical	UAV–Ground	1.8–5.76
[171]	Empirical	UAV–Ground	0.8
[172]	Empirical	UAV–Ground	0.96–0.977 5.030–5.091
[173]	Empirical	UAV–Ground	3.1–5.3
[174]	Semi-Empirical	UAV–Ground	2
[177]	Semi-Empirical	UAV–UAV	2.4

3.1.4 Routing protocols

In this section, we will look at the different routing protocols designed for ad-hoc and FANET networks. Routing protocols are responsible for the path taken by the data transmitted over the network and are the backbone of the system. In FANETs, the use of wireless communication and the characteristics of the nodes make routing protocols very different from traditional routing protocols. This makes routing a major challenge that affects system performance, furthermore, routing protocols must adapt to the system environment and the changing conditions it will face optimizing network performance. Successful implementation of routing protocols requires that it be tested and validated with the widest variety of conditions and that it be easy to adapt. It must also be agile and make decisions quickly, to adapt to different networks, changes in network topology, and characteristics, as well as handle unexpected difficulties.

The different routing protocols for FANETs will be seen to focus on various applications, due to the wide range of possibilities they offer (coverage extension, search and rescue, real-time monitoring applications, etc.). Thus, for real-time video applications, the routing protocols designed will focus on achieving a constant jitter, in addition to satisfying the intrinsic characteristics of FANETs. On the other hand, for coverage extension applications or support for traditional communications networks, routing protocols will focus on providing high reliability, regardless of delay and jitter. However, to combine the strengths of traditional routing protocols and FANETs, one of the biggest challenges for a successful routing protocol is the ability to adapt to the constantly changing network topology and adapt it to the user's needs.

In this subsection, we present the five groups of routing protocols proposed for FANETs and review their advantages, shortcomings, and applications.

Topology-Based routing protocols

These topology-based routing protocols use link information in data transmission. Within this category we can differentiate:

- Static routing protocols*:* This type of routing protocols are the simplest since the network nodes are forced to be static. Because of this, the routing tables are configured to ensure routing throughout the network, but no changes are allowed. Therefore, networks adopting these routing protocols must ensure a fixed topology, as they cannot adapt to changes in the network. The advantages offered by this type of routing protocols are its simplicity once the network is deployed, however, configuring all the nodes can be a laborious task if there are many nodes in the network, in addition, it must be guaranteed that the network will never change. Within this group, Load Carry and Deliver Routing (LCAD) [180], Multi-Level Hierarchical Routing (MLHR) [181], and Data-Centric Routing (DCR) [182] stand out for FANETs. For example, LCAD employs the Store-Carry-Forward (SCF) [183] methodology as a basis, where a UAV node is responsible for capturing information from other ground nodes, and after capturing the information, the UAV physically moves towards a relay or destination node. As can be seen from the examples, these routing protocols are designed for DTN networks, where communication delay is not an important factor and can be used in agricultural/forestry/animal data gathering applications.

- Proactive routing protocols: Proactive routing protocols are distinguished because each node in the network has a routing table that it updates periodically and also shares it with other nodes in the network so that all nodes have the most up-to-date routing table and create their own. Routing tables contain routes/nodes that a message needs to pass through, from a source node, to reach a destination node. Due to the proactivity offered by this type of protocols when the routing tables of all the nodes are shared, the main advantage they offer is their low delay in sending messages since the route that a packet must follow is known. However, their proactive nature brings with it a major disadvantage and that is the high load of sending routing tables when there are a large number of nodes in the network, which can exponentially increase the number of messages sent. Also, a parameter to be considered that will greatly affect the performance of the FANET is the frequency of sending these routing table update messages. A very high frequency will generate a heavy workload on the nodes, with high bandwidth and energy consumption. On the other hand, a very low frequency will not allow capturing the constant changes in the topology of the FANETs. Therefore, the configuration of this forwarding frequency must be fine-tuned, depending on the application to be deployed. We highlight the following proactive routing techniques that have been suggested or modified for FANETs: Optimized Link-State Routing (OLSR) [184], Destination Sequence Distance Vector (DSDV) [185], Better Approach to Mobile Ad Hoc Network (BATMAN) [186], and Directional OLSR (DOLSR) [187]. Because of its ease of use and speedy deployment, OLSR is likely the most widely used routing protocol in ad hoc networks. In OLSR, each node evaluates the links with its directly connected neighbor nodes by means of a metric that can be

communication delay, bandwidth, packet losses, etc. Once the cost of each node with its direct neighbors has been evaluated, it sends these costs by flooding to all nodes in the network. In this way, each node can search for the shortest path to a node in the network by applying the shortest path algorithm (Dijkstra [188]). However, due to this flooding procedure, as with all proactive routing protocols, if one node falls, the update will take considerable time to occur at all nodes in the network, something that for FANETs we see can be a crucial factor. On the other hand, there is DSDV, which is based on the Bellman-Ford algorithm. This algorithm adds several parameters that make it more complex but allow solving the major problem of the routing loop, as well as determining the freshness of the routes. Furthermore, in order to reduce the workload in sending routing tables, nodes only send incremental updates of their routing tables, although they are also allowed to send their full routing table very infrequently. However, when the network topology changes, a large message sending overhead is necessary before updating the routing tables. This makes DSDV unsuitable for very dynamic or large networks. Nevertheless, DSDV is being used in the field of FANETs in various application scenarios, as can be seen in [189]–[191]. Another interesting routing protocol is BATMAN. BATMAN is a recent routing protocol widely employed in MANETs [192]–[194]. Additionally, a modification of BATMAN, BATMAN-Advanced (BATMAN-ADV) [195], was able to provide enhanced performance owing to its direct incorporation into the data transmission protocol stack. BATMAN-ADV has a major advantage over other reactive protocols since it does not follow a flooding technique of the entire network, but rather each node discovers the entire network from the nodes directly connected to them. This allows to better adapt to the requirements of FANETs and has been extensively studied [192], [195]. To meet the specific requirements of FANETs, a variant of OLSR DOLSR [187] was developed. DOLSR has a great advantage over other routing protocols, as its internal configuration parameters and operating principle allow it to reduce the overall network delay. This makes it ideal for real-time applications such as surveillance or search and rescue. Beyond the explained protocols, there are an extensive variety of alternative protocols employed in FANETs, such as Predictive-OLSR (P-OLSR) [196], Mobility and Load-Aware OLSR (ML-OLSR) [197], Contention-Based OLSR (COLSR) [198], Modified-OLSR (M-OLSR) [199], Cartography-Enhanced OLSR (CE-OLSR) [200], Topology Broadcast Based on Reverse-Path Forwarding (TBRPF) [201], Fisheye State Routing (FSR) [202], and Babel [203].

- Reactive routing protocols: An alternative to proactive routing protocols are reactive routing protocols. These protocols use on-demand route discovery processes, i.e., they only find out which path a message should follow just before sending it. This approach reduces the considerable bandwidth consumption caused by the periodic network discovery of proactive routing protocols and

provides a greater dynamism and degree of adjustment to network changes, which is essential in FANETs. However, due to this reactive route discovery process, the initial latency of communication is high since the route is searched after there is a message to send. The basic approach of this protocol suite for searching the route of a message is to flood the network with *RouteRequest* packets. The destination node, upon receiving a *RouteRequest* packet, responds to the source node by sending a *RouteReply* packet, indicating the (backward) route that the received *RouteRequest* packet traversed. As can be seen, this procedure can lead to serious problems of network flooding and infinite message forwarding if no measures are taken to avoid it. Within this group of protocols, we can find several widely used algorithms, such as Dynamic Source Routing (DSR) [130], Ad Hoc On-Demand Distance Vector (AODV) [204], and Time-Slotted AODV (TS-AODV) [205]. DSR stands out for its simplicity since the source node is responsible for indicating the complete route in the header of the message to be sent. In DSR, the source node indicates the route. One of the most widely used reactive routing protocols within MANETs, VANETs, and FANETs is AODV [189], [206]–[214]. The difference between AODV and DSR is that, in AODV, nodes store in their internal memory a routing table with the routes of previously sent messages. If, when a message is to be sent, the route is already in its routing table, this route is used. If this is not the case or if there is an error in the message forwarding, the process of searching for the route to the destination node is initiated. This simple mechanism of saving routes offers AODV a lower communication latency on certain occasions, suitable for applications where routes are being discovered periodically. Focused on FANETs, TS-AODV (based on AODV) uses a time-slot mechanism to reduce packet collisions, decreasing packet losses considerably and increasing the bandwidth available. Other routing protocols seen in the related literature on ad-hoc networks and FANETs, in addition to the aforementioned reactive protocols, are Multicast AODV (MAODV) [215] and AODV Security (AODVSEC) [216].

- Hybrid routing protocols: Hybrid routing protocols emerged to solve the main problems posed by both proactive (high bandwidth and energy consumption and low dynamism) and reactive (long delay in route discovery) protocols, especially in large networks with a large number of nodes. The main approach followed by this set of protocols is to divide the entire network into subnets (or zones) and select the routing protocol (proactive or reactive) that best suits the needs within each zone, and between zones. A proactive routing strategy, for example, might be utilized inside each zone to improve communication performance, while a reactive routing approach could be used for inter-zone communication. Due to this simple approach, in large networks, performance could be greatly improved by being able to maintain small routing tables within each zone, reducing latencies, and being able to anticipate the creation of routes reactively by knowing in advance that a message is going to leave a zone. Among

the hybrid algorithms, the zone routing protocol (ZRP) [217] stands out. For routes beyond the local zone, route discovery is performed using a reactive protocol. The source node sends a route request to the border nodes in its zone with its own address and that of the destination node. Each border node checks whether the destination node is in its local zone. If the destination is not in the local zone of the border nodes, the border nodes add their own address to the route request packet and forward the packet to their own border nodes. This process is repeated until the destination node is found. If the destination node is in the local area, a route reply message is sent to the source node, indicating the route to the destination node. Using this simple approach, ZRP reduces the overhead of sending control messages in large networks if proactive routing protocols were used, as well as reducing the routing delays within an area that would be caused when reactive routing protocols are used for route discovery processes. If we focus on FANETs, we find a set of algorithms that adapt to the requirements needed by these networks, such as the Temporarily Ordered Routing Algorithm (TORA) [218], Rapid-reestablish TORA (RTORA) [219], Hybrid Wireless Mesh Protocol (HWMP) [220], Sharp Hybrid Adaptive Routing Protocol (SHARP) [221], and Hybrid Routing Protocol (HRP) [222].

Position-Based routing protocols

An alternative to topology-based routing protocols are position-based routing protocols. This type of protocols makes use of the information provided by the geographical position of the network nodes to create routes in a smarter way. As a result, this type of protocol is better suited to the needs offered by FANETs with high node mobility. Protocols in this category are grouped into three distinct categories:

- Reactive-Based routing protocols: This type of protocol combines the reactive protocols with the UAV's geographical position to enable them to obtain greater performance. One of the most widely used protocols within this category is the Reactive-Greedy-Reactive (RGR) algorithm [223]. RGR leverages the AODV reactive protocol [204] to discover routes on demand, and for message delivery, it relies on the Greedy Geographic Forwarding (GGF) protocol [224]. Other prominent protocols made for FANETs are Ad Hoc Routing Protocol for Aeronautical Mobile Ad Hoc Networks (ARPAM) [225] and Multipath Doppler Routing (MUDOR) [226].

- Greedy-Based routing protocols: To reduce message delay, this set of protocols uses the position of nodes to find the path with the lowest number of UAVs, using the Greedy Forwarding approach [227]. Thus, Geographic Position Mobility-Oriented Routing (GPMOR) [228] was designed to suit the needs of FANETs. Employing the Gauss-Markov mobility model [124] and the geographic

positions of UAVs, GPMOR predicts the positions of UAVs in the future and selects the next closest forwarding UAV to the receiver to reduce the latency and the number of hops. In addition to this protocol, there are others such as Mobility Prediction-Based Geographic Routing (MPGR) [229], Geographic Load Share Routing (GLSR) [230], Geographic Greedy Perimeter Stateless Routing (GPSR) [231], Greedy-Hull-Greedy (GHG) [232], Greedy-Random-Greedy (GRG) [233], Greedy Distributed Spanning Tree Routing 3D (GDSTR-3D) [234], and UAV Search Mission Protocol (USMP) [231].

- Heterogeneous routing protocols: Heterogeneous networks, networks formed by nodes of different nature (MANETs, VANETs, FANETs, etc.), are a very common type of networks in practice. Heterogeneous routing protocols are focused on this type of networks and are able to exploit this heterogeneity to offer advantages over other types of routing protocols. In this collection of protocols, we can find Connectivity-Based Traffic Density Aware Routing Using UAVs for VANETs (CRUV) [235]. CRUV leverages a FANET to enhance the performance offered by a DTN VANET, enabling VANET interconnection. In addition, the following can also be found: UAV-Assisted VANET Routing Protocol (UVAR) [236], Position-Aware Secure and Efficient Routing (PASER) [237], Cross-Layer Link Quality and Geographical-Aware Beaconless (XLinGo) [238], and Secure UAV Ad Hoc Routing Protocol (SUAP) [239].

Clustering/Hierarchical routing protocols

This category of routing protocols is based on the existence of clusters and leader/head nodes, hierarchically organized. Each cluster has one or more leader nodes, and these are responsible for inter-cluster communication. The selection of these leader nodes is a crucial task since the overall performance of the network will be limited by the performance that these leader nodes can provide. Communication between nodes within a cluster can be done using one of the protocols discussed in the other sections. Particularly noteworthy in this category of protocols made for FANETs are: Clustering Algorithm of UAV Networking (CAUN) [240], and Mobility Prediction Clustering Algorithm for UAV Networking (MPCA) [241]. On the one hand, CAUN has a fairly simple operation: clusters and leader nodes are configured before deploying the FANET based on the application to be deployed and once the network is deployed the clusters are adapted based on real-time conditions/requirements. On the other hand, MPCA tries to predict the changes in the network topology and, depending on the mobility parameters of each UAV, calculates the clusters. Furthermore, it is possible to find other protocols that belong to this category, such as Landmark Ad Hoc Routing (LANMAR) [242], Multi-Meshed Tree Protocol (MMT) [243], Cluster-Based, Location-Aided DSR (CBLADSR) [244], and Disruption Tolerant Mechanism (DTM) [245].

Swarm-Based routing protocols

There are routing protocols based on the behavior of animals and nature. Based on this inspiration, the resulting protocols have characteristics that make them unique and allow them to adapt to situations in a very organic way. Within this group, for FANETs there is a set of swarm-based algorithms, the BeeAdHoc algorithm [246] which bases its operation on the behavior of bees, and the AntHocNet [247], and APAR [248] algorithms, both inspired by the behavior of ants.

Delay-Tolerant Network (DTN) routing protocols

For applications where there are constant node connection outages, partitions, and topology changes, delay-tolerant routing (DTN) protocols were developed. Moreover, these protocols are well suited for FANETs whose applications allow Store-Carry-and-Forward (SCF). In this way, UAVs in a FANET can move with the information to forward it to another UAV. Within this group, the most used protocol is Location-Aware Routing for Opportunistic Delay Tolerant (LAROD) [68]. LAROD was developed by combining the SCF technique with Greedy Forwarding, according to the situation, using the beaconless strategy to reduce network overhead. Also, within this category, the following algorithms are worth mentioning: AeroRP [249], Geographic Routing Protocol for Aircraft Ad Hoc Network (GRAA) [250], Epidemic [251], Maxprop [252], Spray and Wait [253], and Prophet [254].

A summary of the different classes of routing algorithms, their subclasses, and the algorithms discussed in this section is presented in TABLE 3-5.

3.2 Understanding the Impact of Embedded Devices on the Radiation Pattern of UAVs

In the previous sections, none of the works have addressed the impact that the UAV might have on the integrated communications module, affecting the proposed channel models (especially those obtained from real measurements), as well as the performance of the selected routing protocol. Because of this, in this section, we proposed to analyze the impact that an RW-UAV could have on the radiation pattern of a WiFi communications network in the 2.4 GHz and 5 GHz bands.

To this end, this study was divided into three key objectives. The first was to derive the antenna configuration that best addressed the requirements of the U2U and U2G links. Then, secondly, we obtained the radiation pattern that the UAV has on the integrated WiFi communication device, both in the 2.4GHz and 5GHz bands, for the vertical and horizontal polarizations of the electromagnetic fields.

From the obtained results, we quantified the effect (if any) of the UAV motors, the UAV propellers, and the UAV chassis on the radiation pattern.

Finally, in view of the above results, we denoted a series of conditions that must be met for the radiation pattern of an RW-UAV to be considered isotropic (with constant radiation power at all angles), laying the groundwork for simplifying future work related to UAVs and FANETs.

Among the related works in this field, we can find [255], where the authors performed a set of experimental measurements, measuring RSS and UDP throughput in the U2G channel under the WiFi 802.11a standard, in the 5GHz band, using two antennas, for an RW-UAV. The conclusions they reached were that the antennas should be placed parallel to the ground and at a 90° angle to each other. In this way, the antennas could provide a radiation pattern capable of reducing the impact of UAV motion, demonstrating the importance of antenna position selection. An extension of the previous work can be found in [256], where the authors reanalyzed the U2G communication channel of an RW-UAV, performing a set of RSS measurements on the WIFI 802.11a standard, for the 5GHz band.

TABLE 3-5. A summary of routing algorithms and their classes and subclasses. (Table 2, p. 7, [214]).

Type	Subtype	Routing Algorithm
Topology-Based	Static	LCAD [180], MLHR [181], DCR [182]
	Proactive	OLSR [184], DSDV [185], BATMAN [186], BATMAN-ADV [195], DOLSR [187], P-OLSR [196], ML-OLSR [197], COLSR [198], M-OLSR [199], CE-OLSR [200], TBRPF [201], FSR [202], Babel [203]
	Reactive	DSR [130], AODV [204], TS-AODV [205], MAODV [215], AODVSEC [216]
	Hybrid	ZRP [217], TORA [218], RTORA [219], HWMP [220], SHARP [221], HRP [222]
Position-Based	Reactive-Based	RGR [223], GGF [224], ARPAM [225], MUDOR [226]
	Greedy-Based	GPMOR [228], MPGR [229], GLSR [230], GPSR [231], GHG [232], GRG [233], GDSTR-3D [234], USMP [231]
	Heterogeneous	CRUV [235], UVAR [236], PASER [237], XLinGo [238], SUAP [239]
Clustering/ Hierarchical		CAUN [240], MPCA [241], LANMAR [242], MMT [243], CBLADSR [244], DTM [245]
Swarm-Based		BeeAdHoc [246], AntHocNet [247], APAR [248]
Delay-Tolerant Network		LAROD [257], AeroRP [249], GRAA [250], Epidemic [251], Maxprop [252], Spray and Wait [253], Prophet [254]

However, in this study, they employed 3 antennas at the ground node to measure the received power of the UAV. The antennas were placed in a triangular position, to reduce reception nulls. However, in both works the authors did not consider the use of multiple antennas in transmission, i.e., in the UAV, nor did they study the U2U channel, nor the influence that the UAV motors/chassis may have on the radiation pattern.

Other similar works are found in [258], [259], where they also study the U2G channel for several scenarios such as urban, suburban, or rural, in addition to performing measurements varying the antenna positions, as well as varying the distance between nodes. The results they obtain are that the influence of the antenna position is large and should be taken into account before deploying the network and study the different antenna configurations since the UAV could adversely affect the radiation pattern.

As can be seen in the works seen, a major drawback is that they do not consider the external interferences they may have, even if the experiments are performed outdoors, it is still not a totally controlled scenario free of interferences or reflections, and the results may be disturbed. This major problem could be solved by studying in a totally controlled scenario such as an anechoic chamber or a reverberation chamber, places specially designed for the electromagnetic study of devices such as antennas, cell phones, laptops, etc.

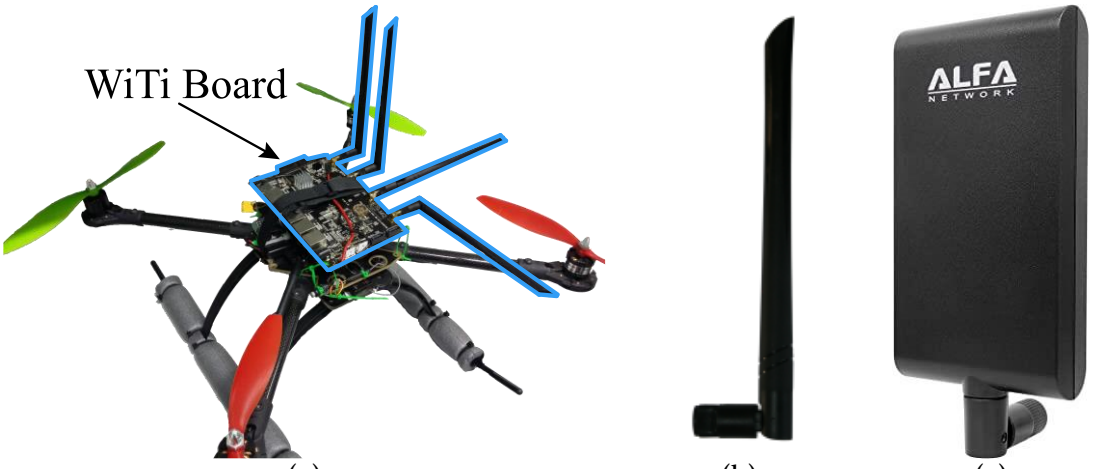
3.2.1 Experimental setup

Prior to studying the comprehensive radiation pattern under a fully controlled environment, in this section we will detail the hardware, software and testbed used. The UAV used in the experiments was the IdeaflyIFLY-4S (Shenzhen Idea-Fly Technology Co., Ltd, Shenzhen, China), a quadcopter widely used in the UAV field for its low weight and high maneuverability, with carbon fiber chassis, brushless motors, and interchangeable propellers.

Attached to this UAV was a wireless communication module consisting of an open firmware router WiTi Board (hereafter WiTi) [260]. WiTi was built on OpenWRT v14.07 with dual band and four antennas, two for each WiFi band. As communication chipsets were the MT7602E chip, which supported 802.11 b/g/n WiFi 2T2R, and the MT7612E chip, which supported 802.11 a/b/g/n/ac WiFi 2T2R, using the 2.4 GHz and 5 GHz bands, respectively. In addition, both communication chips could operate independently and simultaneously. Figure 3-8 shows the UAV and WiTi embedded. Furthermore, to study a broader antenna configuration, two different types of antennas, both dual-band, were used. The first, an omnidirectional antenna with a gain of 5 dBi on both bands (ARS-NT5B), and the second, a directional patch antenna with a gain of 8 dBi for the 2.4 GHz band and 10 dBi for the 5 GHz band (APA-M25).

To analyze the influence of the UAV on the radiation pattern in a controlled environment, the UAV was placed next to the WiTi in a radio frequency anechoic bed. This anechoic chamber had dimensions of 5.2 m × 3.5 m × 2.25 m and was composed of electromagnetic wave isolation cones, which prevented the waves from penetrating the anechoic chamber, as well as preventing the signal from bouncing inside the chamber. The UAV was placed on a turntable with an INN-CO DE 3260-P moving arm that was controlled by an INN-CO CO 2000 digital controller. The R&S ZVL spectrum analyzer, which allows capturing signals up to 6 GHz, together with the R&S HF906 horn antenna with a large receive bandwidth (between 1 GHz and 18 GHz) and constant gain (10 dBi for 2.4 GHz and 11 dBi for 5 GHz) were used as receiving devices.

To analyze the power transmitted by the WiTi and to obtain the influence that the UAV would have, it was decided to use the traffic generator *iperf* v2.0.9 [261] in its default configuration for sending TCP (transport control protocol) traffic. Thus, after launching the traffic session, the WiTi transmitted with maximum power a data transmission session and the signal was captured by the HF906 antenna and analyzed by the ZVL spectrum analyzer connected to it. As a receiving device for the *iperf* data transmission session, a USB-WiFi device was placed inside the anechoic chamber. The power measurements were analyzed with a 10-degree step, making a total of 36. The bandwidth analyzed was 20MHz, centered on the central frequency, with an analyzer resolution bandwidth of 3KHz. The captured signal was averaged, and the average power value was obtained. The distance between the UAV and the HF906 antenna was 3m, with a height to the ground of 1.5m. These distances allowed the far-field condition to be met for both working bands. Figure 3-9 depicts a schematic overview of the testbed. The testbed parameters are summarized in Table 3-6.



(a) (b) (c)

Figure 3-8. UAV and WiTi embedded- device and antennas used: (a) UAV and WiTi; (b) ARS-NT5B antenna; (c) APA-M25 antenna. (Figure 9, p. 14, [124]).

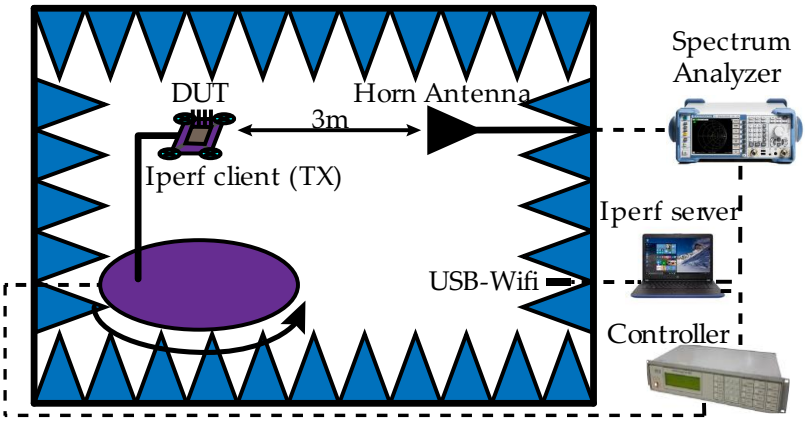


Figure 3-9. Experimental testbed elements (DUT ≡ Device Under Test ≡ UAV + WiTi device). (Figure 10, p. 14, [124]).

TABLE 3-6. A overview of the testbed's parameters and characteristics. (Table 4, p. 14, [124]).

Device or Parameter	Configuration
TX Antennas (Antenna Gain 2.4/5 GHz)	ARS-NT5B (5/5 dBi), APA-M25 (8/10 dBi)
RX Antenna (Antenna Gain 2.4/5 GHz)	HF906 (10/11 dBi)
Distance between TX and RX Antennas	3 m
Traffic Generator	*Iperf* v2.0.9 (TCP default configuration)
Center Frequency	2.462/5.210 GHz
Transmitted BW	40/40 MHz
Captured BW	20/20 MHz
TX Power	20/20 dBm

3.2.2 Results

In this study, we conducted three different experiments. The first included determining the ideal antenna configuration for the WiTi as an isolated device, i.e., the best orientation to provide the most signal power for the U2U and U2G connections, for the 2.4 GHz and 5 GHz WiFi bands. A representation of the U2U and U2G links is shown in Figure 3-10. The second series of studies sought to determine the impact (if any) of the UAV chassis, UAV motors (RPM and vibrations created in the antennas and chassis), and UAV propellers on the radiation pattern in the 2.4 and 5 GHz bands. Following the demonstration of the UAV influence, the final set of experiments aimed to obtain the most accurate radiation pattern of the entire system (UAV + WiTi) using the optimal antenna configuration, that is, the antenna configuration that provided the highest transmitted radiated power for each U2U and U2G communication link.

To achieve the most complete radiation pattern possible, it was decided to study three radiation planes, represented in Figure 3-11 as X-plane, Y-plane, and Z-plane. In addition, within each plane, the vertical and horizontal polarization of the electromagnetic fields was studied to test a larger number of possible antenna configurations. As an example of measurement and what it represents, Figure 3-11 shows each plane analyzed and the position of the WiTi in each plane. To compare

different antenna configurations considering a specific scenario, it was decided to select regions of interest. In Figure 3-12, the regions of interest for the U2U (green) and U2G (yellow) links are highlighted.

We postulate, in particular, that the regions of interest for UAV-to-UAV communications (U2U connections) must be confined in the common horizontal plane (±45°). As a result, the region of interest for U2U linkages would include the whole X plane plus two 90° cones (45° to 135° and -45° to -135°) in the Y and Z planes, as illustrated in Figure 3-12. In turn, we defined the U2G communication region of interest as a 90° cone (from 135° to -135°) in the Y and Z planes from the UAV to the ground. Because of these regions of interest, obtaining the best antenna configuration came down to simply discovering the antennas that radiated the most power in those regions. Note that for different type of network deployment, for example, in the case of satellite-assisted UAV communication, the region of interest for the UAV-satellite link would be different, in which case the antenna configuration that maximizes power transmission in the Y and Z planes at a 90° angle (from -45° to 45°) might be of interest.

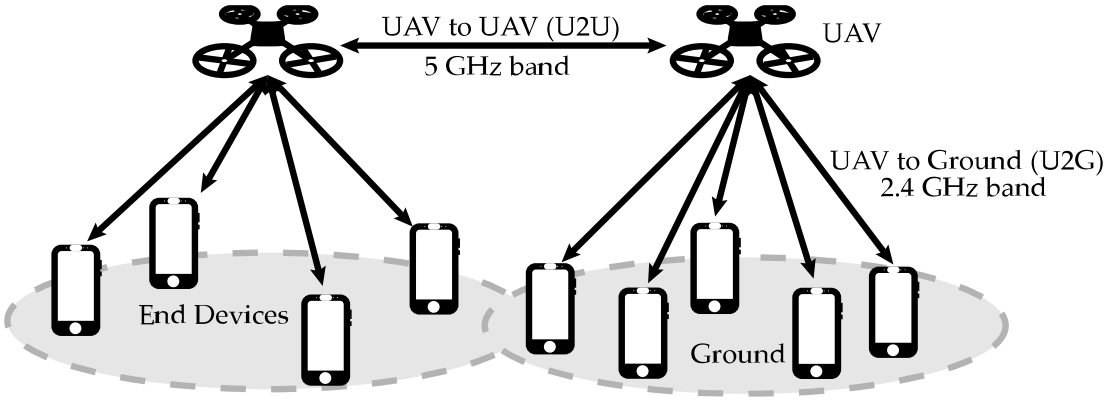


Figure 3-10. Example of FANET communication links employing the 5 GHz band for U2U communication and the 2.4 GHz band for U2G coverage. (Figure 1, p. 3, [124]).

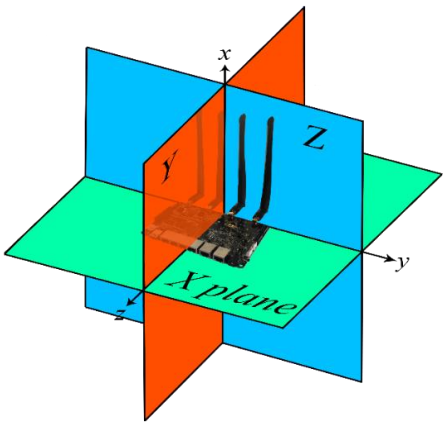
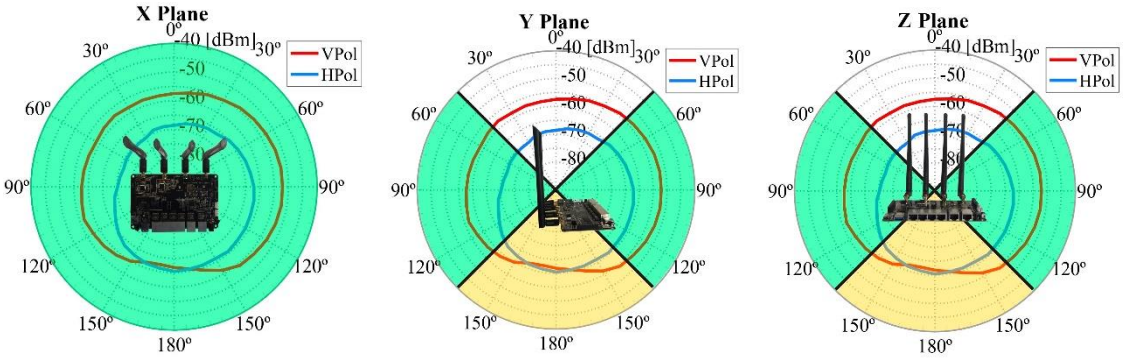


Figure 3-11. Planes under study (X-Plane, Y-Plane, and Z-Plane) and WiTi. (Figure 11, p. 15, [124]).



(a) (b) (c)

Figure 3-12. Example of radiation pattern measurement in the three planes under study and highlighted in color (green for U2U and yellow for U2G) the areas of interest for each communication link: (a) X Plane; (b) Y Plane; (c) Z Plane. (Figure 12, p. 15, [124]).

After performing the first series of experiments, the best antenna configuration obtained was the one shown in Figure 3-13. The radiation patterns we obtained with the antenna configuration previously depicted are shown in Figure 3-14 for the 2.4GHz band and in Figure 3-15 for the 5GHz band.

It can be seen that, by orienting the antennas correctly, the radiated power can be concentrated within the regions of interest. For example, for the U2G link, in the 2.4GHz band shown in Figure 3-14 it can be seen that very concentrated power has been obtained within the area of interest defined for the Y and Z planes (from 135° to -135°), Figures 3-14 b and c. This is achieved by the selection of the APA-M25 directional antennas in the 2.4GHz band. In addition, by placing the antennas perpendicularly, forming a 90-degree angle, and pointing at the nadir, we can take advantage of the polarization diversity and increase the possibilities of having the same polarization between the transmitting and receiving antennas. If we had selected these antennas for the 5GHz band in the U2G link, since the radiation pattern of these antennas is narrower, the entire cone of the area of interest (135° to -135°) would not be covered. The 2.4 GHz band had a gain of 8 dBi, compared to 10 dBi for the 5 GHz band, which would make the radiation lobe narrower and not cover the entire cone of the area of interest.

Furthermore, based on the characteristics of the area of interest of the U2U link (omnidirectional in the X plane, and a 90° cone in the Y and Z planes), the best configuration placed the ARS-NT5B omnidirectional antennas parallel to each other and perpendicular to the X plane. Thus, the results showed an omnidirectional radiation pattern in the X-plane (Figure 3-15 a, VPol), as well as sufficient amplitude in the required regions of interest in the Y- and Z-plane (see Figure 3-15 b and c).

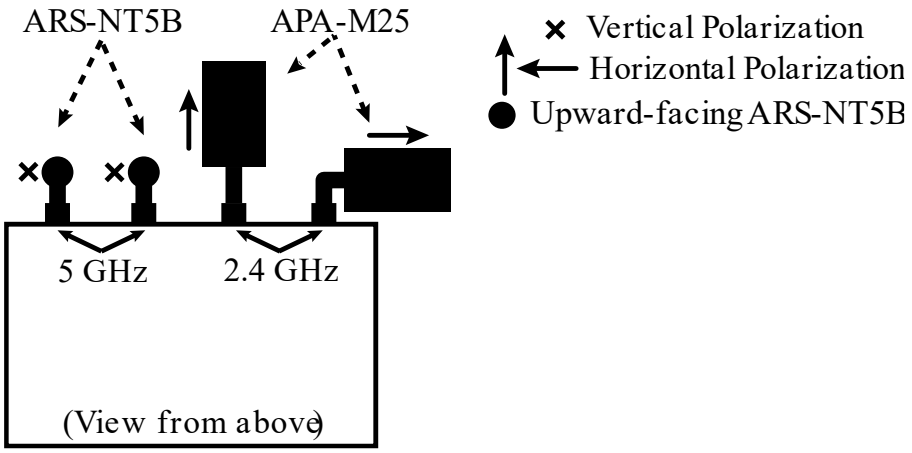


Figure 3-13. Upward view of WiTi's best antenna configurations: 2.4 GHz band for U2G link with APA-M25 antennas in horizontal position with 90° difference and 5 GHz band for U2U link with ARS-NT5B antennas in vertical position. (Figure 13, p. 16, [124]).

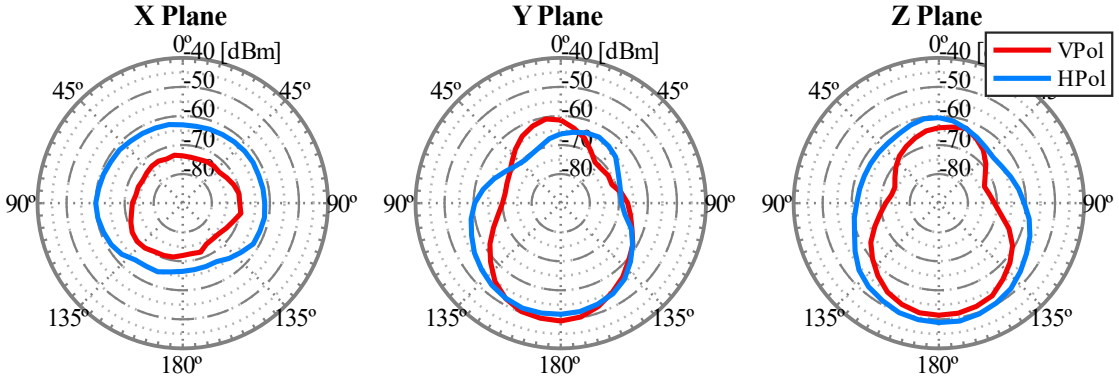


Figure 3-14. Radiation pattern of the isolated WiTi in the U2G link using the antenna setup indicated in Figure 3-13 in the U2G link @ 2.4 GHz: a) X-Plane; (b) Y-Plane; (c) Z-Plane. (Figure 14, p. 16, [124]).

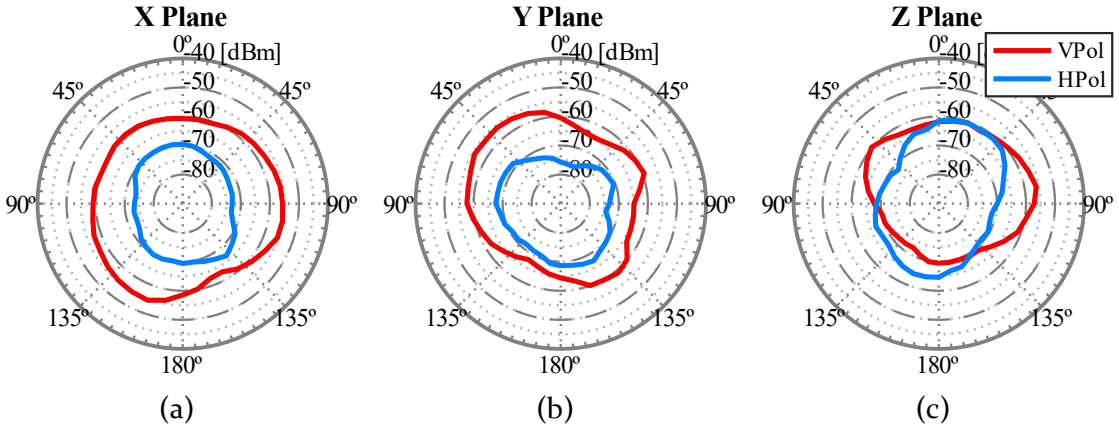
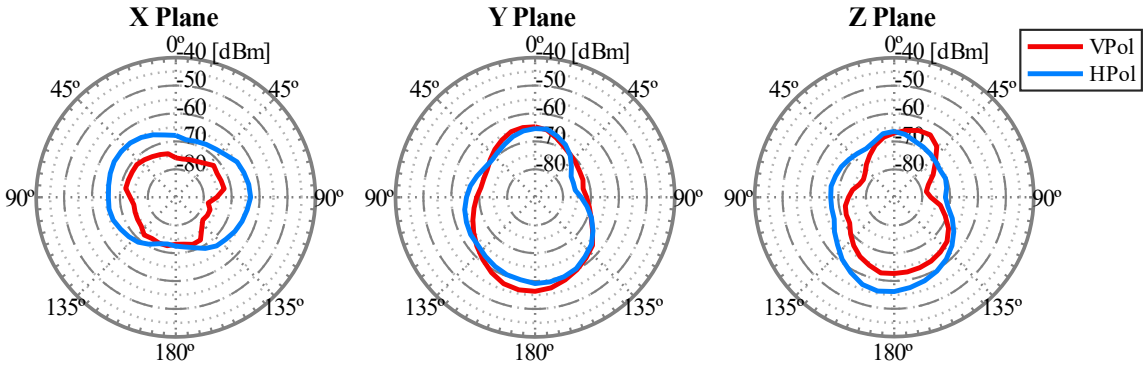


Figure 3-15. Radiation pattern of the isolated WiTi in the U2U link @ 5 GHz using the antenna configuration shown in Figure 3-13: (a) X-Plane; (b) Y-Plane; (c) Z-Plane. (Figure 15, p. 16, [124]).

To analyze the influence of the UAV, the WiTi was placed on board the UAV and the radiation pattern was obtained with the antenna configuration obtained previously. The measurements performed included different states of the UAV, in order to separate the influence of the propellers, engines, and chassis. UAV off, UAV on with the motors at 45 percent power with and without propellers, UAV on at 65 percent power with and without propellers, and UAV on at 95 percent power with and without propellers were the specific states studied.

The results obtained showed that the influence of the propellers and motors was negligible compared to the influence due to the carbon fiber chassis. This is probably partly attributable to propellers being made of plastic and motors are brushless, in addition to the high stability of the whole UAV, with almost no vibrations. These findings may alter if carbon fiber propellers were utilized (which provide more thrust because they can spin faster) or if other types of motors were used.

Regarding the effect of the UAV chassis, it was most noticeable in the Y and Z planes within the range of 135° and -135°. To verify these differences, Figures 3-14, 3-15, along with 3-16, and 3-17 can be compared. As a result of focusing its region of interest inside that area, the U2G connection (Y and Z planes) was the most impacted, with a difference in received power of up to -10 dB. As for the U2U link, for the X-plane in the 5GHz band, within the area of interest, it can be seen in Figure X that the power does not fluctuate significantly, being even higher at some angles. This may be due to the carbon fiber composition of the chassis, which prevents the waves from passing through it and these are directed towards the X-plane.



(a) (b) (c)

Figure 3-16. Radiation pattern of the WiTi on-board the UAV for the antennas configuration shown in Figure 3-13 in the U2G link @ 2.4 GHz: a) X-; (b) Y-Plane; (c) Z-Plane. (Figure 14, p. 16, [124]).

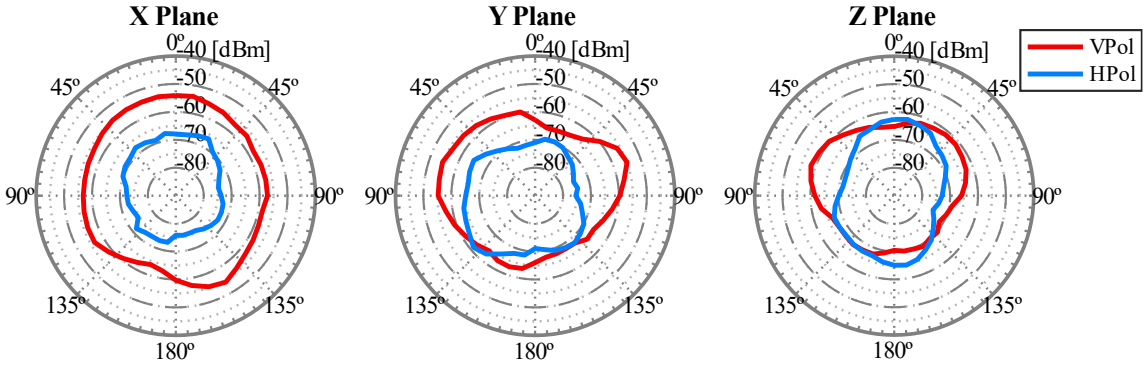


Figure 3-17. Radiation pattern of the WiTi on-board the UAV for the antennas configuration shown in Figure 3-13 in the U2U link @ 5 GHz: (a) X-Plane; (b) Y-Plane; (c) Z-Plane. (Figure 15, p. 16, [124]).

Figure 3-18 illustrates the received power for the full range of angles when WiTi was used alone (without drone) and when used as a communication device onboard a UAV (with drone). In addition, the 2.4 GHz band is illustrated in the X-plane (see Figure 3-18 a) and in the Y-plane (see Figure 3-18 b), and in the 5 GHz band in the X-plane (see Figure 3-18 c) and in the Y-plane (see Figure 3-18 d), for vertical (Vpol) and horizontal (Hpol) polarizations. Since the results obtained in the Z-plane were identical to those in the Y-plane, the Z-plane has been omitted to simplify the visualization, and the Y-plane results can be extrapolated to the Z-plane.

Finally, from the set of experiments performed, we could conclude that the incorporation of the UAV in the proposed models and protocols for FANETs can be seen as additional system losses, as a function of the working angle, as well as a function of the working band. Even for UAVs with configurations similar to those used in this study, the losses can be modeled as constant losses of -10dB in the U2G link for the 2.4GHz band. In this way, it could be guaranteed that the results obtained with the new models and protocols will comply with the received power is sufficient to corroborate the results, as long as the limits of the previously delimited regions of interest are respected. In addition, the most complete and optimal radiation pattern of the UAV together with the WiTi communication device can be found in Figures 3-16 and 3-17. This radiation pattern was specifically optimized for the links and areas of interest shown in Figure 3-12. Furthermore, with the antenna configuration proposed for this purpose, see Figure 3-13, it can be stated in view of the results that the radiation pattern generated by the UAV+WiTi was isotropic for the U2U link, and non-isotropic but constant for the U2G link, both within the regions of interest described above for each communication link.

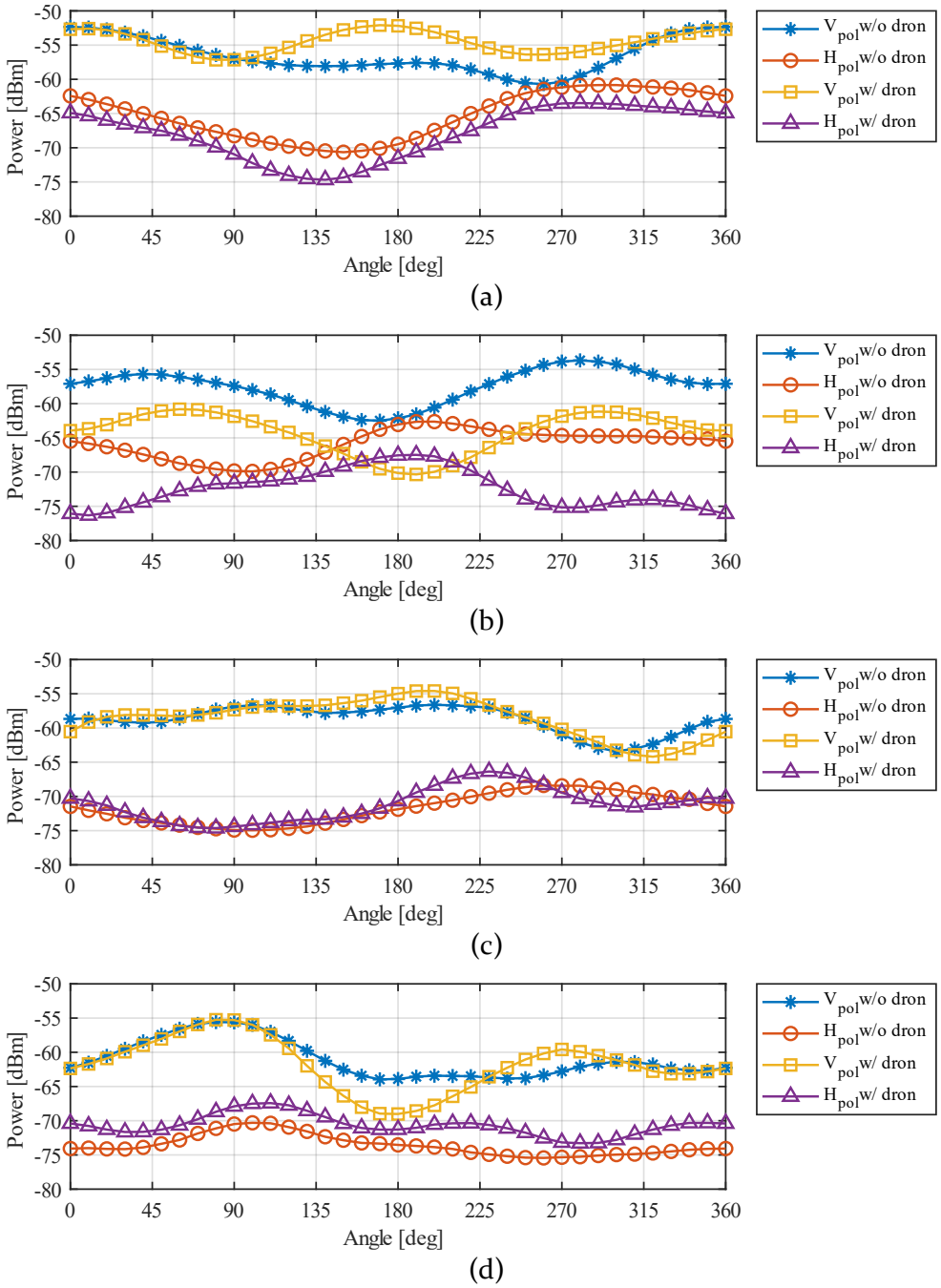


Figure 3-18. Radiation pattern of WiTi with (w/) and without (w/o) UAV (drone): (a) 2.4 GHz band @ X Plane; (b) 2.4 GHz band @ Y Plane; (c) 5 GHz band @ X Plane; (d) 5 GHz @ Y Plane. (Figure 16, p. 18, [124]).

3.2.3 Conclusions

The importance of FANETs is determined by the communication capacity of the nodes that compose it. As we have seen, there is a large number of works that study the use of UAVs as new working tools, modeling the communication channel, or proposing new routing protocols that better adapt to the characteristics and needs of FANETs. However, not considering the influence that the UAV can have on the captured/received power can lead to serious problems such as non-validation of results, poor performances, etc. Due to this lack of works that analyze the previous

mentioned, in this study, we examined the influence that a UAV has on the radiation pattern offered by an integrated WiFi node in the 2.4GHz and 5GHz band in a controlled environment.

For this purpose, using an anechoic chamber during all the experiments, we first obtained the best antenna configuration to suit the needs of the U2U and U2G links in the 5GHz and 2.4GHz bands, respectively. Then, the influence that the UAV can have on the radiation pattern was obtained, transforming the results into an expression that allows modeling the UAV as additional propagation losses. The results showed that the influence is mainly due to the carbon fiber chassis, being the influence of the engines and propellers negligible. Finally, having obtained the influence of the UAV on the radiation pattern, we demonstrated that the radiation pattern of a UAV on-board communication device can be approximated as an isotropic pattern if a correct antenna configuration, previously studied, is used, providing constant radiation within the regions of interest.

3.3 A Comparative Performance Evaluation of Routing Protocols for Flying Ad-Hoc Networks in Real Conditions

A second work performed on FANETs analyzed the throughput and packet loss of several routing protocols in a real deployment of a FANET. In this way, it was analyzed which protocols offered higher performance and thus demonstrate that routing protocols should be selected according to the application to be deployed, before the deployment of the network.

As seen in the previous sections, a large number of routing protocols have been proposed/adapted for FANETs. Each of these routing protocols could be applied in each of the FANET use cases, but because of the broad spectrum of applications that FANETs can perform, routing protocols deliver different performances in different scenarios.

Due to this, in this section we analyzed and compared the performance offered by different routing protocols (Babel, BATMAN-ADV, and OLSR) in terms of throughput and TCP packet loss in a real deployment composed of multiple UAVs, employing the 802.11 WiFi communication standard in the 2.4 and 5 GHz bands.

3.3.1 The importance of real experimental studies

Before developing the experiment, we analyze the proposed works that discuss the performance of various routing protocols in FANETs through a real deployment. It is a challenge to find papers that involve real deployments of FANETs, because of the high cost and complexity of establishing large-scale networks with variable topologies, in addition to the difficulties related to the repeatability of scenarios. That is why we consider this study important since the study through a real and

controlled deployment of a FANET can offer better results, as well as guarantee the performance offered by the proposed protocols/algorithms. By virtue of the veracity offered by the tests in a real deployment of a FANET, we consider it important to expose other previous works to help the reader understand the importance of analyzing the performance of the proposed routing protocols in a deployed FANET.

Among these works is the one carried out by Rosati *et al.* in [262]. The authors compared the performance offered by P-OLSR and OLRS routing protocols in a FANET composed of two small FW-UAVs. The results they obtained showed the advantage offered by P-OLSR, obtaining a higher average throughput, in addition to reducing the number of communication interruptions. This advantage was due to P-OLSR uses the GPS information of the nodes to decide which nodes will be closer to the destination node, thus reducing latencies and increasing throughput.

Moreover, the performance of P-OLSR was also compared with OLSR and Babel by Rosati *et al.* in [196], in a FANET composed of up to three UAVs with high mobility. The results obtained by the authors showed that P-OLSR offered higher throughput and fewer outages than OLSR and Babel. This may be due to the characteristic of the deployed network, where, for high mobility, routing protocols based on the geographical position of the nodes (such as P-OLSR) fit much better than the other tested protocols, which do not consider the mobility of the nodes.

In the work proposed by Lee *et al.* [263], a routing protocol based on the geographical position of the nodes, called Ground Control System-Routing (GCS-R), was developed. This routing protocol was compared with OLSR and DSDV in terms of throughput, stability, and outage time in a FANET network composed of up to six UAVs in a network coverage extension scenario. The results showed a clear advantage of GCS-R over the other tested protocols, however, we considered that this algorithm should have been compared with other algorithms of the same class, such as P-OLSR. Moreover, GCS-R was a centralized algorithm, so if the node in charge of routing was no longer available (something very frequent in FANETs) the routing would stop, and it could also present scalability problems.

Another work that analyzed the performance offered by a FANET is found in [153], where the BATMAN-ADV routing protocol was analyzed in a network composed of two RW-UAVs. The performances offered guaranteed the optimal performance for a coverage extension scenario, in addition to providing a comparison in terms of energy consumption between the Access Point (AP) mode of operation and the ad-hoc mode, with the AP mode offering lower energy consumption.

On the other hand, the work proposed by Kaysina *et al.* [264] showed the performance that a FANET using BATMAN as routing protocol can offer in terms

of coverage and throughput. The results obtained in a network composed of a UAV in three different scenarios showed that the maximum communication distance between two nodes, without packet loss, was 117m.

Finally, Maxa *et al.* showed in [239] a new routing protocol for FANETs with special attention to security, called SUAP. The scenario they deployed consisted of three UAVs and the performance they achieved demonstrated the security that such robust routing protocols can provide. Table 3-7 shows a summary of the papers seen in this section, as well as the number of nodes used in the deployment and the routing protocols used in each job.

TABLE 3-7. Related works involving an actual FANET deployment. The routing protocols used and the number of UAVs are also detailed. (Table 1, p. 3, [214]).

Work.	Routing Protocols	Number of UAVs
Rosati *et al.* [262]	OLSR, P-OLSR	2
Rosati *et al.* [196]	Babel, OLSR, P-OLSR	3
Guillen *et al.* [153]	BATMAN-ADV	2
Lee *et al.* [263]	GCS-R, DSDV, OLSR	6
Kaysina *et al.* [264]	BATMAN	1
Maxa *et al.* [239]	SUAP	2

3.3.2 Experimental setup

To test the performance offered by the different routing protocols selected in a FANET, it was decided to perform the following experiment. The scenario was composed of five nodes, two of them RW-UAV, which were placed 10 meters above the ground. This scenario could correspond to an application of coverage extension over ground nodes on an ad hoc basis due to an overload of the traditional communication network. For coverage extension, each UAV was composed of a communications module (WiTi [260]) capable of deploying a WiFi network in the 2.4GHz and 5GHz bands. Specifically, the IEEE 802.11g standard was used for the 2.4GHz band and the IEEE 802.11a standard for the 5GHz band, using 2 antennas for each band. The antenna configuration and positioning were based on the results obtained in previous work [124].

The deployed scenario can be seen in Figure 3-19, where in the terminal PCs (PC1 and PC2) is where the iperf3 [261] session was executed (one as server and the other as client) to analyze the performance of the entire end-to-end FANET, composed by all the nodes. Finally, there was an intermediate node acting as a communication relay, to provide more complexity to the deployed FANET, acting as an intermediate communication node when distances between nodes required its use. The iperf3 tool was used to obtain the performance achieved by the different routing algorithms, analyzing throughput and packet losses.

More specifically, the selected routing protocols belong to the proactive routing protocol group, namely OLSR, BATMAN-ADV, and Babel. The choice of

these routing protocols has been based on criteria of "state of the art", working experience with them, suitability to the proposed scenario, and compatibility with the communication module.

The terminal PCs (PC1 and PC2) were placed below the UAVs, pretending to be nodes whose coverage is provided by the UAV to which they are connected. UAV1 and PC1 remained static throughout the test, plus the relay node was located 35 m away from PC1 (UAV1).

On the other hand, the motion pattern followed by UAV2 (and PC2) consisted of a linear motion, moving away from UAV1 in 10 m steps. At each step, throughput and packet loss measurements were performed between PC1 and PC2, passing through the UAVs, and through the relay node if the distance between the UAVs required it. The measurements had a duration of 60 seconds. A representation of the deployed scenario can be seen in Figure 3-19.

3.3.3 Results

Once the experiments have been performed, in this section we will show and analyze the results obtained. The throughput results for the 2.4GHz band and the 5GHz band are shown in Table 3-8 and Figure 3-20. Also, the packet loss results are shown in Table 3-9 and Figure 3-21.

Based on the results shown, we were able to conclude that Babel and OLSR protocols provided higher throughput in both the 2.4 GHz and 5 GHz bands. Furthermore, these results demonstrated the inversely proportional relationship between throughput and packet losses.

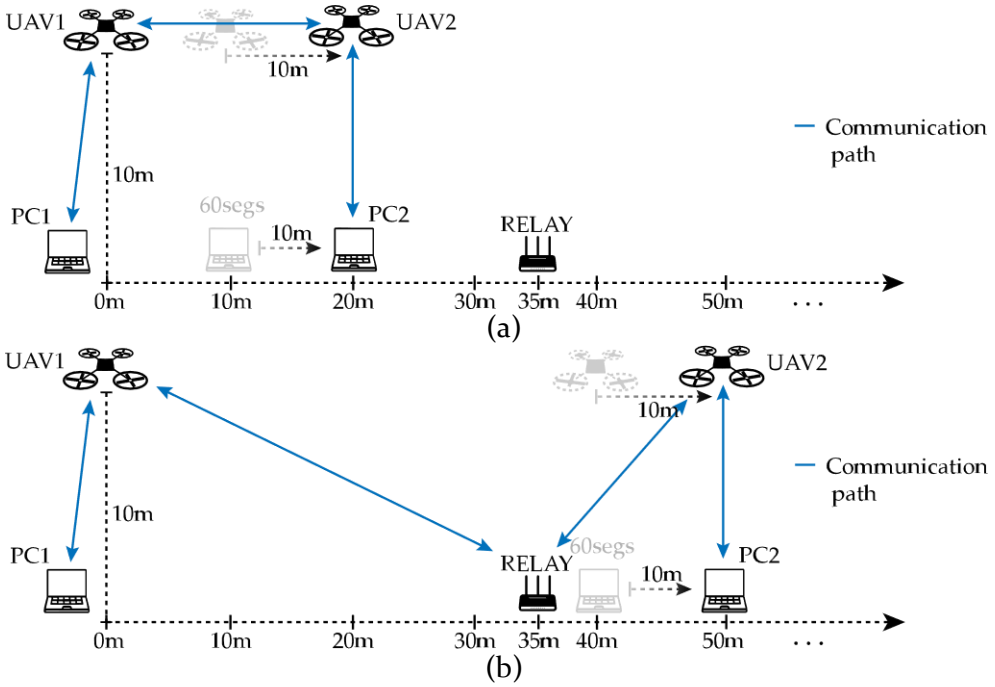


Figure 3-19. Real FANET test bench with relay node: (a) Communication path before 35-meter mark; (b) Communication path after the 35-meter mark. (Figure 2, p. 9, [214]).

On the other hand, we can observe a notable difference in the performance of the results between the 2.4GHz band and the 5GHz band. This is because the 5GHz band has a higher propagation loss constant than the 2.4GHz band, which implies a shorter communication distance. This means that, for example, in the measurement points located at 30 and 40 meters, the 2.4GHz band continues to establish the communication link between the UAVs directly (since there is no change in the trend of the results), however, for the 5GHz band there is a clear change in the trend of the results. This trend change is due to the modification of the communication path between the UAVs to include the relay node at the second measurement point (40 meters) resulting in a UAV-relay-UAV communication.

Analyzing the results obtained in more detail, we can point out that for the 2.4GHz band, the routing protocols presented significantly different performances. Specifically, OLSR uses the communication with the relay node between the 40-meter and 50-meter points, something that we can obtain from the change of trend in the results. However, BATMAN-ADV and Babel do not switch to this communication through the relay node until the measurement point at 70 meters. This may be due to differences in the periodicity of updating the routing tables, with OLSR having the shortest update period.

Lastly, in view of the results shown, it can be concluded that Babel and OLSR protocols present a better performance in terms of throughput and packet loss than those offered by BATMAN-ADV in the scenario described. Therefore, for the coverage extension scenario, these two routing protocols are preferable. Finally, between Babel and OLSR, although the results are very similar, Babel presented more stable results, with less variance, in both frequency bands.

TABLE 3-8. Mean throughput (Mbps) between PCs-UAVs with various gaps for OLSR, BATMAN-ADV, and Babel, for 2.4 and 5 GHz bands. Each measurement had a duration of 60 s. (Table 3, p. 10, [214]).

Routing Protocol	Frequency Band	Distance						
		10 m	20 m	30 m	40 m	50 m	60 m	70 m
OLSR	2.4 GHz	1.01	1.06	1.06	0.76	0.92	0.85	0.00
	5 GHz	0.99	0.74	0.52	0.98	0.81	0.00	0.00
BATMAN-ADV	2.4 GHz	1.06	1.04	0.91	0.83	0.52	0.26	0.76
	5 GHz	0.88	0.15	0.04	0.83	0.59	0.00	0.00
Babel	2.4 GHz	1.06	1.06	1.06	1.04	0.94	0.66	0.85
	5 GHz	0.95	0.65	0.48	0.89	0.72	0.00	0.00

TABLE 3-9. Mean packet loss (%) between PCs-UAVs with various gaps for OLSR, BATMAN-ADV, and Babel, for 2.4 and 5 GHz bands. Each measurement had a duration of 60 s. (Table 4, p. 10, [214]).

