UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH

PhD program in Network Engineering

Contribution to the enhancement of IoT-based application development and optimization of underwater communications, by artificial intelligence, edge computing, and 5G networks and beyond, in smart cities/seas

by:

JUAN CARLOS CEPEDA PACHECO

Ph.D. Advisors:

MARI CARMEN DOMINGO ALADRÉN

Thesis submitted to Universitat Politècnica de Catalunya in accordance with the requirements of the degree of DOCTOR OF PHILOSOPHY IN NETWORK ENGINEERING.

Department of Network Engineering

UNIVERSITAT POLITÈCNICA DE CATALUNYA

BARCELONA, NOVEMBER 26, 2023

Abstract

In the current digital era, interconnectivity between devices and processing power has reached a level never seen before. 6G networks have emerged as a revolutionary breakthrough, promising ultra-fast and reliable connectivity that redefines the way we interact with the digital world. This new generation of networks not only drives communication between devices but is also the backbone of the Internet of Things (IoT). In addition, the learning and adaptive capabilities of Artificial Intelligence systems are driving process automation and efficiency. This enables the potential for applications in diverse fields, from healthcare to logistics and manufacturing. Similarly, Edge Computing complements this landscape by decentralizing data processing, bringing computing capacity closer to the sources of information. This allows for reducing latency and improving efficiency by processing data in real-time, driving critical applications that require instantaneous responses.

This Ph.D. thesis focuses on two important points: 1) Improving the efficiency of applications in smart cities, and 2) Enhancing the efficiency of underwater communications in smart coastal cities by applying artificial intelligence, edge computing, and 5G and beyond. To achieve these objectives, an exhaustive study of the existing literature on 5G and beyond networks (architecture and services), smart cities (enabling technologies), and artificial intelligence (applications and use cases) has been carried out. In addition, technical documentation to obtain an updated view of the different technologies that enable the development of applications based on 5G and beyond has been analyzed. Aiming to generate new and innovative alternatives in the field of tourism, security, improved underwater communications, and marine discovery that promote development to meet the needs of citizens in smart cities and ocean/sea. As a result of this study, the first contribution has emerged. It involves the analysis, design, and implementation of a tourist attraction recommendation system employing a deep learning algorithm tailored for smart cities. The primary objective is to reduce the time it may take a user to search for potential places to visit and to improve how recommendations of tourist attractions are made in a given city.

The second contribution arises in surveillance and security, which consists of a distraction detection system for the prevention of drowning in aquatic places, developed in a 5G and beyond network environment. For this goal, an approach of surveillance cameras capturing images of people in charge of minors in swimming pools or beaches was proposed; and employing an ML algorithm (convolutional neural networks) to classify the type of distraction that a person in charge of a minor may have.

Finally, the third contribution is presented, called reinforcement learning and mobile edge computing for underwater wireless networks based on 6G. In this approach, a submerged edge mobile computing architecture is presented in which an AUV is used as a mobile platform (MEC), in addition, several local AUVs equipped with computational resources that collect tasks from sensor nodes and can make the decision to process them locally or partially or fully offload them to the mobile edge computing AUV device (AUV-MEC). To this end, an algorithm based on deep reinforcement learning (DDPG) is proposed for trajectory control, task offloading strategy, and computational resource allocation, combined with mobile edge computing and AUVs to improve underwater communication; aiming to minimize the sum of maximum processing delays and energy consumption during the whole process of executing a task.

The exhaustive simulations performed enable all these contributions to be compared with other previously existing proposals, demonstrating their effectiveness and performance.

The contributions presented in this doctoral thesis are of singular importance, since to date they continue to be innovative. The contributions presented not only represent significant advances in their respective areas but also lay the groundwork for future research and developments in smart city construction and underwater communications optimization, thereby reinforcing the transformative potential of artificial intelligence, edge computing, and advanced wireless networks in these domains.

ACKNOWLEDGMENTS

First, I wish to express my deep gratitude to God. Every step I take and every achievement I reach become blessings since He guides my life favorably. I acknowledge His constant presence, guiding me along the road He has traced for me at every moment.