Routing Protocol	Frequency Band	Distance						
		10 m	20 m	30 m	40 m	50 m	60 m	70 m
OLSR	2.4 GHz	3.3%	0%	0%	27%	11%	19%	100%
	5 GHz	6.6%	30%	44%	6%	20.4%	100%	100%
BATMAN-ADV	2.4 GHz	0%	0%	13%	22.1%	44%	73%	27%
	5 GHz	14%	84%	97%	21.3%	43%	100%	100%
Babel	2.4 GHz	0%	0%	0%	0%	10%	35%	18%
	5 GHz	8%	36%	45.9%	15%	29.7%	100%	100%

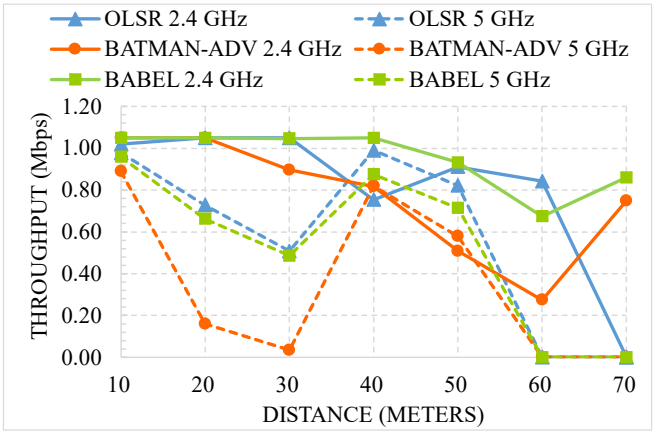


Figure 3-20. Mean throughput (Mbps) vs. distance (meters) (solid line for 2.4GHz and dashed line for 5GHz). (Figure 3, p. 10, [214]).

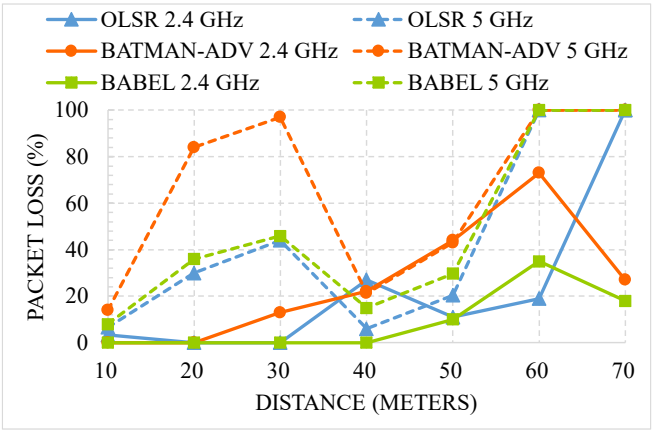


Figure 3-21. Mean packet loss (%) vs. distance (meters) (solid line @ 2.4GHz and dashed line @ 5GHz). (Figure 4, p. 10, [214]).

3.3.4 Conclusions

Due to the constantly changing network topology of FANETs, the correct selection of the routing protocol used is crucial to ensure the success of the deployment, as well as to obtain the best possible network performance.

Because of its significance, we compare the performance of three distinct proactive routing protocols (OLSR, BATMAN-ADV, and Babel) in terms of

throughput and packet loss in this study using an actual FANET implementation. The scenario consisted of five nodes, two of which were RW-UAVs, and it simulated a coverage extension scenario in an emergency or search and rescue situation. The results revealed that Babel outperformed OLSR and BATMAN-ADV in the criteria under consideration.

Additionally, this study corroborated the importance of selecting the appropriate routing protocol for FANETs, depending on the application to be deployed, to obtain the best possible performance. We feel that the routing protocol chosen will be incredibly essential in a few years as this sort of network becomes more ubiquitous in our daily lives and is used to a wide range of applications.

3.4 Conclusions to this chapter

Given the enormous importance that FANETs could offer with the advent of new technologies and communication standards such as 6G for intelligent transportation systems, Smart Cities, and for the Internet of Things (IoT) with applications such as remote monitoring of electronic systems, traffic surveillance, remote control of intelligent services, remote and autonomous sensing, or the extension of point coverage, the correct use of FANETs can offer a great advantage over traditional communication systems. Thus, the benefits that FANETs could have could be immeasurable, such as the reduction of development and deployment costs of new networks, the enormous speed of deployment, the low latency of communications, in applications for extending coverage in situations of natural disasters (earthquakes, tsunamis, fires, etc.), search and rescue, agriculture, intelligent transportation systems, etc.

This chapter reviews the state of the art related to mobility models, positioning, propagation in FANETs, and routing algorithms. Although we have detected a wide variety of contributions within this novel field, further work by the research community is still needed to address the challenges intrinsic to this new type of flying network.

The communication capability of each UAV node in a FANET is limited by the communication device integrated into each UAV. The influence that the UAV can have on the radiation pattern of the communication device is a complex and an open issue, as it depends on a large number of factors (UAV shape, communication technology, antenna types, application, etc.). However, we consider it important to take this issue into account during the network planning process. Therefore, a study of the impact of a UAV on the radiation pattern was conducted in a fully controlled environment. The results demonstrated the significance of this type of analysis, concluding that the influence of the UAV chassis resulted in a decrease in transmitted power of about 10 dB for the UAV-to-

Ground link when compared to when there is no UAV, implying that the power dropped to one-tenth of its original value.

Finally, the last section addresses the question of how the various routing protocols affect the service offered by FANETs. Routing protocols are particularly important, especially in large networks, where the amount of data being sent is large and therefore very important for the proper functioning of the services. Therefore, the choice of the appropriate protocol depends on several factors, such as the context, the application, and the requirements of the service users. Hence, in the second part of this section, we tested and contrasted the performance that three proactive routing protocols for FANETs can offer, presenting the results in terms of throughput and packet losses. The results showed that it is of vital importance to select the routing protocol prior to network deployment, to achieve optimal performance of the entire network.

3.5 Publications associated with this research

The works related to this chapter are as follows:

Articles:

Guillen-Perez, A.; Cano, M.-D., "Flying ad hoc networks: A new domain for network communications," *Sensors,* vol. 18, no. 10, p. 3571, Oct 2018, doi:10.3390/s18103571

2018 Journal Impact Factor (JIF): 3.031. (Q1), Rank: 15/61 in Instrument & Instrumentation.

Guillen-Perez, A.; Montoya, A-M; Sanchez-Aarnoutse, J. C.; Cano, M.-D., "A comparative performance evaluation of routing protocols for flying ad-hoc networks in real conditions," *Appl. Sci.*, vol. 11, no. 10, p. 4363, May 2021, doi: 10.3390/app11104363

2020 Journal Impact Factor (JIF): 2.679. (Q2), Rank: 38/90 in Engineering Multidisciplinary.

Congress:

Guillen-Perez, A.; Sanchez-Iborra, R.; Cano, M.-D., Sanchez-Aarnoutse, J. C.; and Garcia-Haro, J., "WiFi networks on drones," in *2016 ITU Kaleidoscope: ICTs for a Sustainable World (ITU WT)*, Nov. 2016, pp. 1–8, doi: 10.1109/ITU-WT.2016.7805730.

Guillen-Perez, A.; Cano, M.-D., "Comunicaciones Inalámbricas con Vehículos Aéreos no Tripulados", *I Jornadas Doctorales UPCT*, Universidad Politécnica de Cartagena. 2018. Oral communication.

Chapter 4: Smart Cities and Pedestrians

4.1 Introduction

Pedestrians will play a crucial role in the design of intelligent transportation systems and future smart cities. Estimating the number of users in an area offers countless advantages for optimizing a large number of systems: traffic light timing, ambient temperature, urban planning, etc. Table 4-1 summarizes a set of applications where the estimation of the number of users can offer a great advantage.

TABLE 4-1. Crowd estimation area and applications. (Table 1, p. 2, [265]).

Area	Example
Energy efficiency	Adjustment of indoor climate control (supermarkets, offices, etc.) according to the estimated number of people inside, etc.
Urban planning and transportation	Pedestrian patterns, passenger flows, customized traffic planning, etc.
Security	Crowd control (concerts, sporting events, street events, etc.), surveillance, search, rescue, etc.
Better user satisfaction (QoS, QoE, UX)	Improved design services in public spaces such as airports, hospitals, amusement parks, parks, museums, libraries, etc.
Marketing and retail	Traffic patterns in stores (people tracking) or shopping malls (crowd tracking), number of cash registers, etc.

Due to this importance, many methods capable of estimating the number of users both indoors and outdoors have been investigated. So far, user counting or estimation has been based on image processing techniques [266] on the use of radiofrequency (RF) or ultrasound sensors [267] or the use of techniques based on radio frequency signals, such as Bluetooth or WiFi [268]–[270].

If we focus on the use of sensors, its main disadvantage is that it requires a wide deployment over the whole area to be monitored, however, once all the sensors are deployed, the results can outperform some image processing techniques [271].

As for image processing techniques, they have attracted the most attention from researchers, and numerous approaches based on this type of technique can be found to count the number of people. The most common approaches are based on background subtraction to improve person detection, regression-based techniques such as AdaBoost, or CNN classifiers. Despite the accuracy that these types of techniques can offer, they also have several drawbacks and limitations. The main one is privacy and data protection when dealing directly with images. Other limitations are given by situations in which people present similar patterns

to the background, reflections, shadows, or hidden areas, overlapping between people, or extreme far/close distance of people to the sensor. In addition, this type of technique presents a higher computational cost of processing, both during the training of the detection systems and in the inference.

On the other hand, many approaches make use of RF signals emitted by devices carried by users for counting or estimating density in a specific area. The advantages that stand out of this approach are that RF signals can be unaffected by non-conductive obstacles (walls, shelves, trees, etc.), in addition to being able to operate over longer distances and requiring minimal deployment [272]. Within this category, approaches using wireless sensor networks [273], [274], Bluetooth/WiFi [268], [270], cellular [275] are distinguished.

RF-based mechanisms rely on information captured from signals at the physical layer, such as Received Signal Strength Indicator (RSSI), Channel State Information (CSI), Channel Quality Indicator (CQI), or Channel Frequency Response (CFR) (see Table 4-2).

RF-based category can in turn be divided into two subclasses: device-free (see Figure 4-1 a) or device-based (see Figure 4-1 b). While in the first approach, the proposed techniques assume that people leave a signature (fingerprint) on the transmitted signals. The human body absorbs and reflects part of the signal, i.e., it becomes an obstacle and an antenna at the same time, affecting LOS, propagation losses, and increasing the multipath effect. A paper where proposals for device-free activity recognition are analyzed can be found in [269].

TABLE 4-2. Physical Layer Information parameters usually used. (Table 2, p. 2, [265]).

Acronym	Parameter	Description
RSS	Received Signal Strength	The received signal strength in dBm of a received data frame or a beacon is measured at the receiver's antenna.
CSI	Channel State Indicator	Channel measurements depict the amplitudes (signal strength) and phases of every subcarrier.
CQI	Channel Quality Indicator	Current communication channel quality as measured by user equipment in cellular technologies.
CIR	Channel Impulse Response	Temporal linear filter that models the wireless propagation channel.
CFR	Channel Frequency Response	Discrete Fourier Transformation (DFT) of the Channel Impulse Response.

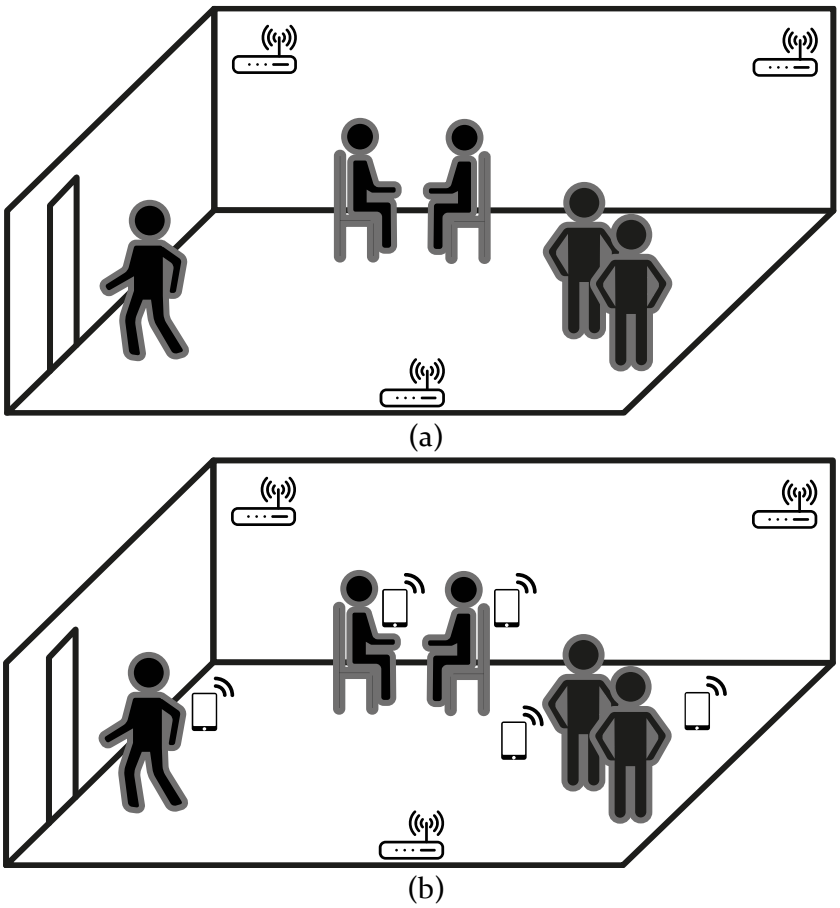


Figure 4-1. RF-based approaches: (a) Device-free group. At least one WiFi Tx and one WiFi Rx are required; then the effect of people on the received signal is processed and associated to the number of people inside; (b) Device-based group. Each person must carry at least one WiFi device from which messages are collected. (Figure 1, p. 3, [265]).

On the other hand, device-based mechanisms focus on using the signals generated by users' mobile devices. In turn, within this approach, we can find two methods of counting users: active and passive. Active methods are those that require the action/response of the mobile device to be counted, and passive methods are those that obtain an estimate of the number of people only by analyzing the Probe Request messages sent periodically by the mobile devices to be counted.

Given that everyone carries a cell phone, that WiFi connectivity is now offered on a large scale, and that cell phones periodically send signals to discover WiFi Access Points (APs) to connect to, it is intuitive to think of a method that uses WiFi signals to distinguish devices and estimate the number of people. Note that the terms mobile device, smartphone, or cell phone will be used interchangeably throughout this paper to refer to a personal mobile communication device that includes a WiFi communication module.

The challenges faced by passive WiFi-based methods are numerous, such as not considering that when a person is not carrying a mobile device or has WiFi

turned off (e.g., for security reasons or to save battery power), there could be a person with one or more WiFi devices. In addition, the use of the radio communication medium involves working with non-deterministic factors that affect electromagnetic signals in an uncontrollable way, such as signal fading, multipath propagation, noise, or distortion. Finally, and most importantly, randomization of the most used identifier for device identification and discrimination, the MAC address, is being encouraged during WiFi network discovery. Specifically, to increase user security and privacy, since iOS 8, Android 6, and Windows 10, device MAC addresses follow a randomization policy. This randomization policy depends on the manufacturer of the communication chip and the state and characteristics of the mobile device. However, the cell phone has indeed become a device that accompanies us permanently and from which countless data about human activity can be extracted [276], [277].

Even though there are works that study this randomization [278]–[281], the study within this field requires more work by the scientific community to obtain a policy that allows the security and privacy of the user, but that allows obtaining an estimate of the number of mobile devices in a room for its correct optimization and use.

4.2 Related works

In this section, we will review the state of the art of the proposed works to estimate, track or count people based on the number of mobile devices using WiFi technology.

4.2.1 Device-Free approach

The device-free approach (see Figure 4-1 a) consists mainly of measuring the influence that a user has on an RF signal. That is, the human body acts as both an obstacle and an antenna, absorbing some of the RF energy and transmitting some. This phenomenon means that each person leaves a trace, or fingerprint, on the propagation of the RF waves. The objective of these methods is based on transmitting a set of RF waves and estimating the number of users based on the behavior of the RF waves received. To study the behavior, many statistical parameters have been studied.

The simplest is the RSS measurement, i.e., the measurement of the power at which a receiver captures RF signals [269], [272], [282]–[289]. It is logical to think that when there are no users in a given area the received signal power will be higher than when there are users. However, RSS-based methods have a major drawback and that is the large variability of the RSS due to interference, noise, multipath, etc. [268]. In addition, other works showed that there is a trade-off value in the chosen frequency in the RF signal, since the higher the frequency, the more

accurate the results can be, but they will have a shorter range, in addition to very high frequencies are greatly affected by obstacles, as well as human absorption [290].

Another widely used parameter is the CSI. The CSI is able to provide higher granularity in the results, as it can separate the received signals into multipath components [291]. However, this parameter has a major drawback and that is that not all commercial WiFi communication cards offer this parameter [292], [293].

4.2.2 Device-Based approach

Within the device-based approach (see Figure 4-1 b) we find the passive methods. These methods are the least intrusive to the user, as they do not force the device to send any messages, but only capture messages sent periodically by mobile devices. These messages are usually Probe Requests sent by mobile devices to discover WiFi networks to connect to. These messages are sent periodically with a sending frequency of 10 Hz (although this frequency may vary depending on the manufacturer, device status, etc. [294]–[296]). A breakdown of the fields included in the Probe Request message can be found in Figure 4-2.

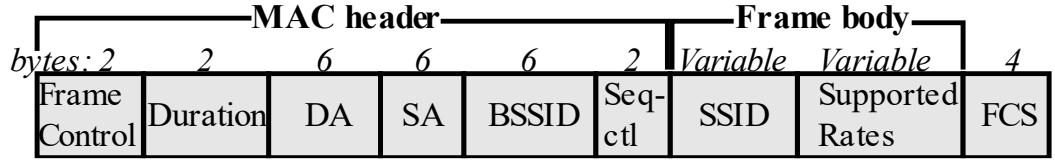


Figure 4-2. Probe Request frame. DA ≡ Destination Address; SA ≡ Source Address (MAC); BSSID ≡ SSID of destination Access Point; Other client's parameters such as supported rates, extended rates, sequence number, fragmentation number, etc. (Figure 2, p. 2, [297]).

As can be seen, within the Probe Request field, the address of the device sending the message, i.e., the MAC address, is sent unencrypted. This MAC address is unique for each device and is used to differentiate between devices. As mentioned above, this MAC address from iOS 8, Android 6, and Windows 10 are recommended to be sent in a randomized manner. However, this randomization depends on the manufacturer, device state, configuration, etc. In addition, numerous studies study this randomization concluding that there are techniques that allow device tracking using the sequence number sent in the frames as well as the MAC [276], [277].

The main works we have found on passive WiFi-based methods are located in indoor environments [268], [269], [298]–[302], achieving accuracy values around 75% in the best case. We have only found a few works performing outdoor experiments [270], [295], [303]–[305]. In this part, we will review the proposed outdoor works, which is the focus of this part of the review.

The work developed by Petre *et al.* [295] consisted of studying pedestrian behavior in an open environment using an algorithm based on passive WiFi. They captured the Probe Request messages sent by mobile devices, and by analyzing the RSS and MAC, they differentiated between static devices and pedestrians, thus obtaining an estimate of the number of pedestrians. In addition, they analyzed the sending behavior of Probe Request messages by various devices, obtaining an average sending period of 3 seconds. The results obtained showed that information on pedestrian behavior (flow) could be obtained with high accuracy, as well as an estimation of user density from the study of the captured Probe Request messages. Undoubtedly, these findings could greatly facilitate the work of urban planners, event organizers, etc., as well as improve the performance of many other systems such as elevators, air conditioning temperature, security, etc.

Another similar work using Probe Request messages can be found in [303]. In this work, they deployed a large number of low-cost capture sensors to analyze large-scale pedestrian flow. The captured messages were sent to a cloud server and processed there. Using large-scale deployment, the obtained results were able to provide a lot of information, such as density in a city, peak/valley hours of pedestrians, flow distribution by time slots, etc., corroborating the importance of passive WiFi-based methods.

An interesting work was proposed by Acuña *et al.* [304]. In this work, they made use of the capture of Probe Request messages to locate people in disaster/emergency scenarios. For this purpose, they proposed the use of a UAV integrating a WiFi receiver, capable of capturing Probe Request messages from the mobile devices of the users to be searched/rescued. In this way, the search process could be accelerated, increasing the chances of surviving emergency situations.

Following a similar approach to the previous work, another algorithm for pedestrian flow estimation based on Probe Request messages was proposed in [305]. However, in this case, they used a Viterbi algorithm to estimate the most probable route followed by pedestrians based on the captures made by the sensors. The results obtained conclude that passive WiFi-based methods are an excellent alternative to consider in large-scale deployments for the study of urban mobility among others, both pedestrians and vehicles, thanks to their low cost of both deployment and maintenance.

From all the works analyzed, we can conclude that WiFi-based passive device-based methods have great applicability in a large number of work areas, and are very promising, being able to offer great information to numerous systems. Moreover, these approaches offer great simplicity, involving low development, operation, and processing cost, which gives them a great advantage over other camera-based methods, or other methods. However, more research should be done in outdoor environments, as well as trying to solve the problem of MAC

randomness by using other messages that can still offer privacy to users, but from which the same information as that provided by Probe Request messages can be obtained.

4.3 A WiFi-based method to count and locate pedestrians in urban traffic scenarios

4.3.1 Introduction

Estimating the number of pedestrians in an open environment such as an intersection is not trivial. However, obtaining this value can provide great information to traffic light control systems, which already consider vehicles but not pedestrians.

Therefore, in this research, we proposed an algorithm that uses a passive method based on WiFi to estimate the number of pedestrians in an urban environment consisting of a traffic-light intersection. With the proposed algorithm it was possible to distinguish between pedestrians walking and those waiting to cross, being the latter the important ones to be considered for the traffic light control system. In addition, pedestrians identified as waiting to cross were able to be positioned by the algorithm at the corner of the intersection where they are waiting to cross.

In this way, the proposed algorithm laid the groundwork for other intelligent traffic control algorithms to take pedestrians into account, as they are the most vulnerable and neglected group in these control systems.

4.3.2 Experimental setup

To test the development of the proposed algorithm, the OMNET++ simulator [306] along with the INET framework [307] was used. The simulation scenario consisted of an urban arterial consisting of two intersections $i \in \{1, 2\}$ (See Figure 4-3). By using two intersections, it was possible to simulate a wide range of behaviors: pedestrians waiting to cross, crossing, walking from one intersection to the other, etc. The simulated scenario can be seen in Figure 4-3, as well as the dimensions used.

In the simulations, each pedestrian was simulated as a mobile device configured as *AdhocHost,* and the devices in charge of capturing the Probe Request messages from the mobiles were configured as *WirelessAPWithSink*. These capture devices were called Data Acquisition Units (DAUs) simulating being at the traffic lights of an intersection, therefore, they were at a height of 6 meters above the ground. There were four DAUs per intersection and were denoted as DAUij, $i \in \{1, 2\}$ and $j \in \{1, 2, 3, 4\}$. The rest of the features used to set up the simulator can be found in Table 4-3.

In order to evaluate the performance of the proposed algorithm, four different sets of experiments were proposed. First, to demonstrate whether there was an influence between the number of pedestrians and the sending of Probe Request messages, an experiment was conducted in which the number of simulated pedestrians was doubled. Next, a set of experiments was performed to adjust the internal parameters of the proposed algorithm. Finally, a last set of experiments was performed in which the performance of the proposed algorithm was evaluated against a large set of pedestrian behaviors. In addition, the impact of the frequency of sending Probe Request messages on the accuracy of the proposed algorithm was studied.

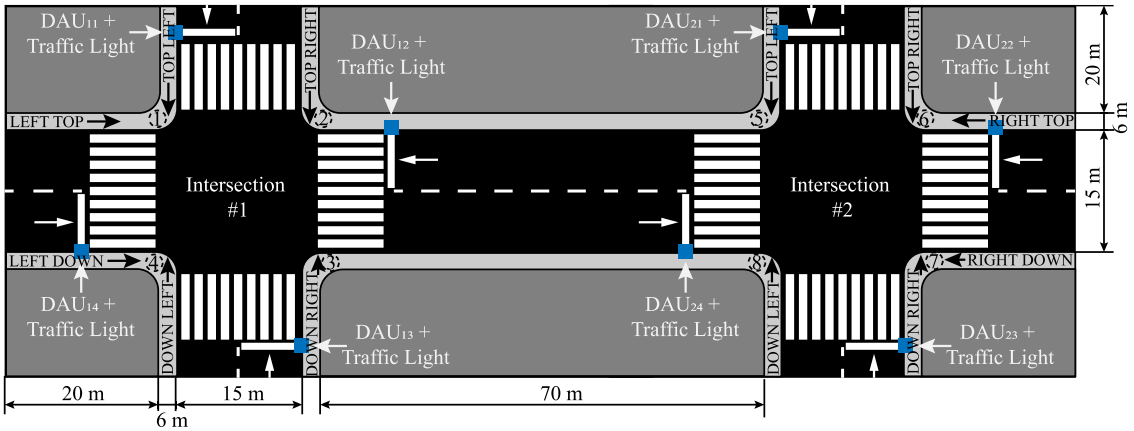


Figure 4-3. Simulated scenario. (Figure 7, p. 5, [297]).

TABLE 4-3. Simulation parameters employed in OMNET++. (Table 1, p. 5, [297]).

Parameters	Value
Simulation tool / Framework	OMNetT++ / INET
Version	5.2.1 / 4.0
Ground-type	Flatground
Obstacle loss	Dielectric Obstacle Loss
Propagation loss	Rayleigh Fading
DAU height	6 m
Mobile devices height	1.5 m
Mobile devices speed	1.39 m/s
Number of mobile nodes	64
Probe Request Period	2 secs
Transmission power	13 dBm
Reception sensibility	-120 dBm
Probability distribution to send Probe Request frames	Normal distribution with variable mean and variances

4.3.3 Our proposal

The proposed algorithm for estimating the number of pedestrians in an urban environment composed of an intersection from Probe Request message captures is detailed below.

To classify pedestrians as static or moving, during each traffic light phase, Probe Request messages sent by mobile devices are captured, and just before each traffic light phase change (Red-Green or Green-Red), all messages are processed, analyzing the temporal behavior of each captured device along with the total time interval of data capture. In this way, the information on the estimation of the number of pedestrians would be available in the next cycle for advanced traffic control systems.

Prior to data processing, all DAUs of each traffic light composing the intersection send the encrypted Probe Request messages to a central DAU, in charge of decrypting the messages and processing all captured devices to obtain their behavior and the position of static pedestrians, waiting to cross. For example, all DAU1j $j \in \{2, 3, 4\}$ send the data to DAU11 of intersection i=1, which acts as the central DAU.

To discriminate between a moving pedestrian and a static pedestrian, waiting to cross, it was decided to place an RSSI threshold in the Probe Request messages. Thus, if a DAU picks up a device with a high received power variance, it means that the device is moving. Otherwise, if all DAUs, that capture a device, have a low received power variance over the entire time interval, then the device is considered static, waiting to cross. This power variance threshold situation is crucial for the correct operation of the algorithm.

Finally, the situation of static pedestrians, waiting to cross, is obtained by observing the DAUij with the highest received power during the capture interval. This is possible due to the large separation between the DAUij and the propagation losses characteristic of the 2.4GHz band.

Through empirical tests, it was observed that mitigating the random components of the RF channel by smoothing the power measurements improved the performance of the discrimination algorithm. For this purpose, a Gaussian filter with an adjustable standard deviation was used. Thus, with an optimal value of the Gaussian filter width, smoothing could mitigate variations due to rudder for static pedestrians, while it could maintain variations in power measurements in situations where the pedestrian was moving.

In summary, the proposed algorithm consists of:

1. Grouping the Probe Request messages sent by the devices to be analyzed in a central DAUij.
2. Obtaining the list of DAUs captured by each pedestrian p (DAUijp).
3. Perform a smoothing using a Gaussian filter of the power received by each DAUijp that captured p.
4. Differentiation between each device is performed using the MAC included in the Probe Request messages.
5. Discrimination between static and moving pedestrians is performed by the variance of the received power over the entire capture interval.
6. Positioning of static pedestrians is performed on the DAUij with the highest average power over the entire capture interval.

The pseudocode of the proposed method is represented in Algorithm 4-1

Algorithm 4-1: *Our Proposal*

The $list\_of\_pedestrians$ detected by all DAUij is obtained.
for pedestrian p *in* $list\_of\_pedestrians$ **do**:
    the $list\_of\_DAUij$ that have detected p is obtained.
    $static = True$ # by default the pedestrian is considered as static
    for $DAUij$ *in* $list\_of\_DAUij$ **do**:
        # The power measurements of the pedestrian captured by the $DAUij$ are
        # obtained. If this power measurement presents a high variance in one
        # $DAUij$ means that this pedestrian p is moving.
        # The power measurements are *smoothed* to reduce noise.
        if $var(smoothed) > var\_max$ **then**:
            $static = False$ # now the pedestrian is considered as moving
        end if
    end for
    if $static == True$ **then**:
        # The pedestrian p is considered as static in all of $list\_of\_DAUij$,
        # then its correct location corresponds to the $DAUij$ that presents
        # the higher power in average.
        $DAUij\_max = get\_DAUijp\_MaxMeanPower(p)$
        $DAUij\_max.add\_Pedestrian(p)$
    end if
end for

4.3.4 Experiments & Results

Experimental set #1

The first set of experiments obtained the possible influence on the behavior of probe request messages sent by mobile devices as a function of the number of mobile devices. For this, a simple scenario (an intersection) was used, and pedestrians followed a simple movement pattern. Figure 4-4 shows the simulated scenario and the movement pattern followed by the pedestrians (1-2-3). The movement pattern was as follows: between the time interval from 0 to 14.5 seconds, pedestrians approached the intersection from (1) and positioned themselves at (2).

Between the time interval of 14.5 and 44.5 seconds (in total, 30 seconds), pedestrians remained static at (2). Finally, at the time interval of 44.5 and 74.5 pedestrians cross the intersection at the crosswalk and move away towards (3) passing next to DAU12. The number of simulated pedestrians doubled in each simulation in an interval between 1 and 64 pedestrians.

The results of the behavior of the average received power of the Probe Request messages can be seen in Figure 4-5. Given the results, it can be stated that neither the number of messages sent by the mobile devices (there was a possibility of channel saturation) nor the average received power had any influence. It can be observed that, in all cases, the power received by the DAU followed the same behavior regardless of the number of people. To saturate the channel, there would have to be a huge number of devices trying to send messages at the same time on the same WiFi channel.

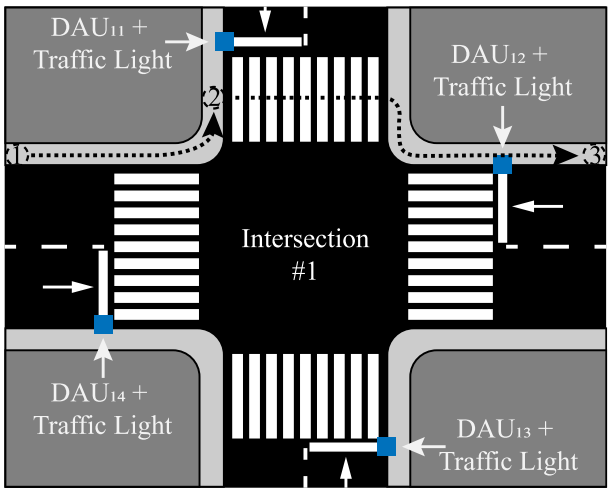


Figure 4-4. Simulated scenario for Experimental set #1 and mobility pattern followed by the simulated pedestrians. (Figure 3, p. 3, [297]).

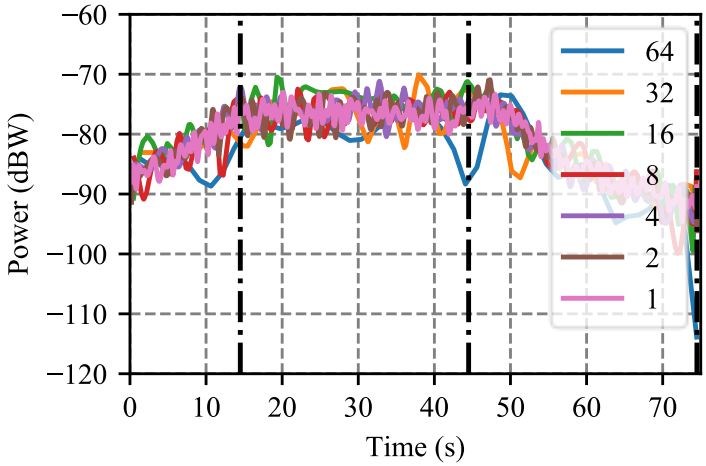


Figure 4-5. Received power in Probe Request frames captured by DAU11 with an incremental number of pedestrians from 1 to 64. (Figure 8, p. 5, [297]).

Experimental set #2

This set of experiments aimed to optimize the internal parameters of the proposed algorithm. That is, to find the optimal values of the variance threshold (*var_max*) and the optimal size of the Gaussian smoothing bell (*std_len*) that would maximize the accuracy of the proposed algorithm. In this way, the random components of the channel could be largely mitigated by the smoothing, in addition to being able to differentiate static and moving pedestrians thanks to the variance in the received power of the Probe Request messages during the entire capture interval.

For this purpose, a group of 64 pedestrians was simulated in the scenario shown in Figure 4-3 following a movement pattern starting at the left-hand scenario (*l*) and ending at the right-hand scenario (*r*), crossing both intersections and using the upper sidewalk (*t*).

After simulating the pedestrians, the optimization process of the *var_max* and *std_len* parameters was carried out using a mesh search to maximize the joint accuracy of the proposed algorithm.

The results of this mesh search optimization process are shown in Figure 4-6. From these results, we can obtain that the optimum values were 4.20 dB and 0.78 for *var_max* and *std_len* respectively.

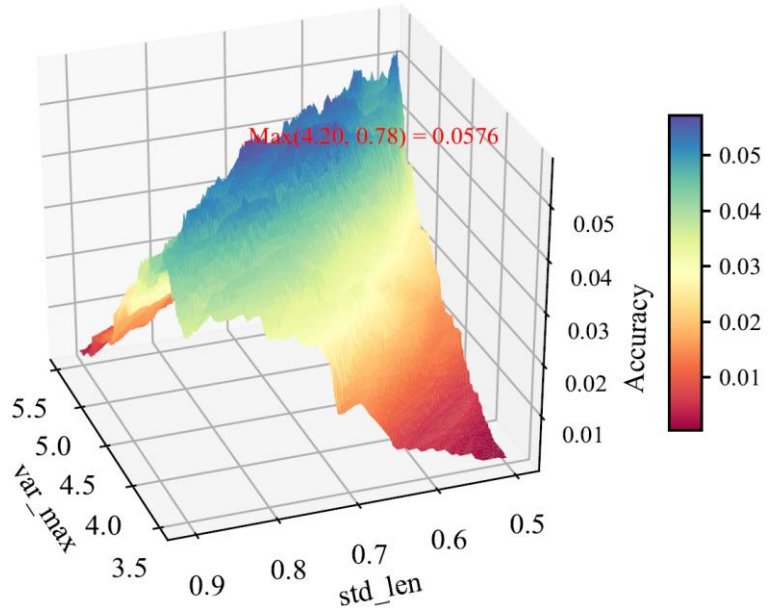


Figure 4-6. Accuracy versus *var_max* and *std_len*. In this case, the accuracy shown is the product of *moving* and *static* accuracies. (Figure 9, p. 6, [297]).

Experimental set #3

The third set of experiments was performed to obtain the overall accuracy of the proposed algorithm.

Before showing the results, we had to define a set of metrics that would allow us to get an idea of the overall performance of the proposed algorithm. In other words, since our algorithm presents different goals (defining the number of static pedestrians, the number of moving pedestrians, and the location of static pedestrians), it was necessary to define the following metrics: *Amoving* represents the detection accuracy of moving pedestrians; *Astatic* represents the detection accuracy of static pedestrians (waiting to cross); *Apositioning* represents the location accuracy of static pedestrians at the appropriate crosswalk (i.e., knowing their exact location under the DAUij where they are waiting). These accuracies are calculated as shown in equations (4-1), (4-2), and (4-3), respectively.

$$A_{moving} = \frac{pedestrians\_detected\_as\_moving}{\text{total\_number\_of\_pedestrians\_moving}} \quad (4\text{-}1)$$

$$A_{static} = \frac{pedestrians\_detected\_as\_waiting}{\text{total\_number\_of\_pedestrians\_waiting}} \quad (4\text{-}2)$$

$$A_{positioning} = \frac{pedestrians\_located\_properly}{pedestrians\_detected\_as\_waiting} \quad (4\text{-}3)$$

To obtain a large set of behaviors and a large data set, 12 groups of pedestrians with different movement behaviors were simulated in the scenario shown in Figure 4-3. Each pedestrian group consisted of 64 pedestrians (mobile devices). These movement behaviors and the associated denotation are detailed below:

1) From left *l* to right *r* on the top sidewalk *t* (crossing the two intersections), it was denoted as direction D*lrt*, and from left *l* to right *r* on the down sidewalk *d* (crossing the two intersections), it was denoted as direction D*lrd*.

2) From right to left on the top sidewalk (crossing the two intersections), it was denoted as direction D*rlt*, and from right to left on the down sidewalk (crossing the two intersections), it was denoted as direction D*rld*.

3) From top to down crossing only intersection i=*1* on the left sidewalk, denoted as direction D*tdl1*; ditto on the right sidewalk denoted as direction D*tdr1*.

4) From down to top traversing only the i=*1* intersection on the left sidewalk, denoted as direction D*dtl1*; ditto on the right sidewalk denoted as direction D*dtr1*.

5) From top to down crossing only intersection i=*2* on the left sidewalk, denoted as direction D*tdl2*; ditto on the right sidewalk denoted as direction D*tdr2*.

6) From down to top crossing only intersection i=*2* on the left sidewalk, denoted as direction D*dtl2*; ditto on the right sidewalk denoted as direction D*dtr2*.

To simplify the simulation, all pedestrians in each 30-second time interval were either static or moving, as shown in Table 4-4 and Table 4-5.

TABLE 4-4. Timestamps and movement states. (Table 2, p. 6, [297]).

Timestamp	Time Interval (s)	State
T0	0 - 14.5	Moving
T1	14.5 - 44.5	Static
T2	44.5 - 74.5	Moving
T3	74.5 - 104.5	Moving
T4	104.5 - 134.5	Moving
T5	134.5 - 164.5	Static

TABLE 4-5. Example of movements D*lrt* and D*tdl1* (D=direction, *l*=left, *r*=right, *t*=top, *d*=down, *1*=intersection 1, *2*=intersection 2). Spots (1), (2), (5), (6) refer to the points marked in Figure 4-3. (Table 3, p. 6, [297]).

Movement	Example	Temporary Mobility Pattern
Horizontal	Dlrt	0s - 14.5s: Left to (1) 14.5s-44.5s: Static in (1) 44.5s-104.5s: (1) to (5) 104.5s-134.5s: (5) to (6) 134.5s-164.5s: Static in (6)
Vertical	Dtdl1	0s - 14.5s: Top to (1) 14.5s-44.5s: Static in (1) 44.5s-74.5s: (1) to bottom 74.5s-134.5s: Bottom to top to (1) 134.5s-164.5s: Static in (1)

The results obtained are shown in Figure 4-7. In view of the results, we can see that, despite the simplicity of the proposed algorithm, we were able to successfully identify moving pedestrians with an accuracy of over 52%. On the other hand, we see that we were able to correctly classify more than 61% of static pedestrians. Finally, pedestrians detected as static were correctly classified at the appropriate location with an accuracy of over 93%.

Experimental set #4

The last set of experiments consisted of analyzing the influence of the Probe Request message sending period on the accuracy of the proposed algorithm. As a reminder, the cycle time of state change of the simulated traffic lights at the intersection, and thus the Probe Request message capture window, was 30 seconds.

For this purpose, several experiments were performed, each time doubling the sending period. The results are shown in Figure 4-8 and as expected, the longer the sending period, the lower the accuracy of the algorithm. Specifically, the accuracies for differentiating static from moving pedestrians remained above 51% when the sending period was equal to or less than 16 seconds. These results may be due to the relationship with the 30-second traffic light cycle.

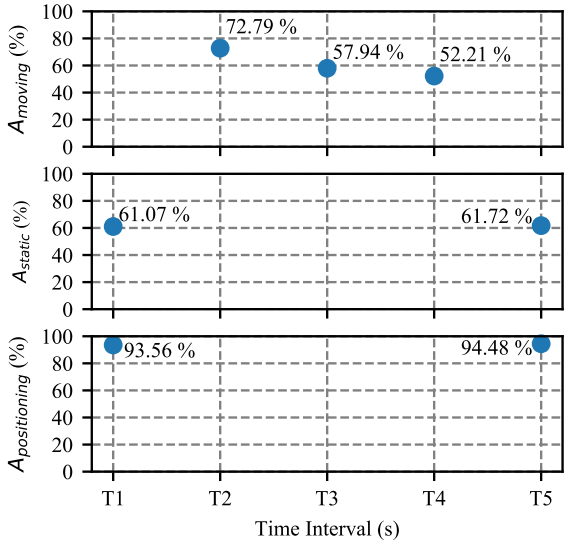


Figure 4-7. Accuracies obtained in the simulated scenario shown in Figure4-4 with 768 pedestrians moving in twelve different directions. At timestamps T2, T3, and T4 all pedestrians are moving, whereas at timestamps T1 and T5 all pedestrians are static. (Figure 10, p. 6, [297]).

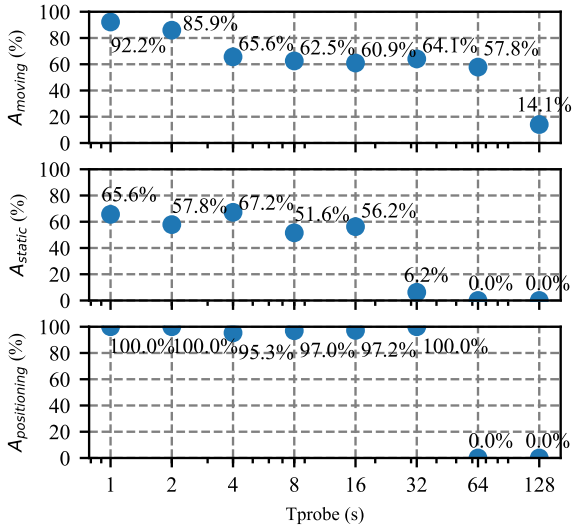


Figure 4-8. Accuracies versus Probe Request sending rate (*Tprobe* represents the time interval between two consecutive Probe Request frames) in the scenario described in Figure 4-4 for timestamps *T1* and *T2*. (Figure 11, p. 7, [297]).

As can be seen, when the sending period is greater than 30 seconds (e.g., 32 seconds) the accuracy decreases significantly because there will be a large number of devices that, by probability, have not sent any Probe Request message within the capture window. On the other hand, it can be seen how the positioning accuracy remains above 95% up to a sending period of 64 seconds.

4.3.5 Conclusions

The role that pedestrians can bring to future smart cities and intelligent traffic control systems is crucial to ensure their successful development. Despite the multitude of approaches to pedestrian detection, the use of WiFi is increasingly boosted by the breakthrough in communication systems and the fact that the cell phone has become an essential device. However, the practice has shown that there are still numerous challenges to be faced in order to achieve a fully reliable and accurate pedestrian detection and positioning system in an outdoor environment such as an urban intersection.

This research proposed a passive WiFi-based method capable of identifying the behavior of pedestrians, in an open environment such as an intersection, as static or moving, and was also able to obtain the position within the intersection of pedestrians detected as static. The results showed that the accuracy levels obtained were comparable, even superior in some cases to those obtained by previous work in indoor scenarios. Particularly noteworthy is the accuracy obtained by our method in the classification of moving pedestrians (higher than 52%), in the classification of static pedestrians (higher than 61%), and in obtaining the position within the intersection that static pedestrians are waiting to cross (higher than 93%).

4.4 Counting and locating people in outdoor environments: a comparative experimental study using WiFi-based passive methods

4.4.1 Introduction

In previous studies [297] we obtained an algorithm capable of classifying pedestrian behavior into static and moving, as well as obtaining the position of pedestrians in an outdoor urban environment such as an intersection. This algorithm was a passive WiFi-based method, which used Probe Request messages sent by mobile devices passively and periodically to find WiFi networks to connect to. To classify and position pedestrians, the RSSI was used by 4 DAUs located at each of the four traffic lights that make up an intersection. See Figure 4-4 for a picture of the proposed scenario.

Given the promising results obtained in the previous research, in this study, it was decided to analyze the performance of the proposed algorithm in a real outdoor deployment. Furthermore, to compare the performance of our algorithm with others, it was decided to compare the performance offered by several classical Machine Learning algorithms, namely with Binary Logistic Regression, Support Vector Classification, Gaussian Naive Bayes, Random Forest, and k-Nearest Neighbors in discriminating pedestrian behavior between static and moving. The

objective of this comparison is to identify whether the simplicity of our proposal implies a reduction in accuracy, compared to other techniques that (initially) would have higher computational requirements.

4.4.2 Experimental setup

The experiments were performed in a rural environment to mitigate possible interference that could be added by other external devices to the experiment. The actual scenario in which the experiments were conducted is shown in Figure 4-9. This scenario simulated the traffic light-controlled traffic intersection depicted in Figure 4-10. A DAU acting as a WiFi sniffer capturing Probe Request messages was placed at each of the four traffic lights and consisted of a low-cost Beaglebone Black development board running Linux 7.8, along with a WiFi USB adapter (TP-Link TL-WN722N). Figure 4-11 shows the devices that made up each DAU. The distance between each DAU is 30 meters and the DAUs were located at a height of 2 meters.

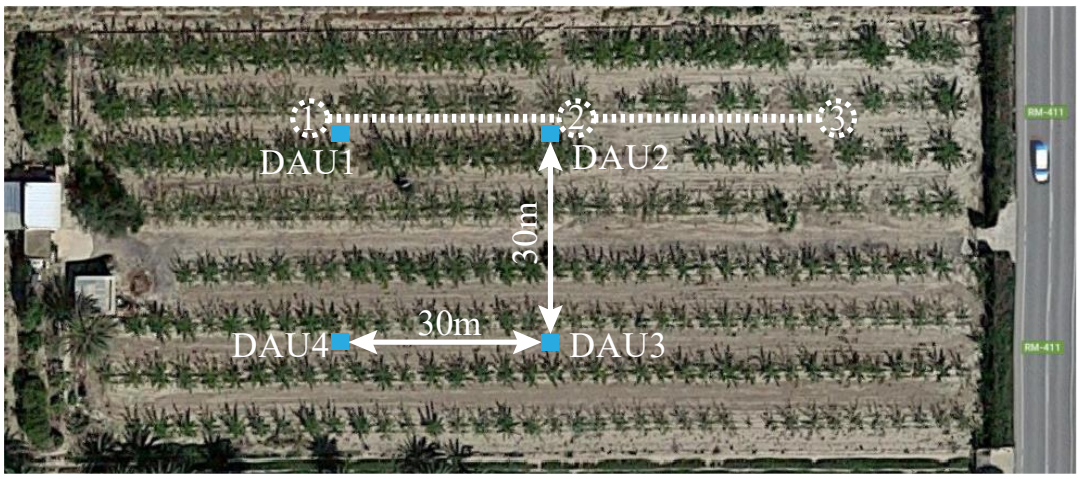


Figure 4-9. The real outdoor scenario where experiments were performed. (Figure 2, p. 3, [294]).

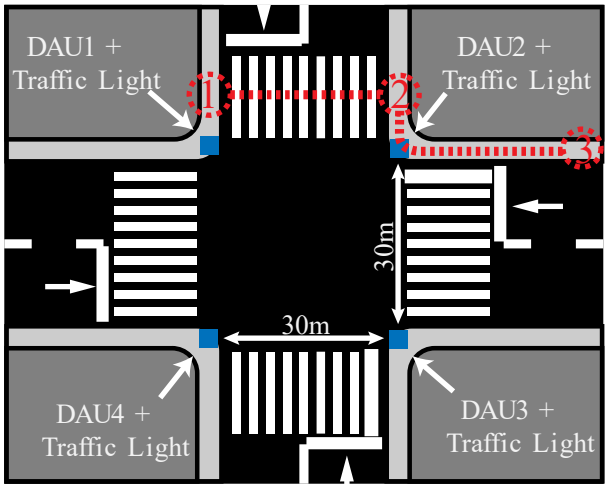
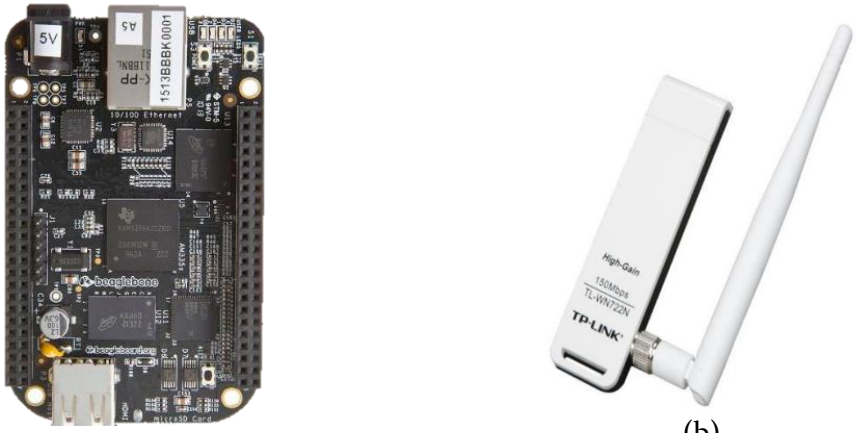


Figure 4-10. A signalized traffic intersection composed of four branches. On each branch, there is a traffic light incorporating a data acquisition unit (DAU). (Figure 3, p. 3, [294]).



(a) (b)

Figure 4-11. Devices used in the experimental tests: (a) Beaglebone Black; (b) WiFi USB adapter TP-Link TL WN722N. (Figure 1, p. 2, [294]).

A total of 14 cell phones were used in the experiments. The details of the cell phones are described in Table 4-6. The cell phones were in passive WiFi network search state, i.e., with WiFi in an activated state without being connected to any network and the display turned off. To simplify the scenario deployment, it was considered that, in each capture time interval, pedestrians are waiting to cross or are in motion. In addition, each pedestrian was considered to be carrying a mobile device and all pedestrians followed the same movement pattern, albeit with small variations in speed around the average (1.38 m/s).

TABLE 4-6. Devices used, brand, model, and version of the operating system (OS). (Table 1, p. 3, [294]).

Brand/Model	S.O Version
Xiaomi Redmi Note 5	Android 8.1.0 MIUI 9.6
Samsung J5 2016 SM-J500F	Android 7.1.2
Samsung Galaxy S4 GT-I9506	Android 7.1.2
Samsung Galaxy S7 Edge	Android 7.0
Xiaomi Redmi Note 4	Android 6.0 MIUI 9.6
Huawei Y6 2017 (x2)	Android 6.0 EMUI 4.1
Vodafone Smart Turbo 7	Android 6.0
Xiaomi Redmi 3	Android 5.1.1 MIUI 9.6
Samsung Galaxy S2 GT-9100	Android 4.1.2
Google Nexus S	Android 4.1.2
Vodafone Smart III 975N (x2)	Android 4.1.1
Samsung Galaxy Ace GT-S5830	Android 2.3.3

The movement pattern followed by pedestrians is represented as a dotted line in Figure 4-9 (1-2-3) and that is simulated to be the movement of pedestrians at the intersection in Figure 4-10 marked with a dotted line (1-2-3). The pedestrian movement pattern, time intervals, and associated timestamps are detailed in Table 4-7.

TABLE 4-7. Time intervals, timestamps, and pedestrians' state. (Table 2, p. 4, [294]).

Time Interval	Timestamp	State
0s-30s	T1	Static on (1)
30s-60s	T2	Static on (1)
60s-90s	T3	Static on (1)
90s-120s	T4	Static on (1)
120s-150s	T5	Moving from (1) to (2)
150s-180s	T6	Moving from (2) to (3)
180s-210s	T7	Moving from (3) to (2)
210s-240s	T8	Moving from (2) to (1)

The experiments were divided into 30-second capture intervals, which placed the experiment in the worst case by providing the minimum time for the algorithm to capture the highest number of messages from cell phones, and there may be mobile devices that, due to their state (screen status, battery, etc.) and a random component, may not have sent any Probe Request messages and therefore cannot be classified/positioned. In scenarios with longer traffic light phases (typically ranging from 30 to 90 seconds), more Probe Request messages could be obtained, which would improve the accuracy of the algorithm (for more information on how the period of sending Probe Request messages affects the performance of the algorithm, see [297]).

To capture the messages, the DAUs used the *tcpdump* tool. In addition, to capture the maximum number of Probe Request messages, the DAUs changed the WiFi channel they cyclically listen every 2 seconds.

For data processing and algorithm execution, a program was developed in Python 3.6. In addition, high-performance scientific libraries such as NumPy [308], SciPy[309], and Pandas [310] were used to speed up the processing. For the implementation of ML algorithms, Scikit-learn library [311] was used, which provides a large number of traditional ML algorithms, and from which Binary Logistic Regression, Support Vector Classification (SVC), Gaussian Naive Bayes, Random Forest, and k-Nearest Neighbors were selected.

A comparison of the ML algorithms versus the proposed algorithm was conducted on pedestrian behavior discrimination (moving or static). The input data for these ML algorithms were the same as used by our proposal, i.e., the power variance of the polling request messages captured at all DAUs. Position classification was further performed by identifying the DAU with the highest average power level during the capture time interval. Please refer to [297] for a complete definition of our approach as well as the metrics used to measure the accuracies.

4.4.3 Results

In this section, we will show the results obtained, as well as a comparison between the accuracy obtained by the different algorithms tested. Once the experiment shown in the previous section has been performed, the results are shown in Figure 4-12. In addition, the mean values of the accuracies obtained for each of the algorithms are shown in Table 4-8.

If we focus on the results obtained by our proposal (Figure 4-12a) we can see that the accuracy in the differentiation of moving pedestrians (Amoving) is higher (on average) than 57%. We can also see that the accuracy in static pedestrian differentiation (Astatic) presents an average value higher than 81%. Finally, as for the positioning accuracy (Apositioning), it presents an average value higher than 85%.

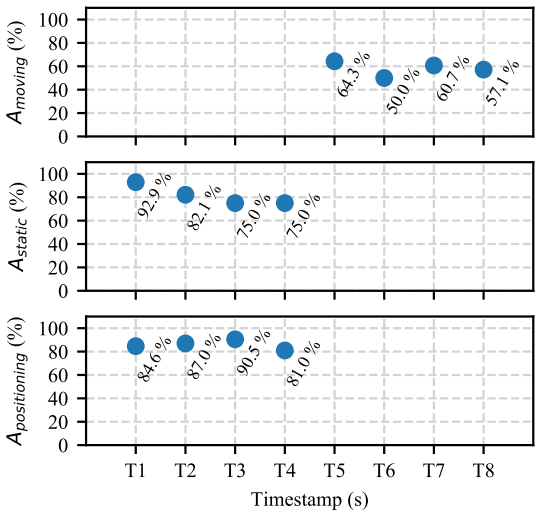
The accuracies obtained significantly exceed those obtained in our previous work based on the simulator. This may be because the RF channel, in reality, presents less randomness and noise than the simulated RF channel. If we compare the results with other works in the related literature, we can observe that our proposal obtains better performance. It should be noted that, to the best of our knowledge, there are no other works that have the same approach for pedestrian counting and localization in urban environments. Therefore, it has had to be purchased with other types of related applications.

If we compare the results obtained with those obtained with traditional ML methods such as Gaussian Naïve Bayes (Figure 4-12b), SVC (Figure 4-12c), Random Forest (Figure 4-12d), Binary Logistic Regression (Figure 4-12e), and *k*NN (Figure 4-12f) the results are similar, even our proposal (Figure 4-12a) outperforms SVC, Gaussian Naïve Bayes, and Binary Logistic Regression in some accuracies. Finally, the results obtained by the Random Forest classifier and the *k*NN stand out, obtaining very high values in all the accuracies.

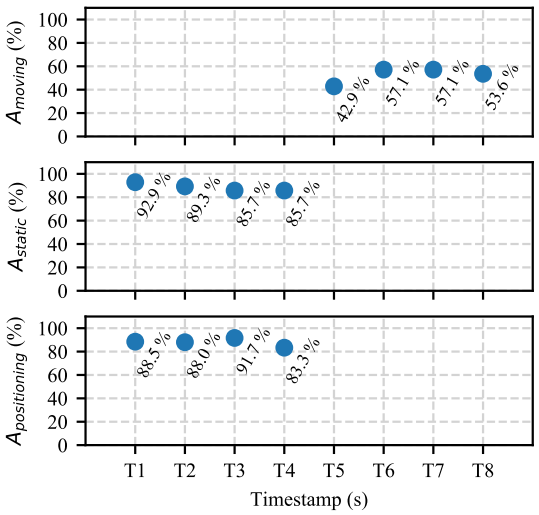
According to the results obtained, we can conclude that the proposed algorithm obtains very promising results in real outdoor scenarios, outperforming some ML algorithms. Only Random Forest and *k*NN offer superior performance.

TABLE 4-8. Average accuracies obtained for each of the tested algorithms and each of the proposed accuracies.

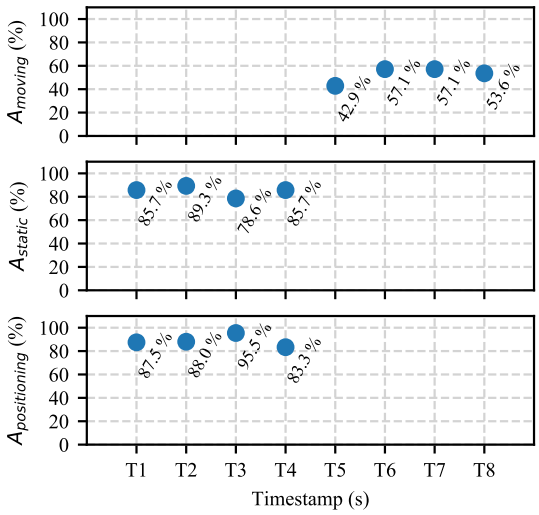
Algorithm	Amoving	Astatic	Apositioning
Our Proposal	57.27	81.25	85.78
Gaussian Naïve Bayes	52.67	88.40	87.88
SVC	52.67	84.82	88.58
Random Forest	87.50	83.93	88.30
Logistic Regression	60.73	75.90	87.23
kNN	90.18	87.50	89.05



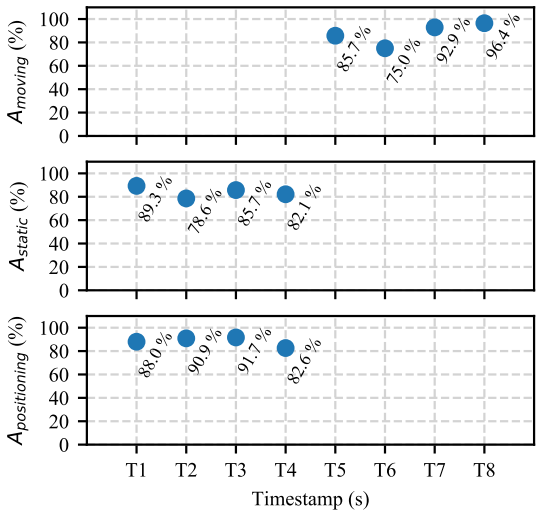
(a) *Our Proposal*



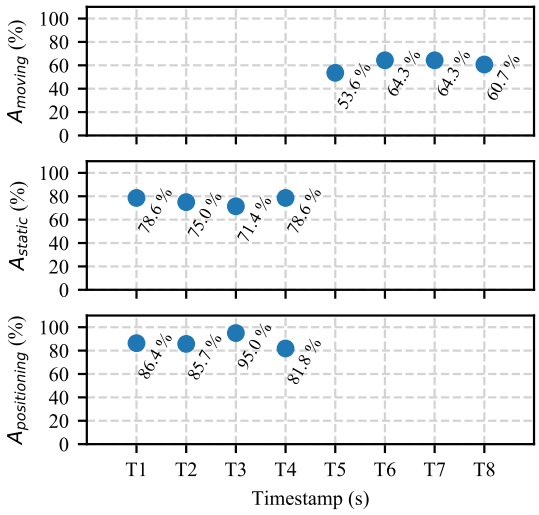
(b) *Gaussian Naïve Bayes*



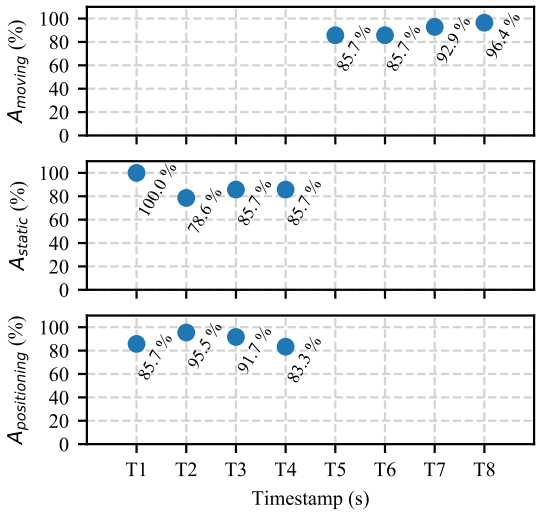
(c) *SVC*



(d) *Random Forest*



(e) *Logistic Regression*



(f) *kNN*

Figure 4-12. Accuracies obtained by the proposed algorithm and the ML algorithms: (a) Our proposal; (b) Gaussian Naïve Bayes; (c) SVC; (d) Random forest; (e) Logistic regression; (f) kNN with k=12; (Figures 4-9, p. 7, [294]).

4.4.4 Conclusions

Passive WiFi-based methods for pedestrian counting are opening up new avenues of development for new intelligent traffic control systems due to their simplicity of application, low cost, and low computational requirements. Until now, these systems only took vehicles into account when calculating the duration of the next traffic light cycle, without considering other road users. Using such WiFi-based solutions, integrating an estimate of the number of pedestrians using the road is much more possible, eliminating the use of cameras or other devices that further violate pedestrians' privacy.

In this study, a real experiment was conducted to show the performance offered by a previously developed passive WiFi-based pedestrian counting algorithm. Specifically, our goal was to classify pedestrian behavior and use this information for intelligent traffic control systems in urban areas. Our algorithm was not only able to differentiate the state of pedestrians (static or moving), but also to position them within the intersection. Furthermore, the performance offered by our approach was compared with other ML algorithms, namely Binary Logistic Regression, Support Vector Classification, Gaussian Naïve Bayes, Random Forest, and k-Nearest Neighbors.

The results showed remarkable performance of our approach, despite the simplicity of our method, as well as results that outperformed the performance offered by several ML algorithms. Only Random Forest and *k*NN offered excellent performance for this classification task.

Further work is needed to determine whether, as expected, the computational cost of ML solutions would be a compromise between lightweight (our proposal) and computational or energy requirements.

4.5 Pedestrian Characterization in Urban Environments Combining WiFi and AI

4.5.1 Introduction

After realizing the advantage that machine learning algorithms could offer for pedestrian behavior classification using a passive WiFi-based algorithm in [294], this paper proposed an improvement in the intelligence of the previously proposed algorithm in [297] using a machine learning algorithm. This new algorithm was named intelligent Pedestrian Characterization using WiFi (iPCW).

4.5.2 Experimental setup

For this algorithm improvement, a simulation setup, simulated scenario, and pedestrian movement pattern similar to previous studies in [297] was employed. That is, OMNET++ [306] simulator was used in combination with INET framework

[307]. The simulated scenario was an urban arterial consisting of two intersections. Each intersection was composed of four traffic lights, where at each traffic light there was a DAU in charge of capturing pedestrian Probe Request messages. The scenario used, as well as all the parameters used in the simulator, can be seen in Figure 4-3, as well as in Tables 4-3.