I would like to thank the Universidad Nacional de Chimborazo and the Secretaría de Educación Superior, Ciencia, Tecnología e Innovación (SENESCYT) of the Republic of Ecuador, thanks to their support, I have been allowed to achieve this accomplishment and allow me to complete this research work.

From the bottom of my heart, I want to thank my parents María Cleofé Pacheco and Carlos Alfonso Cepeda (+), who are always present in my heart and my thoughts, since I was a child knew how to guide my steps and were always by my side supporting me and in their prayers giving me strength to always continue despite the adversities, although at the end of this journey my father is no longer by my side, I know that from heaven he will always be supporting me. To my sister Pauli for her advice, and to my nephews Valeria and Luigi for their best wishes. Special thanks to my wife, friend, and partner Erika Zurita, and my son Gael, who are my driving force to keep going and who bring out the best in me at every moment.

In the academic field, I would like to thank the Universidad Politécnica de Cataluña for opening the academic doors for what was the beginning of this adventure, and I would also like to extend my gratitude and admiration to my supervisor, Prof. Dr. Mari Carmen Domingo Aladrén for her sincere friendship, support, patience, guidance and commitment during all these years of research; thanks to her advice and words of encouragement that kept me focused and motivated in the most difficult moments and thanks also to her recurrent support and contributions that have made this thesis a reality.

To the Ecuadorian friends who welcomed us into their homes, and we were able to share pleasant moments of friendship and companionship.

I would also like to thank all the people we were able to meet during the time we spent in Castelldefels - Barcelona and who somehow influenced our stay in that city and gave us a beautiful memory of all of Spain.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

List of Abbreviations (Acronyms)

2D	Two-Dimensional
3D	Three-Dimensional
3GPP	Third Generation Partnership Project
5G	Fifth-Generation cellular network
5GC	5G Core Network
6G	Sixth-Generation wireless
ADCP	Acoustic Doppler Current Profiler
AE	AutoEncoder
AI	Artificial Intelligence
AMF	Access and Mobility Management Function
AMPS	Advanced Mobile Phone System
AN	Adversarial Network
ANN	Artificial Neural Network
AoI	Age of Information
AR	Augmented Reality
AUSF	Authentication Server Function
AUV	Autonomous Underwater Vehicle
AWGN	Additive White Gaussian Noise
BLE	Bluetooth Low Energy
CB	Content-Based Filtering
CC	Cloud Computing
CDMA	Code Division Multiple Access
CF	Collaborative filtering
CH	Cluster Head
CHF	Charging Function
CNN	Convolutional Neural Network
CP	Control Plane
CPU	Central Processing Unit
CS	Circuit Switching
DDDQN	Double Dueling DQN
DDPG	Deep Deterministic Policy Gradient
DDQN	Double DQN
DF	Demographic Filtering
DI	Directivity Index
DL	Deep learning
DNA	Deoxyribonucleic Acid
DNN	Deep Neural Network
DPG	Deterministic Policy Gradient
DQN	Deep Q Neural Networks
DRL	Deep Reinforcement Learning
DT	Detection Threshold
EC	Edge Computing
EE	Energy-Efficient
eMBB	enhanced Mobile Broadband
E-UTRAN	Evolved UMTS Terrestrial Radio Access Network
FDD	Frequency Division Duplex
FDMA	Frequency Division Multiple Access
FN	False Negatives
FP	False Positives
GBS	Ground Base Station
gNB, or gNodeB	Next-Generation Base Nodes