Twelve pedestrian groups consisting of 64 pedestrians each were simulated, for a total of 768 pedestrians. Each pedestrian group presented a different mobility pattern, although can be grouped in vertical movement and horizontal movement. These mobility patterns (vertical and horizontal) can be found in Table 4-9.

For this project, a variant of the algorithm proposed in [297] was proposed that allowed the integration of an ML algorithm for the discrimination of pedestrian behavior (static or moving). The pseudocode of the algorithm of this proposal can be seen in Algorithm 4-2. A flowchart schematically illustrating the flow of operations and decisions made by the proposed algorithm can also be seen in Figure 4-13.

For data processing, a program was developed in Python 3.6 together with high-performance scientific libraries such as NumPy [308], SciPy[309], and Pandas [310] were used to speed up the processing. For the implementation of the ML algorithms, the Scikit-learn library [311] was used.

The ML algorithms analyzed in this study were Logistic Regressor, Gaussian Naïve Bayes, Support Vector Machine, *k*-Nearest Neighbor, and Random Forest. For a detailed explanation of the working principle, advantages, and disadvantages of the ML algorithms studied in this subsection, see section 3.3 of [265]. Schematically, the working principle of the analyzed ML algorithms is shown in Figure 4-14.

TABLE 4-9. Timestamps of different movements and their behavior (Table 3, p. 6, [265]).

Movement	Example	Time Interval	Timestamp	Behavior
Horizontal	D*lrt*	0 s-30 s	T1	Static in ①
		30 s-60 s	T2	Moving - ① to ②
		60 s-90 s	T3	Moving - ② to ⑤
		90 s-120 s	T4	Static in ⑤
		120 s-150 s	T5	Moving - ⑤ to ⑥
		150 s-180 s	T6	Static in ⑥
Vertical	D*tdl1*	0 s-30 s	T1	Static in ①
		30 s-60 s	T2	Moving - ① to ④
		60 s-90 s	T3	Moving - ④ to ③
		90 s-120 s	T4	Static in ③
		120 s-150 s	T5	Moving - ③ to ④
		150 s-180 s	T6	Static in ④

Algorithm 4-2: *iPCW pseudocode*

```
The list_of_pedestrians detected by all DAUij is obtained.
for pedestrian p in list_of_pedestrians do:
    the list_of_DAUij that have detected p is obtained.
    static = True # by default the pedestrian is considered as static
    for DAUij in list_of_DAUij do:
        # The power measurements are smoothed to reduce noise.
        # The behavior (beh) is obtained. 0=static; 1=moving
        beh = ML_classifier.evaluate(behavior)
        if beh == 1 then: # classifier considers p as moving.
            static = False # now the pedestrian is considered as moving
        end if
    end for
    if static == True then:
        # The pedestrian p is considered as static in all of list_of_DAUij,
        # then its correct location corresponds to the DAUij that presents
        # the higher power in average.
        DAUij_max = get_DAUijp_MaxMeanPower(p)
        DAUij_max.add_Pedestrian(p)
    end if
end for
```

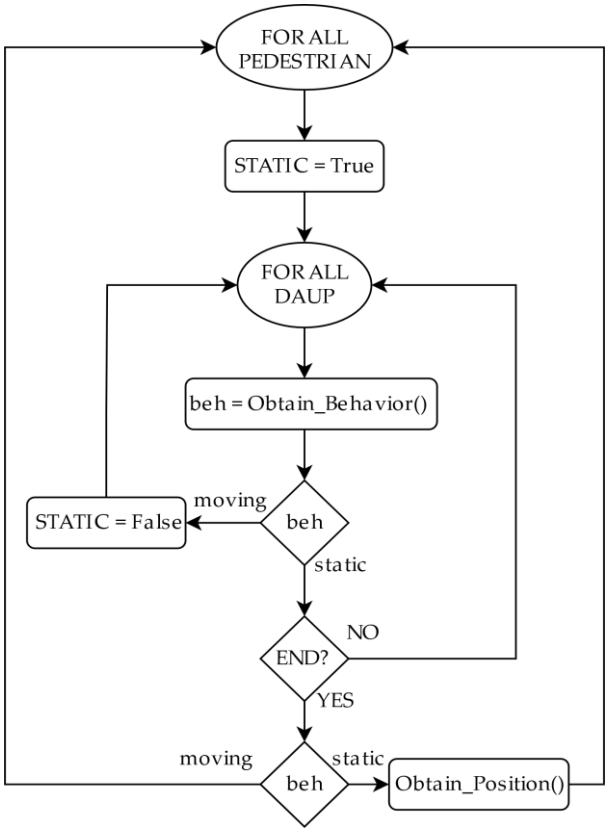


Figure 4-13. Flow diagram of the proposed method (beh=behavior; DAU=Data Acquisition Unit). (Figure 5, p. 7, [265]).

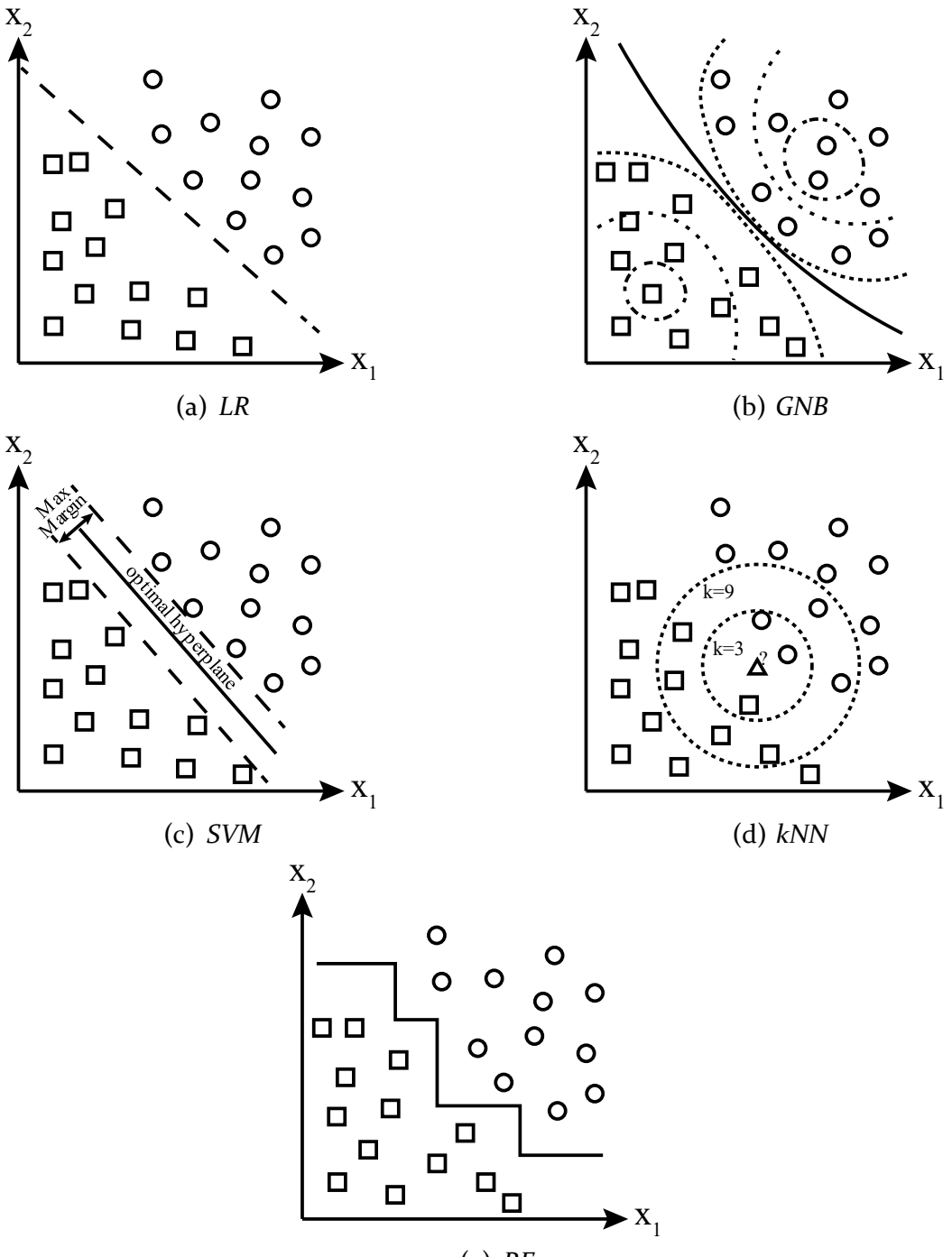


Figure 4-14. How each of the ML algorithms studied works: (a) Logistic Regressor; (b) Gaussian Naïve Bayes; (c) Support Vector Machine; (d) k-Nearest Neighbor; (e) Random Forest. (Figure 6, p. 9, [265]).

For comparing the performance of the different ML algorithms studied, the confusion matrix was used. In each cell of the matrix, the number of predictions made by each algorithm is placed. Each column of the matrix represents the predictions of the ML algorithm, while each row represents the actual class. An example of a binary confusion matrix can be found in Table 4-10.

TABLE 4-10. Example of Binary Confusion Matrix. (Table 6, p. 10, [265]).

	Predicted: negative (0)	Predicted: positive (1)
Actual: negative (0)	TN	FP
Actual: positive (1)	FN	TP

In each cell of the matrix, the acronyms indicated stand for:

- TN (True Negative): the model correctly predicts the negative class.
- TP (True Positive): the model correctly predicts the positive class.
- FN (False Negative): the model incorrectly predicts the negative class.
- FP (False Positive): the model incorrectly predicts the positive class.

From the confusion matrix, it is possible to obtain a large number of metrics that allow us to know in more detail the performance of each algorithm. For our case, we used: Precision, Recall, and F1-score. The equations of these metrics can be seen in Equations 4-4, 4-5, and 4-6.

$$Precision = \frac{TP}{TP + FP} \tag{4-4}$$

$$Recall = \frac{TP}{TP + FN} \tag{4-5}$$

$$F1\_score = \frac{2 * Precision * Recall}{Precision + Recall} \tag{4-6}$$

4.5.3 Experiments & Results

After simulating all the pedestrians in the scenario seen above, we obtained a dataset consisting of 36,864 samples. Each of the samples corresponding to the RSSI temporal evolution of the Probe Request messages during the capture intervals (30 seconds) and came from the behavior of the 768 pedestrians (12 groups x 64 pedestrians/group = 768 pedestrians) in the 6 capture intervals (see Table 4-4) and the 8 DAUs installed at the 2 intersections (4 traffic lights per intersection, see the simulated scenario in Figure 4-3). Half of the temporal samples (18,432) belonged to static pedestrian behaviors and the other half to moving pedestrians. For training and testing, the data set was divided into 23,961 samples (65%) for the training set and the remainder (35%) for the test set. In the training set, there were 12,722 samples from the static class and 11,139 from the moving class, and in the test set, there were 6,508 samples from the static class (timestamps T1, T4, and T6) and 6,395 from the moving class (timestamps T2, T3, and T5).

Experimental set #1

The first set of experiments aimed to obtain the performance of different ML algorithms from a large set of previously generated input data in order to compare their performance. The generated input data set belongs to the statistical parameters of the temporal power behavior of the data set previously extracted from the simulation. These statistical parameters are shown in Table 4-11. In total 27 different features allowed training the various ML algorithms for pedestrian behavior classification.

TABLE 4-11. Statistics used as input features for ML algorithms. (Table 5, p. 10, [265]).

Statistic	
Mean power	
Variance	
From linear regression line	Slope
	Intercept
	r-value
From polynomial regression of degree 2	Coefficient degree 0
	Coefficient degree 1
	Coefficient degree 2
	Residuals
From polynomial regression of degree 3	Coefficient degree 0
	Coefficient degree 1
	Coefficient degree 2
	Coefficient degree 3
	Residuals
From polynomial regression of degree 4	Coefficient degree 0
	Coefficient degree 1
	Coefficient degree 2
	Coefficient degree 3
	Coefficient degree 4
	Residuals
Kurtosis parameter	
25% Quantile	
50% Quantile	
75% Quantile	
Pearson correlation coefficient	
Pearson p-value coefficient	
Skewness parameter	

After an exhaustive training process with grid search with cross-validation of the most important hyperparameters of all the ML algorithms seen above, the results are shown in Tables 4-12 and 4-13. More specifically, Table 4-12 shows the confusion matrix for each of the tested algorithms, as well as Table 4-13 shows a report of the metrics used to evaluate the performance (precision, recall, F1-score). In addition, the computational cost of inferring the results of the test set was

studied. The results of the average time of 100 runs on the test set, as well as their standard deviation, are shown in Table 4-14.

TABLE 4-12. Confusion Matrix in the testing dataset. (Table 8, p. 11, [265]).

		Predicted: Static	Predicted: Moving
LR	True: Static	4294 / 33.28%	2214 / 17.15%
	True: Moving	2083 / 16.14%	4312 / 33.42%
GNB	True: Static	3918 / 30.37%	2590 / 20.07%
	True: Moving	1903 / 14.75%	4492 / 34.81%
SVM	True: Static	4907 / 38.03%	1601 / 12.41%
	True: Moving	2042 / 15.83%	4353 / 33.74%
*k*NN	True: Static	6353 / 49.24%	155 / 1.20%
	True: Moving	448 / 3.47%	5947 / 46.09%
RF	True: Static	6490 / 50.31%	18 / 0.14%
	True: Moving	20 / 0.15%	6373 / 49.39%

LR = Linear Regressor, GNB = Gaussian Naïve Bayes, SVM = Support Vector Machine, *k*NN = k-Nearest Neighbor, RF = Random Forest.

TABLE 4-13. Classification report in the testing dataset. (Table 9, p. 11, [265]).

	State	Precision	Recall	F1-score
LR	Static	0.668	0.659	0.664
	Moving	0.665	0.674	0.669
	avg/total	0.667	0.667	0.667
GNB	Static	0.668	0.612	0.639
	Moving	0.639	0.704	0.669
	avg/total	0.653	0.647	0.654
SVM	Static	0.701	0.754	0.729
	Moving	0.733	0.680	0.706
	avg/total	0.718	0.717	0.718
*k*NN	Static	0.932	0.976	0.954
	Moving	0.975	0.930	0.952
	avg/total	0.954	0.953	0.953
RF	Static	0.996	0.997	0.996
	Moving	0.997	0.996	0.996
	avg/total	0.996	0.996	0.996

LR = Linear Regressor, GNB = Gaussian Naïve Bayes, SVM = Support Vector Machine, *k*NN = k-Nearest Neighbor, RF = Random Forest.

TABLE 4-14. Execution Time in the testing dataset. (Table 10, p. 11, [265]).

	Execution time (mean ± std) (100 runs)
LR	7.238 ms ± 0.245 ms per run
GNB	108.008 ms ± 0.191 ms per run
SVM	92970.154 ms ± 5779.215 ms per run
*k*NN	35244.375 ms ± 2799.993 ms per run
RF	311.806 ms ± 10.147 ms per run

LR = Linear Regressor, GNB = Gaussian Naïve Bayes, SVM = Support Vector Machine, *k*NN = k-Nearest Neighbor, RF = Random Forest.

Once the first set of experiments was performed, the results showed that the best classifier, with a large difference in performance and execution time, was the Random Forest classifier. Random Forest was able to obtain accuracy, recall, and F1 score values higher than 99% on the test dataset, in addition to having a run time of fewer than 0.5 seconds.

Experimental set #2

This second set of experiments consisted of reducing the complexity offered by the Random Forest classification algorithm by selecting the input features that provide the most information for classification. More specifically, a feature selection procedure called Recursive Feature Elimination with Cross-Validation (RFECV) was used. Thus, by selecting the input features that are most important for classifying pedestrian behavior, both the accuracies (precision, recall, F1-score) and the execution time are optimized. This is because the classification noise is reduced by eliminating features that do not (or the least) provide information for classification.

The results of the RFECV process showed that the best classification performance occurred when 5 input features were selected. More specifically, these features are shown in Table 4-15. In addition, the complete RFECV process can be seen in Figure 4-15, where the accuracy of the cross-validation dataset (Cross-validation score) is indicated as a function of the number of features selected.

After this feature selection process, the RF algorithm was re-optimized considering only these selected variables. The results of the inference on the test data set after training can be seen in Table 4-16. In addition, the summary of the analyzed metrics can be seen in Table 4-17 and the computational performance analysis of the average inference time and its standard deviation of 100 runs can be seen in Table 4-18.

TABLE 4-15. Statistics used as input features for RF after RFECV. (Table 11, p. 12, [265]).

Statistic	
From polynomial regression of degree 3	Coefficient degree 0 Coefficient degree 3
From polynomial regression of degree 4	Coefficient degree 0 Coefficient degree 3 Coefficient degree 4

RF = Random Forest.

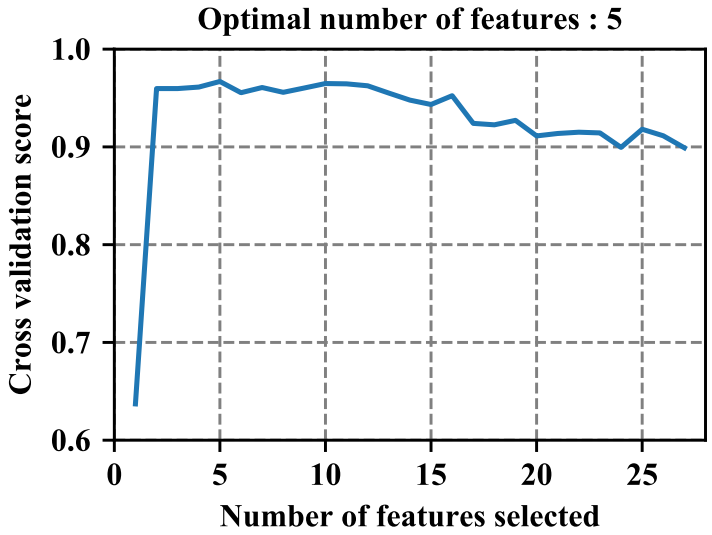


Figure 4-15. RFECV results. The optimal number of features was 5. Increasing the number of features reduces the cross-validation score. This is because not all features provide the same information, apart from possibly conflicting or interfering with classification. (Figure 7, p. 13, [265]).

TABLE 4-16. Confusion Matrix of RF algorithm pre- and post- RFECV the testing dataset. (Table 13, p. 13, [265]).

		Predicted: Static	Predicted: Moving
RF	True: Static	6490 / 50.31%	18 / 0.14%
	True: Moving	20 / 0.15%	6373 / 49.39%
RF with RFECV	True: Static	6495 / 50.34%	13 / 0.10%
	True: Moving	6 / 0.05%	6389 / 49.52%

RF = Random Forest.

TABLE 4-17. Classification report of RF algorithm pre- and post- RFECV in the testing dataset. (Table 14, p. 13, [265]).

	State	Precision	Recall	F1-score
RF	Static	0.9965	0.9972	0.9968
	Moving	0.9973	0.9965	0.9969
	avg/total	0.9969	0.9969	0.9969
RF with RFECV	Static	0.9991	0.9980	0.9985
	Moving	0.9980	0.9991	0.9985
	avg/total	0.9986	0.9986	0.9985

RF = Random Forest.

TABLE 4-18. Execution Time of RF algorithm pre- and post- RFECV in the testing dataset. (Table 15, p. 13, [265]).

	Execution time (mean ± std) (100 runs)
RF	311.8065 ms ± 10.1476 ms per run
RF with RFECV	110.7994 ms ± 0.6019 ms per run

RF = Random Forest.

As can be seen in the results, the advantages offered by the feature selection mechanisms are enormous. We see how the results improve in all the metrics analyzed, achieving accuracies above 99.80%, and reducing the execution time to a third of its original value.

Experimental set #3

Finally, in this last set of experiments, the overall performance provided by iPCW when using RF as a pedestrian discriminator was evaluated. Pedestrian positioning was continued by simply finding the DAUijp with the highest mean power during the capture interval.

To obtain an overall view of the performance offered by iPCW, several metrics were used to obtain the classification accuracy (Aclassification) and positioning accuracy (Apositioning), as shown in Equation 4-7 and Equation 4-8.

$$A_{classification} = \frac{TP + TN}{TP + TN + FP + FN} \quad (4\text{-}7)$$

$$A_{positioning} = \frac{pedestrians\_located\_properly}{pedestrians\_detected\_as\_static} \quad (4\text{-}8)$$

Where:

- TN (True Negative): the classifier (RF) correctly predicts the static state.
- TP (True Positive): the classifier (RF) correctly predicts the moving state.
- FN (False Negative): the classifier (RF) incorrectly predicts the static state.
- FP (False Positive): the classifier (RF) incorrectly predicts the moving state.
- pedestrians_located_properly: indicates pedestrians correctly located under the DAU waiting to cross.
- pedestrians_detected_as_static: indicates pedestrians detected as static by the classifier (RF).

After experimenting with the scenario shown above (Figure 4-3), the results are shown in Figure 4-16. In this figure, we can see both the pedestrian classification accuracy by the RF classifier and the positioning accuracy (timestamps T1, T4, and T6).

Noticeably, the results obtained by iPCW are notable, showing a behavioral discrimination accuracy between static and moving pedestrians of more than 98%. In addition, the positioning accuracy within the intersection for static pedestrians exceeds 92%. This positioning accuracy is only shown at time intervals where static pedestrians were present, i.e., at timestamps T1, T4, and T6. These results are a huge improvement over those obtained in the work in which the base algorithm was proposed, showing the enormous advantages that ML algorithms can offer.

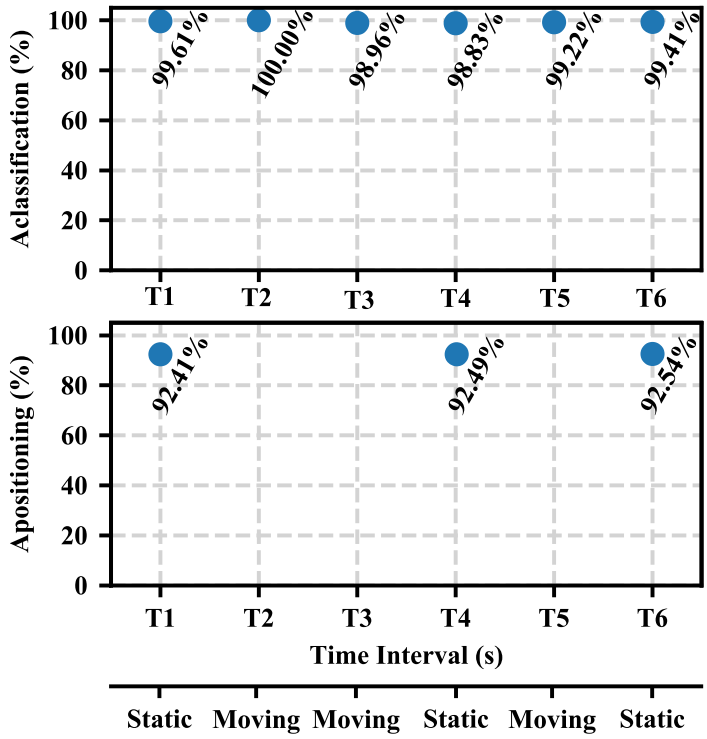


Figure 4-16. Classification accuracy (RF) and positioning accuracy (mean RSS) results. Positioning accuracy only appears in the time intervals where pedestrians were static (T1, T4, and T6), since in the other time intervals (T2, T3, and T5) all pedestrians were moving. (Figure 8, p. 14, [265]).

4.5.4 Conclusions

The advantages of considering pedestrians in intelligent transportation control systems are immense, such as better traffic management, better quality of life for pedestrians, more safety, etc. This study proposed a method based on AI capable of classifying pedestrian behavior in urban environments, as well as positioning pedestrians detected as static within an intersection. The method, called intelligent Pedestrian Characterization using WiFi (iPCW), was a passive device-based method using WiFi technology. More specifically, it used Probe Request messages sent by mobile devices passively and periodically to classify and position pedestrians in open urban environments. This iPCW used an artificial intelligence algorithm called Random Forest, picking out the most useful features that enabled higher performance with RFECV. iPCW was evaluated through intensive computer simulations in an urban scenario consisting of two traffic light-controlled intersections. The results showed excellent performance, with detection accuracy between moving pedestrians and static pedestrians exceeding 98% and positioning accuracy exceeding 92%. Consequently, we humbly consider iPCW to be simple, lightweight, and with great performance. The accuracy levels achieved are comparable (and in some cases even better) than those obtained by other proposals in both indoor and outdoor scenarios. Particularly noteworthy is the

accuracy that our algorithm achieves in discriminating pedestrian behavior (between static and moving), as well as in obtaining their location at the appropriate crosswalk. It is important to keep in mind that our algorithm requires simplicity and lightness so that it can be executed in real-time on low-cost, low-power embedded devices.

Although were identified some issues such as MAC randomness, the period of sending probes request messages or the underestimation of the number of people due to not everyone carries a WiFi-enabled mobile device, the advantages that these systems can offer are enormous.

4.6 Conclusions to this chapter

The potential benefits of taking pedestrians into account for advanced traffic management systems are enormous, such as increased safety, improved flow, improved quality of life for pedestrians, etc.

In this section, we have analyzed the methods used for people counting in both indoor and outdoor environments. Seeing the benefits and advantages that device-based WiFi passive methods could offer, a set of works have been carried out to obtain a series of algorithms capable of classifying and positioning pedestrians in urban environments.

The first study carried out a first simple approach that laid the foundation for subsequent developments. This approach provided over 52% performance in discriminating between static pedestrians and moving pedestrians. In addition, the positioning at the crosswalk that pedestrians were waiting to cross was performed correctly 93% of the time.

Seeing the advantages that the previously proposed algorithm could offer; in the second work, a deployment was performed in a real scenario simulating an urban intersection. In this way, the proposed algorithm could be validated in a real deployment. In addition, the performance of the algorithm was compared with that of other ML classification algorithms. The results showed that the proposed algorithm could offer superior performance to other ML algorithms, as well as to the results of previous work. The behavior discrimination accuracy exceeded 57%, as well as the positioning accuracy was above 90%. However, the ML RF algorithm stood out. RF was able to obtain a positioning accuracy above 83%, as well as a positioning accuracy above 88%.

Considering the benefits that ML algorithms could offer in the discrimination of pedestrian behavior, in the last work seen in this section, the performance offered by several ML algorithms was evaluated in order to obtain the ML algorithm with the best performance. In addition, another important feature is that the selected algorithm should be lightweight when inferring new results. After an

intensive process of optimization and selection of the variables that provide the most information when classifying by RFECV, the algorithm with the best performance was RF. After the selection of RF, a new algorithm was designed and named intelligent Pedestrian Characterization using WiFi (iPCW). The overall performance offered by iPCW was discrimination accuracy better than 98%, as well as positioning accuracy better than 92%.

In conclusion, the systems proposed in this section can provide traffic control systems with the ability to consider pedestrians in their control decisions. In this way, advanced control policies capable of controlling both vehicular and pedestrian traffic can be obtained, maximizing road safety as well as minimizing waiting times.

As a final note, it is worth mentioning that, due to the intended use case of this type of system, the main problem of MAC randomness in mobile devices during the search for WiFi networks to connect to is greatly reduced, since the probability of a device modifying its MAC in a short capture interval is reduced [296]. In our case, the capture intervals are determined by the traffic-light cycles and typically range from 30 to 90 seconds.

4.7 Publications associated with this research

The works related to this chapter are as follows:

Articles:

Guillen-Perez, A.; Cano, M.-D., "Pedestrian Characterization in Urban Environments Combining WiFi and AI," *Int. J. Sens. Networks*, vol. 37, no. 1, p. 48, 2021, doi: 10.1504/IJSNET.2021.117964.

2020 Journal Impact Factor (JIF): 1.302. (Q4), Rank: 80/91 in Telecommunications.

Congress:

Guillen-Perez, A.; Cano, M.-D., "A WiFi-based method to count and locate pedestrians in urban traffic scenarios," in *2018 14th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, Oct. 2018, vol. 2018-Octob, pp. 123–130, doi: 10.1109/WiMOB.2018.8589170.

Guillen-Perez, A.; Cano, M.-D., "Counting and locating people in outdoor environments: a comparative experimental study using WiFi-based passive methods," *ITM Web Conf.*, vol. 24, pp. 1–10, Feb. 2019, doi: 10.1051/itmconf/20192401010.

Chapter 5: Smart Traffic Light Control – AI approach

5.1 Introduction

One of the main causes of traffic poor management at city intersections and approaches is due to inefficient management of traffic light control systems [312]. Following this challenge, ITS emerged. ITS are able to adapt the traffic light cycles controlling intersections based on various parameters (such as real-time lane utilization, air quality, vehicle waiting time, etc.) in order to improve the use of the intersection.

These control systems are experiencing a transformation in the operating paradigm thanks to Smart Cities, big data, the Internet of Things, and artificial intelligence, giving rise to new opportunities to achieve more efficient, sustainable, and environmentally friendly urban mobility [313].

In this section, we present related work proposed to increase the intelligence of intelligent traffic-lights control systems by using different approaches like artificial intelligence algorithms.

5.2 State of the art

Numerous ITS can handle traffic in an intelligent way [314], within which they can be classified by levels of cognitive capability. There are the most basic systems that only control timing in time divisions [315], [316], passing through widely implemented systems such as SCATS [317], SCOOT [318], and Max-pressure [319], which adjust the parameters of the signal timing scheme (signal period, green/red signal ratio and phase difference), up to isolated systems fully capable of self-learning the characteristics of the environment and requiring high computational power [320]–[323].

In the scientific literature, we can find numerous works addressing intelligent traffic control at regulated intersections using artificial intelligence or relying on AI to optimize an advanced control policy. The most important of these are fuzzy logic [324]–[326], reservation and market-based system [327]–[329], neural networks [330]–[332], reinforcement learning [72], [333]–[336] and swarm intelligence and evolutionary computation [337]–[339] that try to solve the traffic management problem by proposing new approaches in traffic light control.

Another way to obtain a simple but advanced control policy is using Evolutionary Algorithms (EA). More specifically, using Genetic Algorithms (GA) for the optimization of very complex control systems, or to obtain an advanced control policy by modeling the phases of the policy on a set of chromosomes of the genetic algorithm.

Within the second group, we can find the work done by Sánchez-Medina *et al.* [340]. They encoded a fuzzy logic controller in the chromosomes of a GA population. After applying the optimization process, the obtained control policy allowed controlling vehicles in simple scenarios. Similar works can be found in [339], [341]. Also, in [342] we can find an approach that unifies the communication between track and GA devices. In this way, more information can be incorporated into the optimization process and better results can be obtained than in previous works.

If we focus on the search for papers that use AE, we can find papers that use Ant Colony Optimization (ACO) processes. ACOs allow solving optimization problems using graphs. Within this category, the approaches proposed by Rehman *et al.* [343] or by Jerry *et al.* [344], where both treated with ACO the traffic control problem, are worth mentioning.

5.3 Study on the influence of traffic signal duty cycle duration at a single intersection under incremental traffic density using traffic simulator

5.3.1 Introduction

The main basis of the operation of advanced traffic light control systems for intersections and intelligent transportation systems is to adapt the cycle time of traffic lights according to vehicular traffic conditions [312]. It should be noted that the cycle time is the sum of all phases through which a traffic light passes (red time, green time, yellow time, and intersection clearance time). However, this operating principle is strongly influenced by a huge number of external parameters that the vast majority of works do not consider. For example, most works consider that it is better to increase the duty cycle of traffic lights when the flow of vehicles increases [338], [345], [346]. However, this statement cannot always be admitted, as it will depend on the condition of the other roads, the number of lanes, the traffic light cycle, the characteristics of vehicles, etc. Moreover, if special attention is not paid to the simulated scenario, intersections could be oversaturated, i.e., the number of vehicles entering an intersection exceeds the number that the intersection itself can support, exceeding what is known as critical flow. Exceeding this critical flow makes the behavior of vehicles and the intersection unpredictable [347]. Figure 5-1 shows the relationship between incoming vehicular flow (veh/h) and outgoing vehicular flow (veh/h) through a typical intersection. As can be seen, when the critical flow is reached, the capacity to serve all vehicles is reduced, until a point is reached where the intersection becomes saturated, and the intersection is blocked (Jam Flow).

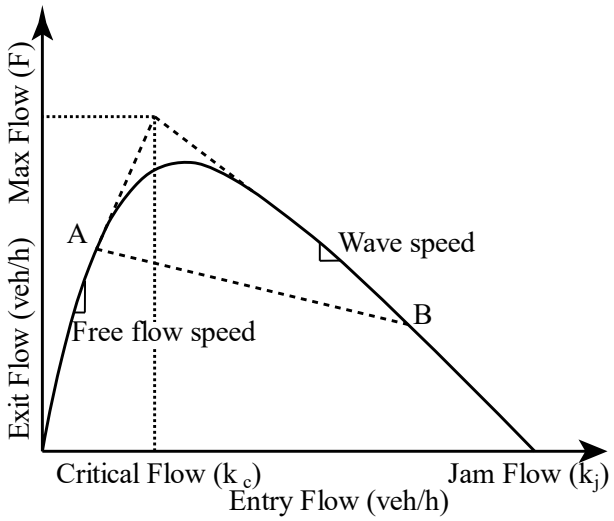


Figure 5-1. Representation of vehicular entry flow (veh/h) and vehicular exit flow (veh/h). (Figure 1, p. 2, [348]).

For the study of new control systems, there are numerous traffic simulators, such as SUMO [349], which allow the behavior of each vehicle to be modeled individually. However, due to the great versatility that this type of microscopic simulator can offer by modifying a large number of internal parameters of the simulation, the complexity of the simulation is high and may not be configured with sufficient rigor.

For that reason, in this paper [348] we investigate the operating bases on which intelligent traffic control systems are based, corroborating the assumptions that are usually made in these works, as well as showing the conditions and limits that the simulations of these proposals must meet in order to be able to rely on these operating bases.

The study aimed to analyze the influence of traffic signal cycle time for intersections as a function of a range of incremental vehicle flow. The influence on a large number of objective variables, such as average waiting time to cross an intersection, average vehicle speed, fuel consumption, and vehicle pollutant emissions, were analyzed. In this way, we were able to obtain the optimal cycle time of the traffic lights controlling an intersection, as a function of each incoming vehicular flow and for each objective variable, and we were also able to corroborate the operating principle on which these systems are based.

5.3.2 Experimental setup

For the experimental development, the SUMO traffic simulator was used, a microscopic traffic simulator, widely studied and developed by the scientific community and used in the study of various fields such as vehicular networks, the study of traffic flows in large cities, and the development of new ITS. The scenario used is shown in Figure 5-2 and is composed of a 4-branch intersection controlled by traffic lights. Each branch is 200 meters long and consists of 2 lanes in each

direction. Between each branch of the intersection, an inbound vehicular flow was defined varying between 600 veh/h and 2000 veh/h for through or right-turn directions. Left turns were not allowed. Table 5-1 shows the simulator input parameters used and the output data that were analyzed.

The simulated vehicle fleet was based on the vehicle fleet data provided by the Dirección General de Tráfico de Madrid (Spain), which can be seen in Table 5-2. The vehicles were configured to meet the air pollution and fuel consumption limitations imposed by the European EURO 5 regulations for more precise results.

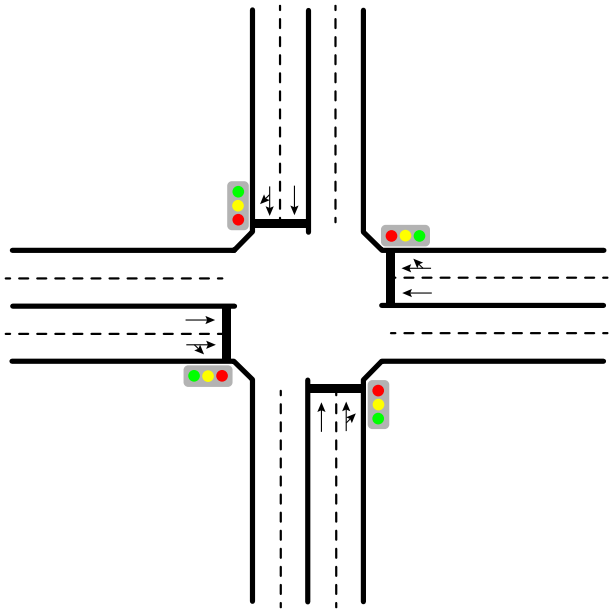


Figure 5-2. Depiction of the simulated intersection. (Figure 2, p. 3, [348]).

TABLE 5-1. Input parameters to the simulator to configure the simulations and output parameters analyzed. (Table 1, p. 3, [348]).

Input/Output	Parameters	Values
Input	Vehicle flow (symmetrical)[1]	{300 - 1000} veh/h/lane
	Yellow time	2 s
	Clearance time	5 s
	Red time	{Green Time}
	Green time	{15 - 100} s
	Cycle	{Green + Yellow + Red + Clearance} = {37 - 207} s
Output	Trip duration	
	Waiting time	
	Average speed	
	CO emitted	
	Fuel consumption	

The simulated vehicular flows were symmetrical, i.e., equal in all lanes of all branches of the intersection. The flow entering each branch at the intersection is twice that indicated since each branch has 2 lanes in each direction.

TABLE 5-2. Vehicle distribution and fuel type used. (Table 2, p. 3, [348]).

Vehicle type	Proportion	Fuel type
Car	30%	Gasoline
Car	40%	Diesel
Motorcycle	10%	Gasoline
Moped	10%	Gasoline
Van	5%	Diesel
Bus	5%	Average of all fuel types

5.3.3 Results

The results obtained from the simulations are shown in Figure 5-3. In this figure, it can be seen the behavior of each of the analyzed metrics as a function of the duration of the green time (between 15 s and 100 s), which directly affects the cycle time of the traffic lights. In addition, each of the lines in each graph represents a different vehicle flow per entry lane (from 300 veh/h/lane to 1000 veh/h/lane). Note that the flow entering the intersection on each branch is twice that indicated in the subfigures in Figure 5-3 since there are two lanes per direction on each branch.

In Figures 5-3 a to 5-3 f it can be noticed that for small flows (less than 500 veh/h/lane) and for very large flows (more than 850 veh/h/lane) the difference in the selection of any green time is very small, as the performance metrics remain the same. However, for the intermediate flow range between 500 veh/h/lane and 850 veh/h/lane, there is a large difference in the analyzed metrics between the green time values.

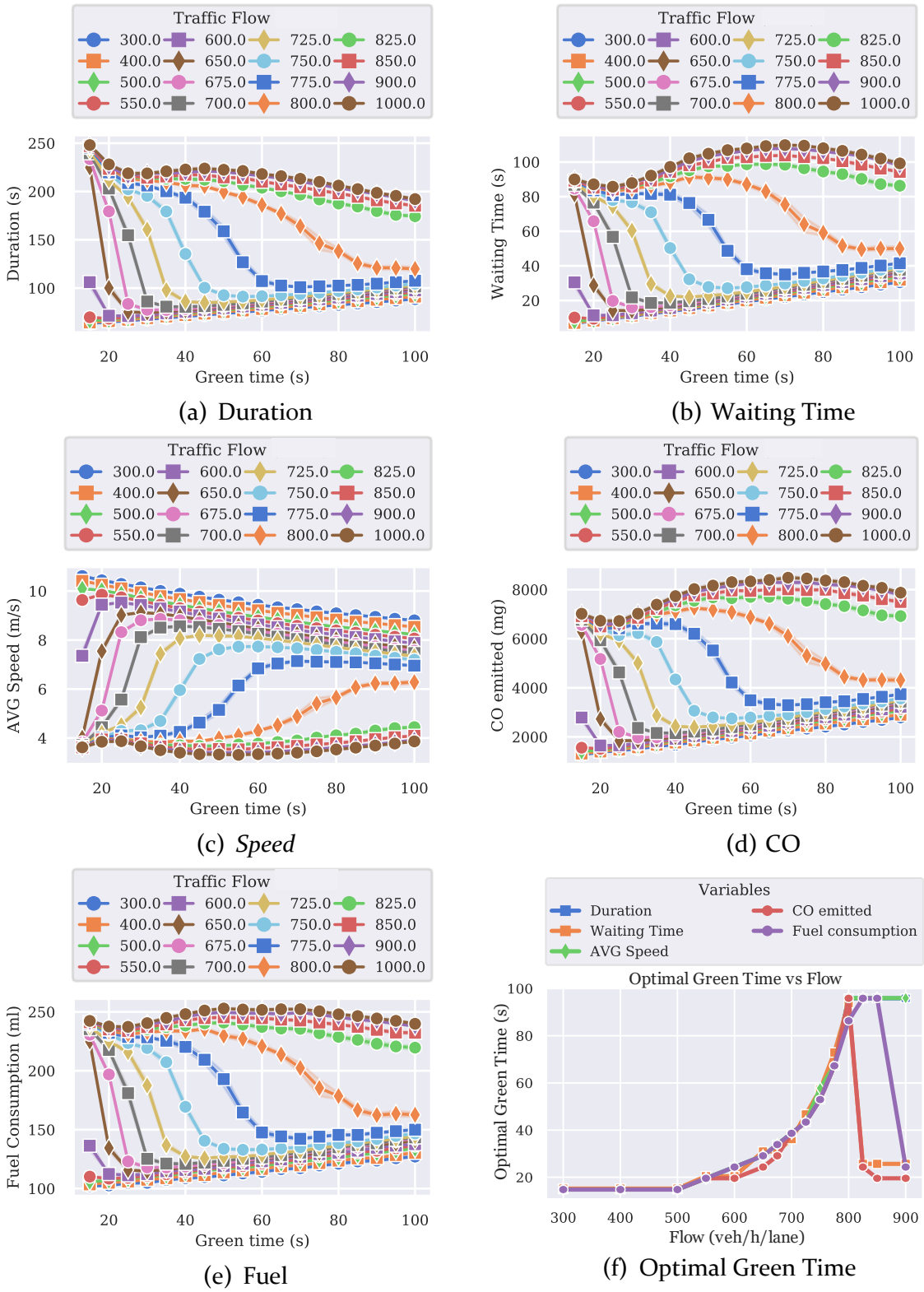
If we focus on this range of flows, we can see in Figure. 5-3 a that increasing the green time to decrease the average travel time benefits (to a greater or lesser extent) traffic flows equal to or greater than 600 veh/h/lane. However, Figure. 5-3 b shows that the same increase in green time is detrimental in terms of average waiting time for flows higher than 800 veh/h/lane. On the other hand, while average speed (Figure. 5-3 c) follows a similar behavior to the average travel time metric, emissions (Figure. 5-3 d) and fuel consumption (Figure 5-3 e) follow the pattern of average waiting time (i.e., limiting efficiency to flows below approximately 800 veh/h per lane).

Finally, Figure 5-3 f shows the optimal green time value for each of the analyzed metrics, as a function of the incoming vehicular flow per lane. In this figure, it can be seen that there are two distinct flow regions. For the region where the vehicular flow is less than 800 veh/h/lane, but greater than 500 veh/h/lane, the behavior of the optimal time is as expected, the higher the vehicular flow, the longer the green time (increase the cycle) of the traffic lights controlling the intersection to optimize the target variables. Below 500 veh/h/lane, the optimal cycle time is always lower than the minimum analyzed (37 s). On the other hand,

if the vehicular flow is higher than 800 veh/h/lane, the behavior of the target variables is no longer predictable, some improve with decreasing green time (waiting time, CO emissions, and fuel consumption) and others improve with increasing green time (travel time and average speed).

5.3.4 Conclusions

In this subsection, we managed to demonstrate that it is possible to alter the performance of vehicular traffic at regulated intersections by modifying traffic light cycles, with particular emphasis on the need to delimit the region in which there is a real benefit and on the importance of identifying this area in work based on computer simulations to avoid results that may not be scientifically rigorous. More specifically, we succeeded in demonstrating that, before the intersection is in a region of oversaturation (in the scenario selected for this study is defined by a vehicular flow of less than 800 veh/h/lane), there is a region in which the studied parameters converge to an optimal value between the range of cycle lengths studied and follows the approach followed by the vast majority of works (the higher the vehicular flow, the more optimal it is to increase the cycle length of traffic lights). However, once this saturation flow is exceeded, the behavior of the analyzed metrics becomes unpredictable. It is interesting to note that there are numerous scientific works based on computer simulations that do not take into account this oversaturation value or do not indicate it, which could compromise the validity of some of the traffic scenarios evaluated.



(a) Duration (b) Waiting Time

(c) *Speed* (d) CO

(e) Fuel (f) Optimal Green Time

Figure 5-3. Results of the experiments carried out. Shown in each of the subfigures (a-e) are the different metrics analyzed as a function of the green time of the traffic light cycle. Each of the lines represents a different incoming vehicular flow: (a) Trip duration: (b) Waiting time; (c) Average speed; (d) CO emitted; (e) Fuel consumption; (f) Optimal green time vs the input flow depending on the parameters studied. (Figures 3-8, p. 4, [348]).

5.4 Intelligent IoT systems for traffic management: a practical application

5.4.1 Introduction

As seen in previous work [348], the basis of intelligent traffic light control systems lies in the modification of the traffic light state depending on the current traffic. Thanks to the enormous advances that have been achieved in the field of AI, the incorporation of many AI algorithms in a myriad of applications have led to enormous improvement and optimization, such as autonomous vehicle control, computer vision, medicine, or NLP [8], [350]–[354].

Within the field of ITS, we can find a wide variety of approaches that apply AI to achieve better performance in numerous areas such as bus route planning, intelligent parking search, or traffic light control [355]–[358].

However, the vast majority of these applications require high computational power to run in real-time, making their deployment on low-cost IoT devices infeasible and having to rely on cloud computing resources [359].

Keeping this problem in mind, in this subsection we decided to address the problem of combining advanced traffic light control systems with AI, taking into account the limitations imposed by low-cost IoT devices and the limitations imposed by the control algorithms themselves, such as the need to run in real-time. As the basis of the control system, the Randomized Early Detection for Dynamic Vehicles (REDVD) algorithm was employed providing a simple but efficient control capable of adapting the phases and cycles of traffic lights at signalized intersections [322].

REDVD was tested in isolated intersections, with simple scenarios, but due to its high degree of adjustment and its multiple parameters to be configured, its behavior in more complex scenarios was not as expected. Therefore, in this study, it was proposed to use a set of AI algorithms called evolutionary algorithms to perform the optimization process of all the parameters. Specifically, a genetic algorithm was employed, which is one of the most popular among EAs due to its simplicity and performance.

These genetic algorithms allowed us to obtain an optimized version of REDVD, which we call iREDVD. iREDVD was able to obtain surprising results in scenarios of high complexity and unknown to iREDVD, allowing us to significantly optimize numerous studied metrics such as vehicle waiting time, average travel time, fuel consumption, and emissions of contaminating particles and pollutant gases, compared to other proposals.

5.4.2 Genetic Algorithms

GAs are a set of procedures inspired by evolutionary theory intended to optimize a set of parameters within a problem in an iterative, intelligent, and fast way, using lightweight procedures [360], [361]. GAs are based on the assumption that, if there is a range of possible solutions to a problem, the best solutions are likely to lie in the solution space defined by the parameter space, and that they are also located spatially close to each other in the parameter space. The main approach followed by GAs is to run a set of iterations where in each iteration new individuals are generated based on the best ones from the previous iteration, in order to find the best individuals within the solution space.

The iterative GA process consists of four phases: population initialization, fitness calculation, selection, and crossover. The GA randomly generates a new population of individuals called the initial population, assigning random values to each of the characteristics/parameters of each individual in the population. In the next phase, the fitness of each individual is evaluated. The fitness is given by the function(s) to be optimized or, alternatively, by an error or fit function. In the selection phase, each individual in the population is ranked according to its degree of fit and then the individuals with the highest degree of fit are selected. Finally, in the crossover phase, the individuals selected in the previous phase are used to create a new population through crossovers, thus obtaining the new population of the same size as the initial one. Moreover, in this phase, random mutations (considered as a noise-type perturbation) are performed on the parameters of the individuals to avoid local optima in the search of the solution space. The whole process is repeated until the convergence of the best solution occurs, i.e., the value of the objective function reaches a certain threshold, or a target number of iterations is met.

A more detailed explanation of each of the phases of genetic algorithms can be found in section 4 of the original paper [313]. The flow diagram of genetic algorithms can be seen in Figure 5-4.

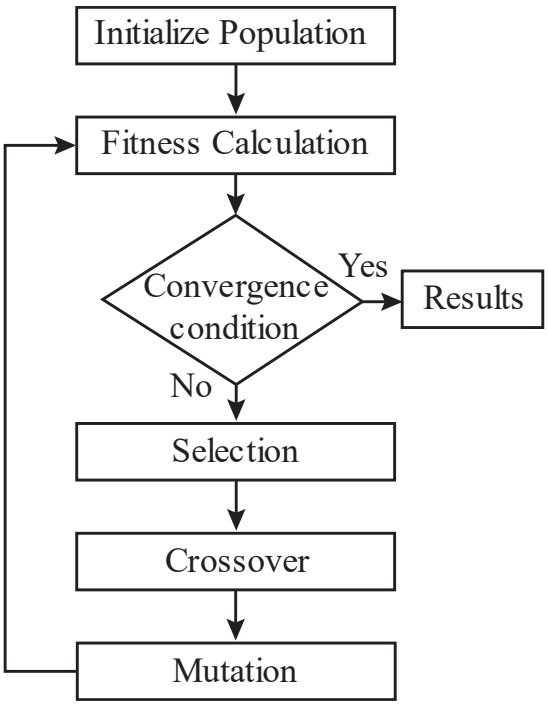


Figure 5-4. A general Genetic Algorithm (GA) procedure. The convergence condition can be different, either several entire generations or a fitness value improvement after a few generations. (Figure 2, p. 4, [313]).

5.4.3 Experimental setup

In order to obtain the most accurate results possible, it was decided to use isolated scenarios, a training scenario for the execution of the genetic algorithm, and two test scenarios to obtain the results. The details of each scenario used can be seen in Table 5-3.

The SUMO simulator v1.0.1 was used for the simulation. Python v3.7 together with TraCI (Traffic Control Interface) was used to develop all the algorithms, as well as to control the traffic. The computer used for the simulations and runs had an Intel Xeon CPU with 16 cores at 2.6GHz.

For training and implementation of the genetic algorithm, a training scenario was designed consisting of a grid of 4x4 intersections (16 in total) with 2 lanes in each direction (see Figure 5-5). There was a distance of 300 meters between each intersection. The vehicular flow was designed to be fluctuating, presenting intervals with low, medium, and high vehicular load (600, 1200, and 1600 veh/hour/lane respectively) in addition to presenting intervals of symmetric and asymmetric traffic. A representation of the simulated vehicular flow in the training scenario can be seen in Figure 5-6. The three flow values (low, medium, and high) can be seen, as well as the times when symmetric (e.g., hours 0, 1, and 2) and asymmetric (e.g., hours 3, 4, and 5) traffic existed. The North (N) and South (S) branches were imposed to have the same vehicular flow, as well as the East (E) and West (W) branches. A total of 10 hours were simulated with the traffic distribution shown in Figure 5-6.

TABLE 5-3. Characteristics of the training and the testing scenarios, (Table 2, p. 5, [313]).

Scenario	Number of Intersections	Intersection Layout	Distance Between Intersections	Simulation Duration
Train	16	4×4	300m	10h
Test1	5	1×5	200m	12h
Test2	100	10×10	250m	12h

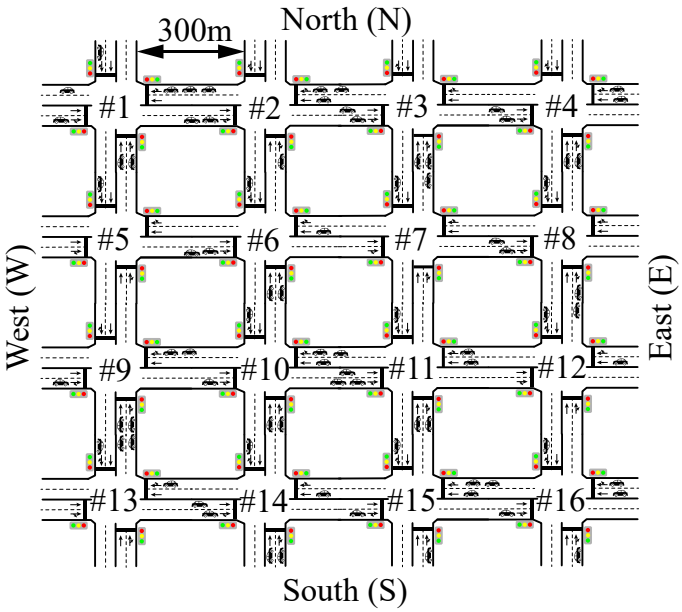


Figure 5-5. Simulated topology for the training scenario: Manhattan 4×4 network with 300m between each intersection. (Figure 3, p. 5, [313]).

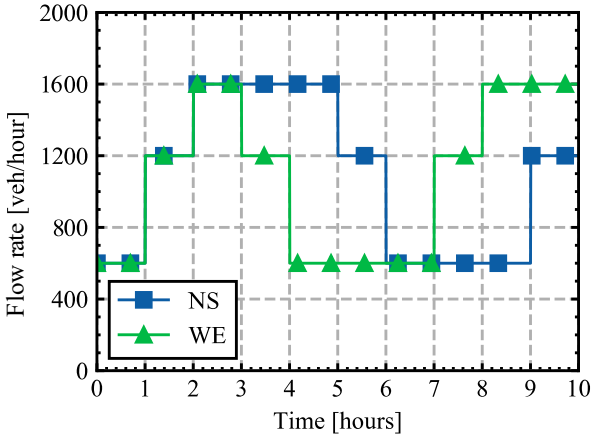
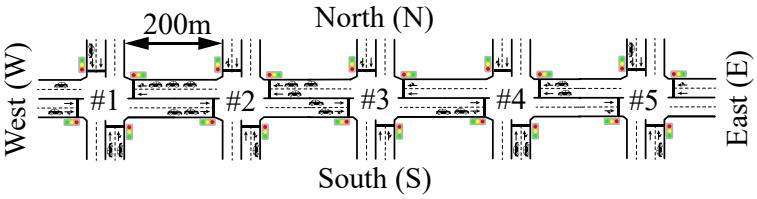


Figure 5-6. Vehicle flow rate per branch used in the training scenario. (Figure 4, p. 5, [313]).

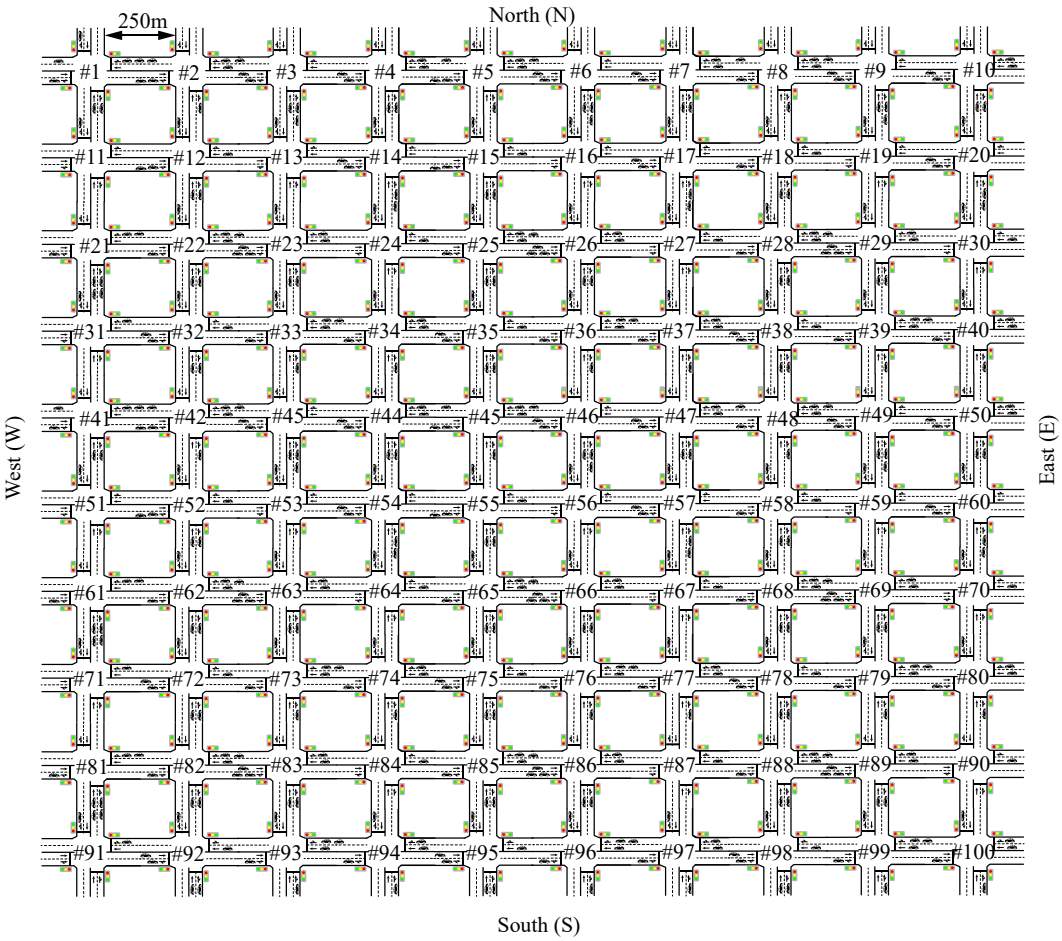
Within the test scenarios, it was decided to design two different scenarios to obtain a greater variety of results. The purpose of these scenarios was to evaluate the performance of iREDVD once trained in scenarios never seen before, with new conditions. Test scenario 1 consisted of a large avenue composed of 5 intersections (1x5) with a separation between intersections of 200 meters. This scenario can be seen in Figure 5-7 a. On the other hand, test scenario 2 consisted of a grid of 10x10

intersections (in total 100 intersections separated by 250 meters). This test scenario number 2 can be seen in Figure 5-7 b.

The same flow distribution was used in the two test scenarios. This can be seen in Figure 5-8. As can be seen, this flow distribution presents a much more chaotic, complex, and realistic behavior, with intervals with low, medium, and high flows (700, 1000, and 1800 veh/hour/lane), and slow and fast variations of these flows. Twelve hours of traffic have been simulated with the flow distribution shown in Figure 5-8.



(a) Test Scenario 1



(b) Test Scenario 2

Figure 5-7. Simulated topology for the testing scenarios: (a) 1x5 network with 200m between each intersection; (b) Manhattan 10x10 network with 250m between each intersection Vehicle flow rate per branch used in the training scenario. (Figure 5, p. 6, [313]).

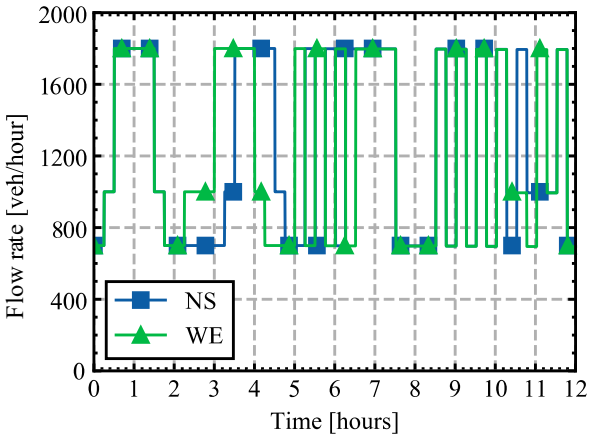


Figure 5-8. Vehicle flow rate per branch used in the testing scenarios. (Figure 6, p. 6, [313]).

It was decided to use the distribution of vehicles provided by the Dirección General de Tráfico for the city of Madrid (Spain) [362] to obtain the most accurate results possible. This distribution can be checked in Table 5-4. In addition, to model vehicle behavior as realistically as possible, it was decided to use the HBEFA pollution model [363]. HBEFA contains a large database of fuel consumption and air pollutant data for a large number of vehicles in a wide variety of states, considering a wide variety of conditions (hot/cold start, temperature, humidity, evaporation, etc.).

TABLE 5-4. Vehicle fleet distribution. (Table 3, p. 5, [313]).

Vehicle type	Proportion	Fuel type
Car	30%	Gasoline
Car	40%	Diesel
Motorcycle	10%	Gasoline
Moped	10%	Gasoline
Van	5%	Diesel
Bus	5%	Average of all fuel types

All intersections were composed of 2 lanes in each direction, where only forward and right turns were allowed. In addition, each intersection was regulated by 4 traffic lights (one for each branch) where the opposite traffic lights were synchronized (North with South and East with West) and the rest in opposite phase, that is, when the North traffic light was green, the South was also green and the East and West traffic lights were red. When North and South were red it was because East and West were green. A representation of the intersections used can be seen in Figure 5-9. In addition, an example cycle for an intersection can be seen in Figure 5-10, where the yellow time interval was set to 3 seconds, and the intersection clearance interval (all red) was set to 2 seconds.

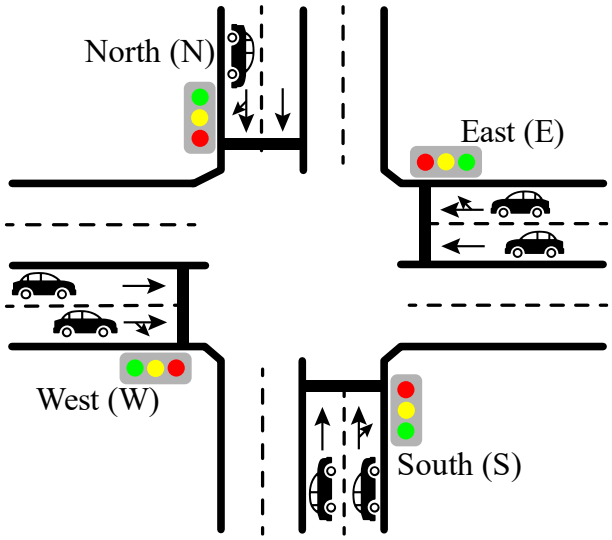


Figure 5-9. An example of a four-way signaled intersection, as used in this study. (Figure 7, p. 6, [313]).

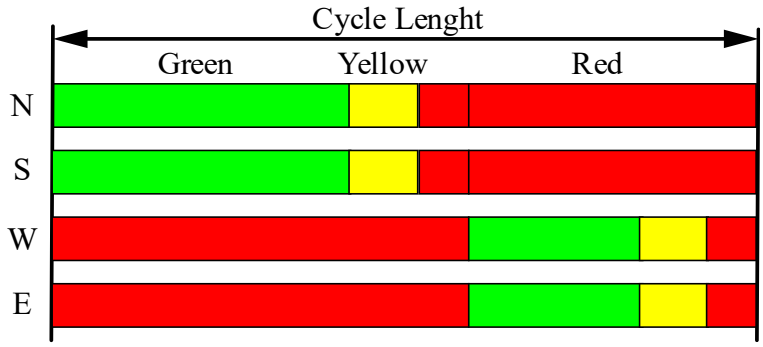


Figure 5-10. Example of a cycle in a traffic light. N=North; S=South; W=West; E=East. (Figure 8, p. 6, [313]).