GPS	Global Positioning System
GPU	Graphics Processing Unit
GRU	Gated Recurrent Units
GSM	Global System for Mobile communication
HD	High Definition
HDCS	Hybrid Data Collection Scheme
IaaS	Infrastructure as a Service
ICT	Information and Communication Technologies
IEEE	Institute of Electrical and Electronics Engineers
IIoT	Industrial Internet of Things
IoT	Internet of Things
IoUT	Internet of Underwater Things
IS-54	Interim Standard-54
IS-95	Interim Standard-95
ITS	Intelligent Transport Systems
kNN	k-Nearest Neighbor
KPI	Key Performance Indicators
LEO	Low Earth Orbit
LoRa	Long Range
LPWAN	Low-Power Wide Area Network
LSTM	Long-Short-Term-Memory
LTE Cat M	Long Term Evolution enhanced Machine Type Communication
MAC	Medium Access Control
MANO	Management and Orchestration
MCC	Mission-Critical Communications
MDP	Markov Decision Process
MEC	Multi-Access Edge Computing
MI	Magnetic Induction
MIMO	Multiple Input Multiple Output
ML	Machine Learning
MLP	Multilayer Perceptron
mMIMO	massive MIMO
mMTC	massive Machine Type Communications
mmWave	millimeter Waves
MTUC	Multi-Level Underwater Computing
NB-IoT	NarrowBand-Internet of Things
NEF	Network Exposure Function
NF	Network Functions
NFV	Network Function Virtualization
NFVI	NFV Infrastructure
NFVO	NFV Orchestrator
NFV	Network Functions Virtualization
NG-RAN	Next Generation Radio Access Network
NLP	Natural Language Processing
NMT	Nordic Mobile Telephone
NN	Neural Networks
NOMA	Non-Orthogonal Multiple Access
NR	5G New Radio
NSA	Non-Standalone
NSI	Network Slice Instance
NSSF	Network Slice Selection Function
OFDMA	Orthogonal Frequency-Division Multiple Access
PaaS	Platform as a Service
PCF	Policy Control Function
PII	Personally Identifiable Information
PoI	Points of Interest
PS	Packet Switching
QoL	Quality of Life

QoS	Quality of Service
RAN	Radio Access Network
RAT	inter-Radio Access Technology
RBM	Restricted Boltzmann Machine
ReLU	Rectified Linear Unit
ResNet	Residual Network
RF	Radio Frequency
RFID	Radio Frequency Identification
RL	Reinforcement Learning
RNN	Recurrent Neural Network
ROV	Remotely Operated Vehicles
RS	Recommender systems
SA	Standalone
SaaS	Software as a Service
SC	Smart Cities
SDN	Software-Defined Networks
SGD	Stochastic Gradient Descent
SL	Source Level
SLA	Service Level Agreement
SMF	Session Management Function
SMS	Short Message Service
SNR	Signal-to-Noise Ratio
SVM	Support Vector Machines
TACS	Total Access Communication System
TDD	Time Division Duplexing
TDMA	Time Division Multiple Access
TN	True Negatives
TP	True Positives
TRS	Tourism Recommendation System
UAC	Underwater Acoustic Communications
UAV	Unmanned Aerial Vehicle
UDM	Unified Data Management
UDR	Unified Data Repository
UE	User Equipment
UMTS	Universal Mobile Telecommunications System
UOWC	Underwater Optical Wireless Communication
UP	User Plane
UPF	User Plane Function
URLLC	Ultra-Reliable Low Latency Communications
USV	Unmanned Surface Vehicle
UWAC	Underwater Wireless Acoustic Communications
UWC	Underwater Wireless Communications
UWSN	Underwater Sensor Networks
V2X	Vehicle-to-Everything
VGG	Visual Geometry Group
VIM	Virtualized Infrastructure Manager
VNF	Virtualized Network Functions
VNFM	VNF Manager
VR	Virtual Reality
WCDMA	Wideband Code Division Multiple Access
Wi-Fi	Wireless Fidelity
WLAN	Wireless Local Area Network
YOLO	You Only Look Once

CHAPTER 1

1 INTRODUCTION

Urbanization is a growing concern. Globally, more people live in urban areas than in rural areas. In 1950, 30% of the world's population was urban. In 2018, 55% of the world's population resided in urban areas and it is projected to reach 68% by 2050 (see Figure 1.1). This continued growth, in line with the overall demographic increase, indicates that the world's urban population will increase by approximately 2.5 billion people in urban areas over the next thirty years. In this context, the imperative to address environmental, social, and economic sustainability through a holistic and comprehensive approach becomes crucial. It’s an essential requirement to stay ahead of the fast urban growth rate, which is currently straining the resources of modern cities. While the world continues to urbanize, successful urban growth management is the primary driver of sustainable development, especially in low-income and lower-middle-income countries, where urbanization is expected to experience its most rapid expansion between now and 2050 [1][2]. Population growth has an undeniable correlation with the tendency toward chaos and disorganization in urban areas. Government agencies recognize the critical role that technology is destined to play as a nexus, particularly in the implementation of technological solutions that foster balanced, safe, efficient, and healthy urban environments.