To compare the performance that iREDVD was able to offer, it was decided to use other widely used traffic light control algorithms, such as the fixed cycle time (FX) and green wave (GW) algorithms, which feature an offset between neighboring intersections to facilitate traffic. The cycle times analyzed were 30, 45, and 60 seconds, as they are the most commonly used in real scenarios. The nomenclature used to denote each algorithm analyzed was expressed as the algorithm employed coupled with the cycle time; thus, for example, GW45 corresponds to the green wave algorithm with a cycle time of 45 seconds. In all cases, 10 experiments were performed for each algorithm studied, obtaining the mean value and its standard deviation for each metric studied.

The metric used to test the fitness of the solutions found by the genetic algorithm was the waiting time per vehicle intersection. In addition, to show the overall performance of the tested algorithms, other metrics such as travel time, average speed, CO, CO2, HC, PMx and NOx emissions, and fuel consumption were analyzed.

5.4.4 Results

Training scenario

Under this scenario, the process of determining the best parameters of the REDVD was carried out using the genetic algorithm explained above, calling this new version of the control algorithm iREDVD. The evolution of the fitness level (average waiting time per intersection) reached by iREDVD during the generations of the genetic algorithm can be seen in Figure 5-11. Starting from generation 5, the results are very promising, but in order to find a robust solution set, it was decided to simulate up to 20 simulations. Once optimized, the control process performed by iREDVD is immediate, not having to run an artificial intelligence algorithm, as it relies on a lightweight and simple control algorithm.

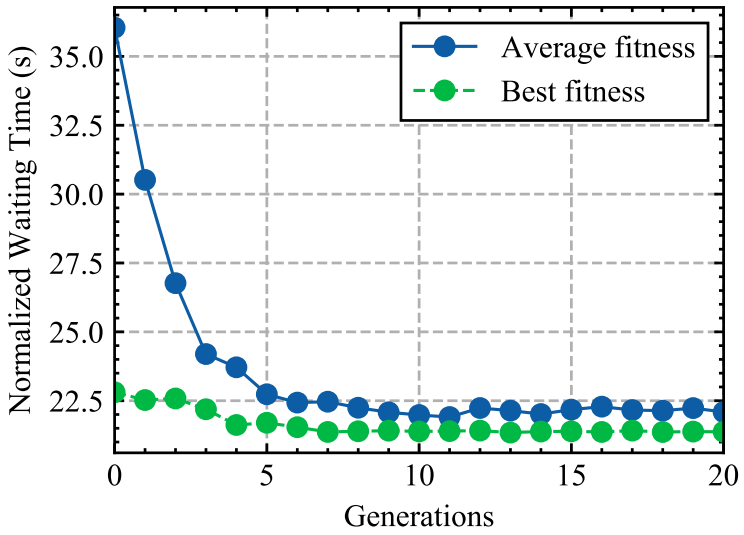


Figure 5-11. Optimization process. "Best fitness" is the individual with the lowest normalized waiting time and "Average fitness" is the average normalized waiting time of all populations. X-Axis represents the generations. (Figure 9, p. 7, [313]).

The values obtained by the genetic algorithm can be seen in Table 5-5. In this table, very interesting values can be seen. For example, the parameter *limdec* controls how many times the green time of the traffic lights should be reduced consecutively to reduce the total cycle time of the intersection. It can be observed that *limdec* presented a value equal to 1, which means that when a reduction in traffic flow is detected, even if it is slight, it is better to shorten the total cycle time quickly. Another interesting value is found in the parameter *liminc*, which indicates how many times the green time of the traffic light must increase in order to increase the total cycle time of the intersection. As can be noted, the value obtained was 5, which indicates that the iREDVD must be very safe (the flow must be much higher than what the intersection is capable of handling) before the total cycle time increases. More information about the parameters of REDVD algorithm can be found in [322].

TABLE 5-5. REDVD parameters. (Table 4, p. 5, [313]).

Parameter	Optimal Value
minth	10
maxth	25
min_greentime	15
max_greentime	60
min_cycle	45
max_cycle	130
delta	10
delta_cycle	15
maxp	0.85
wq	0.7
limin	5
limdec	1

After the optimization process, the performance of iREDVD was analyzed and compared with the other previously mentioned control algorithms. The results in this scenario can be seen in Table 5-6.

Comparing iREDVD with the traditional control algorithms (Fixed 30, Fixed 45, Fixed 60, GreenWave 30, GreenWave 45, and GreenWave 60), it can be evidenced that iREDVD was able to obtain an improvement of more than 50% (reduction of almost 25 seconds) in the optimized metric (average vehicle waiting time per intersection). In other indirectly optimized metrics improvements ranging from 7% to 32% can be seen, being especially remarkable the reduction of more than 32% in CO emissions, as well as 20% in HC and 17% in PMx.

Concerning the original REDV algorithm proposed in [322], an even greater improvement is noticed than with previous algorithms. This reinforces the need to adjust the parameters of advanced control algorithms in complex scenarios, which can lead to catastrophic results if the algorithms are not adjusted correctly.

Lastly, compared to the original REDVD variant, also presented in [322], there was an improvement of about 27% of the optimized metric. This meant a reduction of 8 seconds in the average waiting time per intersection. Observing other indirectly optimized metrics, highlights how iREDVD can outperform REDVD in all these metrics in improvements ranging from 7% to 45%. Furthermore, the reduction in fuel consumption (7%) and the reduction in pollutant emissions (7%-13%) are also remarkable.

Testing scenarios

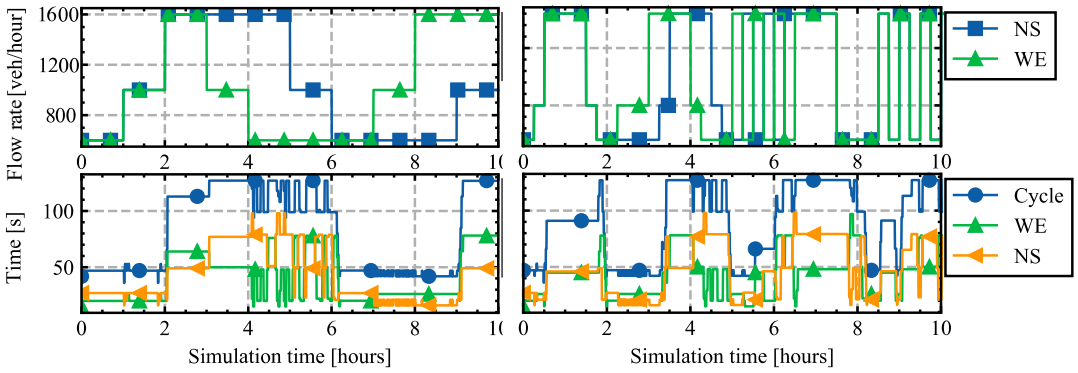
Under this scenario, the process of determining the best parameters of the REDVD was carried out.

After carrying out the optimization process and demonstrating the advantages that iREDVD could offer over other control algorithms, iREDVD was tested in two test scenarios. The results obtained can be seen in Table 5-7 and Table 5-8. It is worth noting that the improvements offered by iREDVD are very similar to those obtained in the training scenario, demonstrating that there is no over-fitting of iREDVD parameters over the training scenario. This also corroborates the flexibility of iREDVD to adjust to new conditions never seen before, showing its robustness to be applied in real deployments.

From the results tables, we can highlight how iREDVD was able to reduce the average waiting time between 34% and 49% in test scenario 1, and between 78% and 82% in test scenario 2. This improvement was also reflected in other metrics analyzed, such as the reduction of CO emissions by around 25% or the reduction of fuel consumption by around 6% and 17%.

Final remarks

The exceptional performance that iREDVD was able to provide is mainly due to the quick cycle time adjustment as a function of the flow through each intersection. This behavior can be seen in Figure 5-12a and Figure 5-12b, where it is highlighted how iREDVD proactively adapted the traffic light cycle time as a function of the simulated traffic flow, both in the training scenario (Figure 5-12 a) and in the test scenario (Figure 5-12 b) in the face of unknown situations. It can be observed that the traffic light cycle increases as vehicle flow increases, and vice versa, as well as allowing controlling the branches independently, adjusting to the traffic conditions on each branch for symmetrical (Figure 5-12 b hour 1) and asymmetrical (Figure 5-12 b hour 4) traffic. This independence allowed iREDVD a better distribution of the total cycle time, as well as a notable reduction of the waiting time seen previously.



(a) *Training scenario flow* (b) *Testing scenario flow*

Figure 5-12. Vehicular flow rate simulated, time for each branch, and total cycle for intersection 1 vs. simulation time: (a) Train scenario; (b) Test scenarios (Figure 10, p. 7, [313]).

TABLE 5-6. Training scenario results. (Table 5, p. 8, [313]).

Algorithm		Avg. norm. waiting time (s)	Avg. trip time (s)	Avg speed (m/s)	Avg. CO emissions (mg)	Avg. CO_2 emissions (g)	Avg. HC emissions (mg)	Avg. PMx emissions (mg)	Avg. NOx emissions (mg)	Avg. Fuel consumption (ml)
Traditional	FX30	51,5 ± 0,11	213,96 ± 0,23	7,25 ± 0,01	6642,68 ± 21,51	783,90 ± 4,49	131,83 ± 1,08	62,75 ± 0,56	3988,72 ± 31,07	315,59 ± 1,79
	FX45	48,3 ± 0,04	201,81 ± 0,09	7,61 ± 0.02	6184,13 ± 24,62	739,80 ± 3,07	122,93 ± 0,22	58,39 ± 0,14	3749,9 ± 15,14	297,69 ± 1,11
	FX60	51,64 ± 0,06	204,76 ± 0,10	7,53 ± 0.01	6463,22 ± 50,19	739,23 ± 2,78	124,46 ± 1,42	58,87 ± 0,69	3754,17 ± 26,77	297,47 ± 1,19
	GW30	46,13 ± 0,08	203,29 ± 0,11	7,65 ± 0.04	6016,83 ± 36,41	751,81 ± 2,05	122,43 ± 0,38	58,54 ± 0,16	3788,61 ± 14,89	302,59 ± 0,71
	GW45	50,02 ± 0,04	206,29 ± 0,07	7,64 ± 0.01	6427,96 ± 20.95	748,24 ± 2,40	125,86 ± 0,89	59,63 ± 0,43	3796,72 ± 19,92	301,2 ± 0,99
	GW60	47,4 ± 0,07	199,45 ± 0,07	7,82 ± 0.02	6178,43 ± 18,74	725,66 ± 5,34	120,27 ± 1,06	56,98 ± 0,52	3665,26 ± 34,72	291,95 ± 2,11
REDV original		62,07 ± 4,01	234,22 ± 7,61	7,22 ± 0,09	7560,28 ± 363,68	819,66 ± 15,15	143,19 ± 5,02	67,75 ± 2,21	4221,82 ± 96,23	4221,82 ± 96,23
REDVD original		29,58 ± 0,33	184,06 ± 0,67	8,27 ± 0,03	4708,8 ± 27,36	723,41 ± 1,67	107,39 ± 1,00	52,51 ± 0,51	3573,78 ± 16,83	3573,78 ± 16,83
iREDVD		**21,37 ± 1,86**	**167,2 ± 2,91**	**9,14 ± 0,16**	**4074,01 ± 135,67**	**672,11 ± 10,13**	**96,01 ± 2,35**	**46,96 ± 1,15**	**3275,5 ± 59,96**	**3275,5 ± 59,96**
Improvement of **iREDVD** vs traditional. [a]										
Improvement Abs		-24,76	-32,25	1,32	-1942,82	-53,54	-24,27	-10,02	-389,76	-21,83
Improvement %		53,67	16,16	16,87	32,28	7,37	20,18	17,58	10,63	7,47
Improvement of **iREDVD** vs REDV orig.										
Improvement Abs		-40,7	-67,02	1,92	-3486,27	-147,54	-47,19	-20,79	-946,32	-59,92
Improvement %		65,57	28,61	26,59	46,11	18,01	32,96	30,68	22,42	18,15
Improvement of **iREDVD** vs REDVD orig.										
Improvement Abs		-8,21	-16,86	0,87	-634,79	-51,29	-11,39	-5,55	-298,28	-20,85
Improvement %		27,75	9,16	10,51	13,48	7,09	10,60	10,56	8,34	7,16

[a] Improvement compared to the best traditional algorithm. Note: *avg* ± *std* of 10 tests.

TABLE 5-7. Testing scenario 1 results. (Table 6, p. 9, [313]).

Algorithm		Avg. norm. waiting time (s)	Avg. trip time (s)	Avg speed (m/s)	Avg. CO emissions (mg)	Avg. CO_2 emissions (g)	Avg. HC emissions (mg)	Avg. PMx emissions (mg)	Avg. NOx emissions (mg)	Avg. Fuel consumption (ml)
Traditional	FX30	21,22 ± 0,25	169,9 ± 1,21	7,34 ± 0,02	5395,04 ± 65,59	597,56 ± 4,11	102,91 ± 0,73	48,99 ± 0,32	3073,18 ± 22,29	240,61 ± 1,6
	FX45	19,2 ± 0,03	150,47 ± 0,16	7,72 ± 0,01	4756,42 ± 23,49	551,99 ± 1,54	92,4 ± 0,27	44,06 ± 0,15	2814,95 ± 5,68	222,12 ± 0,58
	FX60	21,74 ± 0,01	153,05 ± 0,08	7,58 ± 0,02	5047,54 ± 20,13	555,22 ± 3,02	95,51 ± 0,8	45,26 ± 0,42	2844,12 ± 23,67	223,56 ± 1,17
	GW30	17,68 ± 0,39	154,05 ± 1,79	7,81 ± 0,04	4586,92 ± 69,72	558,36 ± 3,49	91,28 ± 0,91	43,83 ± 0,41	2829,06 ± 20,56	224,73 ± 1,4
	GW45	19,25 ± 0,03	151,67 ± 0,15	7,83 ± 0,05	4803,28 ± 19,67	554,43 ± 3,27	92,62 ± 0,74	44,21 ± 0,38	2825,08 ± 21,72	223,09 ± 1,28
	GW60	21,36 ± 0,02	151,66 ± 0,12	7,75 ± 0,03	4919,4 ± 18,56	555,01 ± 3,09	93,9 ± 1,19	44,67 ± 0,55	2839,66 ± 22,56	223,32 ± 1,19
REDV original		25,38 ± 0,71	152,55 ± 2,94	7,73 ± 0,05	4663,73 ± 120,77	547,49 ± 5,15	91,42 ± 1,91	43,61 ± 0,86	2789,18 ± 34,95	220,32 ± 2,14
REDVD original		17,66 ± 0,47	172,52 ± 2,99	7,49 ± 0,08	4771,9 ± 84,21	580,91 ± 7,70	95,48 ± 1,86	45,93 ± 0,91	2953,28 ± 51,28	233,81 ± 3,08
iREDVD		**11,6 ± 0,27**	**132,6 ± 1,65**	**8,38 ± 0,06**	**3406,27 ± 66,39**	**511,34 ± 3,07**	**77,09 ± 1,56**	**37,62 ± 0,7**	**2536,01 ± 23,01**	**205,74 ± 1,36**
Improvement of **iREDVD** vs traditional. [a]										
Improvement Abs		-6,08	-17,87	0,55	-1180,65	-40,64	-14,19	-6,21	-278,94	-16,38
Improvement %		34,38	11,87	7,02	25,73	7,36	15,54	14,16	9,91	7,37
Improvement of **iREDVD** vs REDV orig.										
Improvement Abs		-13,78	-19,95	0,65	-1257,46	-36,14	-14,33	-5,99	-253,17	-14,58
Improvement %		54,29	13,07	8,40	26,96	6,60	15,67	13,73	9,07	6,61
Improvement of **iREDVD** vs REDVD orig.										
Improvement Abs		-6,06	-39,92	0,89	-1365,63	-69,57	-18,39	-8,31	-417,27	-28,07
Improvement %		34,31	23,13	11,88	28,61	11,97	19,26	18,09	14,13	12,0

v Improvement compared to the best traditional algorithm. Note: *avg* ± *std* of 10 tests.

TABLE 5-8. Testing scenario 2 results. (Table 7, p. 10, [313]).

Algorithm		Avg. norm. waiting time (s)	Avg. trip time (s)	Avg speed (m/s)	Avg. CO emissions (mg)	Avg. CO_2 emissions (g)	Avg. HC emissions (mg)	Avg. PMx emissions (mg)	Avg. NOx emissions (mg)	Avg. Fuel consumption (ml)
Traditional	FX30	139,05 ± 0,04	428,63 ± 0,43	6,49 ± 0,01	14273,45 ± 55,48	1508,16 ± 10,14	266,46 ± 1,25	125,60 ± 0,33	7738,98 ± 7,48	608,09 ± 0,87
	FX45	131,08 ± 0,07	400,10 ± 0,85	6,85 ± 0,02	13064,96 ± 44,84	1415,19 ± 4,25	246,19 ± 0,84	116,05 ± 1,15	7274,22 ± 10,44	569,66 ± 2,15
	FX60	134,60 ± 0,21	399.23 ± 0,37	6,90 ± 0,05	13388,89 ± 14,35	1396,32 ± 7,19	246,41 ± 3,44	115,58 ± 0,88	7181,57 ± 8,21	561,98 ± 1,49
	GW30	122,31 ± 0,18	403,24 ± 0,59	6,69 ± 0,05	12634,05 ± 43,01	1443,93 ± 13,25	245,45 ± 2,25	116,67 ± 0,91	7738,98 ± 4,97	581,48 ± 0,45
	GW45	136,20 ± 1,01	407,89 ± 0,89	6,97 ± 0,02	13394,84 ± 24,12	1435,91 ± 6,84	252,15 ± 0,98	118,97 ± 1,01	7274,22 ± 11,02	577,83 ± 1,04
	GW60	122,79 ± 0,50	382,22 ± 1,21	7,28 ± 0,01	12415,55 ± 33,21	1348,19 ± 9,50	231,86 ± 1,21	109,01 ± 2,45	7185,77 ± 6,81	542,51 ± 0,76
REDV original		140,69 ± 2,15	421,36 ± 8,16	5,98 ± 0,11	15584,88 ± 33,86	1550,40 ± 3,46	308,69 ± 3,48	130,18 ± 2,38	7911,04 ± 15,01	624,14 ± 1,24
REDVD original		99,22 ± 8,33	374,45 ± 5,74	8,12 ± 0,18	8847,37 ± 76,84	1321,84 ± 2,62	212,11 ± 1,12	99,01 ± 1,22	6812,88 ± 21,66	539,69 ± 4,18
iREDVD		**21,45 ± 0,24**	**286,85 ± 0,15**	**9,39 ± 0,11**	**6764,93 ± 27,11**	**1286,49 ± 0,96**	**178,71 ± 1,52**	**89,27 ± 0,69**	**6249,30 ± 25,21**	**517,07 ± 0,59**
Improvement of **iREDVD** vs traditional. [a]										
Improvement Abs		-101,34	-95,37	2,11	-5650,62	-61,70	-53,15	-19,74	-932,27	-25,44
Improvement %		82,46	24,95	28,98	45,51	4,58	22,92	18,11	12,98	4,69
Improvement of **iREDVD** vs REDV orig.										
Improvement Abs		-119,24	-134,53	3,41	-8829,95	-263,91	-128,98	-40,91	-1661,74	-107,07
Improvement %		84,75	31,93	57,02	56,62	17,02	42,11	31,43	21,01	17,15
Improvement of **iREDVD** vs REDVD orig.										
Improvement Abs		-77,77	-87,60	1,27	-2082,44	-35,35	-33,39	-9,74	-563,58	-22,62
Improvement %		78,38	23,39	16,64	23,54	2,67	15,74	9,83	8,27	4,19

[a] Improvement compared to the best traditional algorithm. Note: *avg* ± *std* of 10 tests.

5.4.5 Conclusions

The use of advanced traffic light control systems integrated in ITS require perfect tuning to control traffic efficiently and safely. As this research study has shown, poor tuning of control algorithms can lead to catastrophic situations, resulting in traffic situations that ITS are unable to manage.

Considering this imperative to control multiple unexpected situations and anticipate unexpected events, the work seen in this section used an evolutionary algorithm in order to find the best parameter settings for the advanced traffic light control algorithm REDVD. After this optimization process using GAs, the new optimized version of REDVD (iREDVD) was tested in highly complex traffic scenarios and with traffic scenarios never seen during the optimization process, thus eliminating the training scenario bias. iREDVD was compared both to widely used traditional control algorithms (Fixed and Green Wave) and to the versions on which iREDVD is based (REDV and REDVD). The metric directly optimized by the genetic algorithm was the average waiting time per vehicle intersection, in addition, other indirectly optimized metrics such as travel time, travel speed, emissions (namely CO, CO2, HC, PMx, and NOx), and fuel consumption were analyzed.

In light of the results, we have been able to draw several conclusions. The first and most important is the importance of tuning the internal parameters of advanced control algorithms such as REDVD. As can be seen in Tables 5-6, 5-7, and 5-8, when we compare REDVD without optimization with iREDVD we can see that the travel time metric can be reduced by 23%, in addition to many other improvements, demonstrating the importance of optimization.

On the other hand, if we compare iREDVD with widely used traditional control techniques, improvements range from 34% to 80% in several metrics, showing reductions of between 6 and 100 seconds in the vehicle waiting time per intersection in the scenarios analyzed.

Finally, if we compare iREDVD with the original REDVD, we can see that it was able to reduce the waiting time at each intersection by 34% to 78% in the test scenarios. In addition, iREDVD was able to achieve a reduction in both pollutant emissions and fuel consumption between 2%-28% and 4%-12% of vehicles, respectively.

Based on the results obtained in this section, in the design and deployment of new traffic light control systems in large-scale intersections iREDVD can offer a great advantage over other control systems by allowing to control traffic in a very efficient and anticipatory way through the deployment of low cost, low requirements, and low power IoT devices, outperforming not only iREDVD but also other known traffic management methods in all the metrics analyzed.

5.5 Conclusions to this chapter

ITS are set to revolutionize the smart cities of the future, largely through the incorporation of new technologies such as IoT devices, big data, AI, and new 5G/6G communication standards.

Due to the constant increase in the number of vehicles, the massification of large cities, the enormous progress of intelligent control systems, and the great development of mobile communication systems, there is a great demand for intelligent systems capable of orchestrating all users of public roads efficiently and safely. If we focus on intersections regulated by traffic lights, it is imperative to respect safety by improving vehicular flow, obtaining control policies capable of anticipating congestion situations, improving traffic flow, and decreasing pollution and fuel use in cities, which is extremely imperative.

As this section has shown, the basis of these control systems lies in modifying the cycle time of the traffic lights controlling the intersections according to the incoming flow at each intersection to maximize the use of the intersection in an efficient manner. In addition, it has been evidenced that a major issue is the correct optimization and adjustment of these traffic control systems, which leads to complicated situations if comprehensive traffic management is not carried out. This fine-tuning process can be performed through multiple procedures, such as the one seen in this section using genetic algorithms. Thanks to this fine-tuning, it would be possible to achieve an advanced control system, called iREDVD in the previous paper, capable of greatly improving other widely used advanced and traditional control systems.

In conclusion, ITS will bring to future smart cities new advanced tools for the orchestration of all vehicles, which will increase safety, reduce waiting times, and air pollution due to vehicle emissions, and improve the quality of life for all citizens.

5.6 Publications associated with this research

The works related to this chapter are as follows:

Articles:

Guillen-Perez, A.; Cano, M.-D., "Intelligent IoT systems for traffic management: A practical application," *IET Intell. Transp. Syst*., vol. 15, no. 2, pp. 273–285, Feb. 2021, doi: 10.1049/itr2.12021.

2020 Journal Impact Factor (JIF): 2.496. (Q2), Rank: 135/273 in Engineering, Electrical & Electronic.

Congress:

Guillen-Perez, A.; Cano, M.-D., "Optimización de un Sistema Inteligente de Control y Gestión de Transporte en Intersecciones por medio de un Algoritmo Genético", *V Jornadas Doctorales UPCT*, Universidad Politécnica de Cartagena. 2019. Oral communication.

Guillen-Perez, A.; Cano, M.-D., "Influencia del ciclo de trabajo de los semáforos en una intersección simple en múltiples parámetros ante una densidad de tráfico incremental," in *XIV Jornadas de Ingeniería Telemática (JITEL 2019)*, 2019, no. JITEL, pp. 22–24, [Online]. Available: http://jitel2019.i3a.es/.

Chapter 6: Interoperability of Connected Autonomous Vehicles and Intelligent Transportation Systems

6.1 Introduction

After analyzing the different alternatives for vehicle management at urban intersections based on the breakthrough of AI algorithms and communication protocols, this chapter analyzes the Autonomous Intersection Management (AIM) approach. In addition, several AIM approaches using DRL for vehicular control are proposed.

The future success of autonomous vehicles will depend on advances in various components of the driving and control system, as well as understanding and handling unpredictable situations that may arise in complex driving environments. The application of Multi-Agent Deep Reinforcement Learning (MADRL) will enable the development of advanced systems capable of adapting to myriad situations and acting collectively and proactively, anticipating dangerous situations, and ultimately avoiding accidents and increasing fluidity. Using MADRL, new and interesting AIMs capable of controlling autonomous vehicles could be developed with extraordinary results, minimizing waiting times, eliminating accidents, and improving the quality of life of users.

6.1.1 Autonomous Intersection Management

Traffic intersections are the main points of congestion in urban areas and are among the most important places for the planning and management of urban traffic flows, as they are very complex and risky scenarios for urban mobility, due to the hazard of accidents or unexpected interactions between vehicles, pedestrians, or other actors. The main requirement that an intersection must meet is to regulate the way vehicles pass through it in such a way that safety is guaranteed, and flow can be guaranteed.

The development of ITS has become a key strategy to improve transportation safety, as well as to reduce congestion in cities. The advancement and progress of AI is increasing the intelligence and deployment of autonomous vehicles. Autonomous vehicles are expected to contribute significantly to road safety, reducing fatalities and injuries due to human error. The main idea is a system of Connected Autonomous Vehicles (CAVs), which can be interconnected with each other to improve control through collective intelligence.

AIM is an emerging technology that enables advanced control of CAVs at urban intersections through centralized control via an AIM algorithm. This advanced control allows control of the right-of-way of individual vehicles through the application of multi-agent and multivariable optimization rules while

improving the performance offered by traditional intersection control techniques, such as those based on traffic lights, by minimizing collisions and optimizing the overall traffic flow.

AIMs operate mainly based on two modules:

Conflict module

The conflict module determines whether there will be a conflict (accident) between two vehicles that want to cross an intersection. That is, from the routes that the vehicles want to follow, the conflict module will predict the paths that the vehicles will follow, and if two or more vehicles coincide at some point of the intersection in space-time, it will identify that there is at least one conflict. This module can follow several approaches for the identification of conflicts between vehicles: *i) intersection-based* [327], [364], [365], *ii) tile-based* [366]–[369], *iii) conflict point-based* [370]–[374], and *iv) vehicle-based* [375]–[378]. A simple sketch of each approach is shown in Figure 6-1.

The *intersection based* approach laid the foundation for early AIM [327]. In this approach, only one vehicle was allowed within the intersection at a time, regardless of the route taken by the vehicles. Although this approach was very simple, it had serious obvious disadvantages.

Following this, *tile based* and *conflict-point based* approaches emerged. The former checked conflicts within a grid, where two vehicles could not coincide in the same tile simultaneously in space-time. The main problem with this approach is that there was a large trade-off between the granularity chosen, the overall intersection performance, and the complexity of conflict checking. The second approach only considered the conflict points of the routes followed by the vehicles. If two vehicles coincided at the same conflict point simultaneously in space-time, a conflict was identified. This second approach reduced the complexity of conflict checking; however, due to the variable geometry of the vehicles, unexpected collisions could occur.

Finally, a much more complex approach is the *vehicle based* approach. In this approach, vehicles have total freedom of movement within the intersection, and the AIM is in charge of finding them a valid route (that does not coincide in space-time with that of other vehicles already assigned) within the intersection according to their requirements. This option is undoubtedly the one that offers more freedom at the cost of enormous computational capacity requirements since it turns the problem into a multidimensional and multiagent problem of enormous complexity.

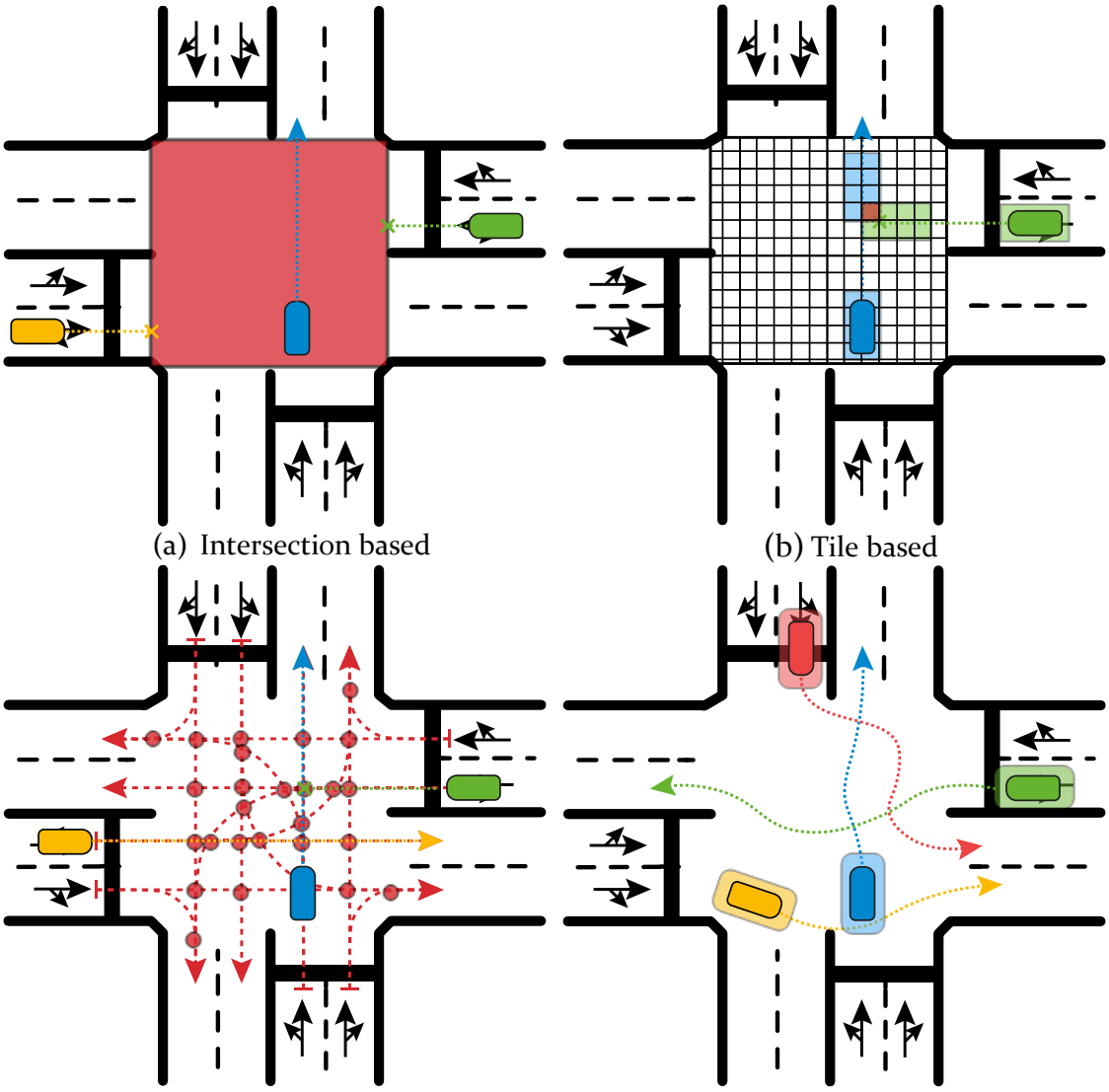


Figure 6-1. Approaches developed for the conflict module of AIM: (a) Intersection based; (b) Tile based; (c) Conflict-Point based; (d) Vehicle based. (Figure 2, p. 3, [379]).

Priority module

Once conflicts between vehicles have been identified, the priority module is responsible for deciding what to do with those vehicles to resolve them. The priority module acts on the state of the vehicles (e.g., their speed, acceleration, etc.) to try to resolve conflicts as fairly as possible, ensuring that no vehicle is stuck infinitely.

In addition, this module is responsible for obtaining a right-of-way for vehicles, assigning priorities among the vehicles wishing to cross the intersection following a priority policy. This policy can follow different approaches:

- On a First-Come, First-Served (FCFS) basis [327], [370], [380], [381].
- Based on vehicle status (speed, waiting time, etc.) or intersection status (number of vehicles per branch, average waiting time per branch, etc.). For

example, the Fast First Service (FFS) policy [372], where vehicles arriving the fastest at the intersection have the highest priority, or the Long Queue First (LQF) policy [368], where vehicles in the longest entry queue have the highest priority.

- Using heuristics such as Dynamic Programming (DP) or Mixed Integer Linear Programming (MILP), where the policy is found using a series of equations and conditions that model the behavior of the intersection and vehicles [366], [374], [378], [382]–[385].
- Through auctions [364], [386], in which vehicles are charged with bidding for right-of-way priority based on an internal economy for the AIM. Vehicles with the highest bid pass first.
- Making use of artificial intelligence mechanisms such as genetic algorithms [387] or RL [368].

A representation of AIM with both modules (conflict and priority) can be seen in Figure 6-2.

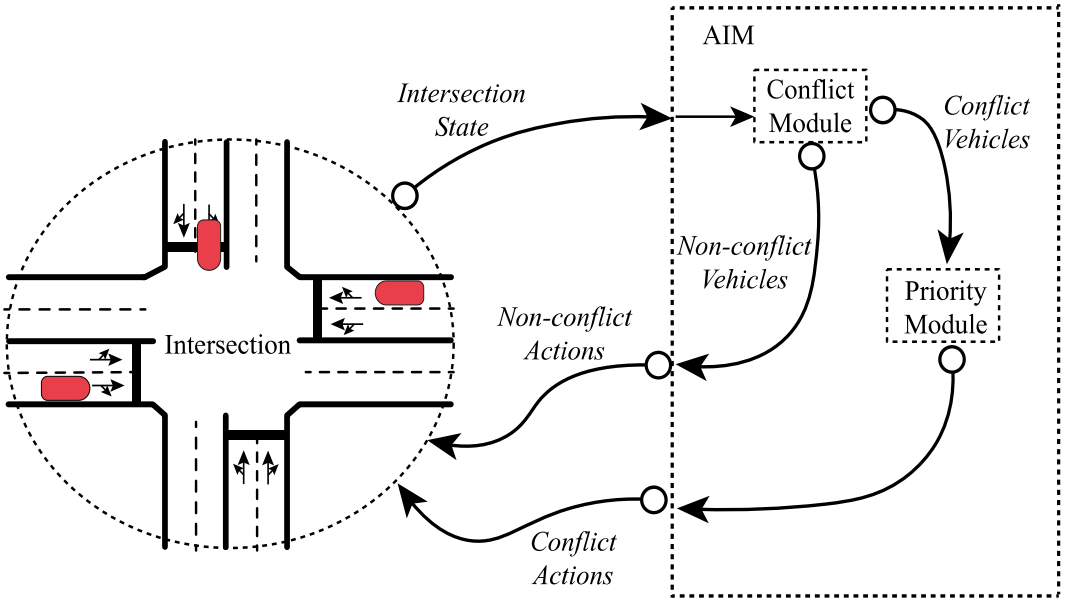


Figure 6-2. AIM basic operation. AIM includes a *Conflict module* and a *Priority module* to control CAVs. (Figure 1, p. 3, [379]).

6.1.1. Deep Reinforcement Learning

RL [11] is a domain of machine learning in which an agent learns to perform a task in an environment where it can perform an action and will receive a reward for the action performed. The agent's goal is to perform actions that maximize the accumulated rewards during the entire task, which is known as expected discounted total rewards.

When an RL problem satisfies the Markov property, i.e., the future state depends only on the current state and current actions, but not on the past, it can be formulated as a Markov Decision Process (MDP) and can be defined by the 5-tuple $\langle S, A, R, T, \gamma \rangle$, where S represents a set of states of an environment, A represents the set of actions that the agent can take, T is the transition function $T: S \times A \times S \rightarrow [0,1]$ that determines the transition probability from any state $s \in S$ to any state $s' \in S$ when the action $a \in A$ is taken. R is the reward function $R: S \times A \times S \rightarrow \mathbb{R}$ and $\gamma \in (0,1]$ represents the discount factor that adjusts the trade-off between immediate and future rewards.

Resolving an MDP generates a policy $\pi : S \rightarrow A$, which maps the states $s \in S$ to the actions $a \in A$. An optimal policy π^* maximizes the expected discounted total reward for all states. This approach to finding the optimal policy can be formulated by the state-action value function (*Q-function*) and can be found in Equation 6-1. This *Q-function* determines the expected reward (R) by starting from the state s, taking the action a, and following the policy π.

$$Q^{\pi}(s,a) = \mathbb{E}^{\pi}\left[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t)\right] \qquad (6\text{-}1)$$

Focusing on DRL, the incorporation of NN to traditional RL algorithms was a major breakthrough, greatly speeding up the learning process of these algorithms and allowing their application to tasks that previously seemed impossible because NNs can act functions to approximate the policy to be learned.

Within DRL there are several approaches to find the optimal policy, but the most analyzed in recent years is based on a paradigm composed of two types of NN for continuous control problems: actor and critic. Both the actor (the policy that decides what action to take for each state) and the critic (given a state and an action decided by the actor, it predicts what expected reward, or Q-value, is obtained, indicating to the actor whether the action to be taken can be good or not) are modeled by NNs.

Twin Delayed Deep Deterministic Policy Gradients (TD3)

One of the most advanced algorithms within the actor/critic approach is TD3 [388]. TD3 has become one of the most widely used RL algorithms in continuous control tasks such as robotics or autonomous control. TD3 improves the Deep Deterministic Policy Gradient (DDPG) [39] algorithm in several aspects:

- TD3 uses two critical networks (two *Q-functions*) instead of one (hence "twin") and uses the lower *Q-value* during training, which gives a better approximation, thus improving the stability of the whole algorithm.
- TD3 updates the actor policy (and the target networks) less frequently than those of the critics. In the original paper, the authors recommended

updating the actor policy every two updates of the critics. In this way, TD3 delays the actor updates, the delayed part.

- TD3 adds a small noise to the target action. This makes it more difficult for the policy to exploit *Q-function* errors by smoothing Q along with the action changes.

The TD3 algorithm can be seen as outlined in Algorithm 6-1.

Algorithm 6-1: *TD3*

Initialize critic networks $Q_{\theta_1}, Q_{\theta_2}$ and actor-network π_ϕ with random parameters; θ_1, θ_2, ϕ.
Initialize target networks $\theta_1' \leftarrow \theta_1, \theta_2' \leftarrow \theta_2, \phi' \leftarrow \phi$
Initialize replay buffer $\mathcal{B}$
for timestep $t \in \{1, \dots, T\}$ **do**:
  Select action with exploration noise $a \sim \pi_\phi(s) + \varepsilon,\ \varepsilon \sim \mathcal{N}(o, \sigma)$ and observe reward r and new state s'
  Store transition tuple (s, a, r, s') in $\mathcal{B}$
  Sample mini-batch of N transitions (s, a, r, s') from $\mathcal{B}$
  $\tilde{a} \leftarrow \pi_{\phi'}(s') + \varepsilon,\ \varepsilon \sim clip(\mathcal{N}(o, \sigma), -c, c)$.
  $y \leftarrow r + \gamma\, min_{i=1,2}(Q_{\theta_i'}(s', \tilde{a}))$.
  Update critics $\theta_i \leftarrow argmin_{\theta_i}\left(N^{-1} \sum \left(y - (Q_{\theta_i}(s, a)\right)^2\right)$
  it $t\ mod\ d$ **then**:
    Update ϕ by the deterministic policy gradient:
    $\nabla_\phi J(\phi) = N^{-1} \sum \nabla_a Q_{\theta_1}(s, a)|_{a=\pi_\phi(s)} \nabla_\phi \pi_\phi(s)$
    Update target networks:
    $\theta_i' \leftarrow \tau\theta_i + (1 - \tau)\theta_i'$.
    $\phi' \leftarrow \tau\phi + (1 - \tau)\phi'$.
  end if
end for

Prioritized Experience Replay (PER)

Prioritized Experience Replay (PER) [53] is a technique used in DRL to speed up training. In DRL, a *replay buffer* is used to store the previous experiences seen during training. These experiences are composed by the 4-tuple including states, actions, rewards, and next states (s_t, a_t, r_t, s_{t+1}) are used to approximate the *Q*-values by the DNN. What PER proposes is to use more frequently those actions from which more can be learned. The experiences that PER considers most important to learn from are those in which the error made between the predicted *Q*-value (*Q(s,a)*)and the actual *Q*-value (*Q*(s,a)*) is high. Therefore, by employing PER, the network is able to focus on those experiences where it predicts with high error and minimize its error, reducing the global error and increasing accuracy, reducing training time.

Curriculum Learning for Reinforcement Learning

Curriculum learning [389] consists of training an intelligent agent on a task sequentially, increasing the complexity of the task to be performed little by little. For example, in tasks where a robot has to learn to jump over an obstacle, a curriculum can include: training the robot to stand, then to walk, then to run, and finally to jump over the obstacle. Adoption of the curriculum has been found to accelerate the speed of convergence and may even outperform the final policy performance, although designing an efficient and effective curriculum is not easy; in fact, a bad curriculum can even hinder learning. One of the most widely employed methods is curriculum-based learning through Self-Play. In this approach, the complexity of the scenario to be solved by the agent increases when a policy is reached that obtains stable results, i.e., when after some simulations, the agent is not able to improve the solution found. For example, in object manipulation and classification tasks, different objects can be added when the results of the task are satisfactory and stable results are obtained.

6.2 State of the art

In this section, we will look at works related to the field of AIM.

Autonomous Intersection Management (AIM)

AIMs emerged as intersection traffic control alternatives to traffic lights. These AIMs propose centralized intersection-level control of autonomous vehicles crossing each intersection. At each intersection there is an Intersection Manager (IM) that controls all vehicles crossing the intersection together, achieving higher throughput than if each vehicle is left to act individually.

One of the first works to propose this was developed by Dresner *et al.* [390]. In this work, they proposed that the IM follow the *"First Come, First Served"* (FCFS) policy to find the right-of-way of vehicles. The operation of their proposal consisted of each vehicle asking the IM to reserve a space-time within the intersection. If this reservation did not conflict with any vehicle, the request was accepted, and the vehicle had to follow the route it had requested. In case of potential conflicts with another route already assigned, the vehicle received a denial of the request and had to slow down and try again later to find available time-space. The results showed that the proposed AIM improved flow and lost time over traditional control systems (stop, traffic lights), although it was only tested with simple intersections and low traffic flows.

In later works [391], [392] proposed improvements to this protocol such as the consideration of non-autonomous vehicles (FCFS-LIGHT) as well as emergency vehicles (FCFS-EMERG). Finally, they proposed [393] a capable

algorithm between each of the previously proposed algorithms depending on the needs and the state of each intersection.

In [370], FCFS protocol was further studied and tested against an optimized traditional traffic light signal. The results indicated that FCFS was able to significantly reduce the delay up to 90%.

Huan *et al.* presented an improvement of the FCFS in [394]. In this study, the authors proposed to modify the request denial message, allowing the IM to indicate in this message the recommended deceleration rate to reduce the waiting time. The proposed algorithm showed a waiting time reduction of more than 85%, as well as a fuel consumption reduction of more than 50%.

Since FCFS was not able to group vehicles into platoons and take advantage of the benefits they offer, mechanisms were proposed in [395], [396] that allowed grouping requests from vehicles with similar characteristics (same route, same destinations, etc.). The results showed an improvement in both FCFS and traffic light control, increasing the flow x2 and reducing the time loss up to 85%.

A different approach was followed by Carlino *et al.* in [364]. In this research, they proposed the use of auctions to obtain the right-of-way of vehicles. Vehicles with the highest bids passed first. In this work, a great deal of work was done to test the proposal by testing it in four scenarios simulating four urban cities. The results they obtained showed superior performance of their proposal in three of the four scenarios tested against FCFS and traffic-light based control algorithms. However, despite the goodness of their work, they had serious problems. The main problem is the approach followed using bids, which involve intrinsic economics, with the risk that some vehicles will be left without food with the risk that they will experience indefinite waiting times, in addition to generating a market economy of the currency used, inflation, discrimination, etc.

Within the field of DRL we can find the work proposed by We *et al.* [368]. In this study, they used RL to obtain the right-of-way for each vehicle. The results compared with FCFS as well as with a variant proposed by them, Longest-Queue-First (LQF), showed that thanks to the use of DRL their proposal could reduce the delay by more than 60%.

While the proposed AIMs were impressive, the work of Levin *et al.* [397] showed that, under certain conditions, these new AIMs could provide inadequate behaviors that could lead to inappropriate results. To this end, they subjected several AIMs to various scenarios, such as burst traffic, asymmetric traffic, heavy traffic, or traffic traveling on the main avenue and several secondary approaches. They indicated that AIM control policies, such as FCFS, required detailed and in-depth study before being operational in real control systems. In addition, they noted that the study of these new AIMs required a comprehensive simulation

analysis, in which a large number of scenarios were tested to ensure adequate performance.

From this context, it makes a lot of sense to take advantage of the benefits that DRL can offer for the development of new AIMs, as this approach allows the AIM to learn and acquire an advanced and deep control policy through trial and error. In addition, it is expected that the use of DRL will provide a safer and faster solution that will help overcome the limitations of existing AIM algorithms. In the following sections, we will look at the work related to the proposed AIM.

6.3 RAIM

6.3.1 Introduction

The first approach to an AIM based entirely on deep reinforcement learning was carried out in the work [398]. This research analyzed the use of DRL to design an advanced control policy implementing an AIM, controlling autonomous vehicles within urban intersections.

The main advantage of using DRL is obvious: finding an advanced control policy, based on trial and error, improving the performance of other proposed AIM policies. Thus, this new approach made it possible to unify the advantages offered by DRL with the great opportunities that AIM could offer in the development of new vehicle control systems. This first approach was called Reinforced Autonomous Intersection Management (RAIM).

6.3.2 RAIM - Reinforced Autonomous Intersection Management

RAIM unified the conflict module and the priority module into a single controller, formed by a deep neural network, which would determine the action to be carried out by each vehicle at each time interval, depending on the state of each of the vehicles at the intersection.

The RAIM was considered an *ego*-centered policy. That is, it calculated the action to be performed by the vehicles individually, based on the state of the *ego*-vehicle and the state of the other vehicles. It was something like, based on the state of the *ego*-vehicle and the state of the cooperating vehicles, what action the *ego*-vehicle should perform so that the overall performance would improve. Because of this *ego*-centric policy, RAIM needed to calculate as many speeds as there were vehicles to control.

RAIM considered the status of vehicles that planned to cross each intersection and determined the speed to be followed by the vehicles during the next time interval. This speed ensured collision avoidance and minimized travel time. Due to technical constraints, the RAIM had a limitation with the number of vehicles to be considered simultaneously, allowing a maximum of up to 32 vehicles

in total (the *ego*-vehicle and up to 31 additional vehicles). For situations where there were no additional 31 vehicles, a zero-fill policy was followed because the format of the input data to the control neural network had to have the same form (*n_features* x 32 vehicles).

The features of each vehicle used to indicate its status can be seen in Table 6-1. After correctly encoding the input parameters, each vehicle inputted 14 features (*n_features*) into the first layer of the controller's neural network. The total number of input features for the controller was 448 features (32 vehicles x 14 features/vehicle).

TABLE 6-1. Input Variables and Meaning. (Table 1, p. 5, [398]).

Variable	Description
Relative pos x	Relative position to the center of the intersection on the x-axis
Relative pos y	Relative position to the center of the intersection on the y-axis
Speed	Vehicle speed
Angle	Vehicle orientation angle
Lane	Lane of approach (left lane, right lane)
Way	Way the vehicle will follow (right turn, straight path, left turn)
Queue	Intersection branch through which the vehicle is approaching (North, South, East, West)

The parameters with continuous range (such as position, speed, and angle) were normalized between -1 and 1, and the parameters with discrete values (lane, way, queue) were encoded with one-hot encoding. There are only two lanes because we consider 2 lanes per branch.

The neural network responsible for modeling the RAIM control policy was composed of 4 fully connected layers. The characteristics of each layer can be found in Table 6-2. In addition, a graphical representation of the neural network architecture designed for RAIM can be seen in Figure 6-3. The *ego*-vehicle features are fed into the first 14 neurons of the input layer, and the rest of the neurons are used for the features of the other vehicles.

For the optimization of the controller modeled by the neural network seen above, the reinforcement learning algorithm TD3 [388] was used. In addition, to accelerate the training process, techniques such as curriculum-based learning through *Self-Play* [389] and PER [53] were employed. TD3 is the evolution of the DDPG algorithm, widely studied within the scientific community, and which has become one of the most popular algorithms for continuous control problems within the fields of robotics and autonomous driving. TD3 bases its training on the existence of two different types of networks, *Actor* and *Critic*. The *Actor*-network tries to predict the best action to perform based on the current state of the environment (*Actor*, learns the policy). On the other hand, the *Critic*-network tries to predict what will be the expected future reward obtained by the action predicted by the *Actor*-network (*Critic*, learns the action-value function, *Q-function*).

TABLE 6-2. RAIM neural network architecture summary.

Layer	Number of neurons	Activation function
Input Layer	448	ReLU
Hidden Layer 1	128	ReLU
Hidden Layer 2	64	ReLU
Output Layer	1	Sigmoid

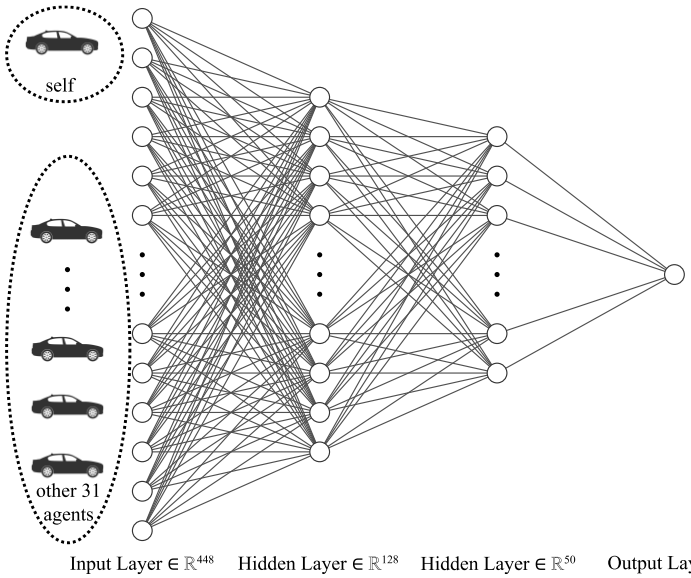


Figure 6-3. RAIM policy neural network architecture. (Figure 2, p. 4, [398]).

6.3.3 Experimental setup

To simulate vehicle behavior as close to reality as possible, the SUMO open-source microscopic traffic simulator was used. SUMO has been widely studied and used in numerous works by the scientific community. For the development and implementation of RAIM, Python 3.7 together with Pytorch 1.5 was used to train the neural network. A 16-core AMD processor and an Nvidia RTX 2080TI graphics card were used as testbed.

Two scenarios were designed, *i)* a training scenario in which RAIM optimization was performed, and *ii)* a test scenario in which the performance offered by RAIM was compared with other traditional traffic light control algorithms (FT) and advanced adaptive control algorithms (iREDVD [313]).

Training scenario

The testing scenario designed consisted of a 4-branch urban intersection, with 2 lanes in each direction. Left turns, right turns, and crossing the intersection in a straight line were allowed. A representation of the simulated intersection can be seen in Figure 6-4.

Each simulation had a duration of 5 minutes, in which the flow of vehicles was increased if, after a series of simulations, the stability objectives were met, following the procedure of learning by curriculum. In each simulation, the

behavior of the vehicles was randomized to obtain a wide range of experiences analyzed by RAIM in order to learn a stable and advanced controller model able to cope with a wide variety of different states never seen before. RAIM began to take control of the vehicles when they approached within 100 meters of the center of the intersection and stopped controlling them when they crossed the intersection. A summary of the parameters used in the simulation can be found in Table 6-3.

As a reward signal, the following rewards were designed:

- -100 (strong negative reward). When the vehicle was involved in a collision.
- +100 (strong positive reward). When crossing the intersection.
- -timestep (weak negative reward). To encourage crossing the intersection as fast as possible, without collision. This reward was given at each simulation timestep.

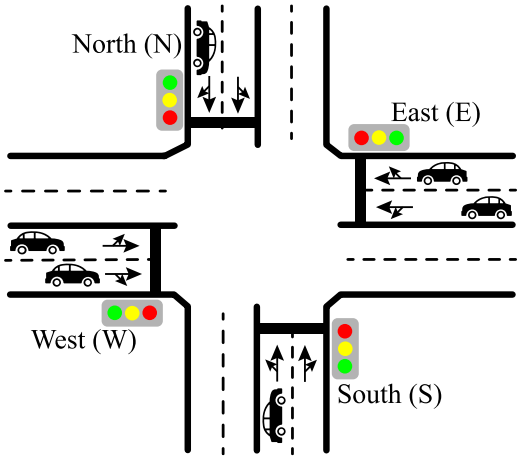


Figure 6-4. Train scenario simulated. (Figure 3, p. 4, [398]).

TABLE 6-3. Summary of simulation setup and RAIM parameters. (Table 2, p. 6, [398]).

Simulator	Simulation timestep	0.25 segs
	Flow step	50 veh/hour
	Minimum flow	25 veh/hour
	Maximum flow	625 veh/hour
	Test duration	5 mins
	Test scenario	4 branches and 2 lanes/way
	Control distance	100 meters
RAIM	Batch size	128
	Gamma	0.99
	Tau	4×10^{-3}
	Learning rate actor	1×10^{-5}
	Learning rate critics	1×10^{-4}
	Weigh decay	1×10^{-6}
	Policy noise	0.2
	Policy Noise Clip	0.3
	Optimizer epochs	3
	TD3 update actor every	2
	PER Memory Size	2^{20}

Testing scenario

The testing scenario consisted of an intersection similar to the training scenario. However, the difference lies in the simulated vehicular flow pattern. This time a fluctuating vehicular flow was simulated, with periods of high, low, symmetrical, and asymmetrical flow. In this way, it was simulated an intersection where more realistic flow has to be coped with. Each simulation was performed 10 times, with a simulated duration of 14 hours. The simulated vehicular flow can be seen in Figure 6-5. As can be seen, there are multiple different situations, low, medium, and high flows, asymmetric and symmetric flows, slow and fast flow variations, etc. This allowed us to evaluate the optimized RAIM control policy in a large number of circumstances as close to reality as possible.

The algorithms with which the RAIM was compared were a traditional fixed traffic light (FT) control algorithm, with different total cycle times (30 s., 45 s., and 60 s.) and the advanced adaptive traffic light control algorithm iREDVD [313] based on queuing theory. The metrics analyzed were several: travel time, waiting time, time loss due to congestion, speed, and energy consumption).

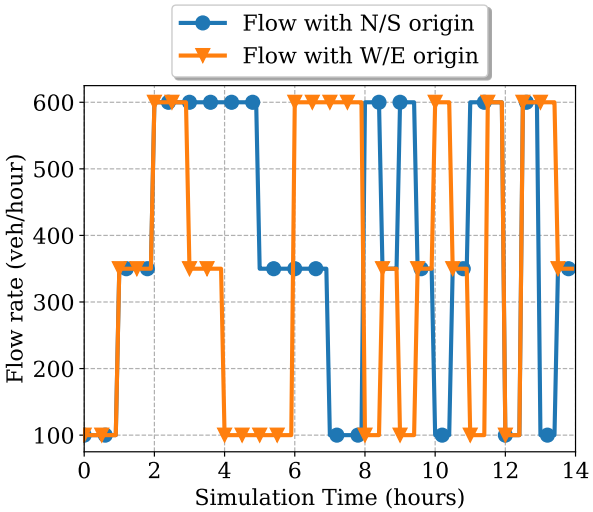


Figure 6-5. Vehicle flow-rate per branch used in the testing scenario. Low = 100 veh/h; Med = 350 veh/hour; High = 600 veh/hour. (Figure 7, p. 7, [398]).

6.3.4 Results

In this section, we present the results obtained in the training scenario, as well as the comparison and discussion of the results obtained in the testing scenario.

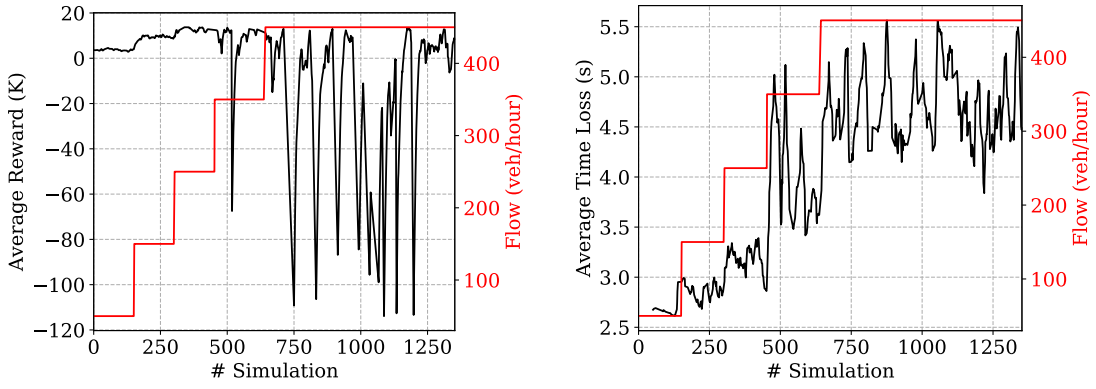
Training scenario

After performing the training process, the results obtained can be seen in Figure 6-6. In this figure, the temporal behavior during the optimization process of the metrics: average reward, and average time loss due to congestion can be found. The values shown are the mean values of the simulations performed, smoothed with a moving average.

The results obtained showed excellent stability as a consequence of the set of algorithms used (PER and curricular learning). It can be observed how, due to the curricular learning process, when the results begin to stabilize, the vehicular flow increases. After this increase, outliers are observed, which are smoothed out by the use of TD3 and PER.

Testing scenario

After the optimization process was performed in the training scenario, the test scenario was simulated. The results obtained by both RAIM and the aforementioned control algorithms can be seen in Table 6-4. As can be seen, RAIM was able to provide superior performance to the other algorithms analyzed in all the metrics studied. RAIM was able to reduce travel time by up to 52%, as well as reduce time loss by up to 84%. Focusing on the energy consumption metrics, the use of RAIM resulted in vehicles using up to 71% less energy. The results obtained, while remarkable, are due to the extensive training and advanced control policy learned by the DRL algorithm, TD3.



(a) (b)

Figure 6-6. Training results. We plot the smoothed mean with an exponential moving average. RAIM was training for 82 hours: (a) Reward; (b) Time Loss. (Figure 5, p. 8, [398]).

TABLE 6-4. Testing scenario results. (Table 3, p. 7, [398]).

Algorithm	Travel time (s)	Waiting time (s)	Time loss (s)	Speed (m/s)	Elect. Cons. (KW/h)
FT (30)	62.48 ± 3.58	10.98 ± 3.72	18.98 ± 4.11	9.96 ± 6.44	1.84 ± 0.25
FT (45)	66.98 ± 4.55	15.21 ± 3.84	23.40 ± 5.54	9.57 ± 5.51	1.79 ± 0.18
FT (60)	71.32 ± 6.31	19.39 ± 4.41	27.78 ± 4.97	9.27 ± 4.15	1.94 ± 0.33
iREDVD	57.49 ± 3.55	5.86 ± 2.21	13.96 ± 3.88	10.49 ± 4.81	1.66 ± 0.12
RAIM	33.89 ± 3.10	0.71 ± 0.19	4.41 ± 1.44	11.81 ± 3.36	0.55 ± 0.09
Improvem.					
Abs	[37.43, 23.60]	[18.66, 5.15]	[23.37, 9.55]	[2.54, 1.32]	[1.39, 1.11]
%	[52.59, 41.05]	[96.34, 87.88]	[84.13, 68.41]	[127.40, 112.58]	[71.65, 66.87]

The improvement shows the range between [best-case, worst-case]. The improvement is shown as an absolute value (abs) in the corresponding units and in percentage value (%). The results indicate the mean and std values (mean ± std).

6.3.5 Conclusions

With the advance of artificial intelligence, numerous fields such as robotics, autonomous vehicles, and intelligent transportation systems are undergoing tremendous development in recent years. In this paper, the first AIM design approach based on deep reinforcement learning was presented, which was termed Reinforced Autonomous Intersection Management (RAIM). RAIM was a novel system employing a controller modeled by neural networks and trained by deep reinforcement learning, which was capable of controlling vehicles at an urban intersection in order to maximize vehicular flow while maintaining safety. The performance of RAIM showed significant results in numerous metrics, such as reducing travel time by 52%, waiting time by 96%, time loss by 84%, speed by 127%, and electrical energy consumption by 71% compared to other traffic light control algorithms.

6.4 Multi-Agent Deep Reinforcement Learning to Manage Connected Autonomous Vehicles at Tomorrow's Intersections

6.4.1 Introduction

Due to the great potential that deep reinforcement learning applied to systems such as AIM has shown to offer [398], this study proposed a significant improvement of RAIM by removing the strict constraints imposed on the architecture developed for RAIM. In this section [379], the proposed system was called advanced Reinforced AIM (*adv*.RAIM) and made use of recurrent neural networks, more specifically LSTM networks to cope with the limitation of the number of vehicles to be considered simultaneously. Thus, when gaining control of a vehicle, *adv*.RAIM considers all the other vehicles in the intersection to obtain the optimal action to ensure the safety of all vehicles. The performance of *adv*.RAIM was extensively evaluated in a variety of realistic and very complex scenarios.

6.4.2 *adv*.RAIM – advanced Reinforced Autonomous Intersection Management

adv.RAIM was based on the knowledge obtained from RAIM, and is trained by DRL, using insights from multi-agent systems, along with other advanced methods such as curriculum- learning through *Self-Play* and PER, to accelerate training and model the complex dynamics of the environment to control autonomous vehicles at intersections. *adv*.RAIM introduced significant improvements over RAIM, such as:

- The control network architecture was modified by including a recurrent Long Short-Term Memory (LSTM) module to analyze the influence of the other vehicles in front of the *ego*-vehicle and include it in the calculation of

the *ego*-vehicle's speed. In this way, all vehicles at the intersection could be taken into account simultaneously and an even more advanced control policy that takes all vehicles into account for decision making could be obtained. In this way, the constraint of always using the same number of vehicles (previously 32) was removed so that the number of input parameters to the neural network would match the number of neurons in the first layer. Thanks to the LSTM module, the characteristics of the vehicles were encoded in such a way as to obtain the conflicts between the *ego*-vehicle and the other vehicles. This module was called the state/conflict encoder module.