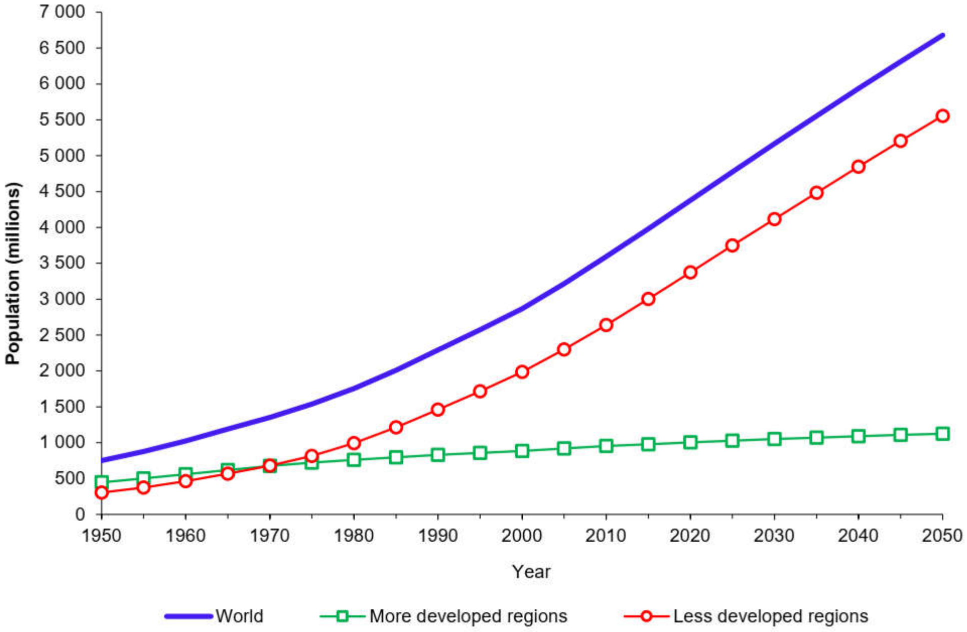


Figure 1.1. Estimated and projected urban populations of the world [2].