- Secondly, the complexity of the scenarios, both training, and testing, was considerably increased, going from a maximum flow of 450 veh/h/lane to 1200 veh/h/lane and from 2 lanes to 3 lanes per direction, increasing exponentially the complexity and allowing to maximize the advantages offered by DRL over traditional and other AIM techniques.
- Finally, *adv*.RAIM was compared with more recently published algorithms, such as an intelligent traffic light control system (iREDVD) and a previously proposed AIM.

As state variables for each vehicle, *adv*.RAIM used those listed in Table 6-5. Variables with a continuous range (such as position, speed, and angle) were coded by normalizing their values in the range of [-1, 1], and discrete variables (i.e., lane, track, tail) were coded using the one-shot encoder.

TABLE 6-5. Input Features and Meaning. (Table 1, p. 5, [379]).

Variable	Description
Relative pos x	Relative position to the center of the intersection on the x-axis
Relative pos y	Relative position to the center of the intersection on the y-axis
Speed	Vehicle speed
Angle	Vehicle orientation angle
Lane	Lane of approach (left lane, center lane, right lane)
Way	Way the vehicle will follow (right turn, straight path, left turn)
Queue	Intersection branch through which the vehicle is approaching (North, South, East, West)

As in RAIM, TD3, PER and Curriculum Learning through *Self-Play* were used for controller optimization. The architecture of *adv*.RAIM can be seen in Figure 6-7.

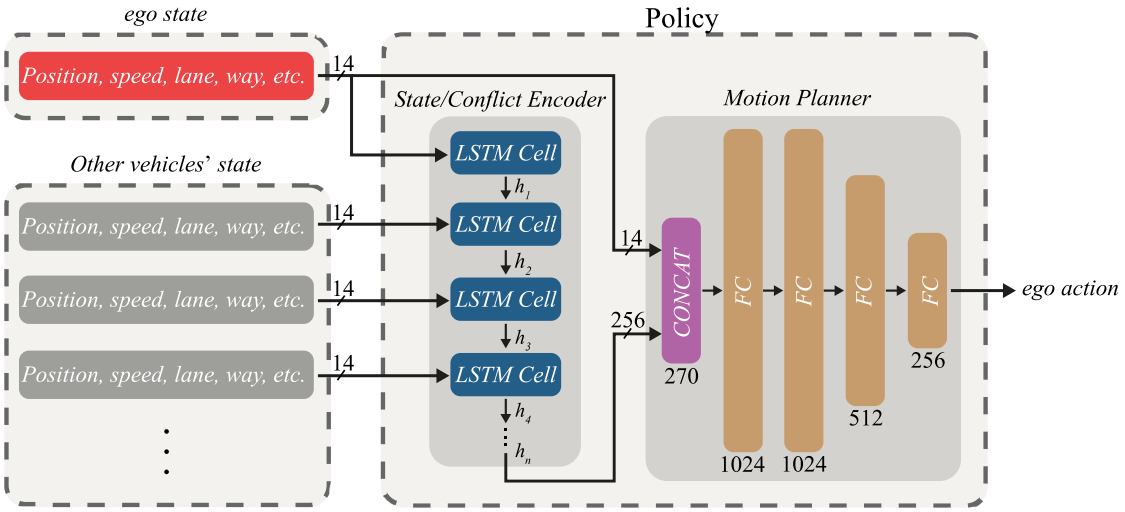


Figure 6-7. New advanced RAIM (*adv.*RAIM) network. The *LSTM Cell* allows us to control a variable number of vehicles, obtaining the possible conflicts between the *ego*-vehicle and the rest of the vehicles at the output of the State/Conflict Encoder. The policy output is the speed that the *ego*-vehicle must follow in the next timestep. Note that there is only one *LSTM cell* that is iteratively fed with the features of each vehicle (14), starting with *ego*-vehicle's state, and continuing with other vehicles' state. The intermediate output (h_x) was set to 256 parameters. (Figure 3, p. 6, [379]).

The pseudocode for *adv.*RAIM inference can be seen in Algorithm 6-2.

Algorithm 6-2: *adv.RAIM*

```
# Reset the environment.
for timestep t ∈ {0, ..., Max_episodie} do:
    # Obtain the vehicles currently being simulated.
    vehicles = sim.get_current_vehicles()
    # For each vehicle, the desired speed for the next timestep is obtained.
    # This loop is repeated for each vehicle in the intersection.
    # Clear input_params.
    for vehicle ego_veh ∈ vehicles do:
        # Obtain the params of ego_veh (position, speed, lane, etc.) and append to
        input_params.
        # Obtain the params of other vehicles.
        for vehicle veh ∈ vehicles do:
            # Obtain the params of veh ≠ ego_veh and add to input_params.
        end for
        # Obtain new speed of ego_veh using input_params and actor net of TD3
        and set new_speed.
        new_speed = actor_net(input_params)
        sim.set_veh_speed(ego_veh, new_speed)
    end for
end for
```

6.4.3 Experimental setup

SUMO was used as a simulator. A computer with a 16-core Intel processor and an Nvidia RTX 2080TI graphics card was used. *adv*.RAIM was implemented in Python 3.7 using PyTorch 1.5.0 framework. The setup consisted of two scenarios, one for training and one for testing.

Training scenario

The training scenario consisted of an intersection composed of 4 branches (north, east, south, and west), where each branch had 3 lanes in each direction. In addition, right and left turns were allowed, as well as through the intersection. The representation of the simulated intersection can be seen in Figure 6-8.

Following the methodology employed in the previous work, *adv*.RAIM was trained following learning with the curriculum through *Self-Play*. The conditions for increasing flow were to keep reward scores stable over the last 150 simulations. The metrics analyzed were average reward per vehicle, the number of collisions, and average time loss due to congestion per vehicle. The reward signal is identical to that of previous work. Each simulation had a simulated duration of 5 minutes, where in each simulation the flow was kept constant. In addition, to increase the number of states seen by *adv*.RAIM during training, in each simulation the random seed was changed, so the vehicles followed random routes in each simulation. A summary of the parameters used in the simulation can be seen in Table 6-6.

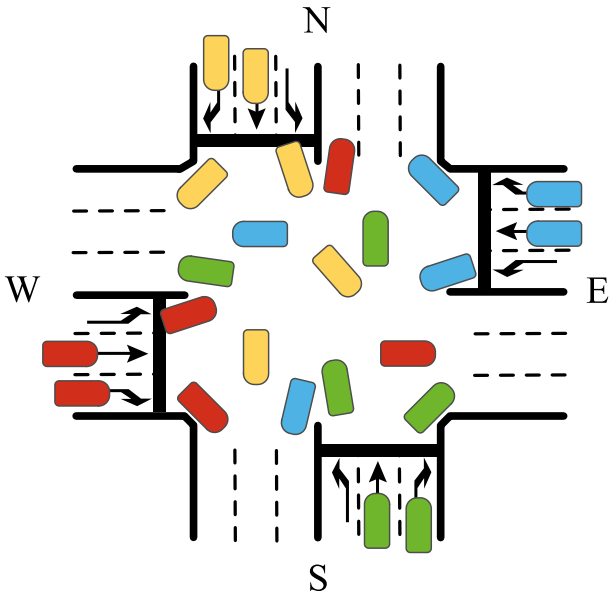


Figure 6-8. Simulated intersection with 4 approaches and 3 lanes/approach, where the movements go straight, turn right, and turn left are allowed. (Figure 4, p. 7, [379]).

TABLE 6-6. Summary of simulation setup and RAIM parameters. (Table 2, p. 6, [379]).

Simulator	Simulation timestep	0.25 segs
	Flow step	100 veh/hour
	Minimum flow	200 veh/hour
	Train duration	5 mins/simulation
	Test duration	720 mins/simulation
	Test scenario	4 branches, 3 lanes/way, and all ways.
	Control distance	100 meters
RAIM	Batch size	128
	Gamma	0.99
	Tau	4×10^{-3}
	Learning rate actor	1×10^{-5}
	Learning rate critics	1×10^{-4}
	Weigh decay	1×10^{-6}
	Policy noise	0.2
	Policy Noise Clip	0.3
	Optimizer epochs	3
	TD3 update actor every	2
	PER Memory Size	2^{20}

Testing scenario

For the testing scenario, an intersection was simulated as in the training scenario (4 branches, 3 lanes per branch). In each simulation, 12 hours of traffic were simulated, following the vehicular flow indicated in Figure 6-9. As can be seen in the previous figure, the simulated flow was separated according to the traffic origin (vertical flow or horizontal flow), in addition, the multiple flow variations can be seen, presenting periods of low (200 veh/h/lane), medium (600 veh/h/lane) and high (1200 veh/h/lane) flow, along with symmetric and asymmetric traffic as well as slow and fast flow variations. Each simulation was performed 10 times, obtaining the mean values and standard deviation. The metrics analyzed were multiple, such as travel time, waiting time, and time loss due to intersection congestion, as well as pollution and fuel/energy consumption metrics (CO, CO2, HC, PMx, NOx, and fuel and electricity). The vehicle distribution included 35% diesel vehicles, 35% gasoline vehicles, and 30% electric vehicles.

Using this scenario, the performance offered by *adv*.RAIM could be thoroughly tested and compared with that of other control algorithms. The algorithms with which *adv*.RAIM was compared were: traditional fixed-time traffic light (FT) control algorithms with different phase durations for each branch (10, 15, 20, and 30 seconds), an advanced adaptive traffic light control algorithm (iREDVD [313]), an AIM algorithm previously proposed by Qian *et al.* [372] and the previous version of RAIM [398]. The performance of other metrics of interest is shown because the control policy found by adv.RAIM allowed to optimize them.

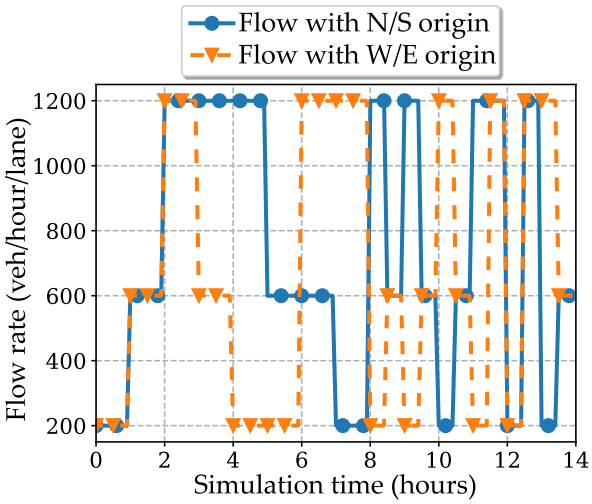


Figure 6-9. Vehicle flow-rate per lane used in the testing scenario. Low = 200 veh/hour/lane; Med = 600 veh/hour/lane; High = 1200 veh/hour./lane (Figure 5, p. 7, [379]).

6.4.4 Results

In this section the results obtained in the training scenario will be presented, accompanied by a detailed comparative analysis of the results of the testing scenario.

Training scenario

After the training process, the record of the analyzed metrics can be seen in Figure 6-10. In Figure 6-10 a is the number of collisions metric, in Figure 6-10 b is the average reward metric per vehicle in each simulation, and finally, in Figure 6-10 c can be seen the average time loss per vehicle metric. In addition, in each of these metrics, the simulated flow metric (veh/hour) can be seen in red. It can be seen how this metric has been increasing as *adv*.RAIM was learning to control the vehicles, as well as that at the beginning the scenario faced by *adv*.RAIM was simpler than at the end, where more simulations were required to increase the simulated flow. The results shown show the moving average (blue line) of the 3 runs, as well as the 90% confidence interval (blue shaded area).

From the results obtained in the training scenario we can see that by means of curriculum learning through *Self-Play* and PER, the analyzed metrics offered high stability, showing outliers that broke the trend when there was a change in the simulated flow, but showing a fast convergence to stable values after a few simulations. Thus, *adv*.RAIM was able to handle a larger number of vehicles simultaneously and learns to deal with them to further optimize each metric.

Looking at the number of collisions metric (Figure 6-10 a), the trend displayed shows a very robust trend, showing a clear understanding by *adv*.RAIM of the large negative reward of collisions from about simulation 750 onwards.

On the other hand, if we focus on the metric of average reward received per vehicle (Figure 6-10 b) we see a negative trend. At first instance this may seem a bad behavior, but if we analyze the evolution of the scenarios we see that this is because, as the number of simulated vehicles increases, the intersection is increasingly congested and vehicles can't circulate at a higher speed than when there is a lower flow, having to reduce their speed (on average) progressively and losing some time (the minimum) to maximize the average reward received by each vehicle.

Finally, if we look at the time loss metric (Figure 6-10 c), we can see an increasing trend, but it occurs similarly to what happened with the reward metric. As the flow of vehicles increases, the number of simulated vehicles increases, and the average time loss per vehicle due to congestion increases. The *adv*.RAIM training, despite the techniques used, lasted 14 days.

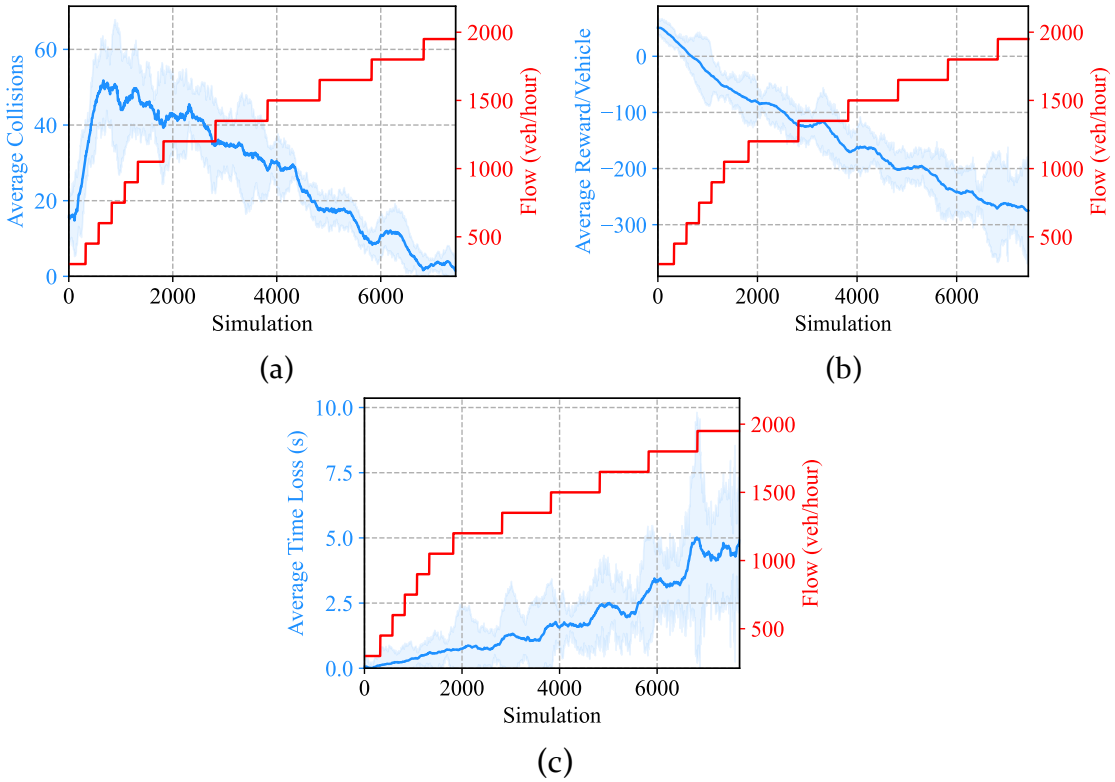


Figure 6-10. Training results. We plot the smoothed mean with an exponential moving average and 90% confidence interval across 3 seeds. The right vertical red axis and the curve in red show the simulated vehicular flow (veh/hour). The left vertical blue axis and the blue curve show each of the metrics studied: (a) Average number of collisions; (b) Average reward per vehicle; (c) Average Time Loss. (Figure 6, p. 7, [379]).

Testing scenario

After training, *adv*.RAIM was compared with other control algorithms in the test scenario. The results are presented in Table 6-7 (directly optimized metrics) and Table 6-8 (indirectly optimized metrics).

If we focus on the traffic light control algorithms (FT and iREDVD), we can see that, for all the analyzed metrics, *adv*.RAIM showed its superiority in performance. More specifically, if we focus on the advanced iREDVD algorithm, *adv*.RAIM was able to reduce time loss by up to 95%, in addition to reducing travel time by up to 59%. On the other hand, indirectly, significant reductions in pollutant gas emissions (CO, CO2, HC, PMx, and NOx) and consumption (fuel and electricity) of up to 37%, 13%, 28%, 37%, 50%, 21%, and 27% respectively are observed.

More interesting is if the performance obtained from *adv*.RAIM is compared with those of the AIM algorithm proposed by Qian *et al.* in [21]. The results obtained showed similar performance. However, the algorithm proposed by Qian *et al.* had a major disadvantage and that is that they had to know in advance the scheduling of all vehicles to obtain the optimal passing policy. In case there was a new vehicle, or it could not adjust to the speeds imposed by [21], the system had to recalculate from scratch all the passing priorities, with no possibility to readjust the policy. Consequently, this prevented it from being able to easily adapt to changing conditions in the environment, which is very common at intersections where cyclists, pedestrians, emergency vehicles, accidents, etc. may occur. In the case of *adv*.RAIM, this problem is avoided because it can be trained to take into account these occurrences and provide optimal solutions, considering all vehicles and actors (cyclists, pedestrians, emergency vehicles, etc.) in each time interval and obtain for each vehicle the optimal speed that guarantees the highest expected reward (eliminating collisions and minimizing time loss). With RAIM we can see that the performance is similar to *adv*.RAIM although slightly lower because not all simulated vehicles are considered simultaneously.

TABLE 6-7. Testing scenario results, metrics directly optimized. (Table 3, p. 8, [379]).

	Algorithm	Travel Time (s)	Waiting Time (s)	Time loss (s)
Traffic light	FT10	72.75 ± 7.44	30.91 ± 4.91	43.77 ± 9.88
	FT15	54.67 ± 8.11	17.59 ± 3.74	25.58 ± 6.12
	FT20	56.04 ± 6.91	19.02 ± 3.89	27.05 ± 7.91
	FT30	60.56 ± 7.92	23.56 ± 4.98	31.27 ± 8.89
	iREDVD [313]	43.23 ± 6.11	10.14 ± 3.33	18.64 ± 4.41
Qian *et al.* [372]		32.49 ± 3.39	2.55 ± 0.83	4.87 ± 1.22
RAIM [9]		31.44 ± 2.71	1.86 ± 0.94	3.08 ± 1.12
adv.RAIM		29.88 ± 3.01	1.14 ± 0.71	2.16 ± 0.46

For all metrics, the lower the better. No collisions were recorded. [avg. ± std. of 10 simulations].

TABLE 6-8. Testing scenario results, metrics indirectly optimized (Table 4, p. 8, [379]).

	Algorithm	CO emiss. (g)	CO_2 emiss. (g)	HC emiss. (mg)	PMx emiss. (mg)	NOx emiss. (mg)	Fuel cons. (ml)	Elect. cons. (W)
Traffic light	FT10	*1.58 ± 0.28*	*80.56 ± 18.84*	*9.78 ± 2.47*	*3.32 ± 0.98*	*240.13 ± 55.12*	*33.03 ± 5.58*	*32.57 ± 4.55*
	FT15	*1.04 ± 0.31*	*76.21 ± 16.81*	*7.88 ± 2.88*	*2.22 ± 0.88*	*18.56 ± 53.34*	*27.77 ± 5.21*	*32.08 ± 4.15*
	FT20	*1.27 ± 0.24*	*70.81 ± 14.99*	*8.08 ± 2.32*	*2.46 ± 0.91*	*189.49 ± 51.66*	*28.95 ± 5.01*	*31.54 ± 2.94*
	FT30	*1.38 ± 0.33*	*74.60 ± 19.22*	*8.69 ± 2.12*	*2.82 ± 0.92*	*211.44 ± 50.22*	*30.48 ± 4.88*	*31.74 ± 2.71*
	iREDVD [313]	*1.01 ± 0.21*	*70.70 ± 20.21*	*7.54 ± 1.99*	*2.21 ± 0.89*	*154.43 ± 45.22*	*27.21 ± 4.64*	*29.11 ± 3.33*
	Qian *et al.* [372]	*0.99 ± 0.09*	*70.44 ± 16.55*	*7.04 ± 1.58*	*2.18 ± 0.66*	*125.44 ± 30.31*	*26.84 ± 3.12*	*23.84 ± 1.98*
	RAIM [9]	*1.00 ± 0.10*	*70.08 ± 13.47*	*7.02 ± 0.97*	*2.11 ± 0.47*	*121.71 ± 44.19*	*26.42 ± 4.71*	*23.89 ± 3.01*
	adv.RAIM	***0.99 ± 0.08***	***69.84 ± 12.01***	***7.01 ± 0.91***	***2.07 ± 0.43***	***119.42 ± 13.42***	***25.99 ± 4.01***	***23.71 ± 1.48***

For all metrics, the lower the better. No collisions were recorded. [avg. ± std. of 10 simulations].

6.4.5 Conclusions

In this study, a significant improvement of an intersection vehicle control algorithm called *adv*.RAIM was presented, which has been based on RAIM for the development of this new version, adding important improvements to solve the problems presented by RAIM. The results have shown that *adv*.RAIM achieved notable results, overcoming the most important disadvantages of traditional AIM, managing to control autonomous vehicles in extremely challenging scenarios, and achieving outstanding results thanks to the coexistence of DRL techniques such as TD3, PER, and curriculum-based training techniques. Furthermore, by comparing adv.RAIM with traditional and advanced traffic-light based control techniques, was possible to evidence the benefits that an AIM trained by DRL can offer.

6.5 Learning from Oracle Demonstrations – A new approach to develop Autonomous Intersection Management control algorithms based on Multi-Agent Deep Reinforcement Learning

6.5.1 Introduction

As we have seen in previous work, the development of advanced autonomous vehicle intersection control systems using deep reinforcement learning offers great advantages such as reduction of travel time, elimination of waiting times, reducing collisions, etc. However, they require extensive training so that, through trial and error, the AIM can find an advanced optimal policy that can perform in a wide variety of situations. This development may be feasible in simple scenarios, but

when the number of variables to consider increases: different types of vehicles, each with unique characteristics, pedestrians, bicyclists, emergency vehicles, multiple lanes, etc. the complexity of the policy to be learned increases enormously, and with it, the training time [399].

In the scientific literature, different solutions can be found to speed up learning. Such as Imitation Learning (IL) [400], Imitation from Observation (IfO) [401], or Learning from Demonstration (LfD) [402]. Each of them has its advantages and disadvantages, but the most promising is LfD, where there are several papers showing how LfD speeds up training in several tasks [402]–[404].

In LfD, there is an expert agent that indicates which action the training agent should perform for each state. Thus, from this demonstration provided by the expert agent (*demonstration* = {*env_state*, *optimal_action*}), the training agent can use supervised learning to optimize its policy. However, not all simulators and simulation scenarios have an expert from which to extract what action to perform (demonstration) in each possible state, like in traffic simulators used to develop new AIMs (e.g., SUMO).

Therefore, in this research [405], a new approach was proposed that allowed an IL-trained agent to act as an expert agent and thus leverage the advantages offered by LfD. This new agent was called Oracle and hence this new training approach was called Learning from Oracle Demonstrations (LfOD).

Thus, in environments where there is no expert, or is embedded/hidden in the programming of the simulator (as in traffic simulators), from which to extract demonstrations, an agent can be trained using IL to imitates the behavior of the hidden expert and act as an Oracle from which to extract demonstrations and train a new agent by means of DRL.

LfOD was implemented on top of the TD3 DRL algorithm, incorporating significant changes to exploit Oracle most efficiently during training. Because of these changes in TD3, it was decided to call this new version of TD3, TD3fOD.

6.5.2 Imitation Learning

IL is a learning technique that emerged as an alternative to RL for agent training. In IL there is an expert agent from which are extracted demonstrations (*state-action* tuples) of the desired behavior. These demonstrations are used to train a new agent by supervised learning, learning to "*imitate*" the behavior of the expert agent.

The main advantages of IL over RL are that it reduces training time and eliminates the problem of reward shaping, where in RL a reward signal had to be hand-designed to allow stable and consistent training. Within IL we can find several approaches:

Behavioral Cloning (BC)

BC [400], [406], [407] exploits the demonstrations extracted from the expert to train an agent through SL and obtain a policy that clones the behavior of the expert in the observed states. This form of IL is the simplest and easiest to apply, presenting excellent results in tasks where the expert agent is able to explore the entire state-action space. However, in most tasks, BC can be problematic because the expert does not explore the entire state-action space, but only visits the states that have previously resulted from taking an optimal action. This problem leads to a situation where, when an agent trained using BC encounters a state never seen during SL due to the accumulation of errors, the agent's behavior can lead to dangerous situations from which it can never recover. This problem is known as compound errors [400] and an example of its behavior can be seen in Figure 6-11. In this figure, the optimal policy taught by the expert agent to navigate the circuit can be observed. The trained agent learns by IL means to mimic the behavior of the expert, and when it proceeds to make decisions, it may make a small error in each decision making. After a series of decisions, it will reach a state never seen during training that can lead to dangerous situations. In this example, the vehicle would collide with the outside of the circuit.

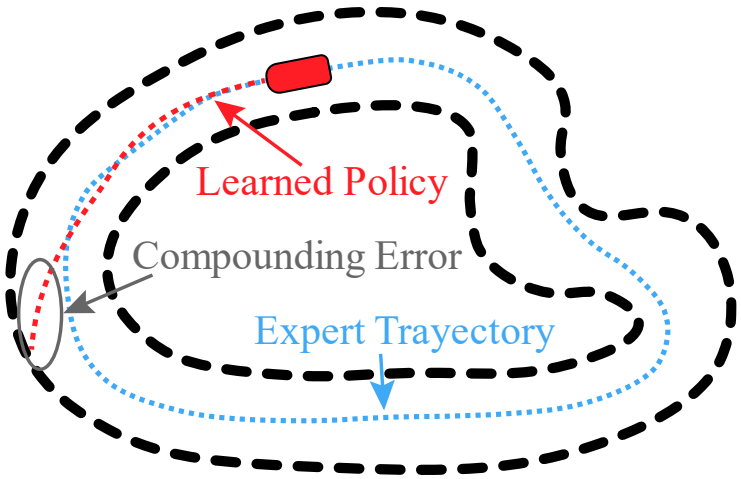


Figure 6-11. Compounding Errors example. (Figure 1, p. 3, [405]).

Direct Policy Learning (DPL) via Interactive Demonstrator

DPL [408] incorporates a training loop to the BC where the expert agent is asked what action it would have taken in the states visited by the learner agent, extracting the optimal demonstrations from the states visited by the learner agent. That is, DPL would start as a BC that collects the demonstrations of an expert agent. With these demonstrations, an agent would be trained by SL. This agent would roll out the trained policy and the visited states would be stored. Then, from the states visited by the previous agent, feedback is obtained from the expert, asking him

what actions he would have performed in these new states and storing these new demonstrations. These new demonstrations would be added to the rest of the expert's demonstrations and used to train a new learner agent using SL. The DPL training loop can be seen in Figure 6-12. Within this approach, there are several algorithms proposed, among which SEARN (Search-based Structured Prediction). [408], SMILe (Stochastic Mixing Iterative Learning) [409], and DAGGer (Data AGGregation) [400] stand out. Despite the simplicity of these approaches, as in the end, they are learning to mimic the behavior of an expert agent they will not be able to obtain a policy that allows them to improve that expert behavior.

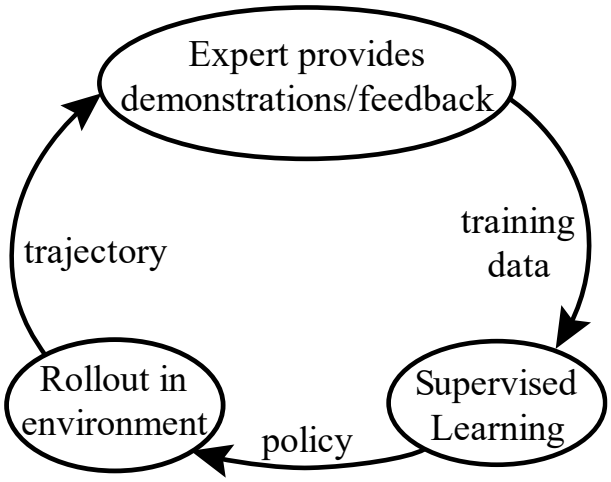


Figure 6-12. The general Direct Policy Learning (DPL) algorithm main loop. (Figure 2, p. 3, [405]).

6.5.3 Learning from Demonstration

Due to the problems presented by IL algorithms, a set of algorithms that take advantage of the demonstrations offered by experts to train a superior policy using RL was proposed. This type of algorithms was called LfD. Basically, LfD uses the BC-trained policy of an expert agent as a starting point. Then, using RL algorithms, it uses the pre-learned policy to discover a policy superior to the expert agent's policy by interacting with the environment, but without forgetting the pre-learned policy.

One of the first proposals for LfD work was in the DeepMind paper Deep Q-Learning from Demonstrations (DQfD) [402]. In this work, the authors were able to unify the strengths of IL together with RL in a very efficient way by employing the Deep Q-Network (DQN) algorithm. The results showed a great acceleration of training on discrete tasks such as Atari games, in addition to the fact that DQfD was able to find a superior policy compared to the expert policy.

Following that, they transferred what they learned to the DDPG algorithm used for scenarios with agents in continuous space actions. Due to modifications

made in DDPG that ensured a stable transition between IL and RL, they named the new algorithm DDPGfD [403]. DDPGfD stored both the demonstrations obtained in the IL phase and the actions performed by the agent in a replay buffer indefinitely. The results obtained by DDPGfD showed a clear acceleration of the training, being able to reduce the number of simulations up to x4 compared to DDPG.

However, DQfD and DDPGfD had the disadvantage that there could be a mismatch in the agent's internal parameters learned during the IL phase when the agent took control. This problem could lead to forgetting everything pre-learned during the IL phase.

6.5.4 TD3 from Oracle Demonstration – TD3fOD

In this paper, we proposed a new approach for the development of LfO algorithms using an oracle trained by BC. This Oracle would act as an expert from which to extract demonstrations to train an agent. To allow the use of the demonstrations offered by the Oracle and to guarantee a smooth training handover between IL and RL, several modifications had to be proposed in the RL algorithm used, which was TD3, which we will see in this section. Following the nomenclature used in previous work, we decided to denote this new approach Learning from Oracle Demonstration (LfOD), and the modification of the RL algorithm used (TD3) was denoted TD3fOD.

TD3 is one of the most powerful and advanced off-policy model-free RL algorithms for continuous tasks. TD3 where several key improvements are added, specifically, Clipped Double-Q Learning, Target Policy Smoothing, and "Delayed" Policy Updates [388], [410]. As many articles have shown, TD3 offers fast convergence for complex tasks where continuous control is required [411], [412]. This is the reason why it was decided to implement the algorithm on TD3 and not on other algorithms such as SAC [413], PPO [57], or A3C [414].

For the training of the Oracle (from which we can extract new demonstrations and ask about what action to take in each state) we used BC on the experiences collected from the hidden expert of the simulator. To capture the experiences of the hidden expert what was done is to obtain the state of the actors in two consecutive time instants so that we could extract the actions that the hidden agent performed in the previous instant for each of the actors (e.g., for each vehicle, reduce, maintain, or increase its speed). The optimization of the Oracle parameters was performed using SL.

The TD3 actor parameters (θ^{π}_{actor}) were updated using a *soft_update* approach (*soft-copy* of parameters) inspired by that employed by Mnih *et al.* in [47]. In this case, the weights of π_θ network (θ^{π}_{actor}) were updated as shown in

Equation 6-2. As can be seen in that equation, the influence of the Oracle on the TD3 actor is controlled by the parameter τ_1.

$$\theta^{\pi}_{actor} = \tau_1 \times \theta^{\pi}_{oracle} + (1-\tau_1) \times \theta^{\pi}_{actor} \quad with\ 0 < \tau_1 < 1 \tag{6-2}$$

Using the *soft_update*, the actor is forced to learn more slowly than the Oracle, improving the stability of the training. Furthermore, so that the influence of the Oracle on the actor is progressively reduced throughout the simulations, the parameter τ_1 followed the expression shown in Equation 6-3. The behavior throughout the simulations of $\tau_1$can be seen in Figure 6-13.

$$\tau_1 = sigmoid\left(-\frac{sim - th}{th/5}\right) \tag{6-3}$$

The parameter th in Equation 6-3 controlled the smoothness of the transition and the number of simulations from which learning by RL was considered more important than learning by Oracle via *soft_update*. This parameter th should be adjusted depending on the complexity of the scenario.

As can be seen in Equations 6-2, 6-3, and Figure 6-113, at the beginning of the training ($sim << th$), the parameter $\tau_1$has values close to 1. This implies that the parameters of π_θ (θ^{π}_{actor}) will be very similar to the parameters of the Oracle. After several simulations, the influence of the Oracle on $\pi_\theta$is progressively reduced, until reaching a simulation where $sim >> th$ and $\tau_1$has a value close to 0, which causes the Oracle term to cancel out, nullifying the changes in $\pi_\theta$due to the *soft_update*.

In addition to the incorporation of Oracle in the training of the RL agent, the following modifications to the RL algorithm, TD3, had to be proposed:

1. Modification of TD3 error equation so that the importance of the error produced by the RL actions increases progressively as the number of simulations performed increases. More specifically, Equation 6-4 is as follows.

$$\nabla_\theta \frac{1}{\mathcal{B}} \sum_{s \in \mathcal{B}} \tau_2 Q_{\phi_1}(s, \pi_\theta(s)) \tag{6-4}$$

Where τ_2 modifies the importance of Q_{ϕ_1} (*Q-values* of critic Q1) on the update of π_θ.

$\tau_2$is defined by Equation 6-5 and its behavior throughout simulations can be seen in Figure 6-13.

$$\tau_2 = 1 - \tau_1 \tag{6-5}$$

2. Use of 2 replay buffers with PER. One replay buffer was used to store experiences extracted from Oracle (Imitation Buffer) and the other was used to store experiences performed using RL (RL Buffer). Although PER was designed to be used for RL, the original PER paper [53] indicates that it can also be used for SL, speeding up training. The size of both replay buffers was of fixed length, removing older experiences when new ones were obtained, and the replay buffer was full. The Oracle used the experiences stored in the Imitation Buffer for SL training.

3. Finally, β factor was added, which allowed the TD3 agent (π_θ) to control the agent spontaneously, during a timestep. This parameter indicated the probability that the control of the agents was performed by the TD3 agent instead of the simulator (hidden expert). This probability increased smoothly exponentially until finally, the TD3 agent had full control of all agents. Thanks to this parameter, it was possible to guarantee high stability at the beginning of the training and a gradual and smooth transition from BC to RL. In addition, it allowed Oracle to explore a wider set of states at the beginning of training since the action taken by π_θ could be considered as "sticky actions", or noise actions, with all the benefits this can offer, and reducing compounding errors.

TD3fOD algorithm is divided into Algorithm 6-3, Algorithm 6-4, and Algorithm 6-5.

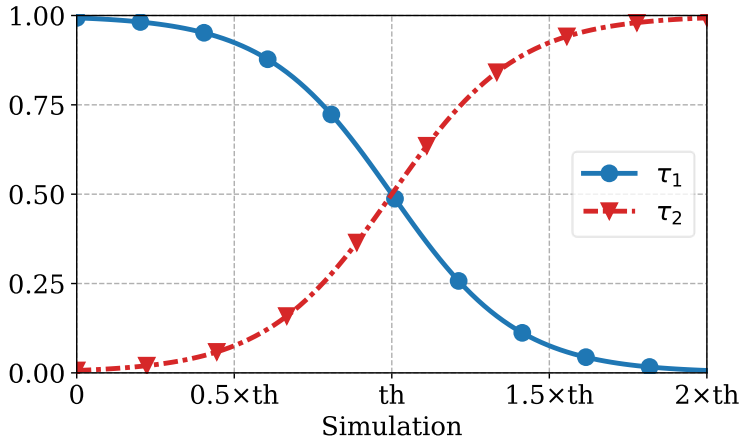


Figure 6-13. Evolution of τ_1 and τ_2 throughout the simulations in function of *th* parameter. (Figure 3, p. 6, [405]).

Algorithm 6-3: *TD3fOD*

Input: env Environment; θ^{π}_{actor} initial actor policy parameters; $\theta^{\pi'}_{actor}$ initial actor policy target parameters.

Input: ϕ_1^Q initial Q1-function parameters; $\phi_1^{Q'}$ initial Q1-function target parameters.

Input: ϕ_2^Q initial Q2-function parameters; $\phi_2^{Q'}$ initial Q2-function target parameters.

Input: θ^{π}_{oracle} initial Oracle policy parameters; $\mathcal{D}^{imitation}$ Imitation replay buffer; $\mathcal{D}^{reinforcement}$ Reinforcement replay buffer; p coeff. to expert decay; $warmup\_simulations$ number of simulations to pre-train actor, Q-functions, and Oracle; $imitation\_learn\_every$ every each timestep $run\ imitation\_module$; $reinforcement\_learn\_every$ every each timestep $run\ reinforcement\_module$;

for simulation $sim \in \{1, \dots, N_{simulations}\}$ **do**:
  # run simulation(sim):
  $\beta = p^{sim - warmup\_simulations}$
  for timestep $t \in \{1, \dots, Max_{episodie}\}$ **do**:
    $expert\_control = True$ **if** $random(0,1) < \beta$ **else** $False$
    obtain state $s_t \ \forall\ agent$
    if $expert\_control$ **then**:
      # Expert[1] selects the actions $a_t \ \forall\ agent$
    else:
      # Actor selects the actions $a_t \ \forall\ agent$
      $a_t = \pi_\theta(s_t)$
      $env.actions(a_t)$
    end if
    # Get next state and reward $\forall\ agent$
    $r_t, d_t, s_{t+1} \leftarrow env.step()$
    if $expert\_control$ **then**:
      # Obtain $a_t$comparing s_t and $s_{t+1} \ \forall\ agent$
      # Store $(s_t, a_t)\ in\ \mathcal{D}^{imitation}$
    end if
    # Store transition $(s_t, a_t, r_t, s_{t+1}, d_t)\ in\ \mathcal{D}^{reinforcement}$
    if t ***mod*** $imitation\_learn\_every = 0$ **then:**
      $run\ imitation\_module(sim)$
    end if
    if t ***mod*** $reinforcement\_learn\_every = 0$ **then:**
      $run\ reinforcement\_module(sim)$
    end if
  end for
end for

[1] We can't ask to an expert to obtain the actions.

Algorithm 6-4: *Imitation Module*

Run imitation module (sim):
for epoch $e \in \{1, \dots, N_{imitation\_epochs}\}$ **do**:
  for $mini\_batches\ \mathcal{B}$ in $\mathcal{D}^{imitation}$ **do**:
    $actions, states = unzip(\mathcal{B})$
    $actions\_pred = \pi_{oracle}(states)$
    # Update oracle params (θ^{π}_{oracle}) with one step of gradient descent.
  end for
end for
# $soft\_update(\pi_{oracle}, \pi_{\theta}, \tau_1)$:
$\theta^{\pi}_{actor} = \tau_1 \cdot \theta^{\pi}_{actor} + (1 - \tau_1) \cdot \theta^{\pi}_{oracle}$

Algorithm 6-5: *TD3 Reinforcement Module*

Run reinforcement module (sim):
for epoch $e \in \{1, \dots, N_{reinforcement\_epochs}\}$ **do**:
  # Sample a $mini\_batch\ \mathcal{B}$ from in $\mathcal{D}^{reinforcement}$
  $s_t, a_t, r_t, s_{t+1}, d_t = unzip(\mathcal{B})$
  $a'_t = (\pi'_{\theta}(s_t) + \epsilon).clamp(a_{min}, a_{max})$
  # Update Q-functions parameters ϕ_1^Q and ϕ_2^Q as in *TD3*
  if e ***mod*** $update\_actor\_every = 0$ **then**:
    $\tau_2 = 1 - \tau_1$
    # Update actor policy with one step of gradient ascent with equation 4 and using τ_2
    # Update target networks as in *TD3* and using τ_3
    # $soft\_update(\phi_1^Q, \phi_1^{Q'}, \tau_3)$
    # $soft\_update(\phi_2^Q, \phi_2^{Q'}, \tau_3)$
    # $soft\_update(\pi_{\theta}, \pi'_{\theta}, \tau_2)$
  end if
end for

6.5.5 Experimental setup

Our approach to training agents from demonstrations extracted from an Oracle focused on environments and simulators where there is no agent (or it is hidden) from which to extract demonstrations. Such simulators are found, for example, in traffic simulators, where internally vehicles have a set of hand-designed rules that they comply with in order to follow imposed traffic rules. For this purpose, in this study, it was decided to use the SUMO traffic simulator.

In addition, to provide robustness to the proposed algorithm, it was decided to train a new AIM by RL, using TD3fOD. Therefore, this new algorithm was named RAIM over TDF3fOD (RAIMfOD). For the design of RAIMfOD and its training, a computer with a 16-core Intel processor was used, along with an Nvidia RTX 2080TI GPU. RAIMfOD was programmed with Python 3.7 and the DL framework Pytorch 1.5.0.

RAIM over TD3fOD (RAIMfOD)

TD3fOD was used to train a new AIM system. Since RL was used to train the vehicle control policy, this algorithm was named RAIMfOD. This algorithm was based on previous work [379] that showed the benefits that RL algorithms could offer for the development of a new AIM, however, these systems required extensive training time. But by using the LfOD offered by TD3fOD, the development and research of these RL-based AIMs could be greatly accelerated.

Training scenario

The training scenario consisted of a 4-branch intersection with 3 lanes in each direction. Right turns, left turns, and straight ahead were allowed. Each simulation simulated 5 minutes of vehicular traffic with a constant flow of 1200 veh/hour. A representation of the simulated intersection can be seen in Figure 6-14.

As a reward signal for the RL algorithm, a signal was designed in which each agent (vehicle) received: a *+10* (strongly positive reward) when crossing the intersection, a *-10* (strongly negative reward) when the vehicle collided with another vehicle, and *-timestep* every time step simulated to encourage them to cross the intersection as fast as possible. The design of this reward signal was intended to find a policy that would allow vehicles to cross the intersection as fast as possible (reducing time loss) but without colliding with other vehicles. The training was performed 3 times, modifying the random seed. A summary of the hyperparameters used for both the simulator and TD3fOD can be found in Table 6-9.

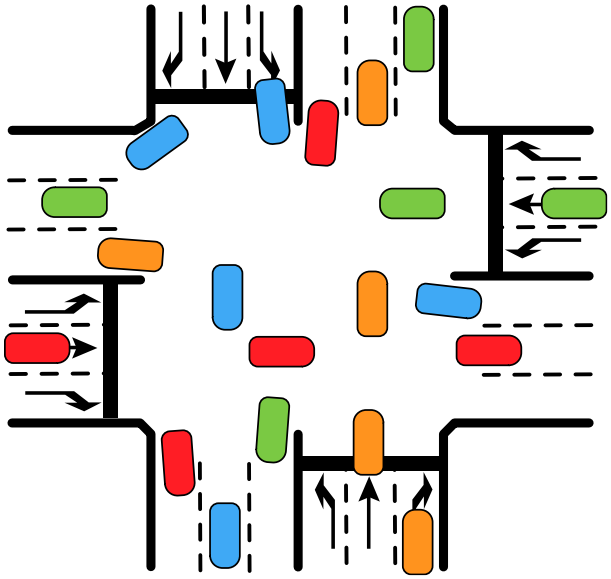


Figure 6-14. Representation of simulated intersection with 4 approaches and 3 lanes/approach, where the movements go straight, turn right, and turn left were allowed. (Figure 4, p. 8, [405]).

TABLE 6-9. Hyperparameters used in simulator and TD3fOD. (Table 1, p. 9, [405]).

Simulator	Simulation timestep	0.25 segs
	Flow	1200 veh/hour
	Train duration	5 mins/simulation
	Test scenario	4 branches, 3 lanes/way, and all ways.
	Control distance	100 meters
RAIM	Batch size	64
	TD3 Gamma	0.99
	τ_3	4×10^{-3}
	Learning rate actor	1×10^{-5}
	Learning rate oracle	1×10^{-4}
	Learning rate critics	1×10^{-4}
	Weigh decay	1×10^{-8}
	Action Range	[-1, 1]
	Policy Noise	0.15
	Policy Noise Clip	0.1
	$N_{reinforcement\_epochs}$	200
	TD3 update actor every	2
	$\mathcal{D}^{reinforcement}\ size$	$2^{20} \approx 1\times10^{6}$
	$reinforcement\_learn\_every$	15
	$N_{imitation\_epochs}$	5
	$\mathcal{D}^{imitation}\ size$	$2^{17} \approx 1\times10^{5}$
	$imitation\_learn\_every$	15
	$warmup\_simulations$	100
	p	0.995
	th	250

Testing scenario

RAIMfOD and TD3fOD were tested against simulated traffic for 14 hours following the time distribution of the simulated flow shown in Figure 6-15. The test scenario differed from the training scenario in both simulated flow and duration. In this case, this scenario presented a flow distribution by sections that presented multiple variations from low flows (500 veh/h), medium flows (1000 veh/h), and high flows (2000 veh/h), as well as symmetric or asymmetric traffic flows with respect to the North/South (N/S) and West/East (W/E) origin branches.

The intersection control algorithms with which RAIMfOD was compared were: No control, stop signal, fixed cycle time (FT) traffic light (with various durations studied: 30 s., 60 s. and 90 s.), an advanced traffic-light based control algorithm (iREDVD [313]) and the previous RAIM proposal [379].

The metrics studied to compare the performance of all algorithms were: travel time, waiting time, time loss due to congestion, and pollution and consumption metrics (CO, CO2, HC, PMx, NOx, and fuel and electricity). The distribution of vehicles used was as follows: 35% diesel cars, 35% gasoline cars, and 30% electric cars.

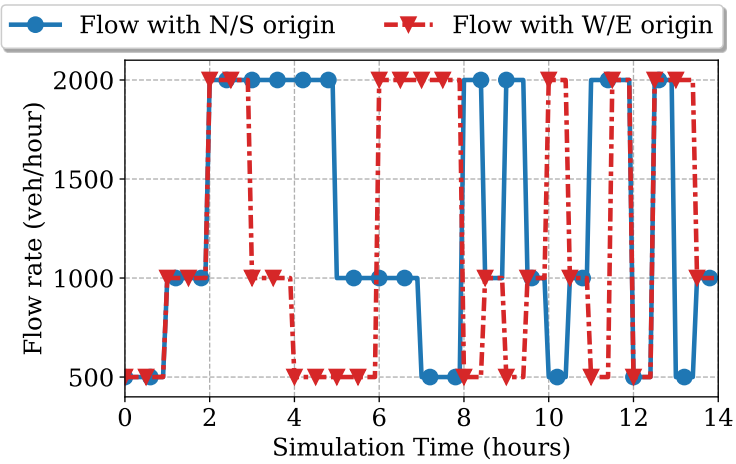


Figure 6-15. Flow test scenario distribution representation. (Figure 5, p. 9, [405]).

6.5.6 Results

This section will show the results obtained during the training of RAIMfOD in the training scenario, as well as the comparison of the performance offered with other control algorithms in the testing scenario.

Training scenario

The metrics analyzed in the training scenario were overall reward and time loss. Time loss was due to driving below the ideal speed at which the vehicle would travel if there were no intersection, sampled each simulated time step, and defined as shown in Equation 6-4.

$$\text{Timeloss} = \text{timestep} \times \left(1 - \frac{speed}{idea\_speed}\right) \tag{6-4}$$

The training results can be seen in Figure 6-16. Figure 6-16 a shows the average reward metric per simulation and Figure 6-16 b shows the lost time metric.

In Figure 6-16, the high performance that TD3fOD was able to offer in several aspects can be seen. The first is the increase in training speedup that TD3fOD offers. If we compare RAIMfOD with the original RAIM it is evident that the training speedup is between x3-x4, reducing the number of simulations needed in that range to achieve similar or even better results. If time loss metrics are taken into account, RAIMfOD was able to reduce time loss to less than 20 seconds after 200 simulations. However, the original RAIM required more than 1500 simulations to achieve the same, even allowing to obtain preliminary results much earlier, reducing the number of simulations by x7. With the reward metric, this same superior performance is observed, obtaining excellent results in 7 times fewer simulations.

On the other hand, we can see that the variance of the results obtained with RAIMfOD is much lower than that of the original RAIM, indicating that the policy

learned during training was much more stable, performing more reasonable actions. In the metrics we can see differentiated the three phases of the TD3fOD training:

1 Between simulation 0 and 100 yielded the pre-training stage and it can be seen that the results are noisy, indicating the filling of the Imitation and RL replay buffers.

2 From simulation 100 to simulation 250, approximately, the pre-training stage ended and training with TD3fOD began. This start can be seen in the change in the trend of the metrics analyzed at the beginning of this stage. In this range of simulations, the smooth transition in "*soft-copy*" between Oracle learning and TD3 RL was taking place, with the simulator performing most of the actions and the TD3 actor acting as "*sticky action*".

3 From simulation 250, curves τ_1 and τ_2 intercepted and most of the actions were performed by TD3 actor, which allowed to discover a better control policy to further optimize the results. This behavior highlighted the excellent performance offered by LfOD, allowing to reach a policy that outperforms the one offered by the expert through a smoothed step from a policy pre-trained by an Oracle, to the policy learned by RL.

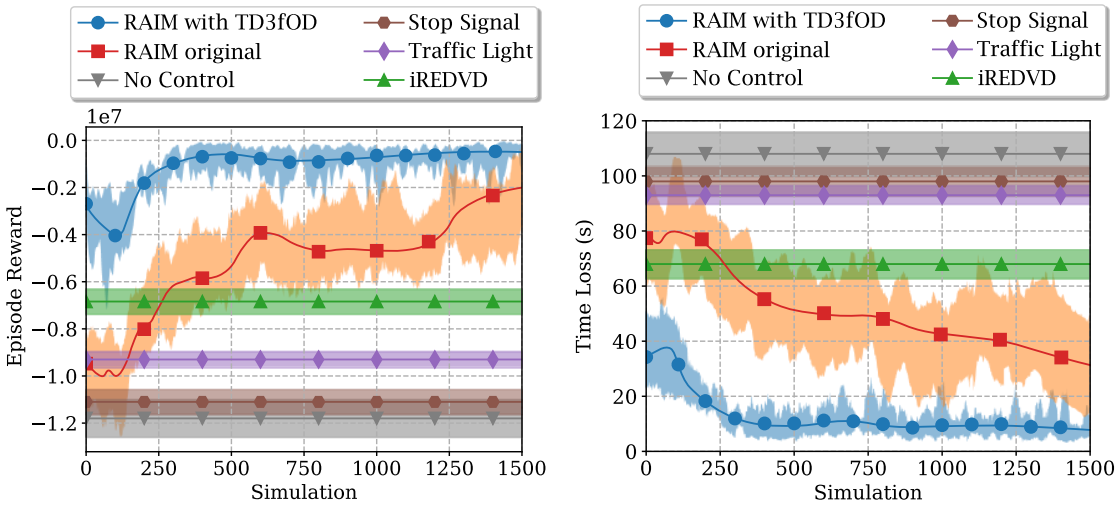


Figure 6-16. Training results: (a) Evolution of episode reward, the more, the better; (b) Time Loss evolution, the less, the better. RAIM with TD3fOD was able to learn a robust policy faster than the original RAIM. We plot the smoothed mean with an exponential moving average and 90% confidence interval across 3 seeds. (Figure 6, p. 10, [405]).

Testing scenario

The testing scenario allowed us to compare the performance obtained by RAIMfOD, compared to other control algorithms. In this way, we were able to demonstrate the capacity of generalization and adaptation to new scenarios that the policy learned by TD3fOD possessed. Thus, we demonstrated the advantages offered by RL and LfOD. The testing scenario was run 10 times, and then the average and std. were used to compare the performance. The results obtained are shown in Table 6-10. In this table it can be seen that RAIM with TD3fOD and original RAIM obtain very similar results, demonstrating that both algorithms find solutions with similar performance, but thanks to LfOD, RAIMfOD finds a policy much earlier, with a much lower variance of the results, as could be seen in Figure 6-16.

The results shown in Table 6-10 show a reduction in time loss between 95% and 86%, with waiting time reduction between 97% and 93%. This can be summarized as a reduction in travel time between 72% and 53%. Focusing now on the emission of pollutant gases, a significant improvement is achieved in all metrics, all of which can be reduced by more than 50%. Finally, considering the fuel and electricity consumption of vehicles, a reduction of between 29% and 5% is achieved for combustion vehicles, and between 34% and 24% for electric vehicles.

The results demonstrated the enormous potential of LfOD, as well as the use of RL-based AIM systems for autonomous vehicle control.

TABLE 6-10. Testing scenario results. (Table 3, p. 7, [405]).

Algorithm		Travel Time (s)	Waiting Time (s)	Time loss (s)	CO emiss. (g)	CO2 emiss. (g)	HC emiss. (mg)	PMx emiss. (mg)	Nox emiss. (mg)	Fuel cons. (ml)	Elect. Cons. (W)
Traditional	No control	121.42 ± 11.41	83.32 ± 17.53	91.44 ± 23.96	2.47 ± 1.65	111.78 ± 68.32	13.97 ± 7.65	5.21 ± 1.11	375.87 ± 51.54	41.24 ± 10.32	33,97 ± 3,21
	Stop Signal	107.53 ± 8.94	72.65 ± 12.42	78.15 ± 19.85	1.74 ± 0.99	81.70 ± 32.44	11.65 ± 6.01	3.52 ± 0.99	272.65 ± 37.95	35.33 ± 8.99	35,44 ± 2,88
	TL30	86.85 ± 4.43	50.07 ± 15.21	55.46 ± 17.83	1.08 ± 0.64	69.45 ± 23.64	7.32 ± 3.23	2.61 ± 0.86	194.99 ± 31.43	30.55 ± 7.66	32,39 ± 2,97
	TL60	81.69 ± 4.87	45.72 ± 12.65	50.07 ± 12.32	1.40 ± 0.75	82.82 ± 33.55	8.62 ± 2.31	2.85 ± 0.89	211.53 ± 32.32	33.96 ± 7.32	32,03 ± 3,02
	TL90	91.52 ± 9.87	52.69 ± 16.98	61.46 ± 19.26	1.33 ± 0.83	79.15 ± 38.52	8.76 ± 2.78	3.36 ± 0.92	231.53 ± 37.45	31.42 ± 8.31	36,64 ± 3,04
iREDVD		72,06 ± 6,75	27.12 ± 9.98	32.25 ± 15.23	1.49 ± 0.43	81.34 ± 23.42	9.00 ± 3.05	3.18 ± 0.67	221.94 ± 33.95	35.60 ± 9.85	36.50 ± 2.87
RAIM		36,91 ± 9,86	2.31 ± 2.05	6.21 ± 3.22	1.10 ± 0.88	65.52 ± 33.88	7.55 ± 2.03	2.74 ± 0.55	124.48 ± 24.92	29.69 ± 4.32	26.40 ± 2.66
RAIM over TD3fOD		33,24 ± 3,21	1.86 ± 0.99	4.21 ± 1.21	1.02 ± 0.23	52.54 ± 10.21	6.79 ± 1.68	2.47 ± 0.21	136.85 ± 14.33	28.88 ± 1.27	24.03 ± 1.09
Improvement											
Abs		[-88.19, -38.82]	[-81.46, -25.26]	[-87.23, -28.04]	[-1.45, -0.06]	[-58.24, -16.91]	[-7.18, -0.53]	[-2.74, -0.14]	[-239.02, -58.14]	[-12.36, -1.67]	[-12.61, -8.01]
%		[72.62, 53.87]	[97.77, 93.14]	[95.40, 86.95]	[58.70, 5.56]	[53.01, 24.35]	[51.40, 7.24]	[52.59, 5.36]	[63.59, 29.82]	[29.97, 5.47]	[34.42, 24.98]

The improvement shows the range between [best-case, worst-case]. It is shown as an absolute value (abs) in the corresponding units and percentage value (%). [avg. ± std. of 10 simulations]. For all metrics, the lower the better. TLXX means Traffic Light with total cycle duration XX segs.

6.5.7 Conclusions

The eventual success of future autonomous vehicles will depend on advances in driving systems. RL has proven capable of outperforming other traditional vehicle control systems, however, RL systems require extensive training. LfD provides tools capable of finding control policies in a much more efficient manner. However, not all training environments have an expert from which to extract demonstrations. Therefore, in this paper, we propose the use of an Oracle trained by IL that can be used to extract demonstrations and thus teach from a demonstration to an agent by RL. This original approach allows leveraging the use of LfD in environments where there is no expert to get feedback from. In this way, LfD can be used to train an agent in a much faster way, able to obtain a superior policy than the expert could offer and with a low variance in the results.

The algorithm modified in this study to leverage the demonstrations offered by the Oracle was TD3, and following the nomenclature used in the previous algorithms proposed for LfD, it was named TD3 from Oracle Demonstrations (TD3fOD). The modifications made in TD3 were: *i)* incorporation of an Oracle trained by IL from the states extracted from the simulator, *ii)* inclusion of several parameters for a smooth and progressive transition between LfOD and RL, and *iii)* use of two repetition buffers, one for the demonstrations to train the Oracle and another for the RL, in addition to the use of PER to speed up learning.

TD3fOD was implemented in the SUMO traffic simulator to accelerate the learning of an autonomous vehicle control system at traffic intersections, i.e., an AIM. The only AIM utilizing RL we know was RAIM [379], and for this reason, this algorithm was used over TD3fOD (RAIM over TD3fOD, RAIMfOD).

The results obtained in the training scenario showed that TD3fOD achieved significantly faster learning compared to TD3, allowing to find quicker control policy, speeding up the training between x3 and x4. In addition, the policy found outperformed the one offered by Oracle (hidden expert policy) and offered a significantly lower variance, allowing more robust results to be obtained.

When looking now at the RAIM over TD3fOD, in the testing scenario, it can be noted that the RAIM over TD3fOD allows obtaining results that even improve those obtained by the RAIM, improving in all metrics, and obtaining a lower variance. These results highlight the advantages offered by LfOD. RAIM over TD3fOD allows a reduction of waiting time between 97% and 93%, allowing a reduction of up to 50% in the emission of polluting gases, when compared to other traditional vehicle control techniques such as traffic lights, as well as other advanced traffic lights techniques such as iREDVD. Finally, regarding fuel consumption, combustion vehicles reduce their fuel consumption by up to 29% and electric vehicles by up to 34%.

6.6 AIM5LA: A Latency-Aware Deep Reinforcement Learning-Based Autonomous Intersection Management system for 5G Communication Networks

6.6.1 Introduction

Having developed the framework on which the control systems of connected autonomous vehicles would rely, this research developed a mechanism capable of considering the latency of 5G communication systems in AIMs. Due to the lack of works studying the impact that the communication network can have on the decentralized control of CAVs by AIMs, it was decided to investigate a novel latency-aware DRL-based AIM for the 5G communication network, which was named AIM5LA [415].

AIM5LA was the first AIM to take into account the latency inherent in the 5G communications network to achieve a robust and resilient multi-agent control policy using deep reinforcement learning. In addition to taking into account the experienced latency history, AIM5LA predicted the future latency behavior of each vehicle to provide enhanced security and improve traffic flow.

Within the related literature, the most similar articles are [416]–[419], where the work done by Zheng *et al.* in [417] stands out. In this study, the authors presented an AIM that was able to consider the latency of the network they used for vehicle communication. This AIM was based on a set of strict rules and the results showed that when the traffic was medium-high (> 360 veh/h), or there was asymmetric traffic, the results obtained by the proposed AIM were worse than when working with traffic lights. Other interesting works are shown in[418], [419]. In this case, another system capable of considering the latency of the wireless network used for vehicle communication is proposed. In this work, the authors designed an AIM based on FCFS as a heuristic to select the passing priority. However, both AIMs were based on fixed control rules that did not allow adapting the control policy, nor continuous learning. Moreover, the proposed systems were only tested in scenarios where the traffic density was extremely low (< 300 veh/h). None of the works found work with advanced mobile wireless communication networks, such as 5G, but use WiFi or Bluetooth networks.

It is clear that, so far, work on AIMs has focused on improving vehicular flow as well as reducing wasted time, but very few have considered the influence of latency inherent in the wireless communication systems used to communicate CAVs and AIMs. Therefore, we believe that addressing the study of latency in AIMs using advanced multi-agent DRL-based approaches could offer a great advantage over other techniques. Specifically, in this study, AIM5LA was able to learn autonomously without the need of any control rule and be able to obtain advanced control techniques that could offer high security and the possibility to learn from

the environment under control to perfectly adapt its behavior to the conditions and properties.

To the best of our knowledge, this was the first latency-aware AIM trained entirely by deep reinforcement learning. By latency-aware, I mean that AIM5LA was able to learn a control policy through DRL that was able to cope with constant changes in communication latencies and thus internally model the behavior of CAVs as a function of their latency and adapt the control of these CAVs to avoid collisions.

By taking communication latencies into account, AIM5LA prevents an unexpected delay in communication from causing a CAV to receive a command later than desired, leading to unexpected situations, including collisions.

A representation of an AIM can be seen in Figure 6-17. Note that AIM refers to the control algorithms of the CAVs and IM to the node in charge of the communication between the CAVs and the AIM.

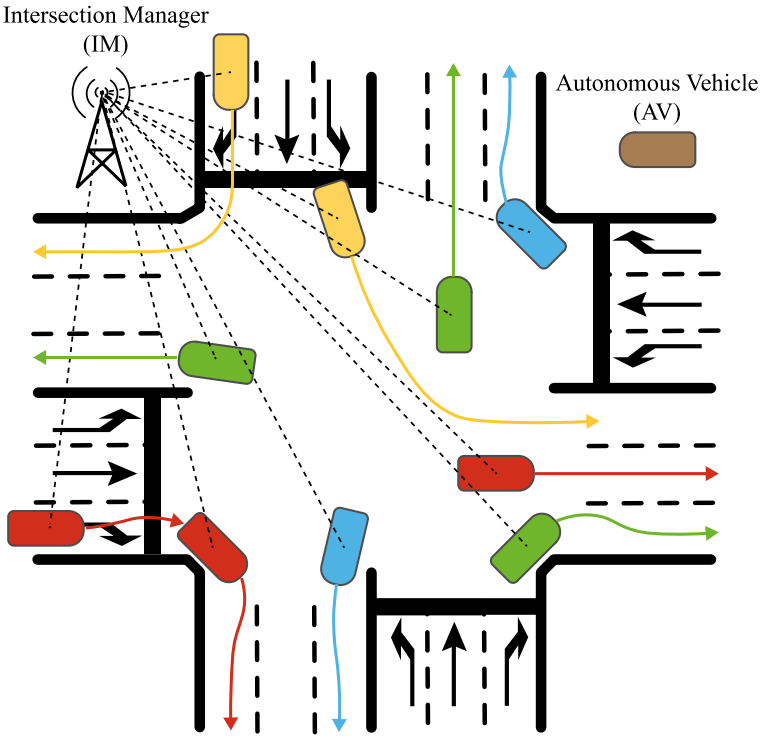


Figure 6-17. Example of autonomous intersection management (AIM). The Intersection Manager (IM) communicates with the Connected Autonomous Vehicles (CAVs) through a wireless communication network and guides them on the action to be taken by each CAV. The AIM runs inside the IM. (Figure 1, p. 2, [415]).