Sustainable urbanization requires cities to generate adequate revenues and provide the necessary water and sanitation, energy, transportation, and communications infrastructure. To address these demands, cities are increasingly integrating Information and Communication Technologies (ICT) into their development strategies. Likewise, the integration of the Internet of Things (IoT) infrastructure [3], which is a unique dynamic "network" of interconnected smart objects [4], with self-configuration capabilities, allows interaction and communication between humans and objects anytime, anywhere, and with anybody or anything. Furthermore, IoT can generate an unprecedented volume of events and data, commonly known as Big Data, which can be sent to the cloud for further analysis. Combined with Artificial Intelligence (AI), this data can be harnessed to develop software applications that gather information from both citizens and urban infrastructures [5]. The aim is to efficiently manage information to enhance the basic needs of citizens, businesses, and institutions, transforming cities into Smart Cities (SC). The successful deployment of a smart city application is based on several essential components, the first of them being wireless connectivity, which must be reliable and ubiquitous to ensure optimal performance. Although there is no one-size-fits-all solution, Low-Power Wide Area Network (LPWAN) technologies emerge as the ideal choice for most applications, due to their cost-effectiveness and ubiquity. Among the most high-profile LPWAN technologies are Long Term Evolution enhanced Machine Type Communication (LTE Cat M), NarrowBand-Internet of Things (NB-IoT), Long Range (LoRa), and Bluetooth which provide robust and efficient connectivity [6]. Significant milestones to promote the evolution of mobile communications are the continuous development of 5G technology and the future arrival of 6G. These new technologies promise not only extremely fast data transmission speeds, but also reduced latency, massive device connectivity, and ultra-high reliability [7]. Also, 6G is expected to use a more advanced frequency spectrum that would increase data throughput 100 to 1000 times faster than 5G [8]. With even faster transmission speeds, near-zero latency, and the ability to support a variety of advanced applications, connectivity is elevated to an unprecedented level [9]. These technological advances become a key driver in the economic and social sectors. They enable new innovative applications and accelerate the deployment of more advanced and efficient solutions. Cities worldwide are adopting AI as a fundamental tool for real-time analysis of urban data, aiming to obtain highly useful information [5]. An emerging trend in this area is the generation of three-dimensional simulations, known as "digital twins", capable of representing a highly detailed virtual replica of a city in its current state [10][11]. The main goal is to explore and evaluate hypothetical scenarios of how they could look in the future in several aspects such as infrastructure development, traffic improvement and sustainable mobility, more efficient public services, safe and protected territories, environmental care, education, culture, and tourism, among others [12]. Furthermore, in smart coastal cities, the combination of the Internet of Underwater Things (IoUT) and AI enables the monitoring of coastal ecosystems by collecting accurate data such as water temperature, salinity [13], and various environmental variables. In particular, data obtained from IoUT sensors can be used to detect noticeable alterations in environmental conditions, such as variations in water temperature or pH levels. These variations can be clear indicators of the presence of pollutants or harmful substances in the marine ecosystem. The implementation of AI is an essential resource in the detection of changes in the abundance of marine life. This is particularly valuable in conservation terms, as it enables, to identify early the decrease of populations of endangered species. The deployment of AI-enabled cameras and sensor systems is advocated as an effective strategy for identifying illegal fishing, monitoring human activities that may harm coastal ecosystems, and assessing the impact of tourism in fragile environments [14]. Overall, AI is a remarkably effective tool for monitoring and protecting urban areas and coastal ecosystems. Its ability to collect and analyze data related to environmental conditions and human activities enables the early identification of areas at risk, providing an informed basis for the implementation of preventive measures. This proactive approach contributes to the mitigation of negative impacts, thereby promoting the long-term sustainability and preservation of these environments.

New challenges arise with each technological advance. Currently, there are new applications that require low latency such as Virtual Reality (VR), which is an advanced human-computer interface that simulates a realistic environment [15], and Augmented Reality (AR), which integrates virtual information with the real world [16]. Furthermore, mission-critical applications, need very low latency communications, very high reliability, and high availability, since they require immediate responses, in high-risk situations. Similarly, real-time applications, are capable of monitoring and responding immediately to user requirements or controlling an external environment [17] such as public safety, environmental care, traffic monitoring, and autonomous vehicles, among others. Edge Computing (EC) has emerged as a fundamental element in the context of smart cities to address these needs, providing effective and quick responses to the demands of real-time solutions [18].

Edge Computing currently represents a vitally important technological paradigm. This approach is based on the fundamental premise of storing data as close as possible to its origin, which leads to significant benefits in terms of bandwidth efficiency and minimization of response time [19]. This model of processing information as close as possible to the user's physical location, for an agile response, has become a key element for the development of this sort of infrastructure in public and private organizations. One of the most outstanding characteristics of Edge Computing lies in its integration with Machine Learning (ML) based systems, allowing it to constantly learn and evolve towards the future at the same speed as human beings. The functionality of a peripheral network brings several substantial benefits in the context of information and communication technologies. This approach enables interconnected devices to provide real-time responses and make autonomous decisions, as opposed to the previous paradigm that required sending information from one point to another, pending authorization.

The seamless interconnection between Edge Computing and 5G networks is essential for their optimal operation and to reach the full potential of both technologies. In the Big Data context, where the quantity of information can be overwhelming, Edge Computing performs a crucial function by identifying, filtering, and forwarding to the cloud only relevant data for processing. The capability of continuous learning through ML and AI algorithms enables devices to refine their decisions intelligently over time, dynamically adapting to patterns and changes in the environment. This continuous enhancement in decision-making contributes to the operational efficiency and resource optimization of urban areas [20]. For this reason, the development of new applications based on IoT, IoUT, and AI will always be a great contribution to the development of society, therefore this research contributes to the development of Smart Cities generating proposals for current scenarios, as well as for future work.

1.1 Motivation

The motivations for the particular development of this thesis are as follows:

- Improving the efficiency of applications in smart cities using artificial intelligence, edge computing, and 5G and beyond.
- Improving the underwater communication efficiency in smart coastal cities using artificial intelligence, edge computing, and 5G and beyond.

These motivations are described in more depth below.

1.1.1 Motivation 1: Improving the efficiency of applications in smart cities using artificial intelligence, edge computing, and 5G and beyond.

Application development is a continuous activity for technological evolution. Each time new and better applications appear aiming to deliver new experiences to the user. Therefore, information and communication technology has emerged as a fundamental component in the development and evolution of cities, an urban paradigm that intends to improve the quality of life for citizens through the application of advanced technological solutions. The deployment of these solutions is always a challenge. With the emergence of new technologies (IoT, AI, 5G and beyond) more efficient, reliable, and sustainable solutions can be developed.

Therefore, the present research has focused on designing and improving the efficiency of new and innovative deep learning-based applications for the development of smart cities. These applications integrate new technologies such as AI, edge computing, and 5G networks and beyond. AI is considered to play an essential role in application optimization. AI algorithms can analyze large data sets in real-time to manage fields such as traffic and mobility, energy management, security, healthcare, culture, and tourism. Meanwhile, Edge Computing, enables data processing and application execution close to the source, reducing latency and improving processing speed. This capability provides rapid responses to emergency situations, such as traffic accidents or natural disasters. Furthermore, 5G technology provides ultra-fast and reliable connectivity, facilitating seamless communication between devices and systems. This capability is essential for critical applications, such as autonomous vehicles, health monitoring systems, environmental sensors, etc. Overall, these technologies contribute to boosting the efficiency of applications as they contribute to decision-making and better adaptation to the changing needs of an ever-evolving city.

1.1.2 Motivation 2: Improving the efficiency of underwater communications in smart coastal cities, through artificial intelligence, edge computing, and 5G and beyond.

Underwater communications are a complex task and have increasingly led to new researchers becoming interested in this field, introducing new challenges. The seabed can be a hostile and dangerous place. Therefore, human beings have developed many tools and mechanisms to gain access to it. In contrast to terrestrial communications, underwater communications are faced with different issues such as bandwidth, latency, and jitter, which are factors that affect their performance [21]. Also, energy consumption savings emerge as a critical factor [22], since recharging underwater sensors and devices can be problematic and expensive. In multiple underwater scenarios integration of new technologies such as (IoUT, AI, 5G and beyond) allows the information collected on the seabed to be stored in cloud servers for further processing. This integration may lead to new challenges and discoveries.

Based on the aforementioned, the development of alternatives to improve the efficiency of underwater communications is a challenge that has also been the focus of the development of this research and that remains largely unexplored. In this framework, the implementation of artificial intelligence algorithms is presented as a strategy for the analysis of collected data and the optimal management of the resources of underwater devices. Likewise, the integration of edge computing enables the execution of computational operations close to the requesting devices. This approach reduces latency compared to cloud processing. It also aims to improve the efficiency of subsea communications and provide agile solutions to applications that require immediate responses.

1.2 Objectives

The central objective of this PhD thesis is to boost the implementation of IoT-based applications and the optimization (improvement) of underwater communications, by artificial intelligence, edge computing, and 5G networks and beyond, contributing to the technological development in smart cities and ocean/seas.

To meet the stated core objective, we will divide it into two specific objectives.

- Improving the efficiency of applications in smart cities using artificial intelligence, edge computing, and 5G and beyond.

- Improving the efficiency of underwater communications in smart coastal cities using artificial intelligence, edge computing, and 5G and beyond.

The following specific objectives have been identified to achieve the above two objectives.