6.6.2 AIM5LA

AIM5LA is a major improvement over its predecessor *adv*.RAIM [379]. In this case, it includes the following improvements and modifications with respect to *adv*.RAIM:

1. Included the previous latency experienced by the *ego*-vehicle to be controlled in the previous interval.
2. Used a novel latency prediction module that predicts the latency experienced by the *ego*-vehicle during the next control interval, based on a Transformer deep neural network and the history of latencies experienced, as well as the number of AVs to be controlled simultaneously.
3. An LSTM coding network was used to consider the latency experienced by other VAs at the intersection.
4. Finally, the set of messages, the time intervals of each message, as well as the communication protocol to implement AIM5LA were proposed.

Once these modifications were included, the architecture of the AIM5LA controller was as shown in Figure 6-18.

As can be seen in Figure 6-18, AIM5LA includes a latency prediction module composed of a Transformer-based network [420] (referred to as "*Deep Transformer*" in Figure 6-18). This module is responsible for predicting the latency that the *ego*-vehicle would experience in the next time interval based on the history of latencies experienced by that vehicle and the number of AVs at the intersection (as we will see later, the number of AVs with a 5G communication module affects the latency).

On the other hand, there is the latency encoder module, composed of a stacked LSTM network. This network handles encoding the previous latencies of the other vehicles, in order to take into account these behaviors during the *ego*-vehicle control.

Finally, the message exchange protocol necessary for the operation of AIM5LA is proposed. By means of this message exchange protocol, AIM5LA can calculate the latency existing in the communication channel with each of the CAVs. All calculations and times are based on the adv.RAIM control interval that, every 250 ms, updates the action to be performed by each vehicle. This protocol includes the following messages:

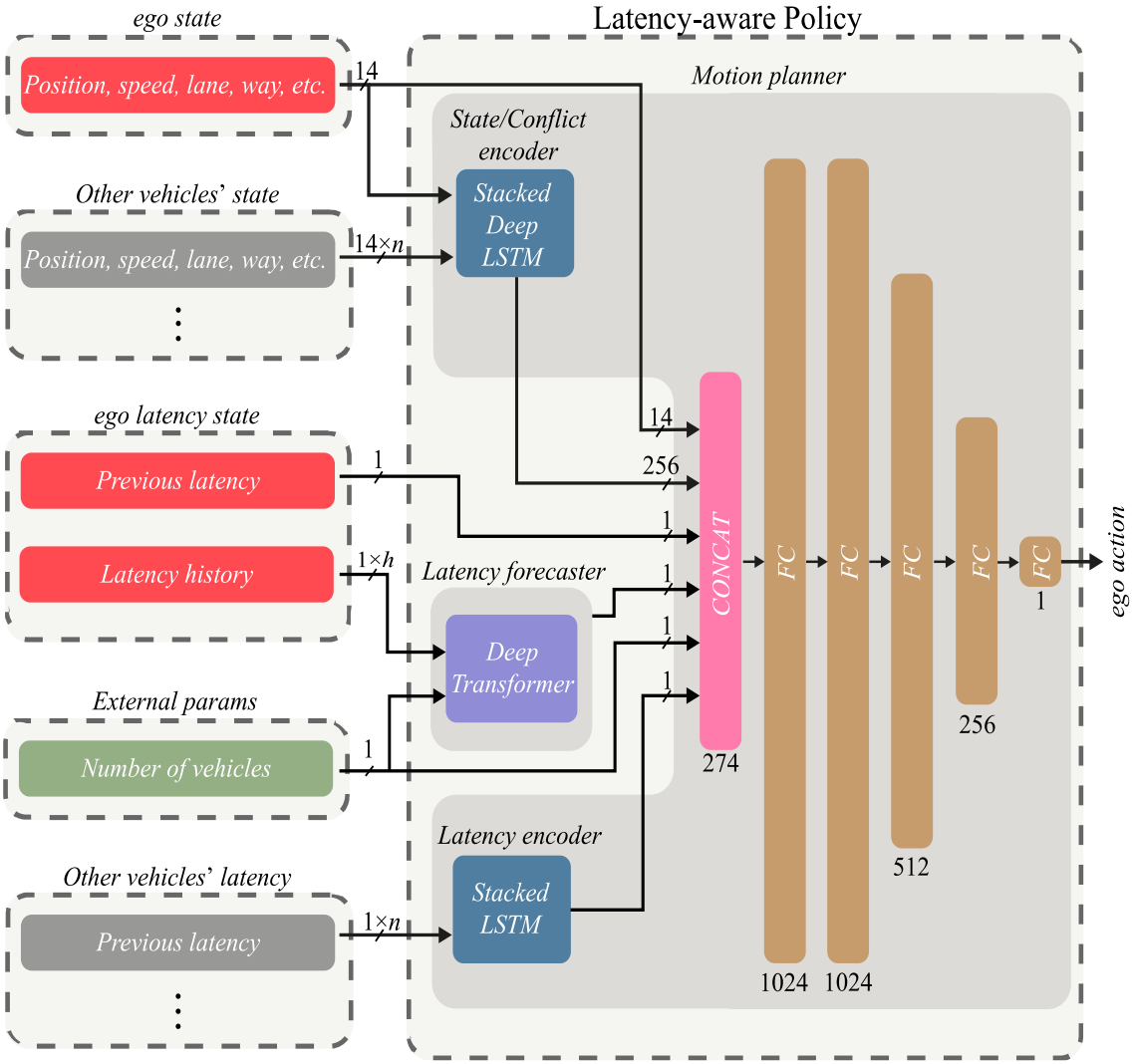


Figure 6-18. AIM5LA architecture with latency forecaster module based on Deep Transformer model and Latency encoder based on Stacked LSTM network. (Figure 2, p. 4, [415]).

Probe:

The *probe* message is sent by the IM at each intersection and contains information related to the IM (communication characteristics, geographical location, etc.) and the timestamp used to estimate the communication channel latency. This timestamp is the internal clock time of the IM before the message is sent. This message is periodically transmitted during the broadcast period for safe management without saturating the communication channel. Based on the control intervals of other works [368], [398], and *adv*.RAIM [379] development experience, the corresponding period was 250 ms. This means updating the vehicle's state (speed, acceleration, route, etc.) every 250 ms (4 times per second). As we saw in the previous study, reducing the update time (for example, 100 ms) increased the operating load and limited the system's real-time operation. On the other hand, if the update period is long (for example, 500 ms), the time elapsed between the two

updates is too long and there is a control error, and the security of the system is compromised.

When each vehicle receives a *probe* message, it checks that its geographical location and route match the managed IM and responds to that message with a *request* message. This periodicity allows all CAVs to be managed consistently and individually in a sequential manner, as well as secure and fast management, without saturating the processing or communication channel.

Request:

The CAVs send these messages to the IM to indicate that they want to be controlled at the next control interval (250ms). The CAV waits a random time *t* to send the *request* message. This random time t is used to reduce the possibility of message overlap when attempting to access multiple vehicles to 5G URLLC channel resources and to exclude this overlap from the latency calculation. After a series of preliminary tests in the Simu5G simulator [421], we found that the optimal value of the random time t corresponds to a continuous uniform distribution with a minimum value of 0 ms and a maximum value of 50 ms. This reduces the possibility of overlap with other CAVs, while allowing all control requests to be processed in real-time.

The *request* message contains the vehicle identifier, along with the expected random time *t* to send the control request message, which is appended to the resulting IM control message. In addition, it contains the internal parameters required for coordinated control. These are geographic location (x, y), speed, acceleration, traffic lanes (left, center, right), routes of interest (crossing, left turn, right turn), vehicle type, physical characteristics (width, length, weight), technical characteristics (maximum speed, maximum acceleration, etc.).

This message can be used to calculate the latency of the wireless channel. When the IM receives a *request* message from a CAV, it must compare the timestamp contained in the message with the internal timestamp to calculate the latency along the entire IM-CAV-IM path. In addition, the calculated latency history allows evaluating the latency behavior as well as the jitter. Lastly, the capture time of the internal parameters (position, speed, road, etc.) of each vehicle is excluded from the link latency calculation thanks to the random time *t* foreseen for sending the *request* message.

Action:

This message is sent to each CAV managed by the IM and indicates the *action* to be taken at that time until the next control interval. This message may include speed, acceleration, steering position, geographic route to follow, speed and acceleration profile, and more. The calculation of this action takes into account the

possible latencies of the wireless communication channel and the effect of other CAVs at the intersection. In this case, the AIM5LA indicates the *ego*-vehicle driving speed in the following control intervals.

The message exchange diagram is shown in Figure 6-19. The time allotted to listen to the *request* messages is 100 milliseconds. This value takes into account the maximum of 50 ms expected to respond to the *probe* message and the possible latency in higher saturation scenarios scenario [421], [422]. In order not to miss the request, the upper limit was set to 100 ms. In this figure, from the point of view of the AIM, after sending the *probe* message (at time *T1*), enough time is given to reply to the CAVs (*t_V1*, *t_V2*, and *t_V3)* taking into account the latency and random time *t*.

Therefore, AIM can calculate the average latency $Latency_{T_{i+1}}^{Vx}$ of the vehicle Vx for the next time interval T_{i+1} as shown in Equation 6-5. Where $T\_Request^{Vx}$ is the timestamp at which the *request* message is received from vehicle Vx; t^{Vx} is the random time t that vehicle Vx has waited to send the *request* message; and T_i is the timestamp at which the AIM sent the *probe* message.

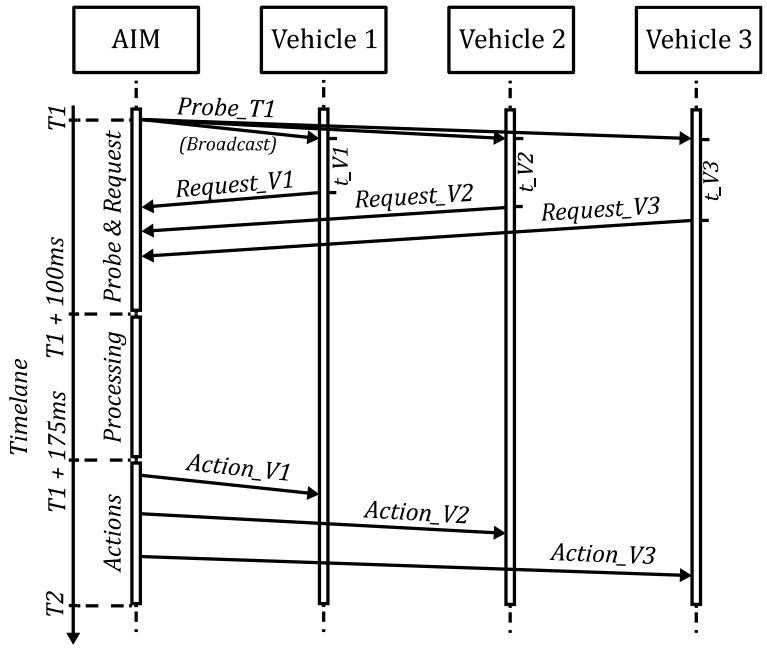


Figure 6-19. Proposed message diagram for AIM5LA. In this example, we need to control three vehicles. *Probe* messages are broadcast messages. (Figure 3, p. 5, [415]).

$$Latency_{T_{i+1}}^{Vx} = \frac{T\_Request^{Vx} - t^{Vx} - T_i}{2} \qquad (6\text{-}5)$$

From that point on, the AIM will start processing the *request* to get the best possible *actions* that the CAVs should perform during the next control interval. It keeps it up to date with all other sources of information, such as cameras, people counters, noise sensors, etc. Up to 75 ms are allocated for this stage. This time is based on the time required to perform the inference of the DL models employed, as well as the remaining time of the self-imposed control period (250 ms – 100 ms of the *probe+request* messages). Due to the use of concurrent computing, knowledge distillation, and half-precision operations, the actual drawing time is expected to be much shorter than at this time.

Finally, the actions to be taken in the next time interval are sent to each CAV. This operation lasts up to 75 ms. This is the remaining time of the control interval period (250 ms *control interval* – 100 ms of *probe+request* – 75 ms of *processing*). Then the process starts again, but this time T2 (= T1 + 250 ms [*control interval*]).

6.6.3 Experimental Setup

In this section, we will see the testbed developed, as well as the different experiments performed.

Testbed

To evaluate the performance of AIM5LA and the proposed message protocol, several experiments were designed. The first experiment was used to optimize the latency prediction module in a simple urban scenario using the Simu5G 5G simulator [421].

Then, a complex urban scenario used to train AIM5LA using DRL was designed. In this way, AIM5LA was able to find an advanced control policy that took into account the latency of wireless communication.

Finally, a third experiment was designed to show the performance of AIM5LA in a different urban scenario, never seen by the algorithm during the training phase. This experiment aimed to demonstrate that AIM5LA found a robust control policy, despite facing unknown traffic and latency situations.

AIM5LA employs DRL, in addition to other advanced techniques and algorithms such as TD3 [388], PER [53], and curriculum learning [389] to accelerate training and maximize the learned knowledge, to find the control policy that allows it to control CAVs safely.

The 5G simulator Simu5G 1.1.0 [421] was used to simulate communication channels and message exchange protocols with OMNeT++ 5.6.2 [306] and INET 4.2.2 [307]. Simu5G is an OMNeT++ library for fully evaluating the performance of 5G networks. The 5G mobile protocol is a well-designed and studied protocol with the expected ultra-low latency (1 ms.), so it was decided to investigate the latency.

SUMO 1.8.0 was used together with TraCI for vehicle behavior simulation and control.

Python 3.8.10 and PyTorch 1.9.0 [349] and CUDA [423] along with the DRL framework were used to develop AIM5LA, using an Intel 8-core/16-thread processor (i7-11700k) as hardware. Nvidia RTX 3080 graphics card.

Experiments

First, we optimized the latency prediction module in *Experiment #1*. This experiment consisted of a simple two-lane, traffic light-controlled intersection with a 60-second cycle time, and an incremental number of vehicles. By varying the number of vehicles, we were able to analyze latency as a function of the number of 5G devices [421], [422]. 1, 4, 16, 16, 16, 64, 128, and 256 vehicles were simulated in each run. The simulations were run 10 times, and in each run, the vehicles took random routes (turn right, go straight, or turn left). The communication latency between the vehicles (*nrCar* in Simu5g) and the 5G base station (*gNodeB* in Simu5g) was measured, obtaining the average latency and the standard deviation.

The *gNodeB* was located at the intersection edge, at 20 meters height. To compute the latency, the above message exchange protocol was implemented in Omnet++, in which the *gNodeB* periodically (every 250ms) sent the *probe* message which, when received by the CAVs, was randomly responded with a *request* message after waiting a random time *t*. As shown in the previous section, the random time *t* followed a uniform distribution with a minimum value of 0 ms and a maximum value of 50 ms. Each simulation ran for 300 seconds, the first 250 seconds of the simulation were used as the training dataset and the last 50 seconds were used as the test dataset, splitting the dataset 83%/17%.

The latency prediction module is composed of a transformer-based network (specialized for time series or text analysis) as well as 3 fully connected layers (FC) DNN with 64, 16, and 1 neurons each. The transformer network considers *h* historical latencies (*h* in Figure 6-20) to predict the expected latency in the next timestep. This value of historical latencies was optimized during the experiments. Furthermore, since the latency may depend on the number of CAVs (5G devices) at the intersection, this parameter was concatenated with the expected future latency predicted by the transformer network and used as input to the FC DNN. The representation of the predictor architecture is shown in Figure 6-20. The results of this optimization can be seen in the next section.

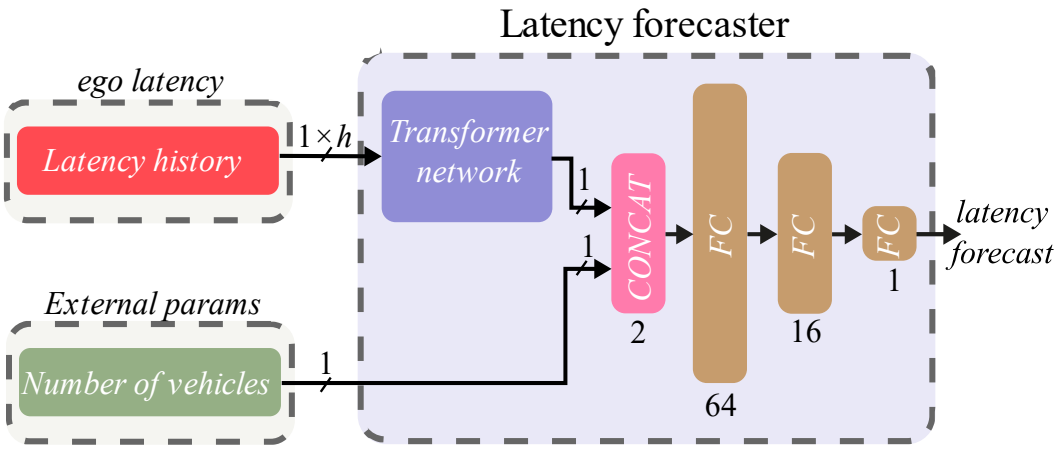


Figure 6-20. The latency prediction module architecture consisted of a transformer network and three fully connected layers. The input size of the transformer network depended on the size of the considered window (h), which determined how many previous latencies were considered for prediction. The output was combined with the variable number of simulated (controlled) vehicles and the results were fed into the FC DNN. (Figure 4, p. 6, [415]).

Experiment #2 was developed to optimize the AIM5LA control algorithm using DRL. This scenario included an intersection with AIM responsible for controlling the CAVs through the intersection. The scenario was composed of four branches (north, south, east, and west) with a length of 200 meters, three lanes per direction, and the possibility of left turn, straight ahead, or right turn. The CAVs requested AIM control at a distance of 100 meters from the intersection. A representation of the simulated scenario is shown in Figure 6-17. To obtain different situations challenging the control algorithm, the flow diagram of the simulation is shown in Figure 6-21 a. It shows the number of simulated vehicles per lane per simulated hour for the North (N), South (S), West (W), and East (E) branches. On the opposite branches (NS and WE), the flow is symmetrical. The reward function is defined in *adv*.RAIM [379]. We performed three optimization runs and the results showed the moving average of the mean and the standard deviation.

Finally, *Experiment #3* included a Manhattan grid consisting of 100 intersections (10 × 10) to demonstrate the performance of AIM5LA. Each intersection had a separate AIM controlling the vehicles passing through it. The intersections were 250 meters apart and the configuration is similar to that of training *Experiment #2* (4 branches, 3 lanes, left, straight, and right allowed). In addition, to account for unique situations, other flow behaviors were modeled as shown in Figure 6-21 b. This scenario allowed demonstrating the capabilities of AIM5LA in a situation never seen before. A depiction of the simulated scenario can be found in Figure 6-22.

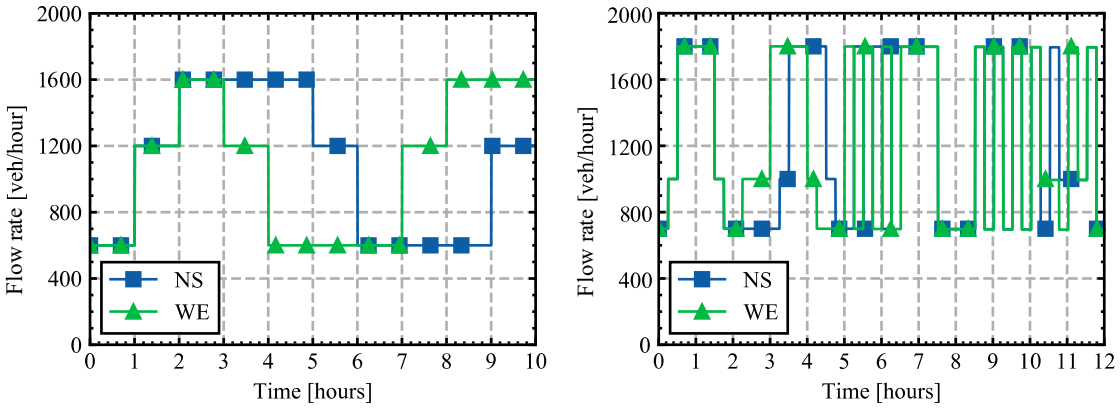


Figure 6-21. Vehicle flow rate per branch: (a) Experiment #2; (b) Experiment #3. (Figure 6, p. 7, [415]).

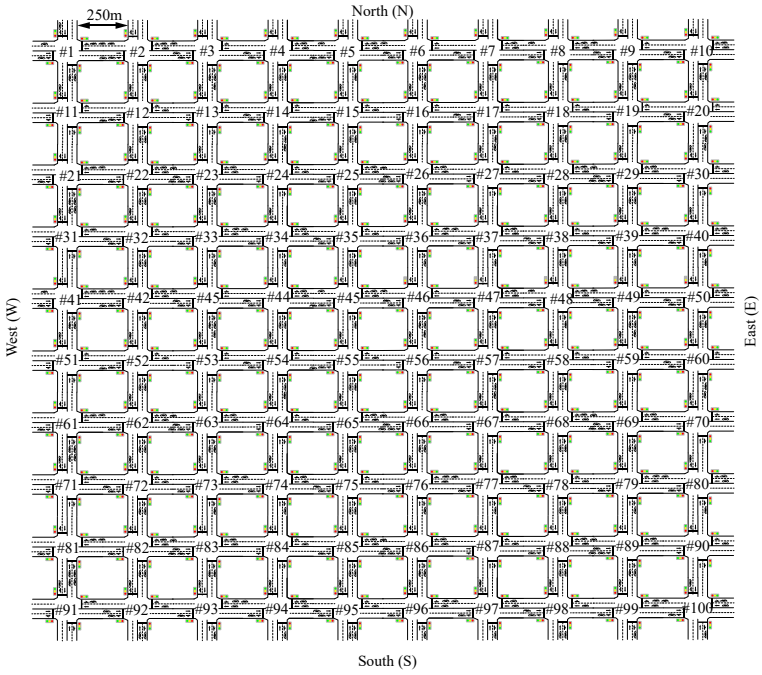


Figure 6-22. Simulated topology for Experiment #3: 10×10 Manhattan network with 250 meters between each intersection. (Figure 5, p. 7, [415]).

SUMO provides a complete set of metrics after each simulation. The metrics analyzed include time loss due to congested intersections, number of collisions, average waiting time, and various pollutant gas metrics (CO2, PMx), as well as fuel and electricity consumption. Although some of the metrics shown are not directly optimized during training, they are indirectly optimized through the overall optimization process.

The performance of AIM5LA was compared with that of other AIM algorithms such as *adv*.RAIM [379], the one published by Andert *et al.* [418], and two preliminary versions of AIM5LA: using only the latency of the *ego*-vehicle at the previous time point (denoted AIM5LA_v0.1); and AIM5LA_v0.1 + the latency

prediction module (denoted AIM5LA_v0.2). It was also compared with the results obtained with traffic light-based control techniques; in particular, fixed time control (FX) systems with different cycle lengths (30, 60, and 90 seconds) and advanced intelligent traffic light control technology (iREDVD) [313]. A fixed traffic signal system assigns a fixed green time to each branch of an intersection where the right-of-way (green light) is applied. This priority passes through all branches with a fixed cycle time.

6.6.4 Results

In this section, the results are shown and discussed.

Experiment #1 – Forecast module optimization

After running *Experiment #1*, Figure 6-23 shows the mean latency value and its standard deviation as a function of the number of simulated vehicles. These results show that the latency value is highly dependent on the number of vehicles, and if we compare the scenario of 1 vehicle with the scenario with 256 vehicles simulated simultaneously, the latency value can increase up to a factor of 10. This increase in latency can be due to many factors such as: saturation of the communication channel, reduction of the available bandwidth, overlapping of requests with other devices, etc. A summary of the results can also be found in Table 6-11.

TABLE 6-11. Results of *Experiment #1*. (Table 2, p. 7, [415]).

Number of vehicles	Average latency	STD latency
1	1.51	0.48
4	2.43	0.61
16	4.85	1.05
64	6.48	1.93
128	10.24	1.95
256	15.63	2.12

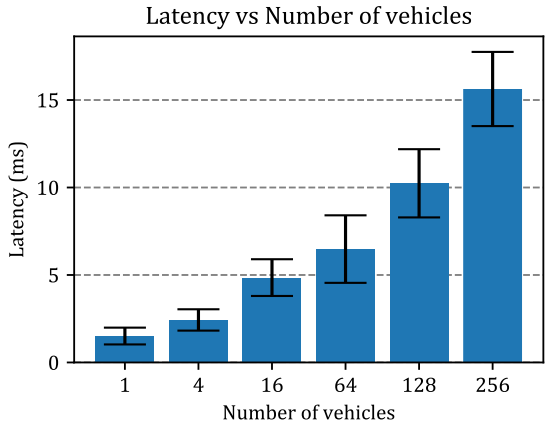


Figure 6-23. Latency vs Number of vehicles in *Experiment #1*. Plots the mean and standard deviation values for each group (1, 4, 16, 64, 128, 128, 256 simultaneous vehicles). (Figure 7, p. 7, [415]).

The latency temporary evolution for a random vehicle in each of the simulated scenarios can be seen in Figure 6-24. From this figure, the dependence of the number of AVs on the experienced latency can be observed. Furthermore, one can see that the more devices simulated at the same time, the higher the variance of the latency (~jitter).

After simulating the behavior of the CAVs and obtaining the observed latency, we had a dataset that was employed to train the latency prediction module. To split the dataset, the first 250 seconds of latency were used as the training dataset, and the last 50 seconds were used as the validation dataset. Figure 6-25 shows the latency behavior for one vehicle in the 4-vehicle scenario, as well as the division of the dataset into training (< Message #1000, blue line) and test (>= Message #1000, red line).

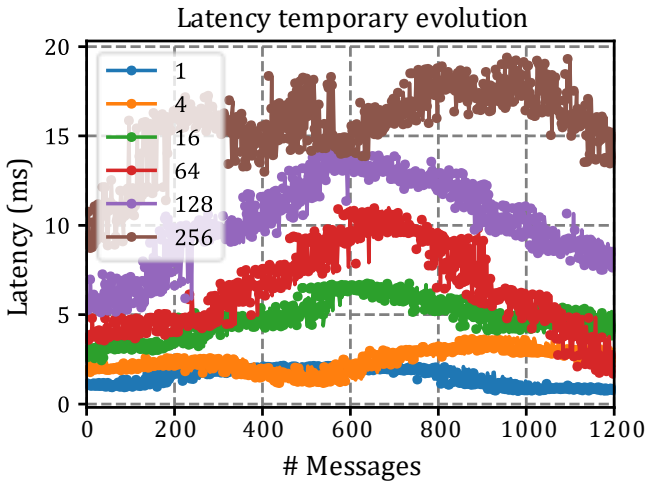


Figure 6-24. Time evolution of latency for a random vehicle in each simulation group. The legend shows the number of vehicles simulated at the same time. (Figure 8, p. 7, [415]).

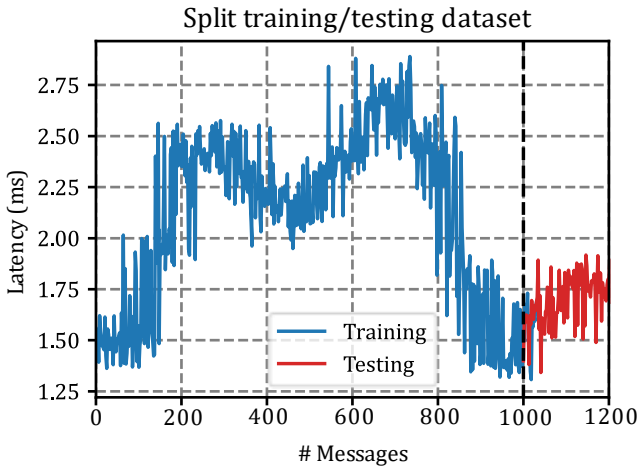


Figure 6-25. Example of dataset division for a vehicle in the scenario of 4 simultaneous vehicles; Up to latency sample (#Message) 1000 was used to comprise the training dataset, blue line. From latency sample (#Message) 1000 was used for the validation dataset, red line. (Figure 9, p. 8, [415]).

The training and testing results can be found in Figure 6-26. It has plotted the root mean square error (RMSE) metric [43] as a function of the historical latency window size (*h* in Figure 6-20) used in the transformer-based module. Additionally, are shown both when the number of vehicles parameter (w/ nvehs) was considered and when it was not considered (w/o nvehs) as an input parameter to the FC DNN of the latency forecasting module.

From these results shown in Figure 6-26, it can be extracted that the optimal historical latency window size (*h*) was 10 samples since they presented the smallest RMSE with the smallest window size. Therefore, for the latency prediction module, 10 previous samples were considered for latency prediction at the next time instant, as well as the number of simultaneous vehicles. With this configuration, the mean value and standard deviation in terms of RMSE were 0.4551 ± 0.0264 in the training set and 0.3702 ± 0.0301 in the validation set.

To visually illustrate what these results mean, in Figure 6-27 we depict the behavior of the predictor module during both the training and validation phases. These results were notably accurate, being able to strongly predict the latency value in both the training and validation datasets. Figure 6-27 presents the result for the 4-vehicle scenario (Figure 6-27 a) and the 128-vehicle scenario (Figure 6-27 b).

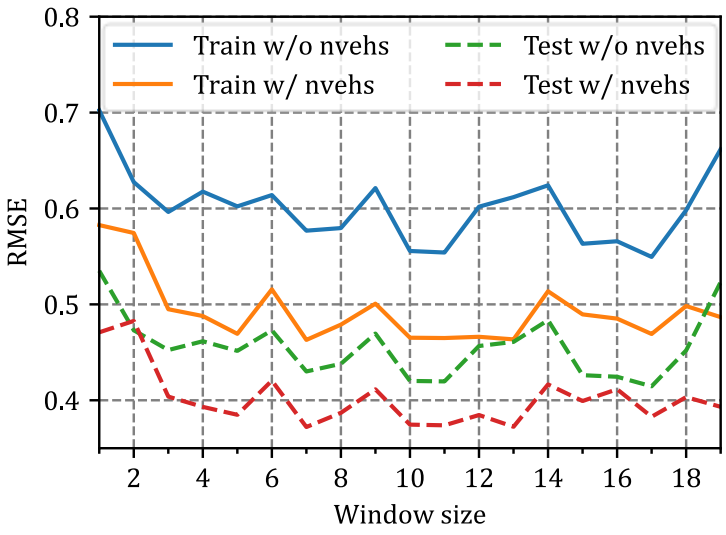
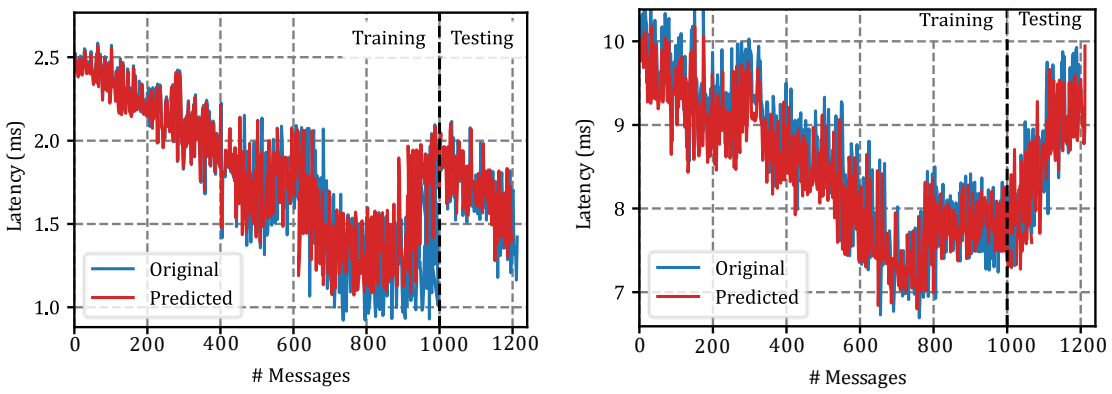


Figure 6-26. RSME of the latency forecasting module for the training and validation datasets as a function of the size of the historical latency window considered (*h*). The results are presented with the variable vehicle number (w/ nvehs) and without it (w/o nvehs) as input parameters in the latency prediction module. (Figure 10, p. 8, [415]).



(a) (b)

Figure 6-27. Training and validation of latency forecast module. RMSE: 0.4551 ± 0.0264 / 0.3702 ± 0.0301 (training / validation). (a) 4 vehicles scenario; (b) 128 vehicles scenario. (Figure 11, p. 7, [415]).

Experiment #2 – Training AIM5LA

After training the latency predictor module, in Experiment #2, AIM5LA was trained using DRL, employing the message exchange protocol described previously. Figure 6-28 shows the evolution of the metrics analyzed during training: reward (Figure 6-28 a), time loss (Figure 6-28 b), and number of collisions (Figure 6-28 c). Depicted are the mean (solid line) and standard deviation (shaded area) of the three runs. The shown plots have been smoothed using a 100-sample moving average for the sake of clarity.

The training results reveal the superior performance of AIM5LA, which was able to eliminate the number of collisions from about simulation 10^6 onwards. After eliminating collisions (the most penalizing factor for it in the reward function), AIM5LA fine optimizes the control strategy and adjusts the control of the CAVs to reduce the time loss. This behavior can be seen in the reward metric, which showed a significant improvement as the number of collisions decreased, and after removing crashes, the reward continued to increase, albeit to a lesser extent. The AIM5LA optimization required 21 days (~500 hours) of simulation and training, run 3 times using an 8-core/16-thread Intel processor (i7-11700k) and an Nvidia RTX 3080 graphics card.

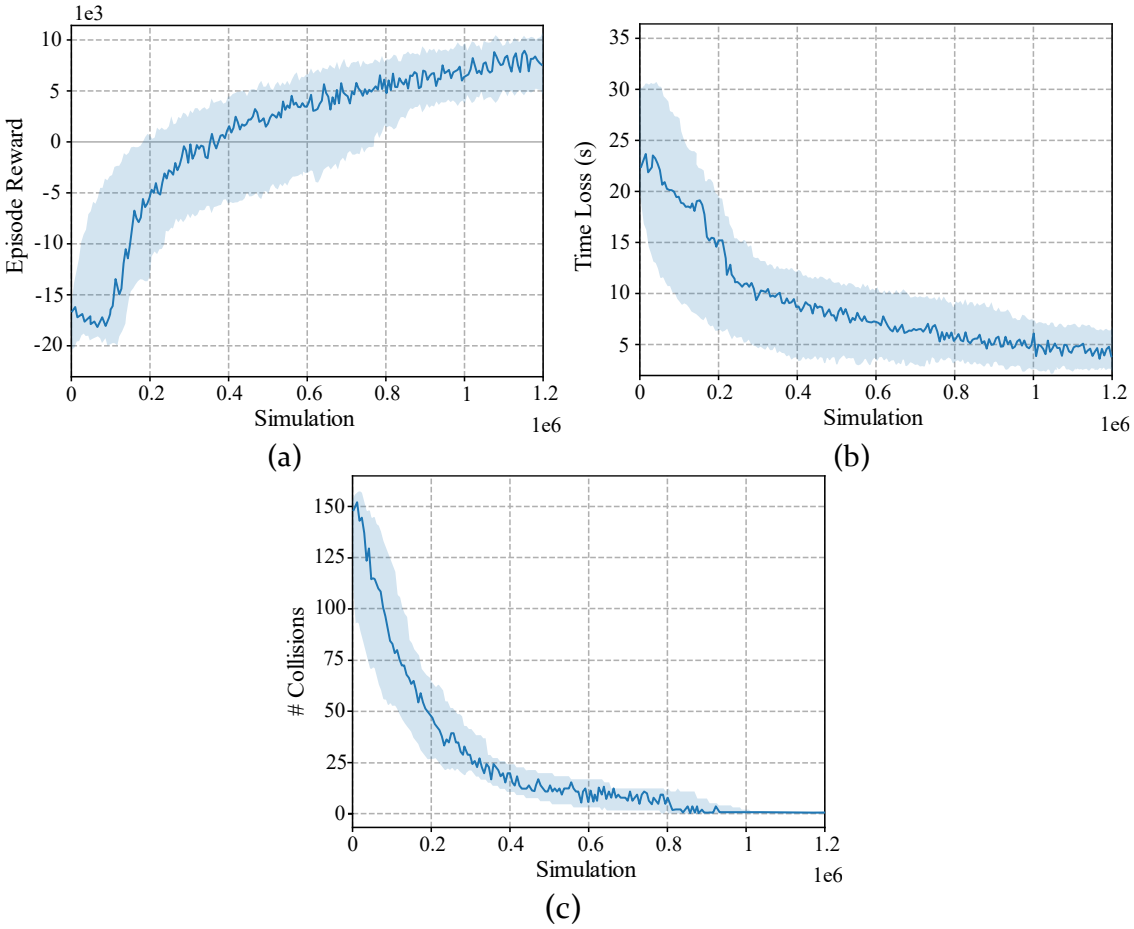


Figure 6-28. Results of AIM5LA training. Measurements show the mean (solid line) and standard deviation (shaded area) of the three runs. (a) Average episode reward; (b) Time loss; (c) Number of collisions. (Figure 12, p. 9, [415]).

Experiment #3 – Benchmarking AIM5LA

Lastly, in Experiment #3, the trained AIM5LA was benchmarked against other AIM and traffic-light based protocols. The results are shown in Table 6-12.

AIM5LA obtained outstanding results, as it is the only AIM protocol that reduced collisions in the testing scenario. When considering other metrics such as lost time, waiting time, contamination, or fuel consumption, AIM5LA achieved similar performance to other AIM methods such as Andert *et al.* [418] or, for example, RAIM [379]. However, keep in mind that the performance provided by AIM5LA is constrained by conflict resolution, so achieving similar results adds significant value to our work.

Furthermore, when we compare AIM5LA with AIM5LA v0.1 (which only takes into account the last latency of the *ego*-vehicle), it is clear that AIM5LA_v0.1 fails to resolve some conflicts due to the channel instability in its latency behavior, and the influence of other CAVs. Finally, compared to AIM5LA_v0.2 (i.e. AIM5LA_v0.1 plus the latency prediction module), the results improve considerably, however, there are still some conflicts that it is not able to deal with.

Overall, AIM5LA performs excellently in all the measures analyzed, reducing the time loss by 92.25% or the waiting time by more than 99.52% compared to other traditional traffic light (FX) algorithms. When comparing AIM5LA with other advanced traffic light control methods (iREDVD [42]), the results showed that AIM5LA was able to significantly reduce all metrics, such as time loss (84.5%), waiting time (99.2%), fuel consumption (35.6%) or gas emission pollution (48.5%).

Finally, comparing the performance of AIM5LA with other AIM methods such as *adv*.RAIM and Andert *et al.* we can conclude that the performance of our proposal is similar to these methods, although AIM5LA stands out when comparing the number of collisions, being the only one able to eliminate them.

It should be noted that in the original work where *adv*.RAIM was proposed, no conflicts occurred due to the non-existence of communication latencies, as this was not taken into account when setting up the simulation. However, when testing its operation in a more realistic 5G scenario, it was observed that latent collisions existed (see Table 6-12), indicating the importance of this factor.

TABLE 6-12. Testing scenario results. (Table 3, p. 10, [415]).

Algorithm	Time Loss (s)	Collisions	Waiting Time (s)	CO_2 emiss. (g)	PMx emiss. (mg)	Fuel cons. (ml)	Elect. cons. (W)
FX30	79.6 ± 8.9	0 ± 0	61.3 ± 7.9	101.7 ± 15.1	78.4 ± 8.5	391.1 ± 64.5	103.4 ± 11.5
FX60	70.1 ± 11.4	0 ± 0	50.7 ± 6.1	89.6 ± 9.8	66.9 ± 7.2	333.7 ± 42.6	99.7 ± 9.6
FX90	72.5 ± 7.4	0 ± 0	55.6 ± 7.0	95.8 ± 8.4	72.5 ± 7.8	351.5 ± 39.9	101.8 ± 9.8
iREDVD [313]	34.9 ± 3.3	0 ± 0	32.2 ± 4.4	53.4 ± 3.2	39.7 ± 6.1	205.2 ± 13.1	66.2 ± 4.4
adv.RAIM [379]	5.1 ± 1.2	49.9 ± 9.8	0.3 ± 0.1	25.4 ± 2.6	18.9 ± 2.8	124.4 ± 12.3	33.7 ± 3.8
Andert et al. [418]	4.9 ± 1.1	27.1 ± 3.1	0.2 ± 0.1	26.9 ± 2.3	17.5 ± 1.9	118.9 ± 18.7	31.6 ± 3.3
AIM5LA_v0.1	4.1 ± 1.4	32.1 ± 4.9	0.2 ± 0.1	26.1 ± 1.2	17.8 ± 1.9	119.1 ± 15.2	30.2 ± 3.2
AIM5LA_v0.2	4.8 ± 1.9	3.2 ± 2.6	0.3 ± 0.1	26.9 ± 1.4	18.4 ± 1.8	126.4 ± 18.4	32.3 ± 2.9
AIM5LA	5.4 ± 1.2	0 ± 0	0.3 ± 0.1	27.5 ± 1.9	19.2 ± 2.1	131.8 ± 17.4	34.2 ± 3.6

6.6.5 Conclusions

Urban environments are challenging environments where Connected Autonomous Vehicles (CAVs) are concerned. In particular, intersections are very complex scenarios, as CAVs face the challenge of crossing several different traffic flows, with different actors and multiple situations. There are multiple approaches to try to solve this problem, such as the use of path planning algorithms or hierarchy-based control, however, one approach that is gaining attention within the scientific community is Autonomous Intersection Management (AIM). AIMs can control all the CAVs crossing an intersection, centralizing the control in an intersection manager (IM), which is a great advantage over other approaches, since it allows centralizing all the knowledge and taking actions jointly, allowing optimizing different metrics jointly. However, most AIMs presented to date ignore

the latency that exists in the wireless communication networks used to communicate CAVs with IMs. Due to the latency inherent in the 5G communication network, AIMs that are capable of modeling the latency of the wireless network are needed to improve the control of CAVs, achieving a robust and fault-tolerant control policy.

To address this challenge, this paper presents a new latency-aware deep reinforcement learning-based AIM, called AIM5LA, for 5G wireless communication network. AIM5LA is a unique system that employs deep reinforcement learning, capable of modeling the communication delay between CAVs and the IM, forecasting the latency for each CAV using a Transformer-based deep neural network, as well as considering all CAVs simultaneously, creating an effective adaptive control policy, achieving seamless and resilient multi-agent control.

AIM5LA demonstrates impressive results across a wide range of metrics and different scenarios when is compared with other AIM [418] as well as other advanced traffic light-based control techniques such as iREDVD [313]. The results show that, unlike other AIM methods, AIM5LA can eliminate accidents. Comparing AIM5LA with traditional traffic light-based control methods, AIM5LA can reduce time loss by 92% and waiting time by more than 99%. In addition, AIM5LA is able to reduce lost time by 84% compared to advanced adaptive traffic light control methods, such as iREDVD, and achieve significant reductions in other metrics such as waiting time (99%) and fuel consumption (35%) or pollutant emission (48%).

6.7 6G Communications Network Framework in the context of Edge-Decentralized Cooperative Autonomous Driving Systems

6.7.1 Introduction

Because of the potential benefits of cooperative autonomous driving systems, such as AIM, and the unstoppable development of 5G/6G communication systems, a paper proposing a framework for solving the current problem of orchestrating cooperative control of autonomous vehicles using the future sixth-generation (6G) mobile communication network was proposed [424].

To this end, we suggested the group of specialized components that the future 6G network should integrate into its architecture to accelerate the development of decentralized cooperative control systems at the edge for Connected Autonomous Vehicles (CAVs). These systems make use of AI techniques and specialized AI hardware at different levels of the 6G communication architecture to guarantee the correct operation of the system, as well as its security and continuous and distributed training. This framework

enables the agile development of new Advanced Driving Assistance Systems (ADAS) and Cooperative-ADAS (C-ADAS) based on 6G and AI, as well as providing an efficient way to address the challenges and requirements that these new applications may require.

Within the literature related to research on 6G, connected autonomous vehicles, and other 6G framework proposals, the following works can be highlighted. The University of Oulu in Finland published a white paper on 6G in 2019 [425] identifying key drivers, research requirements, and challenges. The authors outline that 6G is expected to exceed terabits per second (> 1 Tbps) with extremely high-reliability communication (> 99.99999%) [426], ultra-low latency (< 0.1 ms), high-resolution location accuracy (at the centimeter level), and high-precision synchronization between devices (within 1 μs).

W. Saad *et al.* pointed out in [427] that the most important technologies for the effective development of 6G are THz-band communications, integrated terrestrial-space networks (ISTN), the use of reconfigurable intelligent surface (RIS), visible light communication (VLC), AI and distributed computing. On the other hand, the 6G roadmap proposed by Letaief *et al.* [428] highlights the importance of the use of AI in most of the communication network architecture levels, showing the importance that AI has had in recent years within the telecommunications field.

On the other hand, Yang *et al.* highlighted in [429] the requirements of 6G and highlighted the most promising technologies that will enable the evolution from 5G to 6G, such as the inclusion of AI and big, the use of mmWave and THz band communications, and ultrafast multiband transmission. Finally, in [430], the authors suggested several use cases that could be offered by the future 6G communications network, such as augmented reality, holographic telepresence, and autonomous mobility, based on the use of several of the advances proposed in [429].

AI will undoubtedly be part of the future 6G communications network at multiple levels of the architecture, thanks to the huge advances in recent years in deep learning, along with natural language processing and reinforcement learning.

These advances will enable the performance of all the existing core network functions, but also significantly extend the value of these networks by enabling completely new use cases such as cooperative vehicle control, tactile Internet, augmented reality, or holographic telepresence. In addition, these advances will make it possible to design solutions to problems that remain open today, such as latency prediction, intelligent noise reduction, or super-accurate positioning and synchronization.

Regarding the proposed work on the development of connected autonomous vehicles (CAVs) [431] using the 6G network, Zhifeng Yuan *et al.* evaluated in [432] the efficiency of a full-duplex 6G V2V communication. The results obtained using a simulator showed that the communication method they employed used 20% of the time-frequency resources.

On the other hand, [433] demonstrated the feasibility of using 6G for cooperative autonomous driving by studying the performance of vehicular networks over 6G. In addition, he showed an AI-based approach for cooperative autonomous driving.

Finally, examines the evolving technology of key enabling technologies for CAV deployment and explores two particular research directions: 6G for CAVs and CAVs for 6G. The former explores how various key enablers of 6G, such as terahertz, cell-free communication, edge intelligence, and artificial intelligence, can be used to provide key services for CAVs. The latter explores how CAVs can contribute to the efficient deployment and operation of future 6G systems. The authors argue that the marriage of CAVs and 6G networks will bring very important innovations to both. Therefore, joint designation of both could be a very effective means of achieving significant advances in either area, suggesting that this should be taken into account in the initial design phase of both. These outcomes highlight the significance of combining 6G and CAV.

6.7.2 Key Enabling Technologies

This section describes the basic technologies that enable the development and implementation of intelligent autonomous control systems for CAVs via 6G.

Connected Autonomous Vehicles (CAVs)

In recent years, many promising applications of CAVs have been proposed that can significantly improve road safety and efficiency, as well as traffic congestion and energy consumption, such as cooperative platooning [434], intelligent parking search [355], cooperative autonomous intersections [398], and cooperative multi-source sensing [435]. 6G networks for CAV control must meet the strict requirements to ensure vehicle safety and connectivity already desired for 6G. These rigorous requirements lead to important challenges that require fresh ideas and new communication technologies to surpass the existing ones.

Many studies have been conducted in recent years on Vehicle-to-Anything (V2X) communication paradigms, either through Direct Short-Range Communication (DSRC) or cellular V2X (C-V2X) [436]. The debate between these two communication standards currently opens up a wide range of research that could significantly improve both standards and address their drawbacks. While DSRC is the main communication standard in vehicular networks, using the IEEE

802.11p standard, cellular networks are a functional and more accessible solution on the market to provide ubiquitous and reliable connectivity in urban and rural regions.

DSRC employs the IEEE 802.11p standard, an evolution of the IEEE 802.11 standard that enables wireless access in vehicular environments, emphasizing the PHY and MAC layers. Nevertheless, IEEE 802.11p has significant limitations because of random channel access, coupled with the absence of QoS guarantees, the potential for infinite latency on the physical channel, or the need for large-scale deployments of IEEE 802.11p infrastructure [437].

Numerous investigations have indicated that C-V2X exhibits better capabilities with respect to the capacity, coverage, range, scalability, number of supported devices, and security [438]–[441]. Moreover, C-V2X utilizes the full range of C-V2X services (V2V, V2I, V2P, V2N) with end-to-end application support, which can reduce the overall cost due to the availability of infrastructure already present [442]. In addition, there is significant experience in cellular technologies among automotive manufacturers, which will facilitate the integration of C-V2X chips in vehicles. Approximately 210 million vehicles worldwide are currently equipped with mobile communications, 81 million of which are 4G, with an estimated average annual growth of 30% [443].

Finally, one of the strengths of C-V2X is its support for multiple operating modes, which can cover a wide spectrum of scenarios. C-V2X can operate through the *Uu* interface using cellular networks, or through the *PC5* interface for direct communication between C-V2X devices in the unlicensed 5.9 GHz band. The *Uu* interface is intended for V2N (Vehicle-to-Network) communication, while the *PC5* interface covers V2V, V2I, and V2P communication, which does not require a cellular-type infrastructure. These two interfaces can be seen in Figure 6-29.

Based on these observations, in our opinion, 6G cellular solutions can play a vital role in the development of CAVs due to their multiple advantages, such as pre-existing infrastructure, high reliability, low latency, multiple communication modes, high security, and high performance.

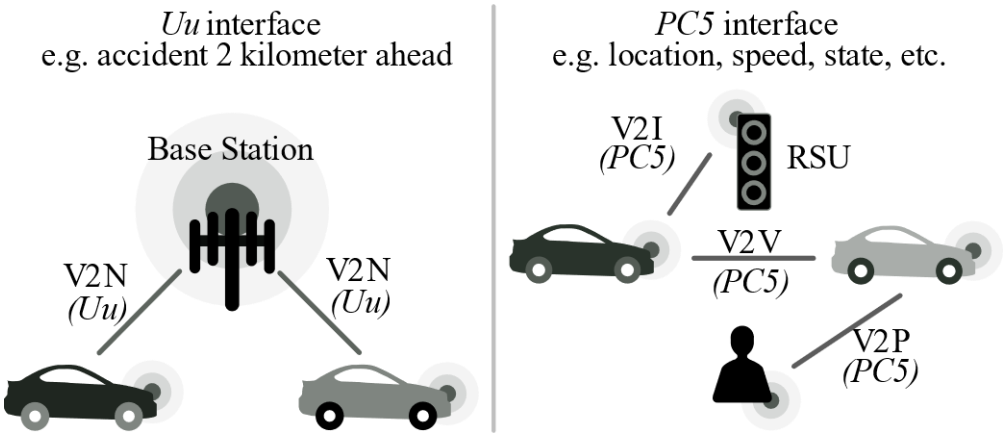


Figure 6-29. Diagram of C-V2X communication types. RSU ≡ Road-Side Unit (traffic lights, signals, etc.). (Figure 2, p. 5, [424]).

Edge Intelligence

The use of CAV in mobile network scenarios can benefit from Edge Intelligence (EI). EI enables the outsourcing of many vehicle tasks that can be processed in powerful edge servers co-located to the base stations to which they are connected.

EI is conceptualized as a method of collecting, analyzing, and processing data as close as possible to where it is collected on the network. . Real-time applications are ensured by analyzing data directly at the edge via EI. With this approach, EI can be defined more conveniently as a "decentralized cloud". EI enables a technological solution that meets the physical world at the edge, creating exciting new possibilities in technologies such as the IoT.

With EI, all data collected by multiple sensors in a CAV can be transmitted and stored directly to the Edge Nodes (ENs) of the mobile network. CAVs can leverage the benefits of ENs' performance in terms of speed, low latency, and high throughput to provide powerful and optimized analysis of collected data for real-time multi-agent applications. As a result, vehicles using EI can achieve higher inference accuracy, lower latency, and better collective intelligence because they can interact with other vehicles connected to the same EN without the need for a communication link. This allows them to react in emergency situations or facilitate platooning to improve traffic flow, among other applications.

Autonomous Unmanned Aerial Vehicles (AUAV)

Unmanned Aerial Vehicles (UAVs) combined with AI could play a key role in the implementation and development of autonomous vehicles controlled cooperatively through 6G. In view of the accelerated global development of UAV technology, the extensive availability of high-speed and low-cost network infrastructure, and the diverse UAV platforms with different characteristics and functionalities [124], UAVs can be ideal platforms to support a wide range of 6G network applications.

There are many areas where UAVs could be applied, such as in support of areas where more coverage is needed or where the overhead of an area needs to be reduced (e.g., a higher density of network-connected devices). Furthermore, these UAVs could perform their role as intelligent edge nodes, alleviating the overhead of terrestrial base stations or offering additional facilities. Beyond that, focused on cooperative autonomous driving of autonomous vehicles, UAVs could help create real-time, high-resolution three-dimensional maps of the area they control or help control, collecting images, depth maps, environmental data, noise pollution, etc.

The possibilities for UAVs in future 6G communication networks are unlimited and combined with improvements in UAV autonomy and increased UAV intelligence, the future of Autonomous UAVs (AUAVs) supporting communication networks in a cooperative, intelligent and proactive manner offers unlimited possibilities.

6.7.3 Connected Autonomous Vehicles Framework over 6G

This section presents the proposed framework to promote the development of intelligent autonomous control system applications for Connected Autonomous Vehicles (CAVs) on 6G.

Autonomous Vehicles (AV)

To develop cooperative autonomous driving with 6G, vehicles should be able to recognize and understand their environment and their own state (position, speed, route, load, etc.), communicate their perceptions, respond to coordinated external commands and provide information, and ensure the safety of passengers and other vehicles.

In order to comprehend the environment, vehicles must be equipped with sophisticated sensors that provide high resolution and enhanced awareness of the environment, e.g., on-board cameras, RADAR, LIDAR, IMU (inertial measurement unit), IR/ultrasonic sensors, GPS, GALILEO, GLONASS, etc. The cameras shall have a high resolution covering a 360° field of view to provide a detailed view of the current state of the entire environment. LIDAR sensors will be able to "see" in emergency situations, low light, glare, etc., and measure short distances with high accuracy. RADAR will be able to measure long distances with high accuracy and range. Ultrasonic/infrared sensors can be used to detect very close objects. Finally, the use of multiple tracking systems, such as GPS, GALILEO, and GLONASS, will reduce tracking errors to a few centimeters and ensure a high degree of tracking reliability, which is extremely important to avoid accidents due to position errors.

These sensors enable the vehicle to recognize its environment and react to external factors. For example, if an object is suddenly detected in front of the vehicle, the emergency steering system must automatically steer the vehicle away

from the obstacle to soften the impact or, if unavoidable, mitigate the consequences (airbag activation, emergency braking, emergency lighting, door unlocking, etc.).

One of the most important features of 6G cooperative autonomous driving is the ability to detect and react to other vehicles or road users (such as pedestrians or bicyclists) in advance. AUAVs can track other road users via cameras, 6G cellular signals, or pedestrian-like triggers (activated by signals at traffic control points, such as traffic lights). It can then immediately alert the vehicle when the user's path may cross, and the vehicle can take this into account as if it were a cooperative autonomous control system.

Therefore, besides the sensors, CAVs need a 6G communication module. This module will be responsible for sending and receiving information from the 6G network. In this case, the focus is on URLLC. In addition, this module can use network slicing to leverage eMBB use cases to transmit or receive video streams or other high-bandwidth-intensive applications. The data collected by all these sensors are sent to the nearest base station (usually a PicoCell base station), which forwards it to an iNB (intelligent-NodeB), referred to as the Vehicle Edge Node (VEN), located in the Vehicle Local Edge Cloud (VLEC) with higher processing capacity.

For further privacy and security protection, data collected by cameras and sensors can be preprocessed in the vehicle to remove personal or sensitive data [444]. It is our belief that the vehicle has some processing capability in emergencies. However, to reduce vehicle energy consumption, the devices performing actions are disabled and only activated when no control information is received. When these actions are executed locally, they are locally optimal because they do not take into account other non-connected vehicles in their surroundings.

Vehicle Edge Computing (VEC) and Vehicle Edge Nodes (VEN)

Taking Mobile Edge Computing (MEC) for 5G as a reference, we propose Vehicle Edge Nodes (VENs) for 6G and autonomous driving, which form the Vehicle Edge Cloud (VEC) and are responsible for cooperative decision making for autonomous driving, as close as possible to the vehicle to reduce control latency and improve safety. Thus, VECs become Intelligent Edge Nodes (iENs) responsible for cooperative control of autonomous driving vehicles. In addition to 6G communication base stations, iENs must have specialized AI hardware (neural processors, GPUs, TPUs, FPGAs, ASICs, etc.) to tune and infer neural networks, as well as specialized databases to store the local experience of the CAVs connected to them, so that the autonomous driving of vehicles can be coordinated based on the real-time data received from the vehicles connected to each VEN. This specific hardware is expected to be used in the 6G network.

The most important data to be processed in a VEN is the data from the vehicles connected to each VEN, i.e., processing information from the nearest vehicles and controlling them. In this way, cooperative intelligence is ensured by combining information from multiple vehicles in the same VEN and considering only those vehicles that can interact with each other based on their proximity to the VEN (through attention mechanisms, clustering, reinforcement learning, etc.), ensuring that each vehicle and user is related to the cooperative control of other vehicles.

One of the most challenging issues in developing efficient VEN-based solutions is how to deal with different users and different types of vehicle applications while leveraging collective intelligence. We believe that there can be special intelligence mechanisms that share the workload among existing VENs so that the focus is only on the vehicles operated by each VEC. In addition, VENs could use mechanisms such as network slicing to parallelize the various cooperative control applications they perform for vehicles.

PicoCells

The layer below the VECs is the communication PicoCells, which are designed for Massive-URLLC (mURLLC). This greatly reduces latency and improves mobile network coverage. In addition to traditional terrestrial PicoCells, this can include AUAVs and Low Earth Orbit (LEO) satellites, CAVs (with *PC5* interfaces), and IoT devices, sensors, or nodes on the public road, such as traffic lights, streetlights, signs, etc.

One of the new use cases planned for 6G will be mURLLC. This new use case will be an elegant convergence of URLLC and mMTC use cases and presents a trade-off between reliability, latency, power consumption, cost, and scalability, which in turn will require significant changes in network design and key performance indicators (KPIs) to be achieved. Thus, with mURLLC, it is possible to deploy high-density, low-cost devices that have very low power consumption (as in the case of mMTC) but meet very stringent requirements (KPI defined by mURLLC use case) in terms of communication, latency, packet loss, etc., which makes several new applications possible. Since these PicoCells do not need to process the information, but only forward it, they are characterized by low power consumption, low processing load, and low cost, which makes it possible to deploy them in large numbers in large cities.

Vehicle Local Edge Cloud (VLEC)

The group of VENs would be called the Vehicle Local Edge Cloud (VLEC). Located at the edge of the cloud, the VLEC would be part of the VEC and would consist of distributed AI between the individual VENs that comprise each VLEC. This distributed AI would allow ad-hoc deployment of additional VENs when needed,

as it is scalable and can adapt to the needs of each region, for instance by deploying new AUAVs to cover a specific area in case of a possible traffic overload. Thanks to intelligent clustering algorithms, predictive algorithms, etc., the number of VENs composing each VLEC can vary according to demand, allowing cooperative control algorithms to improve long-term trip planning and reduce congestion, travel time, etc. For example, the VENs on a large avenue in a city can form a VLEC.

For cooperative vehicle control, VENs would be responsible for creating a shared multidimensional map of the environment they control in real-time, combining data from all vehicles and all data sources collected locally. Here "local" means that each VEN will process data from the nearest vehicle (to which it is connected) and share it with other VENs in the same VLEC. In addition, each VEN will have a local experiences database. From these local experiences, the system will be able to fine-tune the existing cooperative vehicle control systems to each geographic area, since not all intersections, roads, crossroads, junctions, detours, etc. are identical. In addition to fine-tuning the control algorithm, VENs will supervise the inference and control of the nearest vehicle. To do this, the state that each vehicle should have at the next interval is inferred using a local joint map.

Central Vehicular Cloud (CVC)

At the apex of our proposed framework is the Central Vehicular Cloud (CVC). This CVC would perform global-scale full training for different intelligent vehicle control systems, utilizing the experiences gathered by the VENs locally. In other words, the CVC will use a global experience database (local vehicle status together with a global map) to optimize each intelligent control system. Hence, insights gained from experience can be used optimally, resulting in a highly intelligent system that leverages big data.

This CVC may consist of a single cloud or may be distributed nationwide. We believe that as distributed intelligence continues to advance, there should be multiple CVCs across the country that can learn faster, acquire more knowledge, and ensure resilience through the use of distributed learning (e.g., federated learning).

Consequently, this CVC would include: large processing servers with AI-specific hardware; a database to store local experiences, virtual maps, etc.; another database to store the internal parameters of the neural network that models each intelligent vehicle control system; communication systems; security systems; load balancing; etc.

Multidimensional Virtual Maps

High-definition multidimensional virtual maps (i.e., the information can come from multiple sensors of different nature: RADAR, LIDAR, ultrasound, cameras,

etc.) as a crucial task, since the accuracy of these maps, would offer high performance, precision, granularity, and enormous safety to the different cooperative vehicular control systems. If the environment to be controlled is virtually represented, the control task becomes down to learning the physics of the different vehicles and virtually emulating them for optimal control. This physics knowledge can be done straightforwardly using deep learning and reinforcement learning techniques.

These maps could be created and used in a hierarchical way, where the higher we are in the architectural hierarchy, the more global the map will be, and the lower we are in the hierarchy, the more accurate, granular, and detailed the map will be. Thus, in the CVC the global map of the entire vehicle network could be used throughout the training and optimization process. Meanwhile, in the VENs there would be a very detailed local map of the regions they control that will be used to fine-tune cooperative control systems as well as vehicle inference.

6.7.4 Use Case

In this section we will explain the workflow of the proposed 6G framework, which bases its operation on three main components: ultra-low latency communications, high-reliability communications, and intelligent systems, capable of training and inferring the different artificial intelligence algorithms.

The entire development of the framework has been focused on prioritizing and promoting the highest possible fluidity and safety, as the control system can be very large and complex, having to maximize the functionality of each individual intelligent system (cooperative vehicle control, route planning, intelligent parking management, and search, platooning, etc.). Therefore, we consider it very important that data processing and response be as fast as possible, with the highest level of security allowed.

In order to provide a better overview of the workflow in the proposed framework, in this section, we will show the complete process from data capture by vehicles, data collection at the VENs, overall training of the intelligent control systems performed by the CVCs, fine-tuning of each model in the VENs, and inference and sending the control information to the vehicles through PicoCells.