- To analyze existing literature on 5G networks and beyond (architecture and services), smart cities (enabling technologies), and artificial intelligence (applications and use cases). Collect technical documentation to get an updated overview of the different technologies that enable the development of 5G-based applications in smart cities.
- To perform a comprehensive analysis of existing IoT-based applications that integrate artificial intelligence and 5G networks and beyond. Review relevant publications, to generate new and innovative alternatives in the field of tourism, aquatic childcare, and marine discovery that boost development to meet the needs of citizens in smart cities and Oceans/seas.
- To generate innovative strategies and solutions fundamental to the operation of smart cities and oceans. Conduct research, design, and development of applications that manage data and provide essential information for decision-making and efficient resource management.
- To design modern, adaptive, and intelligent ecosystems by developing applications that integrate artificial intelligence algorithms. Improve data analysis capabilities, automate processes, and enable more informed decision-making in critical situations.
- To evaluate the performance of such solutions using the capabilities of 5G networks and exploring emerging technologies beyond 5G, such as 6G and edge computing. To ensure ultra-fast and reliable connectivity and computing in all contexts, from urban environments to remote maritime areas.

1.3 Summary of Contributions

The main contributions of this thesis are described below, based on the proposed objectives.

- **A tourist attraction recommendation system for smart cities based on deep learning.** This system recommends tourist attractions in two scenarios, 1) when the tourist is outside the city and 2) when the tourist is already in the target city. To this end, user-generated information is collected, which takes into account the particular circumstances of a trip (traveling alone or with a companion, type of companion such as couple, family with children, etc.), as well as user information (age of the traveler/s), hobbies, etc.) to improve the accuracy of the recommendations. According to this perspective, in our research, we rely on a hybrid Tourism Recommendation System (TRS), since each technique, when used separately, has its strengths and weaknesses, but by combining and complementing them with deep learning techniques, we can generate a recommendation system that meets a broader range of user needs. Furthermore, unlike existing applications, this research is not limited to recommendations within a single establishment, environment/event but to a city as a whole. Additionally, it provides on-site tourist attraction recommendations in real-time by recording already visited tourist attractions and using IoT context-related information such as location, availability, and weather forecast. To our knowledge, this is the first recommendation system that makes real-time tourist attraction recommendations based on IoT context-related data and deep learning.

- **Distraction detection system for child drowning prevention, developed in a 5G and beyond network environment.** This system is based on deep learning, which detects and classifies distractions of inattentive parents or caregivers. This approach can be deployed in indoor pools or outdoor locations such as beaches or aquatic recreation sites with the help of unmanned aerial vehicles (UAVs) (drones). The system detects distracted parents/caregivers in charge of a child and alerts them to focus on the monitoring task. For this purpose, we have first focused on generating our own dataset of images of distracted parents/caregivers in a swimming pool. The current applications are based on drowning detection, by searching for and detecting moving objects in the water. When an object does not move, it is detected as drowned, hence, these applications are limited to detecting victims that have sunk to the bottom of the pool and are static in the same location. Although this task remains challenging due to disturbances on the water surface (e.g., water exhibits random and homogeneous bubble movements, which could be easily identified as foreground objects), some applications deal with swimmer detection and tracking that attempt to analyze swimmer observation sequences for possible signs of drowning behavior. The present research differs from previously developed applications by proposing an early warning system that employs surveillance cameras supported by machine learning algorithms that enable accurate and fast detection of possible distractions. When a distracting situation is detected, the system automatically generates alarms or alerts that are immediately transmitted to caregivers or persons in charge of supervising children present in the aquatic environment. To our knowledge, this is the first work that aims to prevent child drowning by detecting and classifying distractions of parents in charge of a child in aquatic recreational areas; it is also the first work that uses digital technologies such as artificial intelligence and modern communication technologies (such as 5G and beyond) to detect and alert distracted parents or caregivers.