Data acquisition and forwarding to VECs

Data acquisition would be carried out by CAVs and some PicoCells containing sensors such as cameras, noise sensors, pollution sensors, etc. This information would be sent to the nearest PicoCell, which would collect all the information coming from all nearby users and eliminate redundant information. With the use of PicoCells, latency and the sending of non-relevant information to higher nodes in the network could be minimized, as well as high reliability and coverage.

After gathering the data and eliminating redundant information, the PicoCells send the information to the VENs using the most reliable and fastest communication technology (terahertz, visible light communication (VLC), or fiber optics). Once the VENs have collected the data, they send it to the VLC for global training of the control systems. The forwarding of this information is not as critical as the previous one, so this forwarding could be done at the mark, periodically, or in packets.

The experiences collected (status of each vehicle + sensors) by each VEN are stored locally in a local experience database for later use during fine-tuning.

Global training

Using the experiences collected by the VECs, which could be stored locally, or distributed among all existing CVC servers, the CVCs would globally train the different AI systems in charge of vehicle control (autonomous intersection control, autonomous vehicle control, etc.) using AI, DL, RL, distributed learning, etc. Moreover, given the advances in distributed learning (such as federated learning), the training of multiple CVCs would be very efficient and scalable. For this global training, the experiences collected at the global level would be used to generate a global map with a low level of detail to train the different intelligent systems efficiently and safely.

It is important to note that once the intelligent systems are trained, a copy of the internal parameters modeling the different advanced control systems would be saved in a specialized database so that the LECs can perform a fine-tuning based on the global optimization performed in the CVC.

Local Fine-Tuning

The fine-tuning procedure is performed by each LEC for each of the control systems. For this purpose, each LEC would have dedicated hardware for this process. The procedure would consist of each LEC loading the parameters of each of the control systems stored in the CVC database, and from the experiences collected locally, a high-definition local map would be created, with a high level of detail. Then, each of the control algorithms would perform a fine-tuning process, limiting the changes that can be made to the parameters. This procedure would allow adjusting each control system to the particularities of each zone at the local level controlled by each LEC (intersections, ramps, lane widths, maximum allowed speed, asphalt roughness, etc.), allowing the optimal performance of each intelligent control system.

LECs clusters (VLECs) would use this tuning process to fine-tune control systems that require the coordination of several LECs (route planning, intelligent parking systems, public transport route distribution, etc.).

Inference and sending data to CAVs

Finally, the inference of each of the advanced control systems would be performed at the LECs, since they are the closest nodes to the vehicles with sufficient computational capacity to form the local map, store the contributed experiences, and compute them. The inferred control data would be sent to the vehicles through mURLLC communication, passing the information from the LECs to the PicoCell and from the PicoCell to the CAVs.

Only in case of emergency (error in the communication systems), the CAVs would be able to infer at the individual level, considering their current state and the sensors they possess. This emergency inference would ensure the safety of all users, which is essential for the proper functioning of CAVs in society, although this inference would not be optimal as it would not be able to take into account the state of all vehicles and users surrounding the vehicle.

Finally, the proposed framework is shown in Figure 6-30.

Practical case, complex urban intersection and connected autonomous vehicles.

To illustrate the use of the proposed framework, we have chosen a complex unsignalized urban intersection. There are several lanes per direction at the intersection, and the CAVs only follow the control of the distributed control system at the VEC, so there are no traffic lights to regulate the vehicle movement. All CAVs are connected to the central controller (VEC) located near the intersection, which allows coordinated control of all CAVs with the lowest latency. All CAVs send their data to VEC via 6G URLLC. Each VEC can train a variety of AI control systems (in this use case, it will be the coordinated control of autonomous cars). After this, the VEC will have a high-resolution map with all the local information needed to localize and tune the control system at the local level, and at the medium level, the VLECs will also tune the needed control system, located at a higher hierarchical level. In addition, VECs send the data to the CVC for training on a large scale. Finally, VEC obtains the vehicle status at the next time interval (because it performs the CAV control task) and sends the data to CAV. In addition, with AUAV, each VEC can have a PicoCell for communication and inference with CAV. The PicoCell can communicate with multiple CAVs and, in case of emergency, uses the data provided by the CAV sensor to infer vehicle status.

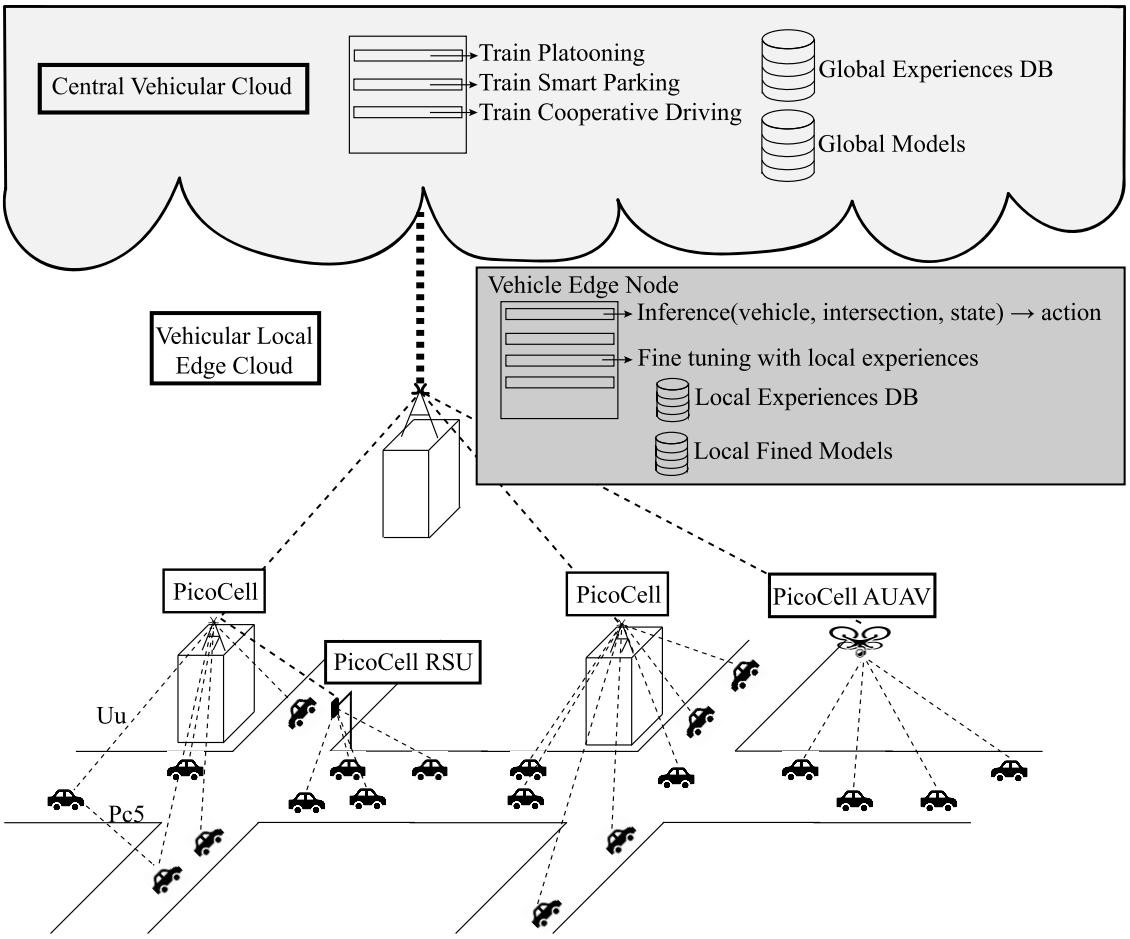


Figure 6-30. Proposed framework. (Figure 4, p. 12, [424]).

6.7.5 Conclusions

With the opportunities offered by CAVs and next-generation 6G communications to deploy distributed control and cooperation services and systems at the seamless edge, in this subsection, we proposed a framework for CAVs to make use of various cooperative autonomous driving systems via specialized 6G nodes. This new set of nodes would be integrated within the 6G network architecture natively in what we define as VEC. The VEC would consist of nodes called VENs, which would be responsible for controlling the CAVs using advanced control systems. However, these nodes do not have enough computational capacity to train all the control systems per complex. Therefore, we envision the existence of one or several CVCs, which through collaborative and distributed learning (such as federated learning) perform the exhaustive training of the different intelligent cooperative control systems using the experiences collected globally. After this training, the NVCs would be able to perform fine-tuning processes to the different advanced control systems based on the experiences collected locally, as well as the inference of each of the control systems for each of the CAVs under their control.

To take advantage of the benefits offered by cooperative control, we propose the existence of intelligent and dynamic clusters of VECs, called VLECs, which

could employ collaborative learning knowledge on a larger scale to optimize different scenarios that require this type of control such as large avenues, perimeter roads, public transportation system, alternative route discovery, etc.

Lastly, the nodes in charge of forwarding the information to/from the CAVs are called PicoCells. These PicoCells are focused on the 6G mURLLC communication use case, with very low cost, low latency, low range, and high reliability, in order to deploy as many of these nodes as possible. By virtue of these characteristics, these PicoCells could be integrated into IoT devices, traffic signals, traffic lights, AUAVs, etc., and even be part of CAVs themselves, which could forward information from other CAVs that do not have access to another PicoCell. The advantage that PicoCells formed by AUAVs could offer is that they would allow an ad-hoc deployment in those areas where an increase of connected devices is foreseen, or that are necessary to alleviate the existing workload in an area or to extend the existing coverage.

The main difference of this framework is that it is specifically designed for cooperative and distributed control of CAVs using 6G communication networks. In addition, it is flexible and extensible and is designed to provide multiple layers of control. The framework uses several sets of specialized intelligent nodes (VEN, CVC, PicoCells) to provide the highest level of performance and security.

In this context, the advantage that 6G offers over 5G, and even Beyond5G (B5G), is the complete integration with specific hardware for processing artificial intelligence-based systems. We consider that these relevant technologies present a degree of maturity too low in 5G/B5G, or are outside the defined scope of 5G/B5G. Therefore, we consider that the incorporation of 6G for this framework is necessary for its proper development, as it requires the fusion of the URLLC capabilities of 5G/B5G, as well as IoT mMTC, together with the AI capabilities of intelligent transport and control systems, as well as high-performance hardware and databases specifically designed for AI, all located at both the CVC and VEN)level, close to the CAVs, ensuring that latency would be minimal and offering high reliability.

6.8 Conclusions to this chapter

CAVs have the opportunity to dramatically improve the quality of life in congested urban environments. However, these benefits cannot be realized if CAVs cannot cope with diverse and complex traffic conditions in a seamless and resilient manner. The severe constraints that urban environments impose on CAVs, especially the high variability of environments, situations, and scenarios, adversely affect the availability and success of these systems. Thanks to advances in various components of driving and control systems, as well as the understanding and handling of unpredictable situations through multi-agent deep reinforcement

learning (MADRL) methods, we believe we are at the tipping point where these systems will achieve a robust and reliable response to such constraints. MADRL enables the development of dynamic systems capable of adapting to an infinite number of situations and acting collectively and proactively, anticipating dangerous situations and ultimately avoiding accidents and increasing fluidity.

To leverage the potential of CAVs, we need a better understanding of CAV control in urban environments, and a suitable system to predict, optimize its performance and train the system in a wide variety of situations, and this is where MADRL excels. In this section, we have explored how MADRL enables the development of wireless-based deep control for traffic control, which shows promising performance in complex environments. In addition, the interoperability of CAVs with ITS over mobile wireless communication networks has been explored.

First, a first approach to the development of an AIM using MADRL was developed. This first AIM was based on an *ego*-centric control policy and was called RAIM (Reinforced-AIM). That is, for each connected autonomous vehicle (CAV) to be controlled (*ego*-vehicle), RAIM calculated the speed at which it should drive to avoid collisions and minimize the waiting time considering the state of the *ego*-vehicle and the other vehicles. The results showed high performance in urban scenarios of medium complexity, outperforming other AIM proposals and traffic light-based control systems.

After this, the previously developed AIM was analyzed in-depth, making it more robust and intelligent, solving several problems it had due to its design. The main problem of the RAIM was that the size of the input variable depended on the number of vehicles to be controlled. To solve this problem, *adv*.RAIM adopted a coding network based on LSTMs. Thus, *adv*.RAIM had as input variables the state of the *ego*-vehicle and the output of the LSTM coding network. To this LSTM coding network was sequentially input the state of each CAV in the intersection and at the output was the coding that modeled the state of all CAVs at the intersection to be controlled. In this case, the results demonstrated excellent performance on numerous metrics, outperforming other proposed AIM and traffic light-based control systems in very complex scenarios.

Due to the complexity of training MADRL systems, in the following work, it was decided to explore different systems to accelerate the training of new MADRL-based systems and, consequently, of new AIM. For this purpose, it was decided to analyze a new training approach using demonstrations. In this case, in this research study, an Oracle was trained by supervised learning, extracting the observations from the traffic simulator SUMO. This Oracle was then used to train the AIM. During training, the AIM kept asking the Oracle what action to take, but less and less, until a point was reached where the AIM no longer asked the Oracle

and obtained a superior control policy to the Oracle. The results of this new training approach allowed the *adv*.RAIM to be trained 5 to 6 times faster, reducing the number of simulations performed and the computational cost.

The design of all the proposed decentralized control systems requires a series of modules and features (hardware, databases, low communication latency, etc.) that make them unique within the communications network. Therefore, another study was to develop the necessary framework to deploy this set of systems (including AIM) within the future 6G mobile communications network. This study analyzed the 6G network as it enables ultra-low latency and high-reliability communication with vehicles, which is very important for AIMs. The conclusions allowed developing future 6G-based decentralized control systems more simply and efficiently, which paves the way for future improvements and implementations.

Finally, due to the importance that communication latency can have on AIMs and the lack of works analyzing this important problem, it was decided to develop a latency-aware AIM for the 5G communication network using MADRL, called AIM5LA. By latency-aware, we mean that AIM5LA was able to model the latency inherent in 5G communication between the CAVs and the intersection manager (IM). In addition to considering the historical latency suffered by each CAV, AIM5LA was able to predict with high accuracy the latency that each vehicle was going to suffer in the next control interval. Thus, in addition to considering these latencies, AIM5LA considered all the CAVs to be controlled to obtain a robust and collision-free control policy. The results showed that AIM5LA was able to eliminate collisions and obtain an efficient control policy, contrary to other AIMs analyzed.

The main conclusions of this chapter are that MADRL is an effective technique for the development of advanced technologies for decentralized and cooperative control of CAVs and that the existing state of the art in this field should be taken into account when designing new AIMs and decentralized cooperative control systems. Moreover, if analyzed in the context of wireless communication technologies, MADRL-based decentralized cooperative systems may be an excellent solution for the development of such systems in the near future, however, there are still tasks to be solved such as the incorporation of external systems (e.g., mobile devices to monitor and communicate with vehicles), the selection of optimal control policies, the incorporation of non-controllable external agents (pedestrians, cyclists, emergency vehicles), the efficient deployment of the systems and verification of their effectiveness, as well as the adaptation of the control method to a diverse range of communication networks. The results of the studies conducted here will contribute to the design of future decentralized control systems more efficiently and robustly.

6.9 Publications associated with this research

The works related to this chapter are as follows:

Articles:

Guillen-Perez, A.; Cano, M.-D., "AIM5LA: A Latency-Aware Deep Reinforcement Learning-Based Autonomous Intersection Management system for 5G Communication Networks," *Sensors*, vol. Accepted, pp. 1–20, 2022.

2020 Journal Impact Factor (JIF): 3.576. (Q2), Rank: 82/273 in Engineering, Electrical & Electronics.

Guillen-Perez, A.; Cano, M.-D., "Multi-Agent Deep Reinforcement Learning to Manage Connected Autonomous Vehicles at Tomorrow's Intersections," *IEEE Trans. Veh. Technol.*, vol. On review, pp. 1–12, 2022.

2020 Journal Impact Factor (JIF): 5.978. (Q1), Rank: 15/91 in Telecommunications; (Q1), Rank: 32/273 in Engineering, Electrical & Electronic.

Guillen-Perez, A.; Cano, M.-D., "Learning from Oracle Demonstrations – A new approach to develop Autonomous Intersection Management control algorithms based on Multi-Agent Deep Reinforcement Learning," *IEEE Access*, vol. On Review, pp. 1–12, 2022.

2020 Journal Impact Factor (JIF): 3.367. (Q2), Rank: 65/161 in Computer Science & Information Systems.

Guillen-Perez, A.; Cano, M.-D., "6G Communications Network Framework in the context of Edge-Decentralized Cooperative Autonomous Driving," *Appl. Sci.*, vol. On Review, pp. 1–12, 2022.

2020 Journal Impact Factor (JIF): 2.679. (Q2), Rank: 38/90 in Engineering Multidisciplinary.

Congress:

Guillen-Perez, A.; Cano, M.-D., "Cómo la superresolución puede ayudar a los vehículos autónomos conectados", *VI Jornadas Doctorales UPCT*, Universidad de Murcia. 2020. Oral communication.

Guillen-Perez, A.; Cano, M.-D., "RAIM: Reinforced Autonomous Intersection Management - AIM based on MADRL," in *NeurIPS 2020 - Workshop Challenges of*

Real-World RL, 2020, pp. 1–12, Accessed: Feb. 16, 2022. [Online]. Available: https://www.researchgate.net/publication/357957238_RAIM_Reinforced_Autonomous_Intersection_Management_-_AIM_based_on_MADRL.

Chapter 7: General Conclusions and Future work

7.1 Introduction

The field of Intelligent Transportation Systems (ITS) is undergoing a major and exciting revolution as a result of huge advances in the field of artificial intelligence, deep learning, reinforcement learning, and the development of new technologies such as autonomous vehicles and advanced communication protocols such as 5G/6G and unmanned aerial vehicles. Developments in these areas have paved the way for the deployment of new traffic management solutions and the future improvement of existing ones. To achieve this goal, it is imperative that we lay the groundwork for the design and constraints of future cognitive ITS.

Achieving cognitive ITS through AI has several exciting challenges. A central challenge in creating intelligent cognitive ITS is how to communicate with the agents around them (vehicles, pedestrians, other ITS, etc.), in an intelligent and timely manner, to take advantage of the goodness offered by collective intelligence. Another central challenge in the creation of intelligent cognitive ITS is how to ensure maximum safety. The fundamental basis for the development of these new ITS is security; any failure in this will mean a drop in the confidence of these ITS.

For the first challenge, the use of advanced communication technologies such as 5G-6G enables the efficient implementation of communications between multiple devices of different characteristics with great ease and security. In addition, new opportunities open up for these systems due to the significant decrease in communication latency. However, the implementation of 5G-6G-like communication between different agents and ITS requires further research and policy coordination to achieve optimal utilization of this technology.

As for the second challenge, ensuring maximum security in the development of new cognitive ITS requires a deeper understanding of the behavioral dynamics of all real-world users and agents, with all that this entails. Consequently, to ensure maximum security, new techniques and control mechanisms are needed to ensure that new cognitive ITS do not ignore security-related information or communicate it incorrectly. This requires the learning of new behaviors, based on models of these behaviors, in order to optimize the learning experience for intelligent cognitive ITS.

The objective of the thesis has been to contribute to the improvement of the cognitive capability of ITS by using cutting-edge techniques and technologies such as WiFi and 5G/6G communication networks, artificial intelligence, multi-agent deep reinforcement learning, and unmanned aerial vehicles, integrating these technologies in applications related to urban mobility and ITS to achieve more efficient and simpler solutions that can be implemented using standard

communication standards. In addition, the use of AI, distributed intelligence, and autonomous vehicles have led to more robust, flexible, and scalable solutions.

To this end, after showing the theoretical background used during this dissertation in Chapter 2, Chapter 3 analyzed the use of unmanned aerial vehicles (UAVs) to contribute to the development of future ITS. To this end, the state of the art of different routing protocols, propagation models, positioning protocols, and mobility models were first reviewed. Then, the impact of the use of unmanned aerial vehicles (UAVs) on the integrated communication modules of these devices was analyzed. The conclusions drawn from this chapter showed the imperative need to develop models and protocols adapted to aerial networks formed by UAVs, due to their unique characteristics of mobility, energy constraints, and network topology. In addition, it was possible to demonstrate the great influence that the UAV chassis can have on the communications module, showing propagation losses of up to 10dB at some propagation angles.

In Chapter 4, the methods used for counting people in indoor and outdoor environments using passive WiFi methods were analyzed. Using artificial intelligence techniques, an algorithm capable of providing excellent performance, exceeding 90% accuracy, was developed. Using these simple algorithms, ITS can use the information provided to adapt control systems and control both vehicle and pedestrian traffic, maximizing road safety as well as minimizing waiting times. However, the main drawback of such methods is the randomness of the MAC of the mobile devices used to differentiate pedestrians. However, the probability of a device modifying its MAC in a short capture interval is small [296].

Chapter 5 shows an intelligent use of traffic control systems based on traffic lights, as well as their principle of operation. In this chapter, an advanced control algorithm called iREDVD has been developed. This control algorithm is based on queue congestion control algorithms and was optimized using genetic algorithms, focused on performing optimization tasks where the space of variables to be explored is extremely large and complex. This optimization using a genetic algorithm allowed iREDVD to obtain excellent results when comparing its performance with other traffic control algorithms Based on the results obtained in this chapter, in the design and deployment of new traffic light control systems at large-scale intersections, iREDVD can offer a great advantage over other control systems by allowing to control traffic in a very efficient and anticipatory way through the deployment of low cost, low requirements, and low power IoT devices, outperforming not only iREDVD but also other known traffic management methods in all the analyzed metrics.

Finally, Chapter 6 shows the use of Multi-Agent Deep Reinforcement Learning (MADRL) for the control of Connected Autonomous Vehicles (CAV), as well as the framework on which these control systems should be based for their

correct implementation. These decentralized cooperative control systems (Autonomous Intersection Management, AIM) for CAVs have the advantage of providing higher controllability and performance, thanks to having an agent at the edge of the communication network that integrates all the knowledge and states, and tells each CAV what action to take to maximize traffic flow, as well as to eliminate accidents. In other words, these MADRL-trained AIMs can provide a way for each CAV to coordinate its actions to maximize traffic flow and eliminate accidents. These control systems offer exciting possibilities for future advances in AIMs and cooperative vehicle control methods. In particular, they may pave the way for new types of automated vehicles and mobile information systems that provide safety and convenience to users. In this chapter, different MADRL system training techniques are explored, as well as a 6G framework on which these cooperative intelligent control systems would rely. In addition, due to the importance that latency can have on vehicular control, a system capable of adapting vehicular control to changes in 5G network latency is proposed.

7.2 Key future research directions

The proposals developed during this thesis make it possible for the cognitive capacity of intelligent transportation systems to increase enormously. However, due to the various challenges associated with them, there is still room for research. The following key research directions are highlighted:

1. The autonomous control problem. This research direction attempts to model and explain the fundamental challenges in the planning, computational, and communication aspects of artificial intelligence-based cooperative autonomous control systems and their integration with real-world road traffic. For example, many tasks have to be defined and performed before the system can be considered fully autonomous. One such task is dealing with potential collisions or conflicts where a critical decision has to be made. For vehicles to be fully autonomous, several models must be developed to estimate the losses associated with a collision.

2. Study of the performance and requirements of the key components of the communications network architecture that will allow CAVs to operate autonomously. This research direction aims to allow the identification of the key components, as well as to study their requirements for the correct deployment of different cooperative control systems on the ITS. In the case of CAVs, these include perception intelligence, control intelligence, decision making and planning, and network and data communication. However, although CAVs are envisioned to operate in the vehicle itself, there are many other parts of the vehicle that need to be considered, such as the surrounding road, the environment, the weather and the number of cars or other road users, emergency vehicles, cyclists, among others.

3. Different methods to solve the problem of decision-making and planning of aerial networks composed of unmanned aerial vehicles (UAVs) for ITS support. Currently, there are many challenges in aerial networks composed of UAVs, which are presented in multiple studies. Such challenges require solutions to decision-making and planning problems in coordination with flight plans, mission planning, and air traffic optimization simulations to ensure a safe flight. Different ways to support decision-making, planning, and coordination of drone-assisted autonomous aerial vehicles in different environments are highlighted, including UAV operations in urban or mixed traffic agglomerations and beyond. Different approaches and implementation requirements are also discussed. For example, different systems are proposed to collect and evaluate sensor data, in addition, to support from ground networks, and from this information, determine decision making.

4. Development of cooperative autonomous control methods for CAVs and UAVs for autonomous vehicle operation. This research direction aims to create systems capable of sensing the environment, responding appropriately to it, and performing decision-making. This will rely on data-driven approaches and automatic reasoning. The relevant research directions are further outlined in the following sections.

In conclusion, the direction of future research should focus on further improving the performance of cooperative ITS, such as MADRL transition based on the capability of multiple very different agents, the use of the latest technologies such as 5G or 6G, high-speed networks, IoT or UAV, by continuously developing knowledge and understanding of the problems to be addressed, in addition to the continued implementation of effective control systems, thus achieving more efficient and simple solutions. We believe that the main focus of future research will be the development of control algorithms capable of learning agent and human behaviors, which will enable new cognitive intelligent ITS to operate in a context-aware, dynamic and safe way.

.

References

[1] J. M. Sussman, *Perspectives on Intelligent Transportation Systems (ITS)*. Boston, MA: Springer US, 2005.

[2] *World Urbanization Prospects: The 2018 Revision*. UN, 2019.

[3] J. Zhang, F. Y. Wang, K. Wang, W. H. Lin, X. Xu, and C. Chen, "Data-driven intelligent transportation systems: A survey," *IEEE Trans. Intell. Transp. Syst.*, vol. 12, no. 4, pp. 1624–1639, 2011.

[4] M. Veres and M. Moussa, "Deep Learning for Intelligent Transportation Systems: A Survey of Emerging Trends," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 8, pp. 3152–3168, 2020.

[5] S. Russel and P. Norvig, *Artificial intelligence—a modern approach 3rd Edition*. 2012.

[6] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Berlin, Heidelberg: Springer-Verlag, 2006.

[7] M. I. Jordan and T. M. Mitchell, "Machine learning: Trends, perspectives, and prospects," *Science (80-.).*, vol. 349, no. 6245, pp. 255–260, Jul. 2015.

[8] Y. Lecun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[9] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.

[10] M. L. Littman, "Reinforcement learning improves behaviour from evaluative feedback," *Nature*, vol. 521, no. 7553, pp. 445–451, May 2015.

[11] R. S. Sutton and A. G. Barto, "Reinforcement Learning: An Introduction," *IEEE Trans. Neural Networks*, vol. 9, no. 5, pp. 1054–1054, Sep. 1998.

[12] J. R. Quinlan, "Induction of decision trees," *Mach. Learn.*, vol. 1, no. 1, pp. 81–106, Mar. 1986.

[13] W. S. McCulloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity," *Bull. Math. Biophys.*, vol. 5, no. 4, pp. 115–133, Dec. 1943.

[14] M. A. Hearst, S. T. Dumais, E. Osuna, J. Platt, and B. Scholkopf, "Support vector machines," *IEEE Intell. Syst. their Appl.*, vol. 13, no. 4, pp. 18–28, Jul. 1998.

[15] D. Opitz and R. Maclin, "Popular Ensemble Methods: An Empirical Study," *J. Artif. Intell. Res.*, vol. 11, pp. 169–198, Aug. 1999.

[16] T. K. Ho, "Random decision forests," in *Proceedings of the International Conference on Document Analysis and Recognition, ICDAR*, 1995, vol. 1, pp. 278–282.

[17] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016.

[18] R. Xu and D. WunschII, "Survey of Clustering Algorithms," *IEEE Trans. Neural Networks*, vol. 16, no. 3, pp. 645–678, May 2005.

[19] G. T. Reddy *et al.*, "Analysis of Dimensionality Reduction Techniques on Big Data," *IEEE Access*, vol. 8, pp. 54776–54788, 2020.

[20] C. Watkins, "Learning From Delayed Rewards," King's College, 1989.

[21] G. Rummery and M. Niranjan, "On-Line Q-Learning Using Connectionist Systems," *Tech. Rep. CUED/F-INFENG/TR 166*, 1994.

[22] R. S. Sutton, "Learning to predict by the methods of temporal differences," *Mach. Learn.*, vol. 3, no. 1, pp. 9–44, Aug. 1988.

[23] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533–536, Oct. 1986.

[24] M. A. Cauchy, "Méthode générale pour la résolution des systèmes d'équations simultanées," *C. R. Hebd. Seances Acad. Sci.*, vol. 25, no. 1, pp. 536–538, 1847.

[25] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997.

[26] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling," in *NIPS 2014 Workshop on Deep Learning*, 2014.

[27] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Trans. Signal Process.*, vol. 45, no. 11, pp. 2673–2681, 1997.

[28] K. Cho *et al.*, "Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014, pp. 1724–1734.

[29] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to Sequence Learning with Neural Networks," in *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, 2014, pp. 3104–3112.

[30] K. Xu *et al.*, "Show, attend and tell: Neural image caption generation with visual attention," in *32nd International Conference on Machine Learning, ICML 2015*, 2015, pp. 2048–2057.

[31] A. Vaswani *et al.*, "Attention Is All You Need," *Adv. Neural Inf. Process. Syst.*, vol. 1, no. 1, pp. 5998–6008, Jun. 2017.

[32] Z. Dai, Z. Yang, Y. Yang, J. Carbonell, Q. Le, and R. Salakhutdinov, "Transformer-XL: Attentive Language Models beyond a Fixed-Length Context," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019, pp. 2978–2988.

[33] J. W. Rae, A. Potapenko, S. M. Jayakumar, and T. P. Lillicrap, "Compressive Transformers for Long-Range Sequence Modelling," in *ICLR 2020 Conference*, 2019.

[34] N. Kitaev, Ł. Kaiser, and A. Levskaya, "Reformer: The Efficient Transformer," Jan. 2020.

[35] I. Beltagy, M. E. Peters, and A. Cohan, "Longformer: The Long-Document Transformer," Apr. 2020.

[36] R. Bellman, "A Markovian Decision Process," *Indiana Univ. Math. J.*, vol. 6, no. 4, pp. 679–684, 1957.

[37] R. Bellman, "On the Theory of Dynamic Programming," *Proc. Natl. Acad. Sci.*, vol. 38, no. 8, pp. 716–719, Aug. 1952.

[38] C. J. C. H. Watkins and P. Dayan, "Q-learning," *Mach. Learn.*, vol. 8, no. 3–4, pp. 279–292, May 1992.

[39] T. P. Lillicrap *et al.*, "Continuous control with deep reinforcement learning," in *4th International Conference on Learning Representations, ICLR 2016 - Conference Track Proceedings*, 2016.

[40] V. R. Konda and J. N. Tsitsiklis, "On Actor-Critic Algorithms," *SIAM J. Control*

Optim., vol. 42, no. 4, pp. 1143–1166, Jan. 2003.

[41] J. Schulman, P. Moritz, S. Levine, M. I. Jordan, and P. Abbeel, "High-dimensional continuous control using generalized advantage estimation," in *4th International Conference on Learning Representations, ICLR 2016 - Conference Track Proceedings*, 2016.

[42] D. Silver, G. Lever, N. Heess, T. Degris, D. Wierstra, and M. Riedmiller, "Deterministic policy gradient algorithms," in *31st International Conference on Machine Learning, ICML 2014*, 2014, pp. 387–395.

[43] A. McCallum, "Efficient Exploration in Reinforcement Learning with Hidden State," *AAAI Fall Symposium on Model-directed Autonomous Systems*. 1997.

[44] M. Tokic, "Adaptive ε-Greedy Exploration in Reinforcement Learning Based on Value Differences," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2010, pp. 203–210.

[45] P. Auer, "Using upper confidence bounds for online learning," in *Proceedings 41st Annual Symposium on Foundations of Computer Science*, 2000, pp. 270–279.

[46] O. Chapelle and L. Li, "An empirical evaluation of Thompson sampling," in *Advances in Neural Information Processing Systems*, 2011, pp. 1–24.

[47] V. Mnih *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, Feb. 2015.

[48] L. J. Lin, "Self-Improving Reactive Agents Based on Reinforcement Learning, Planning and Teaching," *Mach. Learn.*, vol. 8, no. 3, pp. 293–321, 1992.

[49] H. Van Hasselt, "Double Q-learning," in *Advances in Neural Information Processing Systems 23: 24th Annual Conference on Neural Information Processing Systems 2010, NIPS 2010*, 2010, pp. 2613–2621.

[50] H. Van Hasselt, A. Guez, and D. Silver, "Deep reinforcement learning with double Q-Learning," in *30th AAAI Conference on Artificial Intelligence, AAAI 2016*, 2016.

[51] Z. Wang, T. Schaul, M. Hessel, H. Van Hasselt, M. Lanctot, and N. De Frcitas, "Dueling Network Architectures for Deep Reinforcement Learning," in *33rd International Conference on Machine Learning, ICML 2016*, 2016, pp. 1–15.

[52] M. Hessel *et al.*, "Rainbow: Combining Improvements in Deep Reinforcement Learning," in *32nd AAAI Conference on Artificial Intelligence, AAAI 2018*, 2018, pp. 3215–3222.

[53] T. Schaul, J. Quan, I. Antonoglou, and D. Silver, "Prioritized Experience Replay," in *4th International Conference on Learning Representations, ICLR 2016 - Conference Track Proceedings*, 2015, pp. 1–21.

[54] J. Peng and R. J. Williams, "Incremental multi-step Q-learning," *Mach. Learn.*, vol. 22, no. 1–3, pp. 283–290, 1996.

[55] M. Fortunato *et al.*, "Noisy Networks for Exploration," in *6th International Conference on Learning Representations, ICLR 2018 - Conference Track Proceedings*, 2017.

[56] J. Schulman, S. Levine, P. Moritz, M. Jordan, and P. Abbeel, "Trust region policy optimization," in *32nd International Conference on Machine Learning, ICML 2015*, 2015.

[57] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal Policy Optimization Algorithms," Jul. 2017.

[58] Y. Wang, D. Zhang, Y. Liu, B. Dai, and L. H. Lee, "Enhancing transportation systems via deep learning: A survey," *Transp. Res. Part C Emerg. Technol.*, vol. 99, pp. 144–163, Feb. 2019.

[59] A. Haydari and Y. Yilmaz, "Deep Reinforcement Learning for Intelligent Transportation Systems: A Survey," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 1, pp. 11–32, Jan. 2022.

[60] A. K. Haghighat, V. Ravichandra-Mouli, P. Chakraborty, Y. Esfandiari, S. Arabi, and A. Sharma, "Applications of Deep Learning in Intelligent Transportation Systems," *J. Big Data Anal. Transp.*, vol. 2, no. 2, pp. 115–145, Aug. 2020.

[61] T. Yuan, W. Rocha, C. E. Rothenberg, K. Obraczka, C. Barakat, and T. Turletti, "Machine learning for next-generation intelligent transportation systems: A survey," *Trans. Emerg. Telecommun. Technol.*, vol. 1, no. 1, pp. 1–35, Dec. 2021.

[62] Z. Zhu and H. Zhao, "A Survey of Deep RL and IL for Autonomous Driving Policy Learning," *IEEE Trans. Intell. Transp. Syst.*, vol. abs/2101.0, pp. 1–23, Jan. 2021.

[63] H. Nguyen, L. Kieu, T. Wen, and C. Cai, "Deep learning methods in transportation domain: a review," *IET Intell. Transp. Syst.*, vol. 12, no. 9, pp. 998–1004, Nov. 2018.

[64] C.-J. Hoel, K. Driggs-Campbell, K. Wolff, L. Laine, and M. J. Kochenderfer, "Combining Planning and Deep Reinforcement Learning in Tactical Decision Making for Autonomous Driving," *IEEE Trans. Intell. Veh.*, vol. 5, no. 2, pp. 294–305, Jun. 2020.

[65] Y. Ye, X. Zhang, and J. Sun, "Automated vehicle's behavior decision making using deep reinforcement learning and high-fidelity simulation environment," *Transp. Res. Part C Emerg. Technol.*, vol. 107, pp. 155–170, Oct. 2019.

[66] K. Min, H. Kim, and K. Huh, "Deep Distributional Reinforcement Learning Based High-Level Driving Policy Determination," *IEEE Trans. Intell. Veh.*, vol. 4, no. 3, pp. 416–424, Sep. 2019.

[67] B. Chalaki *et al.*, "Zero-Shot Autonomous Vehicle Policy Transfer: From Simulation to Real-World via Adversarial Learning," in *2020 IEEE 16th International Conference on Control & Automation (ICCA)*, 2020, pp. 35–40.

[68] Y. Wu, H. Tan, L. Qin, and B. Ran, "Differential variable speed limits control for freeway recurrent bottlenecks via deep actor-critic algorithm," *Transp. Res. Part C Emerg. Technol.*, vol. 117, no. 1, pp. 100–115, Aug. 2020.

[69] C. Yu *et al.*, "Distributed Multiagent Coordinated Learning for Autonomous Driving in Highways Based on Dynamic Coordination Graphs," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 2, pp. 735–748, Feb. 2020.

[70] J. Zeng, J. Hu, and Y. Zhang, "Adaptive Traffic Signal Control with Deep Recurrent Q-learning," in *2018 IEEE Intelligent Vehicles Symposium (IV)*, 2018, pp. 1215–1220.

[71] M. Gregurić, M. Vujić, C. Alexopoulos, and M. Miletić, "Application of Deep Reinforcement Learning in Traffic Signal Control: An Overview and Impact of Open Traffic Data," *Appl. Sci.*, vol. 10, no. 11, p. 4011, Jun. 2020.

[72] X. Liang, X. Du, G. Wang, and Z. Han, "A Deep Reinforcement Learning Network for Traffic Light Cycle Control," *IEEE Trans. Veh. Technol.*, vol. 68, no. 2, pp. 1243–1253, Feb. 2019.

[73] S. Shi and F. Chen, "Deep Recurrent Q-learning Method for Area Traffic Coordination Control," *J. Adv. Math. Comput. Sci.*, vol. 27, no. 3, pp. 1–11, May 2018.

[74] E. Van Der Pol, "Deep Reinforcement Learning for Coordination in Traffic Light Control," *Master thesis*, no. November 2015, 2016.

[75] S. El-Tantawy and B. Abdulhai, "Multi-Agent Reinforcement Learning for Integrated Network of Adaptive Traffic Signal Controllers (MARLIN-ATSC)," in *2012 15th International IEEE Conference on Intelligent Transportation Systems*, 2012, pp. 319–326.

[76] S. El-Tantawy, B. Abdulhai, and H. Abdelgawad, "Multiagent Reinforcement Learning for Integrated Network of Adaptive Traffic Signal Controllers (MARLIN-ATSC): Methodology and Large-Scale Application on Downtown Toronto," *IEEE Trans. Intell. Transp. Syst.*, vol. 14, no. 3, pp. 1140–1150, Sep. 2013.

[77] M. Khakzar, A. Rakotonirainy, A. Bond, and S. G. Dehkordi, "A Dual Learning Model for Vehicle Trajectory Prediction," *IEEE Access*, vol. 8, pp. 21897–21908, 2020.

[78] J. Sun and J. Kim, "Joint prediction of next location and travel time from urban vehicle trajectories using long short-term memory neural networks," *Transp. Res. Part C Emerg. Technol.*, vol. 128, p. 103114, Jul. 2021.

[79] H. Cui *et al.*, "Deep Kinematic Models for Kinematically Feasible Vehicle Trajectory Predictions," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 10563–10569.

[80] G. Kim, D. Kim, Y. Ahn, and K. Huh, "Hybrid Approach for Vehicle Trajectory Prediction Using Weighted Integration of Multiple Models," *IEEE Access*, vol. 9, pp. 78715–78723, 2021.

[81] S. Lee, D. Ngoduy, and M. Keyvan-Ekbatani, "Integrated deep learning and stochastic car-following model for traffic dynamics on multi-lane freeways," *Transp. Res. Part C Emerg. Technol.*, vol. 106, pp. 360–377, Sep. 2019.

[82] W. Huang, G. Song, H. Hong, and K. Xie, "Deep Architecture for Traffic Flow Prediction: Deep Belief Networks With Multitask Learning," *IEEE Trans. Intell. Transp. Syst.*, vol. 15, no. 5, pp. 2191–2201, Oct. 2014.

[83] Y. Lv, Y. Duan, W. Kang, Z. Li, and F.-Y. Wang, "Traffic Flow Prediction With Big Data: A Deep Learning Approach," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 2, pp. 865–873, 2014.

[84] A. Miglani and N. Kumar, "Deep learning models for traffic flow prediction in autonomous vehicles: A review, solutions, and challenges," *Veh. Commun.*, vol. 20, pp. 100–134, Dec. 2019.

[85] C. Chen, Z. Liu, S. Wan, J. Luan, and Q. Pei, "Traffic Flow Prediction Based on Deep Learning in Internet of Vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 6, pp. 3776–3789, Jun. 2021.

[86] Q. Cheng, L. Yang, W. Wei, and L. Zhiyuan, "Analysis and Forecasting of the Day-to-day Travel Demand Variations for Large- scale Transportation Networks : A Deep Learning Approach," *TRB 2017 Transp. Anal. Contest*, vol. 1, no. 1, pp. 1–18, 2017.

[87] J. Xu, R. Rahmatizadeh, L. Boloni, and D. Turgut, "Real-Time Prediction of Taxi Demand Using Recurrent Neural Networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 8, pp. 2572–2581, Aug. 2018.

[88] T. Liu, W. Wu, Y. Zhu, and W. Tong, "Predicting taxi demands via an attention-based convolutional recurrent neural network," *Knowledge-Based Syst.*, vol. 206, p. 106294, Oct. 2020.

[89] J. Ke, H. Zheng, H. Yang, and X. (Michael) Chen, "Short-term forecasting of passenger demand under on-demand ride services: A spatio-temporal deep learning approach," *Transp. Res. Part C Emerg. Technol.*, vol. 85, pp. 591–608, Dec. 2017.

[90] X. Zhu, J. Li, Z. Liu, and F. Yang, "Location deployment of depots and resource relocation for connected car-sharing systems through mobile edge computing," *Int. J. Distrib. Sens. Networks*, vol. 13, no. 6, p. 15, Jun. 2017.

[91] L. Zhu and N. Laptev, "Deep and Confident Prediction for Time Series at Uber," in *2017 IEEE International Conference on Data Mining Workshops (ICDMW)*, 2017, pp. 103–110.

[92] W. Qu *et al.*, "Short-Term Intersection Traffic Flow Forecasting," *Sustainability*, vol. 12, no. 19, p. 8158, Oct. 2020.

[93] J. Mena-Oreja and J. Gozalvez, "A Comprehensive Evaluation of Deep Learning-Based Techniques for Traffic Prediction," *IEEE Access*, vol. 8, pp. 91188–91212, 2020.

[94] X. Ma, H. Yu, Y. Wang, and Y. Wang, "Large-Scale Transportation Network Congestion Evolution Prediction Using Deep Learning Theory," *PLoS One*, vol. 10, no. 3, p. 10, Mar. 2015.

[95] M. Fouladgar, M. Parchami, R. Elmasri, and A. Ghaderi, "Scalable deep traffic flow neural networks for urban traffic congestion prediction," in *2017 International Joint Conference on Neural Networks (IJCNN)*, 2017, pp. 2251–2258.

[96] Z. Yuan, X. Zhou, T. Yang, J. Tamerius, and R. Mantilla, "Predicting Traffic Accidents Through Heterogeneous Urban Data : A Case Study," in *23rd ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, 2017, pp. 1–15.

[97] G. Pan, L. Fu, and L. Thakali, "Development of a global road safety performance function using deep neural networks," *Int. J. Transp. Sci. Technol.*, vol. 6, no. 3, pp. 159–173, Sep. 2017.

[98] C. Dong, C. Shao, J. Li, and Z. Xiong, "An Improved Deep Learning Model for Traffic Crash Prediction," *J. Adv. Transp.*, vol. 2018, pp. 1–13, Dec. 2018.

[99] Z. Yuan, X. Zhou, and T. Yang, "Hetero-ConvLSTM," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2018, pp. 984–992.

[100] C. Chen, H. Xiang, T. Qiu, C. Wang, Y. Zhou, and V. Chang, "A rear-end collision prediction scheme based on deep learning in the Internet of Vehicles," *J. Parallel Distrib. Comput.*, vol. 117, pp. 192–204, Jul. 2018.

[101] X. Wang, T. Qiu, C. Chen, and N. Chen, "A Neural-Network-Based Real-end Collision Prediction Mechanism for Smart Cities," in *2019 IEEE International Conference on Smart Internet of Things (SmartIoT)*, 2019, pp. 496–497.

[102] Z. Zhang, Q. He, J. Gao, and M. Ni, "A deep learning approach for detecting traffic accidents from social media data," *Transp. Res. Part C Emerg. Technol.*, vol. 86, pp. 580–596, Jan. 2018.

[103] T. Suzuki, H. Kataoka, Y. Aoki, and Y. Satoh, "Anticipating Traffic Accidents with Adaptive Loss and Large-Scale Incident DB," in *2018 IEEE/CVF Conference on*

Computer Vision and Pattern Recognition, 2018, pp. 3521–3529.

[104] M. I. Sameen, B. Pradhan, H. Z. M. Shafri, and H. Bin Hamid, "Applications of Deep Learning in Severity Prediction of Traffic Accidents," in *Lecture Notes in Civil Engineering*, 2019, pp. 793–808.

[105] M. Sameen and B. Pradhan, "Severity Prediction of Traffic Accidents with Recurrent Neural Networks," *Appl. Sci.*, vol. 7, no. 6, p. 476, Jun. 2017.

[106] S. Bang, S. Park, H. Kim, and H. Kim, "Encoder–decoder network for pixel-level road crack detection in black-box images," *Comput. Civ. Infrastruct. Eng.*, vol. 34, no. 8, pp. 713–727, Aug. 2019.

[107] S. Choi and M. Do, "Development of the Road Pavement Deterioration Model Based on the Deep Learning Method," *Electronics*, vol. 9, no. 1, p. 3, Dec. 2019.

[108] R. Roberts, L. Inzerillo, and G. Di Mino, "Exploiting Data Analytics and Deep Learning Systems to Support Pavement Maintenance Decisions," *Appl. Sci.*, vol. 11, no. 6, p. 2458, Mar. 2021.

[109] L. Gao, Y. Yu, Y. Hao Ren, and P. Lu, "Detection of Pavement Maintenance Treatments using Deep-Learning Network," *Transp. Res. Rec. J. Transp. Res. Board*, vol. 2675, no. 9, pp. 1434–1443, Sep. 2021.

[110] L. Yao, Q. Dong, J. Jiang, and F. Ni, "Deep reinforcement learning for long-term pavement maintenance planning," *Comput. Civ. Infrastruct. Eng.*, vol. 35, no. 11, pp. 1230–1245, Nov. 2020.

[111] L. Zhu, F. R. Yu, Y. Wang, B. Ning, and T. Tang, "Big Data Analytics in Intelligent Transportation Systems: A Survey," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 1, pp. 383–398, Jan. 2019.

[112] D. Hahn, A. Munir, and V. Behzadan, "Security and Privacy Issues in Intelligent Transportation Systems: Classification and Challenges," *IEEE Intell. Transp. Syst. Mag.*, vol. 13, no. 1, pp. 181–196, 2021.

[113] J. A. Guerrero-ibanez, S. Zeadally, and J. Contreras-Castillo, "Integration challenges of intelligent transportation systems with connected vehicle, cloud computing, and internet of things technologies," *IEEE Wirel. Commun.*, vol. 22, no. 6, pp. 122–128, Dec. 2015.

[114] A. Lamssaggad, N. Benamar, A. S. Hafid, and M. Msahli, "A Survey on the Current Security Landscape of Intelligent Transportation Systems," *IEEE Access*, vol. 9, pp. 9180–9208, 2021.

[115] E. Mathew, "Intelligent Transport Systems and Its Challenges," in *Advances in Intelligent Systems and Computing*, 2020, pp. 663–672.

[116] Q. Ali, N. Ahmad, A. Malik, G. Ali, and W. Rehman, "Issues, Challenges, and Research Opportunities in Intelligent Transport System for Security and Privacy," *Appl. Sci.*, vol. 8, no. 10, p. 1964, Oct. 2018.

[117] E. Mathew, "Swarm intelligence for intelligent transport systems: opportunities and challenges," in *Swarm Intelligence for Resource Management in Internet of Things*, Elsevier, 2020, pp. 131–145.

[118] Y. Rizk, M. Awad, and E. W. Tunstel, "Decision Making in Multiagent Systems: A Survey," *IEEE Trans. Cogn. Dev. Syst.*, vol. 10, no. 3, pp. 514–529, Sep. 2018.

[119] L. S. Iyer, "AI enabled applications towards intelligent transportation," *Transp. Eng.*,

vol. 5, p. 100083, Sep. 2021.

[120] L. Guevara and F. Auat Cheein, "The Role of 5G Technologies: Challenges in Smart Cities and Intelligent Transportation Systems," *Sustainability*, vol. 12, no. 16, p. 6469, Aug. 2020.

[121] F. Camacho, C. Cárdenas, and D. Muñoz, "Emerging technologies and research challenges for intelligent transportation systems: 5G, HetNets, and SDN," *Int. J. Interact. Des. Manuf.*, vol. 12, no. 1, pp. 327–335, Feb. 2018.

[122] S. Hayat, E. Yanmaz, and R. Muzaffar, "Survey on Unmanned Aerial Vehicle Networks for Civil Applications: A Communications Viewpoint," *IEEE Commun. Surv. Tutorials*, vol. 18, no. 4, pp. 2624–2661, 2016.

[123] G. Cai, J. Dias, and L. Seneviratne, "A Survey of Small-Scale Unmanned Aerial Vehicles: Recent Advances and Future Development Trends," *Unmanned Syst.*, vol. 02, no. 02, pp. 175–199, Apr. 2014.

[124] A. Guillen-Perez and M.-D. Cano, "Flying Ad Hoc Networks: A New Domain for Network Communications," *Sensors*, vol. 18, no. 10, p. 3571, Oct. 2018.

[125] S. Hayat, E. Yanmaz, and C. Bettstetter, "Experimental analysis of multipoint-to-point UAV communications with IEEE 802.11n and 802.11ac," in *2015 IEEE 26th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, 2015, vol. 2015-Decem, pp. 1991–1996.

[126] E. Yanmaz, S. Yahyanejad, B. Rinner, H. Hellwagner, and C. Bettstetter, "Drone networks: Communications, coordination, and sensing," *Ad Hoc Networks*, vol. 68, pp. 1–15, 2018.

[127] E. Yanmaz, C. Costanzo, C. Bettstetter, and W. Elmenreich, "A discrete stochastic process for coverage analysis of autonomous UAV networks," *2010 IEEE Globecom Work. GC'10*, pp. 1777–1782, Dec. 2010.

[128] İ. Bekmezci, O. K. Sahingoz, and Ş. Temel, "Flying Ad-Hoc Networks (FANETs): A survey," *Ad Hoc Networks*, vol. 11, no. 3, pp. 1254–1270, May 2013.

[129] L. Gupta, R. Jain, and G. Vaszkun, "Survey of Important Issues in UAV Communication Networks," *IEEE Commun. Surv. Tutorials*, vol. 18, no. 2, pp. 1123–1152, 2016.

[130] D. B. Johnson and D. A. Maltz, "Dynamic Source Routing in Ad Hoc Wireless Networks," in *Mobile Computing*, Boston, MA: Springer US, 2007, pp. 153–181.

[131] O. Bouachir, A. Abrassart, F. Garcia, and N. Larrieu, "A mobility model for UAV ad hoc network," in *2014 International Conference on Unmanned Aircraft Systems, ICUAS 2014 - Conference Proceedings*, 2014, pp. 383–388.

[132] Y. Wan, K. Namuduri, Y. Zhou, and S. Fu, "A smooth-turn mobility model for airborne networks," *IEEE Trans. Veh. Technol.*, vol. 62, no. 7, pp. 3359–3370, 2013.

[133] J. Lessmann, P. Janacik, L. Lachev, and D. Orfanus, "Comparative study of wireless network simulators," *Proc. - 7th Int. Conf. Networking, ICN 2008*, pp. 517–523, 2008.

[134] M. Sánchez and P. Manzoni, "ANEJOS: A Java based simulator for ad hoc networks," *Futur. Gener. Comput. Syst.*, vol. 17, no. 5, pp. 573–583, 2001.

[135] E. M. Royer, P. M. Melliar-Smith, and L. E. Moser, "An analysis of the optimum node density for ad hoc mobile networks," *IEEE Int. Conf. Commun.*, vol. 3, pp. 857–861, 2001.

[136] F. Bai, N. Sadagopan, and A. Helmy, "The IMPORTANT framework for analyzing the impact of mobility on performance of RouTing protocols for Adhoc NeTworks," *Ad Hoc Networks*, vol. 1, no. 4, pp. 383–403, 2003.

[137] Z. J. Haas and M. R. Pearlman, "The performance of a new routing protocol for the reconfigurable wireless networks," in *International Conference on Communications - Proceedings*, 1998, vol. 1, pp. 156–160.

[138] V. Tolety and T. Camp, "Load reduction in ad hoc networks using mobile servers," 1999.

[139] J. D. M. M. Biomo, T. Kunz, and M. St-Hilaire, "An enhanced Gauss-Markov mobility model for simulations of unmanned aerial ad hoc networks," *2014 7th IFIP Wirel. Mob. Netw. Conf. WMNC 2014*, pp. 1–8, 2014.

[140] D. Broyles, A. Jabbar, and J. P. G. Sterbenz, "Design and analysis of a 3-D gauss-markov mobility model for highly dynamic airborne networks," *Proc. Int. Telemetering Conf.*, vol. 46, no. Itc 2010, pp. 388–404, 2010.

[141] W. Wang, X. Guan, B. Wang, and Y. Wang, "A novel mobility model based on semi-random circular movement in mobile ad hoc networks," *Inf. Sci. (Ny).*, vol. 180, no. 3, pp. 399–413, 2010.

[142] Paparazzi Forum, "Open Source Paparazzi - UAV Project," 2008. [Online]. Available: http://paparazzi.enac.fr/. [Accessed: 24-Jan-2018].

[143] M. Bergamo, R. Hain, K. Kasera, D. Li, R. Ramanathan, and M. Steenstrup, "System design specification for mobile multimedia wireless network (MMWN) (draft)," *DARPA Proj. DAAB07-95-C-D156*, 1996.

[144] X. Li, T. Zhang, and J. Li, "A Particle Swarm Mobility Model for Flying Ad Hoc Networks," *2017 IEEE Glob. Commun. Conf. GLOBECOM 2017 - Proc.*, vol. 2018-Janua, pp. 1–6, 2017.

[145] X. Hong, M. Gerla, G. Pei, and C. C. Chiang, "A group mobility model for ad hoc wireless networks," *Proc. 2nd ACM Int. Work. Model. Anal. Simul. Wirel. Mob. Syst. MSWiM 1999*, pp. 53–60, 1999.

[146] J. Moore and R. Chapman, "Application of particle swarm to multiobjective optimization," 1999.

[147] E. Kieffer, G. Danoy, P. Bouvry, and A. Nagih, "Hybrid mobility model with pheromones for UAV detection task," in *2016 IEEE Symposium Series on Computational Intelligence (SSCI)*, 2016, pp. 1–8.

[148] E. Kuiper and S. Nadjm-Tehrani, "Mobility Models for UAV Group Reconnaissance Applications," in *2006 International Conference on Wireless and Mobile Communications (ICWMC'06)*, 2006, pp. 33–33.

[149] N. Weicker, G. Szabo, K. Weicker, and P. Widmayer, "Evolutionary multiobjective optimization for base station transmitter placement with frequency assignment," *IEEE Trans. Evol. Comput.*, vol. 7, no. 2, pp. 189–203, Apr. 2003.

[150] R. Mathar and T. Niessen, "Optimum positioning of base stations for cellular radio networks," *Wirel. Networks*, vol. 6, no. 6, pp. 421–428, 2000.

[151] K. Tutschku, "Demand-based radio network planning of cellular mobile communication systems," in *Proceedings. IEEE INFOCOM '98, the Conference on Computer Communications. Seventeenth Annual Joint Conference of the IEEE Computer and Communications Societies. Gateway to the 21st Century (Cat.*

No.98CH36169), 1998, vol. 3, pp. 1054–1061.

[152] M. Mozaffari, W. Saad, M. Bennis, and M. Debbah, "Efficient Deployment of Multiple Unmanned Aerial Vehicles for Optimal Wireless Coverage," *IEEE Commun. Lett.*, vol. 20, no. 8, pp. 1647–1650, Aug. 2016.

[153] A. Guillen-Perez, R. Sanchez-Iborra, M.-D. Cano, J. C. Sanchez-Aarnoutse, and J. Garcia-Haro, "WiFi networks on drones," in *2016 ITU Kaleidoscope: ICTs for a Sustainable World (ITU WT)*, 2016, pp. 1–8.

[154] Zhu Han, A. L. Swindlehurst, and K. Liu, "Optimization of MANET connectivity via smart deployment/movement of unmanned air vehicles," *IEEE Trans. Veh. Technol.*, vol. 58, no. 7, pp. 3533–3546, Sep. 2009.

[155] A. Al-Hourani, S. Kandeepan, and S. Lardner, "Optimal LAP Altitude for Maximum Coverage," *IEEE Wirel. Commun. Lett.*, vol. 3, no. 6, pp. 569–572, Dec. 2014.

[156] M. Mozaffari, W. Saad, M. Bennis, and M. Debbah, "Drone Small Cells in the Clouds: Design, Deployment and Performance Analysis," in *2015 IEEE Global Communications Conference (GLOBECOM)*, 2014, pp. 1–6.

[157] M. M. Azari, F. Rosas, K.-C. Chen, and S. Pollin, "Optimal UAV Positioning for Terrestrial-Aerial Communication in Presence of Fading," in *2016 IEEE Global Communications Conference (GLOBECOM)*, 2016, pp. 1–7.

[158] Z. M. Fadlullah, D. Takaishi, H. Nishiyama, N. Kato, and R. Miura, "A dynamic trajectory control algorithm for improving the communication throughput and delay in UAV-aided networks," *IEEE Netw.*, vol. 30, no. 1, pp. 100–105, Jan. 2016.

[159] M. Gruber, "Role of altitude when exploring optimal placement of UAV access points," in *2016 IEEE Wireless Communications and Networking Conference*, 2016, vol. 2016-Septe, no. Wcnc, pp. 1–5.

[160] J. Adler and W.-W. Tso, "'Decision'-Making in Bacteria: Chemotactic Response of Escherichia coli to Conflicting Stimuli," *Science (80-.).*, vol. 184, no. 4143, pp. 1292–1294, Jun. 1974.

[161] R. V. Kulkarni and G. K. Venayagamoorthy, "Bio-inspired Algorithms for Autonomous Deployment and Localization of Sensor Nodes," *IEEE Trans. Syst. Man, Cybern. Part C (Applications Rev.*, vol. 40, no. 6, pp. 663–675, Nov. 2010.

[162] Z. Gaspar and T. Tarnai, "Upper bound of density for packing of equal circles in special domains in the plane," *Period. Polytech. Civ. Eng.*, vol. 44, no. 1, pp. 13–32, 2000.

[163] J. Lyu, Y. Zeng, R. Zhang, and T. J. Lim, "Placement Optimization of UAV-Mounted Mobile Base Stations," *IEEE Commun. Lett.*, vol. 21, no. 3, pp. 604–607, Mar. 2017.

[164] J. Sanchez-Garcia, J. M. Garcia-Campos, S. L. Toral, D. G. Reina, and F. Barrero, "A Self Organising Aerial Ad Hoc Network Mobility Model for Disaster Scenarios," in *2015 International Conference on Developments of E-Systems Engineering (DeSE)*, 2015, pp. 35–40.

[165] J. Chen and D. Gesbert, "Optimal positioning of flying relays for wireless networks: A LOS map approach," in *2017 IEEE International Conference on Communications (ICC)*, 2017, pp. 1–6.

[166] V. V. Chetlur Ravi and H. S. Dhillon, "Downlink coverage probability in a finite network of unmanned aerial vehicle (UAV) base stations," in *2016 IEEE 17th International Workshop on Signal Processing Advances in Wireless Communications*

(SPAWC), 2016, vol. 2016-Augus, pp. 1–5.

[167] V. Sharma, M. Bennis, and R. Kumar, "UAV-Assisted Heterogeneous Networks for Capacity Enhancement," *IEEE Commun. Lett.*, vol. 20, no. 6, pp. 1207–1210, Jun. 2016.

[168] H. T. Friis, "A Note on a Simple Transmission Formula," *Proc. IRE*, vol. 34, no. 5, pp. 254–256, May 1946.

[169] J. Holis and P. Pechac, "Elevation Dependent Shadowing Model for Mobile Communications via High Altitude Platforms in Built-Up Areas," *IEEE Trans. Antennas Propag.*, vol. 56, no. 4, pp. 1078–1084, Apr. 2008.

[170] X. Cai *et al.*, "Low altitude UAV propagation channel modelling," in *2017 11th European Conference on Antennas and Propagation (EUCAP)*, 2017, pp. 1443–1447.

[171] R. Amorim, H. Nguyen, P. Mogensen, I. Z. Kovacs, J. Wigard, and T. B. Sorensen, "Radio Channel Modeling for UAV Communication Over Cellular Networks," *IEEE Wirel. Commun. Lett.*, vol. 6, no. 4, pp. 514–517, Aug. 2017.

[172] D. W. Matolak and R. Sun, "Air-ground channels for UAS: Summary of measurements and models for L- and C-bands," *ICNS 2016 Secur. an Integr. CNS Syst. to Meet Futur. Challenges*, pp. 1–11, 2016.

[173] W. Khawaja, I. Guvenc, and D. Matolak, "UWB Channel Sounding and Modeling for UAV Air-to-Ground Propagation Channels," in *2016 IEEE Global Communications Conference (GLOBECOM)*, 2016, pp. 1–7.

[174] M. Simunek, F. P. Fontan, and P. Pechac, "The UAV Low Elevation Propagation Channel in Urban Areas: Statistical Analysis and Time-Series Generator," *IEEE Trans. Antennas Propag.*, vol. 61, no. 7, pp. 3850–3858, Jul. 2013.

[175] A. Pokkunuru, Q. Zhang, and P. Wang, "Capacity analysis of aerial small cells," in *2017 IEEE International Conference on Communications (ICC)*, 2017, pp. 1–7.

[176] L. R. Pinto, A. Moreira, L. Almeida, and A. Rowe, "Characterizing Multihop Aerial Networks of COTS Multirotors," *IEEE Trans. Ind. Informatics*, vol. 13, no. 2, pp. 898–906, Apr. 2017.

[177] N. Goddemeier and C. Wietfeld, "Investigation of Air-to-Air Channel Characteristics and a UAV Specific Extension to the Rice Model," in *2015 IEEE Globecom Workshops (GC Wkshps)*, 2015, pp. 1–5.

[178] Chun Loo, "A statistical model for a land mobile satellite link," *IEEE Trans. Veh. Technol.*, vol. 34, no. 3, pp. 122–127, Aug. 1985.

[179] K. Daniel, M. Putzke, B. Dusza, and C. Wietfeld, "Three dimensional channel characterization for low altitude aerial vehicles," in *2010 7th International Symposium on Wireless Communication Systems*, 2010, pp. 756–760.

[180] C.-M. Cheng, P.-H. Hsiao, H. T. Kung, and D. Vlah, "Maximizing Throughput of UAV-Relaying Networks with the Load-Carry-and-Deliver Paradigm," in *2007 IEEE Wireless Communications and Networking Conference*, 2007, pp. 4417–4424.

[181] O. K. Sahingoz, "Networking Models in Flying Ad-Hoc Networks (FANETs): Concepts and Challenges," *J. Intell. Robot. Syst.*, vol. 74, no. 1–2, pp. 513–527, Apr. 2014.

[182] J. Ko, A. Mahajan, and R. Sengupta, "A network-centric UAV organization for search and pursuit operations," in *Proceedings, IEEE Aerospace Conference*, 2002, vol. 6, pp. 2697–2713.

[183] P. Kolios, V. Friderikos, and K. Papadaki, "Store carry and forward relay aided cellular networks," *Proc. Annu. Int. Conf. Mob. Comput. Networking, MOBICOM*, pp. 71–72, 2010.

[184] P. Jacquet, P. Mühlethaler, T. Clausen, A. Laouiti, A. Qayyum, and L. Viennot, "Optimized link state routing protocol for ad hoc networks," *Proc. - IEEE Innternational Multi Top. Conf. 2001 Technol. 21st Century, IEEE INMIC 2001*, pp. 62–68, 2001.

[185] C. E. Perkins and P. Bhagwat, "Highly dynamic Destination-Sequenced Distance-Vector routing (DSDV) for mobile computers," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 24, no. 4, pp. 234–244, Oct. 1994.

[186] D. L. Johnson, N. Ntlatlapa, and C. Aichele, "A simple pragmatic approach to mesh routing using BATMAN," in *2nd IFIP International Symposium on Wireless Communications and Information Technology in Developing Countries*, 2008, p. 10.

[187] A. I. Alshbatat and L. Dong, "Cross layer design for mobile Ad-Hoc Unmanned Aerial Vehicle communication networks," in *2010 International Conference on Networking, Sensing and Control (ICNSC)*, 2010, pp. 331–336.

[188] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numer. Math.*, vol. 1, no. 1, pp. 269–271, Dec. 1959.

[189] K. Singh and Anil Kumar Verma, "Experimental analysis of AODV, DSDV and OLSR routing protocol for flying adhoc networks (FANETs)," in *2015 IEEE International Conference on Electrical, Computer and Communication Technologies (ICECCT)*, 2015, pp. 1–4.

[190] M. Yadav, S. K. Gupta, and R. K. Saket, "Multi-hop wireless ad-hoc network routing protocols- a comparative study of DSDV, TORA, DSR and AODV," in *2015 International Conference on Electrical, Electronics, Signals, Communication and Optimization (EESCO)*, 2015, no. January, pp. 1–5.

[191] J. Broch, D. A. Maltz, D. B. Johnson, Y.-C. Hu, and J. Jetcheva, "A performance comparison of multi-hop wireless ad hoc network routing protocols," in *Proceedings of the 4th annual ACM/IEEE international conference on Mobile computing and networking - MobiCom '98*, 1998, pp. 85–97.

[192] R. Sanchez-Iborra, M.-D. Cano, and J. Garcia-Haro, "Performance Evaluation of BATMAN Routing Protocol for VoIP Services: A QoE Perspective," *IEEE Trans. Wirel. Commun.*, vol. 13, no. 9, pp. 4947–4958, Sep. 2014.

[193] E. Kulla, M. Hiyama, M. Ikeda, and L. Barolli, "Comparison of Experimental Results of a MANET Testbed in Different Environments Considering BATMAN Protocol," in *2011 Third International Conference on Intelligent Networking and Collaborative Systems*, 2011, pp. 1–7.

[194] A. Sharma and N. Rajagopalan, "A Comparative Study of B.A.T.M.A.N. and OLSR Routing Protocols for MANETs," in *International Journal of Advanced Trends in Computer Science and Engineering (IJATCSE)*, 2013, p. 5.

[195] J. Pojda, A. Wolff, M. Sbeiti, and C. Wietfeld, "Performance analysis of mesh routing protocols for UAV swarming applications," in *2011 8th International Symposium on Wireless Communication Systems*, 2011, pp. 317–321.

[196] S. Rosati, K. Kruzelecki, L. Traynard, and B. Rimoldi, "Speed-aware routing for UAV ad-hoc networks," in *2013 IEEE Globecom Workshops (GC Wkshps)*, 2013, pp. 1367–1373.

[197] Yu Jiang, Li Dong, Yuwen Wang, Zhenzhen Li, Hong Zhang, and Yi Zheng, "A mobility and load aware OLSR routing protocol for UAV mobile ad-hoc networks," in *2014 International Conference on Information and Communications Technologies (ICT 2014)*, 2014, vol. 2014, no. 650 CP, pp. 1.019-1.019.

[198] Y. L. - and X. L. -, "Cross Layer Optimization for Cooperative Mobile Ad-Hoc UAV Network," *Int. J. Digit. Content Technol. its Appl.*, vol. 6, no. 18, pp. 367–375, Oct. 2012.

[199] A. B. Paul and S. Nandi, "Modified Optimized Link State Routing (M-OLSR) for Wireless Mesh Networks," in *2008 International Conference on Information Technology*, 2008, pp. 147–152.

[200] M. Belhassen, A. Belghith, and M. A. Abid, "Performance evaluation of a cartography enhanced OLSR for mobile multi-hop ad hoc networks," in *2011 Wireless Advanced*, 2011, pp. 149–155.

[201] B. Bellur, R. G. Ogier, and F. L. Templin, "Topology Broadcast Based on Reverse-Path Forwarding (TBRPF)," *ETF Internet Draft*, no. September. 2001.

[202] Guangyu Pei, M. Gerla, and Tsu-Wei Chen, "Fisheye state routing: a routing scheme for ad hoc wireless networks," in *2000 IEEE International Conference on Communications. ICC 2000. Global Convergence Through Communications. Conference Record*, 2000, vol. 1, pp. 70–74.

[203] J. Chroboczek, "The Babel Routing Protocol," Apr. 2011.

[204] C. E. Perkins and E. M. Royer, "Ad-hoc on-demand distance vector routing," in *Proceedings WMCSA'99. Second IEEE Workshop on Mobile Computing Systems and Applications*, 1999, pp. 90–100.

[205] J. H. Forsmann, R. E. Hiromoto, and J. Svoboda, "A time-slotted on-demand routing protocol for mobile ad hoc unmanned vehicle systems," in *Unmanned Systems Technology IX*, 2007, vol. 6561, pp. 98–115.

[206] B. N. Jagdale, "Analysis and Comparison of Distance Vector, DSDV and AODV Protocol of MANET," *Int. J. Distrib. Parallel Syst.*, vol. 3, no. 2, pp. 121–131, Mar. 2012.

[207] T. W. Ching, A. H. M. Aman, W. M. H. Azamuddin, and Z. S. Attarbashi, "Performance Evaluation of AODV Routing Protocol in MANET using NS-3 Simulator," in *2021 3rd International Cyber Resilience Conference (CRC)*, 2021, pp. 1–4.

[208] S. S. A. Emira, K. Y. Youssef, and M. Abouelatta, "Design of Power Efficient Routing Protocol for Smart Livestock Farm Applications," *Adv. Sci. Technol. Eng. Syst. J.*, vol. 5, no. 6, pp. 1719–1726, Dec. 2020.

[209] A. Dahiya and C. R.K, "Performance Evaluation Of Aodv In Vanet Scenario," *Int. J. Comput. Sci. Eng. Appl.*, vol. 3, no. 1, pp. 65–70, Feb. 2013.

[210] P. K. Shrivastava and L. K. Vishwamitra, "Comparative analysis of proactive and reactive routing protocols in VANET environment," *Meas. Sensors*, vol. 16, pp. 1–10, Aug. 2021.

[211] R. Bala and C. R. Krishna, "Scenario Based Performance Analysis of AODV and GPSR Routing Protocols in a VANET," in *2015 IEEE International Conference on Computational Intelligence & Communication Technology*, 2015, pp. 432–437.

[212] D. S. Vasiliev, D. S. Meitis, and A. Abilov, "Simulation-Based Comparison of AODV, OLSR and HWMP Protocols for Flying Ad Hoc Networks," in *Lecture Notes in*

Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 8638 LNCS, Springer, Cham, 2014, pp. 245–252.

[213] R. Bilal and B. M. Khan, "Analysis of mobility models and routing schemes for flying ad-hoc networks (FANETS)," *Int. J. Appl. Eng. Res.*, vol. 12, no. 12, pp. 3263–3269, 2017.

[214] A. Guillen-Perez, A.-M. Montoya, J.-C. Sanchez-Aarnoutse, and M.-D. Cano, "A Comparative Performance Evaluation of Routing Protocols for Flying Ad-Hoc Networks in Real Conditions," *Appl. Sci.*, vol. 11, no. 10, p. 4363, May 2021.

[215] C. E. Perkins and E. M. Royer, "Draft - Multicast Ad hoc On-Demand Distance Vector (MAODV)," *IETF Internet Draft. Draft. txt*, no. July, 2000.

[216] A. Aggarwal, "AODVSEC: A Novel Approach to Secure Ad Hoc on-Demand Distance Vector (AODV) Routing Protocol from Insider Attacks in MANETs," *Int. J. Comput. Networks Commun.*, vol. 4, no. 4, pp. 191–210, Jul. 2012.

[217] P. Haas, ZJ and Pearlman, MR and Samar, *The Zone Routing Protocol (ZRP) for Ad Hoc Networks. IETF draft, July 2002*, no. May 1998. 2002, p. 11.

[218] V. Park and S. Corson, "Temporally-Ordered Routing Algorithm (TORA) Version 1 Functional Specification," *IETF MANET Work. Gr. INTERNET-DRAFT*, no. January 2002, pp. 1–23, 2001.

[219] Z. Zhai, J. Du, and Y. Ren, "The Application and Improvement of Temporally Ordered Routing Algorithm in Swarm Network with Unmanned Aerial Vehicle Nodes," in *The Ninth International Conference on Wireless and Mobile Communications (ICWMC)*, 2013, no. c, pp. 7–12.

[220] Y. Wei, M. B. Blake, and G. R. Madey, "An Operation-Time Simulation Framework for UAV Swarm Configuration and Mission Planning," *Procedia Comput. Sci.*, vol. 18, pp. 1949–1958, 2013.

[221] V. Ramasubramanian, Z. J. Haas, and E. G. Sirer, "SHARP," in *Proceedings of the 4th ACM international symposium on Mobile ad hoc networking & computing - MobiHoc '03*, 2003, p. 303.

[222] G. Pei, M. Gerla, X. Hong, and C.-C. Chiang, "A wireless hierarchical routing protocol with group mobility," in *WCNC. 1999 IEEE Wireless Communications and Networking Conference (Cat. No.99TH8466)*, 1999, vol. 3, pp. 1538–1542.

[223] R. Shirani, "Reactive-greedy-reactive in unmanned aeronautical Ad-hoc networks : a combinational routing mechanism," Carleton University, Ottawa, Ontario, 2011.

[224] R. Shirani, M. St-Hilaire, T. Kunz, Y. Zhou, J. Li, and L. Lamont, "The Performance of Greedy Geographic Forwarding in Unmanned Aeronautical Ad-Hoc Networks," in *2011 Ninth Annual Communication Networks and Services Research Conference*, 2011, pp. 161–166.

[225] M. Iordanakis *et al.*, "Ad-hoc routing protocol for aeronautical mobile ad-hoc networks," in *Fifth International Symposium on Communication Systems, Networks and Digital Signal Processing (CSNDSP)*, 2006, pp. 1–5.

[226] M. Ni, Z. Zhong, H. Wu, and D. Zhao, "A New Stable Clustering Scheme for Highly Mobile Ad Hoc Networks," in *2010 IEEE Wireless Communication and Networking Conference*, 2010, pp. 1–6.

[227] D. P. Dora, S. Kumar, and O. Kaiwartya, "Efficient dynamic caching for geocast

routing in VANETs," in *2015 2nd International Conference on Signal Processing and Integrated Networks (SPIN)*, 2015, pp. 979–983.

[228] L. Lin, Q. Sun, J. Li, and F. Yang, "A novel geographic position mobility oriented routing strategy for UAVs," *J. Comput. Inf. Syst.*, vol. 8, no. 2, pp. 709–716, 2012.

[229] L. Lin, Q. Sun, S. Wang, and F. Yang, "A geographic mobility prediction routing protocol for Ad Hoc UAV Network," in *2012 IEEE Globecom Workshops*, 2012, pp. 1597–1602.

[230] D. Medina, F. Hoffmann, F. Rossetto, and C.-H. Rokitansky, "A Geographic Routing Strategy for North Atlantic In-Flight Internet Access Via Airborne Mesh Networking," *IEEE/ACM Trans. Netw.*, vol. 20, no. 4, pp. 1231–1244, Aug. 2012.

[231] R. L. Lidowski, B. E. Mullins, and R. O. Baldwin, "A novel communications protocol using geographic routing for swarming UAVs performing a Search Mission," in *2009 IEEE International Conference on Pervasive Computing and Communications*, 2009, pp. 1–7.

[232] C. Liu and J. Wu, "Efficient Geometric Routing in Three Dimensional Ad Hoc Networks," in *IEEE INFOCOM 2009 - The 28th Conference on Computer Communications*, 2009, pp. 2751–2755.

[233] R. Flury and R. Wattenhofer, "Randomized 3D Geographic Routing," in *2008 Proceedings IEEE INFOCOM - The 27th Conference on Computer Communications*, 2008, pp. 834–842.

[234] J. Zhou, Y. Chen, B. Leong, and P. S. Sundaramoorthy, "Practical 3D geographic routing for wireless sensor networks," in *Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems - SenSys '10*, 2010, p. 337.

[235] O. S. Oubbati, A. Lakas, N. Lagraa, and M. B. Yagoubi, "CRUV: Connectivity-based traffic density aware routing using UAVs for VANets," in *2015 International Conference on Connected Vehicles and Expo (ICCVE)*, 2015, no. Ivc, pp. 68–73.

[236] O. S. Oubbati, A. Lakas, N. Lagraa, and M. B. Yagoubi, "UVAR: An intersection UAV-assisted VANET routing protocol," in *2016 IEEE Wireless Communications and Networking Conference*, 2016, vol. 2016-Septe, no. WCNC, pp. 1–6.

[237] M. Sbeiti, N. Goddemeier, D. Behnke, and C. Wietfeld, "PASER: Secure and Efficient Routing Approach for Airborne Mesh Networks," *IEEE Trans. Wirel. Commun.*, vol. 15, no. 3, pp. 1950–1964, Mar. 2016.

[238] D. Rosario, Z. Zhao, T. Braun, E. Cerqueira, A. Santos, and I. Alyafawi, "Opportunistic routing for multi-flow video dissemination over Flying Ad-Hoc Networks," in *Proceeding of IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks 2014*, 2014, pp. 1–6.

[239] J.-A. Maxa, M. S. Ben Mahmoud, and N. Larrieu, "Joint Model-Driven design and real experiment-based validation for a secure UAV Ad hoc Network routing protocol," in *2016 Integrated Communications Navigation and Surveillance (ICNS)*, 2016, pp. 1E2-1-1E2-16.

[240] K. Liu, J. Zhang, and T. Zhang, "The clustering algorithm of UAV Networking in Near-space," in *2008 8th International Symposium on Antennas, Propagation and EM Theory*, 2008, pp. 1550–1553.

[241] C. Zang and S. Zang, "Mobility prediction clustering algorithm for UAV networking," in *2011 IEEE GLOBECOM Workshops (GC Wkshps)*, 2011, no. 3, pp.

1158–1161.

[242] P. Guangyu, M. Geria, and Xiaoyan Hong, "LANMAR: landmark routing for large scale wireless ad hoc networks with group mobility," in *2000 First Annual Workshop on Mobile and Ad Hoc Networking and Computing. MobiHOC (Cat. No.00EX444)*, 2000, pp. 11–18.

[243] N. Martin, Y. Al-Mousa, and N. Shenoy, "An Integrated Routing and Medium Access Control Framework for Surveillance Networks of Mobile Devices," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 6522 LNCS, 2011, pp. 315–327.

[244] N. S. - and X. L. -, "A Novel Cluster-Based Location-Aided Routing Protocol for UAV Fleet Networks," *Int. J. Digit. Content Technol. its Appl.*, vol. 6, no. 18, pp. 376–383, Oct. 2012.

[245] B. Fu and L. A. DaSilva, "A Mesh in the Sky: A Routing Protocol for Airborne Networks," in *MILCOM 2007 - IEEE Military Communications Conference*, 2007, pp. 1–7.

[246] A. V. Leonov, "Application of bee colony algorithm for FANET routing," in *2016 17th International Conference of Young Specialists on Micro/Nanotechnologies and Electron Devices (EDM)*, 2016, vol. 2016-Augus, pp. 124–132.

[247] A. V. Leonov, "Modeling of bio-inspired algorithms AntHocNet and BeeAdHoc for Flying Ad Hoc Networks (FANETs)," in *2016 13th International Scientific-Technical Conference on Actual Problems of Electronics Instrument Engineering (APEIE)*, 2016, vol. 2, pp. 90–99.

[248] Y. Yu, L. Ru, W. Chi, Y. Liu, Q. Yu, and K. Fang, "Ant colony optimization based polymorphism-aware routing algorithm for ad hoc UAV network," *Multimed. Tools Appl.*, vol. 75, no. 22, pp. 14451–14476, Nov. 2016.

[249] A. Jabbar and J. P. G. Sterbenz, "AeroRP: A geolocation assisted aeronautical routing protocol for highly dynamic telemetry environments," in *Proceedings of the International Telemetering Conference*, 2009, vol. 45, no. Itc 2009, pp. 1–10.

[250] S. Hyeon and K.-I. Kim, "A new geographic routing protocol for aircraft ad hoc networks," in *29th Digital Avionics Systems Conference*, 2010, pp. 2.E.2-1-2.E.2-8.

[251] J. Whitbeck and V. Conan, "HYMAD: Hybrid DTN-MANET routing for dense and highly dynamic wireless networks," *Comput. Commun.*, vol. 33, no. 13, pp. 1483–1492, Aug. 2010.

[252] J. Burgess, B. Gallagher, D. Jensen, and B. N. Levine, "MaxProp: Routing for Vehicle-Based Disruption-Tolerant Networks," in *Proceedings IEEE INFOCOM 2006. 25TH IEEE International Conference on Computer Communications*, 2006, vol. 6, no. April, pp. 1–11.

[253] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, "Spray and wait," in *Proceeding of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking - WDTN '05*, 2005, pp. 252–259.

[254] A. Lindgren, A. Doria, and O. Schelén, "Probabilistic routing in intermittently connected networks," *ACM SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 7, no. 3, pp. 19–20, Jul. 2003.

[255] E. Yanmaz, R. Kuschnig, and C. Bettstetter, "Channel measurements over 802.11a-based UAV-to-ground links," in *2011 IEEE GLOBECOM Workshops (GC Wkshps)*,

2011, pp. 1280–1284.

[256] E. Yanmaz, R. Kuschnig, and C. Bettstetter, "Achieving air-ground communications in 802.11 networks with three-dimensional aerial mobility," in *2013 Proceedings IEEE INFOCOM*, 2013, pp. 120–124.

[257] E. Kuiper and S. Nadjm-Tehrani, "Geographical Routing With Location Service in Intermittently Connected MANETs," *IEEE Trans. Veh. Technol.*, vol. 60, no. 2, pp. 592–604, Feb. 2011.

[258] C. Cheng, P. Hsiao, H. Kung, and D. Vlah, "Performance Measurement of 802.11a Wireless Links from UAV to Ground Nodes with Various Antenna Orientations," in *Proceedings of 15th International Conference on Computer Communications and Networks*, 2006, pp. 303–308.

[259] D. W. Matolak and R. Sun, "Air-Ground Channel Characterization for Unmanned Aircraft Systems: The Hilly Suburban Environment," in *2014 IEEE 80th Vehicular Technology Conference (VTC2014-Fall)*, 2014, pp. 1–5.

[260] mqmaker, "WiTi board." [Online]. Available: https://goo.gl/bfSvM8. [Accessed: 04-Dec-2017].

[261] "Iperf." [Online]. Available: https://iperf.fr/. [Accessed: 20-May-2016].

[262] S. Rosati, K. Kruzelecki, G. Heitz, D. Floreano, and B. Rimoldi, "Dynamic Routing for Flying Ad Hoc Networks," *IEEE Trans. Veh. Technol.*, vol. 65, no. 3, pp. 1690–1700, Mar. 2016.

[263] J. Lee *et al.*, "Constructing a reliable and fast recoverable network for drones," in *2016 IEEE International Conference on Communications (ICC)*, 2016, pp. 1–6.

[264] I. A. Kaysina, D. S. Vasiliev, A. Abilov, D. S. Meitis, and A. E. Kaysin, "Performance evaluation testbed for emerging relaying and coding algorithms in Flying Ad Hoc Networks," in *2018 Moscow Workshop on Electronic and Networking Technologies (MWENT)*, 2018, pp. 1–5.

[265] A. Guillen-Perez and M. D. Cano, "Pedestrian characterisation in urban environments combining WiFi and AI," *Int. J. Sens. Networks*, vol. 37, no. 1, p. 48, 2021.

[266] J. Garcia, A. Gardel, I. Bravo, J. L. Lazaro, M. Martinez, and D. Rodriguez, "Directional People Counter Based on Head Tracking," *IEEE Trans. Ind. Electron.*, vol. 60, no. 9, pp. 3991–4000, Sep. 2013.

[267] M. Stubenschrott, T. Matyus, and C. Kogler, "Real-Time Estimation of Pedestrian Inflow Rates from Saturated Sensor Counting Data in a Complex Metro Station," in *2015 IEEE 18th International Conference on Intelligent Transportation Systems*, 2015, vol. 2015-Octob, pp. 1958–1963.

[268] Y. Yuan, "Crowd Monitoring Using Mobile Phones," in *2014 Sixth International Conference on Intelligent Human-Machine Systems and Cybernetics*, 2014, vol. 1, pp. 261–264.

[269] E. Cianca, M. De Sanctis, and S. Di Domenico, "Radios as Sensors," *IEEE Internet Things J.*, vol. 4, no. 2, pp. 363–373, Apr. 2017.

[270] L. Schauer, M. Werner, and P. Marcus, "Estimating Crowd Densities and Pedestrian Flows Using Wi-Fi and Bluetooth," in *Proceedings of the 11th International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services*, 2014, pp. 171–177.

[271] H. Li, E. C. L. Chan, X. Guo, J. Xiao, K. Wu, and L. M. Ni, "Wi-Counter: Smartphone-Based People Counter Using Crowdsourced Wi-Fi Signal Data," *IEEE Trans. Human-Machine Syst.*, vol. 45, no. 4, pp. 442–452, Aug. 2015.

[272] C. Xu *et al.*, "SCPL," in *Proceedings of the 12th international conference on Information processing in sensor networks - IPSN '13*, 2013, p. 79.

[273] S. H. Doong, "Spectral Human Flow Counting with RSSI in Wireless Sensor Networks," in *2016 International Conference on Distributed Computing in Sensor Systems (DCOSS)*, 2016, pp. 110–112.

[274] S. Y. Fadhlullah and W. Ismail, "A Statistical Approach in Designing an RF-Based Human Crowd Density Estimation System," *Int. J. Distrib. Sens. Networks*, vol. 12, no. 2, p. 8351017, Feb. 2016.

[275] C.-Y. Chiang, J.-Y. Chuang, J.-K. Chen, C.-C. Hung, W.-H. Chen, and K.-R. Lo, "Estimating instant traffic information by identifying handover patterns of UMTS signals," in *2011 14th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, 2011, pp. 390–395.

[276] K. Farrahi and D. Gatica-Perez, "Probabilistic Mining of Socio-Geographic Routines From Mobile Phone Data," *IEEE J. Sel. Top. Signal Process.*, vol. 4, no. 4, pp. 746–755, Aug. 2010.

[277] D. Anguita, A. Ghio, L. Oneto, X. Parra, and J. L. Reyes-Ortiz, "Human Activity Recognition on Smartphones Using a Multiclass Hardware-Friendly Support Vector Machine," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 7657 LNCS, 2012, pp. 216–223.

[278] M. Vanhoef, C. Matte, M. Cunche, L. S. Cardoso, and F. Piessens, "Why MAC Address Randomization is not Enough," in *Proceedings of the 11th ACM on Asia Conference on Computer and Communications Security*, 2016, pp. 413–424.

[279] M. Uras, R. Cossu, E. Ferrara, O. Bagdasar, A. Liotta, and L. Atzori, "WiFi Probes sniffing: an Artificial Intelligence based approach for MAC addresses de-randomization," in *2020 IEEE 25th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD)*, 2020, pp. 1–6.

[280] J. Martin *et al.*, "A Study of MAC Address Randomization in Mobile Devices and When it Fails," *Proc. Priv. Enhancing Technol.*, vol. 2017, no. 4, pp. 365–383, Oct. 2017.

[281] E. Fenske, D. Brown, J. Martin, T. Mayberry, P. Ryan, and E. Rye, "Three Years Later: A Study of MAC Address Randomization In Mobile Devices And When It Succeeds," *Proc. Priv. Enhancing Technol.*, vol. 2021, no. 3, pp. 164–181, Jul. 2021.

[282] M. Nakatsuka, H. Iwatani, and J. Katto, "A Study on Passive Crowd Density Estimation using Wireless Sensors," in *4th Intl. Conf. on Mobile Computing and Ubiquitous Networking (ICMU)*, 2008, no. 2, pp. 1–6.

[283] T. Yoshida and Y. Taniguchi, "Estimating the number of people using existing WiFi access point based on support vector regression," in *Information (Japan)*, 2016, vol. 19, no. 7A, pp. 2661–2668.

[284] S. Depatla, A. Muralidharan, and Y. Mostofi, "Occupancy Estimation Using Only WiFi Power Measurements," *IEEE J. Sel. Areas Commun.*, vol. 33, no. 7, pp. 1381–1393, Jul. 2015.

[285] W. Xi *et al.*, "Electronic frog eye: Counting crowd using WiFi," in *IEEE INFOCOM 2014 - IEEE Conference on Computer Communications*, 2014, pp. 361–369.

[286] S. Di Domenico, G. Pecoraro, E. Cianca, and M. De Sanctis, "Trained-once device-free crowd counting and occupancy estimation using WiFi: A Doppler spectrum based approach," in *2016 IEEE 12th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, 2016, pp. 1–8.

[287] M. T. Hoang, B. Yuen, X. Dong, T. Lu, R. Westendorp, and K. Reddy, "Recurrent Neural Networks for Accurate RSSI Indoor Localization," *IEEE Internet Things J.*, vol. 6, no. 6, pp. 10639–10651, Dec. 2019.

[288] C.-H. Hsieh, J.-Y. Chen, and B.-H. Nien, "Deep Learning-Based Indoor Localization Using Received Signal Strength and Channel State Information," *IEEE Access*, vol. 7, pp. 33256–33267, 2019.

[289] A. Zanella, "Best Practice in RSS Measurements and Ranging," *IEEE Commun. Surv. Tutorials*, vol. 18, no. 4, pp. 2662–2686, 2016.

[290] K. Woyach, D. Puccinelli, and M. Haenggi, "Sensorless Sensing in Wireless Networks: Implementation and Measurements," in *2006 4th International Symposium on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks*, 2006, pp. 1–8.

[291] Z. Yang, Z. Zhou, and Y. Liu, "From RSSI to CSI," *ACM Comput. Surv.*, vol. 46, no. 2, pp. 1–32, Nov. 2013.

[292] H. Chen, Y. Zhang, W. Li, X. Tao, and P. Zhang, "ConFi: Convolutional Neural Networks Based Indoor Wi-Fi Localization Using Channel State Information," *IEEE Access*, vol. 5, pp. 18066–18074, 2017.

[293] J. P. Conti, T. B. N. da Silveira, and D. P. Araujo, "Dynamic crowd counting via 802.11 MAC layer," in *2016 IEEE International Symposium on Consumer Electronics (ISCE)*, 2016, pp. 13–14.

[294] A. Guillen-Perez and M.-D. Cano, "Counting and locating people in outdoor environments: a comparative experimental study using WiFi-based passive methods," *ITM Web Conf.*, vol. 24, pp. 1–10, Feb. 2019.

[295] A.-C. Petre, C. Chilipirea, M. Baratchi, C. Dobre, and M. van Steen, "WiFi Tracking of Pedestrian Behavior," in *Smart Sensors Networks*, 1st ed., Elsevier, 2017, pp. 309–337.

[296] J. Freudiger, "How talkative is your mobile device?," in *Proceedings of the 8th ACM Conference on Security & Privacy in Wireless and Mobile Networks*, 2015, pp. 1–6.

[297] A. Guillen-Perez and M. D. Cano Banos, "A WiFi-based method to count and locate pedestrians in urban traffic scenarios," in *2018 14th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, 2018, vol. 2018-Octob, pp. 123–130.

[298] A. E. C. Redondi and M. Cesana, "Building up knowledge through passive WiFi probes," *Comput. Commun.*, vol. 117, pp. 1–12, Feb. 2018.

[299] W. Pattanusorn, I. Nilkhamhang, S. Kittipiyakul, K. Ekkachai, and A. Takahashi, "Passenger estimation system using Wi-Fi probe request," in *2016 7th International Conference of Information and Communication Technology for Embedded Systems (IC-ICTES)*, 2016, vol. 2016, pp. 67–72.

[300] E. Vattapparamban, B. S. Ciftler, I. Guvenc, K. Akkaya, and A. Kadri, "Indoor

occupancy tracking in smart buildings using passive sniffing of probe requests," in *2016 IEEE International Conference on Communications Workshops (ICC)*, 2016, pp. 38–44.

[301] L. Sun, S. Chen, Z. Zheng, and L. Xu, "Mobile Device Passive Localization Based on IEEE 802.11 Probe Request Frames," *Mob. Inf. Syst.*, vol. 2017, pp. 1–10, 2017.

[302] B. S. Ciftler, S. Dikmese, I. Guvenc, K. Akkaya, and A. Kadri, "Occupancy Counting With Burst and Intermittent Signals in Smart Buildings," *IEEE Internet Things J.*, vol. 5, no. 2, pp. 724–735, Apr. 2018.

[303] Y. Fukuzaki, M. Mochizuki, K. Murao, and N. Nishio, "A pedestrian flow analysis system using Wi-Fi packet sensors to a real environment," in *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication*, 2014, no. September 2014, pp. 721–730.

[304] V. Acuna, A. Kumbhar, E. Vattapparamban, F. Rajabli, and I. Guvenc, "Localization of WiFi Devices Using Probe Requests Captured at Unmanned Aerial Vehicles," in *2017 IEEE Wireless Communications and Networking Conference (WCNC)*, 2017, pp. 1–6.

[305] A. B. M. Musa and J. Eriksson, "Tracking unmodified smartphones using wi-fi monitors," in *Proceedings of the 10th ACM Conference on Embedded Network Sensor Systems - SenSys '12*, 2012, p. 281.

[306] J. Heidemann and U. S. C. Isi, "OMNeT++ Discrete Event Simulator." [Online]. Available: www.omnetpp.org. [Accessed: 17-Jul-2018].

[307] L. Mészáros, A. Varga, and M. Kirsche, "INET Framework," in *Recent Advances in Network Simulation*, Springer, Cham, 2019, pp. 55–106.

[308] I. Idris, *NumPy Cookbook*. Packt Publishing, 2012.

[309] C. Hill, "SciPy," in *Learning Scientific Programming with Python*, Cambridge: Cambridge University Press, 2016, pp. 333–401.

[310] W. McKinney, *Pandas : a Python Data Analysis Library*, 2nd ed. 2017.

[311] F. Pedregosa *et al.*, "Scikit-learn: Machine learning in Python," *J. Mach. Learn. Res.*, 2011.

[312] A. Pascale, H. T. Lam, and R. Nair, "Characterization of Network Traffic Processes Under Adaptive Traffic Control Systems," *Transp. Res. Procedia*, vol. 9, pp. 205–224, 2015.

[313] A. Guillen-Perez and M. Cano, "Intelligent IoT systems for traffic management: A practical application," *IET Intell. Transp. Syst.*, vol. 15, no. 2, pp. 273–285, Feb. 2021.

[314] Y. Wang, X. Yang, H. Liang, and Y. Liu, "A Review of the Self-Adaptive Traffic Signal Control System Based on Future Traffic Environment," *J. Adv. Transp.*, vol. 2018, pp. 1–12, Jun. 2018.

[315] F. Meggio, A. Donella-deana, and L. A. Pinna, "The Use of Soybean Trypsin Inhibitors as Phosphorylatable Substrates for a Rat Liver Protein Kinase," *J. Biochem.*, vol. 86, no. 1, pp. 261–264, Jul. 1979.

[316] J. D. C. Little, M. D. Kelson, and N. H. Gartner, "Maxband: a Program for Setting Signals on Arteries and Triangular Networks.," *Transp. Res. Rec.*, vol. 1, no. 795, pp. 40–46, 1981.

[317] A. G. Sims and K. W. Dobinson, "The Sydney coordinated adaptive traffic (SCAT)

system philosophy and benefits," *IEEE Trans. Veh. Technol.*, vol. 29, no. 2, pp. 130–137, May 1980.

[318] D. I. Robertson and R. D. Bretherton, "Optimizing networks of traffic signals in real time-the SCOOT method," *IEEE Trans. Veh. Technol.*, vol. 40, no. 1, pp. 11–15, Feb. 1991.

[319] P. Varaiya, "Max pressure control of a network of signalized intersections," *Transp. Res. Part C Emerg. Technol.*, vol. 36, pp. 177–195, Nov. 2013.

[320] J. Hu, M. D. Fontaine, B. B. Park, and J. Ma, "Field Evaluations of an Adaptive Traffic Signal—Using Private-Sector Probe Data," *J. Transp. Eng.*, vol. 142, no. 1, pp. 1–18, Jan. 2016.

[321] N. Jiang, "Optimal Signal Design for Mixed Equilibrium Networks with Autonomous and Regular Vehicles," *J. Adv. Transp.*, vol. 2017, pp. 1–13, 2017.

[322] R. Sanchez-Iborra and M.-D. Cano, "On the Similarities Between Urban Traffic Management and Communication Networks: Application of the Random Early Detection Algorithm for Self-Regulating Intersections," *IEEE Intell. Transp. Syst. Mag.*, vol. 9, no. 4, pp. 48–61, 2017.

[323] R. Sanchez-Iborra, J. F. Ingles-Romero, G. Domenech-Asensi, J. L. Moreno-Cegarra, and M.-D. Cano, "Proactive Intelligent System for Optimizing Traffic Signaling," in *2016 IEEE 14th Intl Conf on Dependable, Autonomic and Secure Computing, 14th Intl Conf on Pervasive Intelligence and Computing, 2nd Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress(DASC/PiCom/DataCom/CyberSciTech)*, 2016, pp. 544–551.

[324] C. P. Pappis and E. H. Mamdani, "A Fuzzy Logic Controller for a Trafc Junction," *IEEE Trans. Syst. Man. Cybern.*, vol. 7, no. 10, pp. 707–717, 1977.

[325] S. Chiu and S. Chand, "Adaptive Traffic Signal Control Using Fuzzy Logic," in *Proceedings. The First IEEE Regional Conference on Aerospace Control Systems,* 1993, pp. 122–126.

[326] J. Niittymäki and M. Pursula, "Signal control using fuzzy logic," *Fuzzy Sets Syst.*, vol. 116, no. 1, pp. 11–22, Nov. 2000.

[327] K. Dresner and P. Stone, "Multiagent traffic management: A reservation-based intersection control mechanism," in *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems, AAMAS 2004*, 2004, vol. 2, pp. 530–537.

[328] S. R. K. Branavan, D. Silver, and R. Barzilay, "Learning to Win by Reading Manuals in a Monte-Carlo Framework," *J. Artif. Intell. Res.*, vol. 43, pp. 661–704, Apr. 2012.

[329] Z. Li, M. V. Chitturi, D. Zheng, A. R. Bill, and D. A. Noyce, "Modeling Reservation-Based Autonomous Intersection Control in VISSIM," *Transp. Res. Rec. J. Transp. Res. Board*, vol. 2381, no. 1, pp. 81–90, Jan. 2013.

[330] C.-H. Wei, "Analysis of artificial neural network models for freeway ramp metering control," *Artif. Intell. Eng.*, vol. 15, no. 3, pp. 241–252, Jul. 2001.

[331] D. Srinivasan, M. C. Choy, and R. L. Cheu, "Neural Networks for Real-Time Traffic Signal Control," *IEEE Trans. Intell. Transp. Syst.*, vol. 7, no. 3, pp. 261–272, Sep. 2006.

[332] M. Tubaishat, Y. Shang, and H. Shi, "Adaptive Traffic Light Control with Wireless Sensor Networks," in *2007 4th IEEE Consumer Communications and Networking Conference*, 2007, pp. 187–191.

[333] S. S. Mousavi, M. Schukat, and E. Howley, "Traffic light control using deep policy-gradient and value-function-based reinforcement learning," *IET Intell. Transp. Syst.*, vol. 11, no. 7, pp. 417–423, Sep. 2017.

[334] W. Genders and S. Razavi, "Evaluating reinforcement learning state representations for adaptive traffic signal control," *Procedia Comput. Sci.*, vol. 130, pp. 26–33, 2018.

[335] P. G. Balaji, X. German, and D. Srinivasan, "Urban traffic signal control using reinforcement learning agents," *IET Intell. Transp. Syst.*, vol. 4, no. 3, pp. 177–181, 2010.

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

[337] J. Sanchez, M. Galan, and E. Rubio, "Applying a Traffic Lights Evolutionary Optimization Technique to a Real Case: 'Las Ramblas' Area in Santa Cruz de Tenerife," *IEEE Trans. Evol. Comput.*, vol. 12, no. 1, pp. 25–40, Feb. 2008.

[338] J. Garcia-Nieto, A. C. Olivera, and E. Alba, "Optimal Cycle Program of Traffic Lights With Particle Swarm Optimization," *IEEE Trans. Evol. Comput.*, vol. 17, no. 6, pp. 823–839, Dec. 2013.

[339] E. Segredo, G. Luque, C. Segura, and E. Alba, "Optimising Real-World Traffic Cycle Programs by Using Evolutionary Computation," *IEEE Access*, vol. 7, pp. 43915–43932, 2019.

[340] J. J. Sanchez-Medina, M. J. Galan-Moreno, and E. Rubio-Royo, "Traffic signal optimization in 'la almozara' district in saragossa under congestion conditions, using genetic algorithms, traffic microsimulation, and cluster computing," *IEEE Trans. Intell. Transp. Syst.*, vol. 11, no. 1, pp. 132–141, Mar. 2010.

[341] K. T. K. Teo, W. Y. Kow, and Y. K. Chin, "Optimization of Traffic Flow within an Urban Traffic Light Intersection with Genetic Algorithm," in *2010 Second International Conference on Computational Intelligence, Modelling and Simulation*, 2010, pp. 172–177.

[342] C. Tang, X. Wei, M. Hao, C. Zhu, R. Wang, and W. Chen, "Traffic Signal Phase Scheduling Based on Device-to-Device Communication," *IEEE Access*, vol. 6, pp. 47636–47645, 2018.

[343] A. Rehman, M. Mazhar Rathore, A. Paul, F. Saeed, and R. W. Ahmad, "Vehicular traffic optimisation and even distribution using ant colony in smart city environment," *IET Intell. Transp. Syst.*, vol. 12, no. 7, pp. 594–601, Sep. 2018.

[344] K. Jerry, K. Yujun, O. Kwasi, Z. Enzhan, and T. Parfait, "NetLogo implementation of an ant colony optimisation solution to the traffic problem," *IET Intell. Transp. Syst.*, vol. 9, no. 9, pp. 862–869, Nov. 2015.

[345] A. Ahmad, R. Arshad, S. A. Mahmud, G. M. Khan, and H. S. Al-Raweshidy, "Earliest-Deadline-Based Scheduling to Reduce Urban Traffic Congestion," *IEEE Trans. Intell. Transp. Syst.*, vol. 15, no. 4, pp. 1510–1526, Aug. 2014.

[346] K. M. Ahmad Yousef, A. Shatnawi, and M. Latayfeh, "Intelligent traffic light scheduling technique using calendar-based history information," *Futur. Gener. Comput. Syst.*, vol. 91, pp. 124–135, Feb. 2019.

[347] P. Ryus, M. Vandehey, L. Elefteriadou, R. Dowling G, and B. Ostrom K, *Highway Capacity Manual 2010*. Transportation Research Board, 2011.

[348] A. Guillen-Perez and M.-D. Cano, "Influencia del ciclo de trabajo de los semáforos en una intersección simple en múltiples parámetros ante una densidad de tráfico incremental," in *XIV Jornadas de Ingeniería Telemática (JITEL 2019)*, 2019, no. JITEL, pp. 22–24.

[349] P. A. Lopez *et al.*, "Microscopic Traffic Simulation using SUMO," in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, 2018, vol. 2018-Novem, pp. 2575–2582.

[350] R. Tang, Y. Lu, L. Liu, L. Mou, O. Vechtomova, and J. Lin, "Distilling task-specific knowledge from BERT into simple neural networks," *arXiv*. 2019.

[351] O. M. Andrychowicz *et al.*, "Learning dexterous in-hand manipulation," *Int. J. Rob. Res.*, vol. 39, no. 1, pp. 3–20, Jan. 2020.

[352] O. Vinyals *et al.*, "Grandmaster level in StarCraft II using multi-agent reinforcement learning," *Nature*, vol. 575, no. 7782, pp. 350–354, Nov. 2019.

[353] E. Talpes *et al.*, "Compute Solution for Tesla's Full Self-Driving Computer," *IEEE Micro*, vol. 40, no. 2, pp. 25–35, Mar. 2020.

[354] A. W. Senior *et al.*, "Improved protein structure prediction using potentials from deep learning," *Nature*, vol. 577, no. 7792, pp. 706–710, Jan. 2020.

[355] S. Wang, M. W. Levin, and R. J. Caverly, "Optimal parking management of connected autonomous vehicles: A control-theoretic approach," *Transp. Res. Part C Emerg. Technol.*, vol. 124, pp. 102–115, Mar. 2021.

[356] N. C. Luong *et al.*, "Applications of Deep Reinforcement Learning in Communications and Networking: A Survey," *IEEE Commun. Surv. Tutorials*, vol. 21, no. 4, pp. 3133–3174, 2019.

[357] T. Fu, C. Wang, and N. Cheng, "Deep-Learning-Based Joint Optimization of Renewable Energy Storage and Routing in Vehicular Energy Network," *IEEE Internet Things J.*, vol. 7, no. 7, pp. 6229–6241, Jul. 2020.

[358] H.-C. Chu, Y.-X. Liao, L. Chang, and Y.-H. Lee, "Traffic Light Cycle Configuration of Single Intersection Based on Modified Q-Learning," *Appl. Sci.*, vol. 9, no. 21, p. 4558, Oct. 2019.

[359] D. McKenney and T. White, "Distributed and adaptive traffic signal control within a realistic traffic simulation," *Eng. Appl. Artif. Intell.*, vol. 26, no. 1, pp. 574–583, Jan. 2013.

[360] P. C. Chu and J. E. Beasley, "A Genetic Algorithm for the Multidimensional Knapsack Problem," *J. Heuristics*, vol. 4, no. 1, pp. 63–86, 1998.

[361] V. Roberge, M. Tarbouchi, and G. Labonte, "Comparison of Parallel Genetic Algorithm and Particle Swarm Optimization for Real-Time UAV Path Planning," *IEEE Trans. Ind. Informatics*, vol. 9, no. 1, pp. 132–141, Feb. 2013.

[362] "Madrid's Vehicle Fleet," 2018. [Online]. Available: bit.ly/2svNegb. [Accessed: 28-Apr-2019].

[363] C. Matzer *et al.*, "Update of Emission Factors for HBEFA Version 4.1," 2019.

[364] D. Carlino, S. D. Boyles, and P. Stone, "Auction-based autonomous intersection management," in *16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013)*, 2013, no. Itsc, pp. 529–534.

[365] M. Bashiri, H. Jafarzadeh, and C. H. Fleming, "PAIM: Platoon-based Autonomous

Intersection Management," in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, 2018, vol. 2018-Novem, pp. 374–380.

[366] Y. Bichiou and H. A. Rakha, "Developing an Optimal Intersection Control System for Automated Connected Vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 5, pp. 1908–1916, May 2019.

[367] P. Dai, K. Liu, Q. Zhuge, E. H. M. Sha, V. C. S. Lee, and S. H. Son, "Quality-of-Experience-Oriented Autonomous Intersection Control in Vehicular Networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 7, pp. 1956–1967, Jul. 2016.

[368] Y. Wu, H. Chen, and F. Zhu, "DCL-AIM: Decentralized coordination learning of autonomous intersection management for connected and automated vehicles," *Transp. Res. Part C Emerg. Technol.*, vol. 103, no. November 2018, pp. 246–260, Jun. 2019.

[369] H. Xu, Y. Zhang, L. Li, and W. Li, "Cooperative Driving at Unsignalized Intersections Using Tree Search," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 11, pp. 4563–4571, Nov. 2020.

[370] D. Fajardo, T.-C. Au, S. T. Waller, P. Stone, and D. Yang, "Automated Intersection Control," *Transp. Res. Rec. J. Transp. Res. Board*, vol. 2259, no. 1, pp. 223–232, Jan. 2011.

[371] M. A. S. Kamal, J. Imura, T. Hayakawa, A. Ohata, and K. Aihara, "A Vehicle-Intersection Coordination Scheme for Smooth Flows of Traffic Without Using Traffic Lights," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 3, pp. 1136–1147, Jun. 2015.

[372] X. Qian, F. Altché, J. Grégoire, and A. Fortelle, "Autonomous Intersection Management systems: criteria, implementation and evaluation," *IET Intell. Transp. Syst.*, vol. 11, no. 3, pp. 182–189, Apr. 2017.

[373] M. W. Levin, H. Fritz, and S. D. Boyles, "On Optimizing Reservation-Based Intersection Controls," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 3, pp. 505–515, Mar. 2017.

[374] A. Mirheli, L. Hajibabai, and A. Hajbabaie, "Development of a signal-head-free intersection control logic in a fully connected and autonomous vehicle environment," *Transp. Res. Part C Emerg. Technol.*, vol. 92, no. May, pp. 412–425, Jul. 2018.

[375] Z. He, L. Zheng, L. Lu, and W. Guan, "Erasing Lane Changes From Roads: A Design of Future Road Intersections," *IEEE Trans. Intell. Veh.*, vol. 3, no. 2, pp. 173–184, Jun. 2018.

[376] B. Li, Y. Zhang, Y. Zhang, N. Jia, and Y. Ge, "Near-Optimal Online Motion Planning of Connected and Automated Vehicles at a Signal-Free and Lane-Free Intersection," in *2018 IEEE Intelligent Vehicles Symposium (IV)*, 2018, vol. 2018-June, no. Iv, pp. 1432–1437.

[377] B. Li and Y. Zhang, "Fault-Tolerant Cooperative Motion Planning of Connected and Automated Vehicles at a Signal-Free and Lane-Free Intersection," *IFAC-PapersOnLine*, vol. 51, no. 24, pp. 60–67, 2018.

[378] A. Mirheli, M. Tajalli, L. Hajibabai, and A. Hajbabaie, "A consensus-based distributed trajectory control in a signal-free intersection," *Transp. Res. Part C Emerg. Technol.*, vol. 100, no. January, pp. 161–176, Mar. 2019.

[379] A. Guillen-Perez and M. Cano, "Multi-Agent Deep Reinforcement Learning to

Manage Connected Autonomous Vehicles at Tomorrow's Intersections," *IEEE Trans. Veh. Technol.*, vol. On review, pp. 1–12, 2021.

[380] L. Chen and C. Englund, "Cooperative Intersection Management: A Survey," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 2, pp. 570–586, Feb. 2016.

[381] X. Zhao, J. Wang, Y. Chen, and G. Yin, "Multi-objective Cooperative Scheduling of CAVs at Non-Signalized Intersection," in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, 2018, vol. 2018-Novem, pp. 3314–3319.

[382] M. W. Levin and D. Rey, "Conflict-point formulation of intersection control for autonomous vehicles," *Transp. Res. Part C Emerg. Technol.*, vol. 85, no. May, pp. 528–547, Dec. 2017.

[383] L. Li and F.-Y. Wang, "Cooperative Driving at Blind Crossings Using Intervehicle Communication," *IEEE Trans. Veh. Technol.*, vol. 55, no. 6, pp. 1712–1724, Nov. 2006.

[384] J. Wu, A. Abbas-Turki, A. Correia, and A. El Moudni, "Discrete Intersection Signal Control," in *2007 IEEE International Conference on Service Operations and Logistics, and Informatics*, 2007, pp. 1–6.

[385] F. Zhu and S. V. Ukkusuri, "A linear programming formulation for autonomous intersection control within a dynamic traffic assignment and connected vehicle environment," *Transp. Res. Part C Emerg. Technol.*, vol. 55, pp. 363–378, Jun. 2015.

[386] M. Vasirani and S. Ossowski, "A market-inspired approach to reservation-based urban road traffic management," in *Proceedings of the International Joint Conference on Autonomous Agents and Multiagent Systems, AAMAS*, 2009, vol. 1, pp. 617–624.

[387] J. Wu, A. Abbas-Turki, and A. El Moudni, "Cooperative driving: an ant colony system for autonomous intersection management," *Appl. Intell.*, vol. 37, no. 2, pp. 207–222, Sep. 2012.

[388] S. Fujimoto, H. Van Hoof, and D. Meger, "Addressing Function Approximation Error in Actor-Critic Methods," in *35th International Conference on Machine Learning, ICML 2018*, 2018, vol. 4, pp. 2587–2601.

[389] Y. Bengio, J. Louradour, R. Collobert, and J. Weston, "Curriculum learning," in *Proceedings of the 26th Annual International Conference on Machine Learning - ICML '09*, 2009, vol. 382, pp. 1–8.

[390] K. Dresner and P. Stone, "A Multiagent Approach to Autonomous Intersection Management," *J. Artif. Intell. Res.*, vol. 31, pp. 591–656, Mar. 2008.

[391] Peter Stone and Kurt Dresner, "Human-usable and emergency vehicle–aware control policies for autonomous intersection management," in *Fourth International Workshop on Agents in Traffic and Transportation (ATT), Hakodate, Japan*, 2016, no. May, pp. 17–25.

[392] K. Dresner and P. Stone, "Sharing the road: Autonomous vehicles meet human drivers," in *IJCAI International Joint Conference on Artificial Intelligence*, 2007, vol. 1, no. January, pp. 1263–1268.

[393] R. Tachet *et al.*, "Revisiting Street Intersections Using Slot-Based Systems," *PLoS One*, vol. 11, no. 3, p. e0149607, Mar. 2016.

[394] S. Huang, A. W. Sadek, and Y. Zhao, "Assessing the Mobility and Environmental Benefits of Reservation-Based Intelligent Intersections Using an Integrated Simulator," *IEEE Trans. Intell. Transp. Syst.*, vol. 13, no. 3, pp. 1201–1214, Sep. 2012.

[395] X. Wei, G. Tan, and N. Ding, "Batch-Light: An adaptive intelligent intersection control policy for autonomous vehicles," in *2014 IEEE International Conference on Progress in Informatics and Computing*, 2014, pp. 98–103.

[396] R. Tachet *et al.*, "Revisiting street intersections using slot based systems Supplementary Information," *PLoS One*, vol. 11, no. 3, pp. 1–13, 2015.

[397] M. W. Levin, S. D. Boyles, and R. Patel, "Paradoxes of reservation-based intersection controls in traffic networks," *Transp. Res. Part A Policy Pract.*, vol. 90, pp. 14–25, Aug. 2016.

[398] A. Guillen-Perez and M.-D. Cano, "RAIM: Reinforced Autonomous Intersection Management - AIM based on MADRL," in *NeurIPS 2020 - Workshop Challenges of Real-World RL*, 2020, pp. 1–12.

[399] D. Silver *et al.*, "Mastering the game of Go without human knowledge," *Nature*, vol. 550, no. 7676, pp. 354–359, Oct. 2017.

[400] S. Ross, G. J. Gordon, and J. A. Bagnell, "A Reduction of Imitation Learning and Structured Prediction to No-Regret Online Learning," in *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2010, vol. 15, pp. 627–635.

[401] Y. Liu, A. Gupta, P. Abbeel, and S. Levine, "Imitation from Observation: Learning to Imitate Behaviors from Raw Video via Context Translation," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 2018, pp. 1118–1125.

[402] T. Hester *et al.*, "Deep Q-learning from Demonstrations," in *32nd AAAI Conference on Artificial Intelligence, AAAI 2018*, 2017, pp. 3223–3230.

[403] M. Vecerik *et al.*, "Leveraging Demonstrations for Deep Reinforcement Learning on Robotics Problems with Sparse Rewards," Jul. 2017.

[404] A. Nair, B. McGrew, M. Andrychowicz, W. Zaremba, and P. Abbeel, "Overcoming Exploration in Reinforcement Learning with Demonstrations," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 2018, pp. 6292–6299.

[405] A. Guillen-Perez and M. Cano, "Learning from Oracle Demonstrations – A new approach to develop Autonomous Intersection Management control algorithms based on Multi-Agent Deep Reinforcement Learning," *Int. J. Intell. Syst.*, vol. On Review, pp. 1–12, 2021.

[406] M. Bain and C. Sammut, "A Framework for Behavioural Cloning," *Mach. Intell.*, vol. 15, no. 1, pp. 103–129, 2001.

[407] S. Daftry, J. A. Bagnell, and M. Hebert, "Learning Transferable Policies for Monocular Reactive MAV Control," in *Springer Proceedings in Advanced Robotics*, vol. 1, 2017, pp. 3–11.

[408] H. Daumé, J. Langford, and D. Marcu, "Search-based structured prediction," *Mach. Learn.*, vol. 75, no. 3, pp. 297–325, Jun. 2009.

[409] S. Ross and J. A. Bagnell, "Efficient reductions for imitation learning," *J. Mach. Learn. Res.*, vol. 9, pp. 661–668, 2010.

[410] A. Raffin, J. Kober, and F. Stulp, "Smooth Exploration for Robotic Reinforcement Learning," in *5th Conference on Robot Learning (CoRL 2021)*, 2020, pp. 1–15.

[411] S. Fujimoto, H. Van Hoof, and D. Meger, "Addressing Function Approximation Error in Actor-Critic Methods," in *35th International Conference on Machine*

Learning, ICML 2018, 2018, vol. 4, pp. 2587–2601.

[412] A. Stooke and P. Abbeel, "rlpyt: A Research Code Base for Deep Reinforcement Learning in PyTorch," Sep. 2019.

[413] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," in *35th International Conference on Machine Learning, ICML 2018*, 2018.

[414] V. Mnih *et al.*, "Asynchronous methods for deep reinforcement learning," in *33rd International Conference on Machine Learning, ICML 2016*, 2016, pp. 1928–1937.

[415] A. Guillen-Perez and M. Cano, "AIM5LA: A Latency-Aware Deep Reinforcement Learning-Based Autonomous Intersection Management system for 5G Communication Networks," *Sensors*, vol. Accepted, pp. 1–20, 2022.

[416] F. Perronnet, A. Abbas-Turki, and A. El Moudni, "A sequenced-based protocol to manage autonomous vehicles at isolated intersections," in *16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013)*, 2013, pp. 1811–1816.

[417] B. Zheng, C.-W. Lin, H. Liang, S. Shiraishi, W. Li, and Q. Zhu, "Delay-Aware Design, Analysis and Verification of Intelligent Intersection Management," in *2017 IEEE International Conference on Smart Computing (SMARTCOMP)*, 2017, pp. 1–8.

[418] E. Andert, M. Khayatian, and A. Shrivastava, "Crossroads," in *Proceedings of the 54th Annual Design Automation Conference 2017*, 2017, vol. Part 12828, pp. 1–6.

[419] M. Khayatian, Y. Lou, M. Mehrabian, and A. Shirvastava, "Crossroads+," *ACM Trans. Cyber-Physical Syst.*, vol. 4, no. 2, pp. 1–28, Apr. 2020.

[420] N. Wu, B. Green, X. Ben, and S. O'Banion, "Deep Transformer Models for Time Series Forecasting: The Influenza Prevalence Case," in *37 th International Conference on Machine Learning*, 2020, vol. abs/2001.0, pp. 1–10.

[421] G. Nardini, D. Sabella, G. Stea, P. Thakkar, and A. Virdis, "Simu5G–An OMNeT++ Library for End-to-End Performance Evaluation of 5G Networks," *IEEE Access*, vol. 8, pp. 181176–181191, 2020.

[422] A. Virdis, G. Nardini, G. Stea, and D. Sabella, "End-to-End Performance Evaluation of MEC Deployments in 5G Scenarios," *J. Sens. Actuator Networks*, vol. 9, no. 4, p. 57, Dec. 2020.

[423] A. Chaudhary, K. S. Chouhan, J. Gajrani, and B. Sharma, "Deep Learning With PyTorch," in *Deep Learning With PyTorch*, 2020, pp. 61–95.

[424] A. Guillen-Perez and M.-D. Cano, "6G Communications Network Framework in the context of Edge-Decentralized Cooperative Autonomous Driving," *Appl. Sci.*, vol. On review, pp. 1–14, 2021.

[425] B. Aazhang *et al.*, *Key drivers and research challenges for 6G ubiquitous wireless intelligence (white paper)*, 1st ed. Oulu: 6G Flagship, University of Oulu, Oulu, 2019.

[426] J. He, K. Yang, and H.-H. Chen, "6G Cellular Networks and Connected Autonomous Vehicles," *IEEE Netw.*, vol. 35, no. 4, pp. 255–261, Jul. 2021.

[427] W. Saad, M. Bennis, and M. Chen, "A Vision of 6G Wireless Systems: Applications, Trends, Technologies, and Open Research Problems," *IEEE Netw.*, vol. 34, no. 3, pp. 134–142, May 2020.

[428] K. B. Letaief, W. Chen, Y. Shi, J. Zhang, and Y.-J. A. Zhang, "The Roadmap to 6G: AI Empowered Wireless Networks," *IEEE Commun. Mag.*, vol. 57, no. 8, pp. 84–90,

Aug. 2019.

[429] P. Yang, Y. Xiao, M. Xiao, and S. Li, "6G Wireless Communications: Vision and Potential Techniques," *IEEE Netw.*, vol. 33, no. 4, pp. 70–75, Jul. 2019.

[430] M. Giordani, M. Polese, M. Mezzavilla, S. Rangan, and M. Zorzi, "Toward 6G Networks: Use Cases and Technologies," *IEEE Commun. Mag.*, vol. 58, no. 3, pp. 55–61, Mar. 2020.

[431] H. Abou-zeid, F. Pervez, A. Adinoyi, M. Aljlayl, and H. Yanikomeroglu, "Cellular V2X Transmission for Connected and Autonomous Vehicles Standardization, Applications, and Enabling Technologies," *IEEE Consum. Electron. Mag.*, vol. 8, no. 6, pp. 91–98, Nov. 2019.

[432] Z. Yuan, Y. Ma, Y. Hu, and W. Li, "High-Efficiency Full-Duplex V2V Communication," in *2020 2nd 6G Wireless Summit (6G SUMMIT)*, 2020, pp. 1–5.

[433] X. Chen, S. Leng, J. He, and L. Zhou, "Deep-Learning-Based Intelligent Intervehicle Distance Control for 6G-Enabled Cooperative Autonomous Driving," *IEEE Internet Things J.*, vol. 8, no. 20, pp. 15180–15190, Oct. 2021.

[434] S. Bang and S. Ahn, "Platooning Strategy for Connected and Autonomous Vehicles: Transition from Light Traffic," *Transp. Res. Rec. J. Transp. Res. Board*, vol. 2623, no. 1, pp. 73–81, Jan. 2017.

[435] P. B. Sujit, D. E. Lucani, and J. B. Sousa, "Bridging Cooperative Sensing and Route Planning of Autonomous Vehicles," *IEEE J. Sel. Areas Commun.*, vol. 30, no. 5, pp. 912–922, Jun. 2012.

[436] G. Naik, B. Choudhury, and J. M. Park, "IEEE 802.11bd 5G NR V2X: Evolution of Radio Access Technologies for V2X Communications," *IEEE Access*, vol. 7, pp. 70169–70184, 2019.

[437] A. Festag, "Standards for vehicular communication—from IEEE 802.11p to 5G," *lektrotechnik und Informationstechnik*, vol. 132, no. 7, pp. 409–416, Nov. 2015.

[438] 5GAA, *The Case for Cellular V2X for Safety and Cooperative Driving 5G Automotive Association WHY V2X?* 2016, pp. 1–8.

[439] A. Papathanassiou, Apostolos Khoryaev, *Cellular V2X as the Essential Enabler of Superior Global Connected Transportation Services*. 2017.

[440] S. Chen *et al.*, "Vehicle-to-Everything (v2x) Services Supported by LTE-Based Systems and 5G," *IEEE Commun. Stand. Mag.*, vol. 1, no. 2, pp. 70–76, 2017.

[441] S.-Y. Lien *et al.*, "3GPP NR Sidelink Transmissions Toward 5G V2X," *IEEE Access*, vol. 8, pp. 35368–35382, 2020.

[442] I. Number, "Study on the Deployment of C-ITS in Europe : Summary Report," 2015.

[443] GSMA, "Connected Living: Mobilising the Internet of Things," *Connected Living Seminar*, 2014. [Online]. Available: https://www.gsma.com/iot/gsma_events/connected-living-seminar-mobilising-the-internet-of-things/. [Accessed: 02-Feb-2021].

[444] M.-D. Cano and A. Cañavate-Sanchez, "Preserving Data Privacy in the Internet of Medical Things Using Dual Signature ECDSA," *Secur. Commun. Networks*, vol. 2020, pp. 1–9, Jun. 2020.

Appendix A: Copyright documentation

All images included in this document have been made by the author, under CC license.

Fe de erratas

PhD Thesis: Contribution To Enhancing The Cognitive Capability Of Intelligent Transportation Systems Using Artificial Intelligence

Antonio Guillén Pérez

En la página 6 del documento, abstract, donde se indica: “The main achievements obtained during the development of this thesis are: *i)* research on the interoperability of VACs”.

Debería indicar: “The main achievements obtained during the development of this thesis are: *i)* research on the interoperability of CAVs”.

En la página 47 del documento, apartado 2.3 Artificial intelligence, donde se indica: “Although there is no formal definition of AI [5], for a better understanding of what IA”.

Debería indicar: “Although there is no formal definition of AI [5], for a better understanding of what AI”.

En la figura 5-3, cuando de hace referencia a “Shown in each of the subfigures (a-e)”,

Debería indicar “Shown in each of the subfigures (a-f)”.

En esta misma figura, Speed no debería aparecer en cursive.