- **Development of a Reinforcement Learning algorithm and mobile edge computing for 6G-based underwater wireless networks.** In this approach, we propose a deep reinforcement learning-based algorithm for trajectory control, task offloading strategy, and computational resource allocation. This algorithm is combined with mobile edge computing and autonomous underwater vehicles to improve underwater communication. The RL algorithm is responsible for deciding the direction and dive speed of the AUVs, the task offloading strategy, and the resource allocation of the MEC server. We choose the overall energy consumption and delay as a reward for the proposed algorithm. There are several studies on the data acquisition step in the marine environment. In some cases, the information is sent directly from the sensor nodes to the sink nodes through acoustic signals and then, in the airborne environment, the data are collected by UAVs through electromagnetic wave propagation. In these approaches, neither delay nor power consumption is considered, simply the objective is to collect information. To reduce the energy consumption of resource-limited devices, the use of aquatic mobile devices (AUVs) is introduced to reduce the transmission distance between the sensor nodes and the AUV, as well as between the AUV and the sink nodes (sea surface). However, since the main objective is only to collect data, it will take time for the information to reach the cloud-based servers and be processed. There is also research that is based on data acquisition, but instead of sending the data to the cloud directly, it is sent to an edge server; in some cases, to AUVs and in other cases to intelligent ocean convergence platforms (ships and buoys) for edge computing purposes. However, these investigations are only concerned with data collection, they do not analyze the offloading of computation, i.e., the desirability of running the computational tasks locally or offloading them (partially or fully) to a MEC server. In the field of edge computing, there is research presenting a network architecture that integrates space-air-land-sea, considering the requirements of edge computing and cloud computing, by implementing RL techniques for the intelligent allocation of joint resources. For this purpose, the information collected by ships and buoys is sent to the MEC server assisted by UAV or low earth orbit (LEO) satellite. Similarly, investigations are proposed in which AUVs are used as information-passing devices from the sensor nodes or the information is sent directly to the surface station (MEC server), which would imply a higher energy consumption. Although these approaches employ data offloading to edge servers located on UAVs - satellites, or at the surface station, in our case we try to bring the edge server (MEC AUV) even closer to the data collecting devices (intermediate AUVs). In this proposed work, based on mobile edge computing, we propose a joint optimization scheme for automatic (full or partial) task offloading, path optimization, and resource allocation, using RL techniques (DDPG algorithm). To the best of our knowledge, this is the first work that proposes a MEC system with AUVs that include IoUT devices, cluster heads, local AUVs, and MEC AUVs. The joint optimization of AUV trajectories, offloading strategy, and resource allocation in an AUV-assisted MEC considering energy efficiency and delay minimization has not been investigated.

1.4 Related Publications

This section lists the publications produced as a result of this research.

Journal Publications:

J1 Cepeda-Pacheco, Juan Carlos; Domingo, Mari Carmen. Deep learning and Internet of Things for tourist attraction recommendations in smart cities. Neural Computing and Applications, 2022, vol. 34, no 10, p. 7691-7709.

J2 Cepeda-Pacheco, Juan Carlos; Domingo, Mari Carmen. Deep learning and 5G and beyond for child drowning prevention in swimming pools. Sensors, 2022, vol. 22, no 19, p. 7684.

J3 Cepeda-Pacheco, Juan Carlos; Domingo, Mari Carmen. Reinforcement Learning and Mobile Edge Computing for 6G-Based Underwater Wireless Networks (submitted for publication to a journal)

1.5 Thesis outline

This thesis is composed of five chapters, which are summarized in Figure 1.2 and described below.

Chapter 1: Introduction.

This chapter presents the introduction, motivation, and objectives for the development of this research work. It provides an overview of the publications developed and a brief description of each of the chapters.

Chapter 2: Background Technologies:

This chapter introduces a description of the key concepts related to the most relevant topics. This overview serves as a source of discussion, enabling the development of new insights and future work. The basis for this analysis is an exhaustive review of existing publications, as well as the study of standards and the compilation of consolidated information.

Chapter 3: Deep learning, Edge computing, and 5G and beyond for smart cities.

This chapter presents the contributions of IoT-based applications that integrate deep neural networks in Edge computing and 5G scenarios and beyond, aiming to meet the requirements of smart cities. We have divided this chapter into two sections:

Deep Neural Network: In this first section the current applications regarding smart tourism are discussed and the first contribution of the present PhD thesis "Internet of Things for tourist attraction recommendations" is presented. A theoretical explanation is developed and detailed simulations of this contribution are also presented. Furthermore, a comparison with other algorithms is conducted to check its effectiveness.

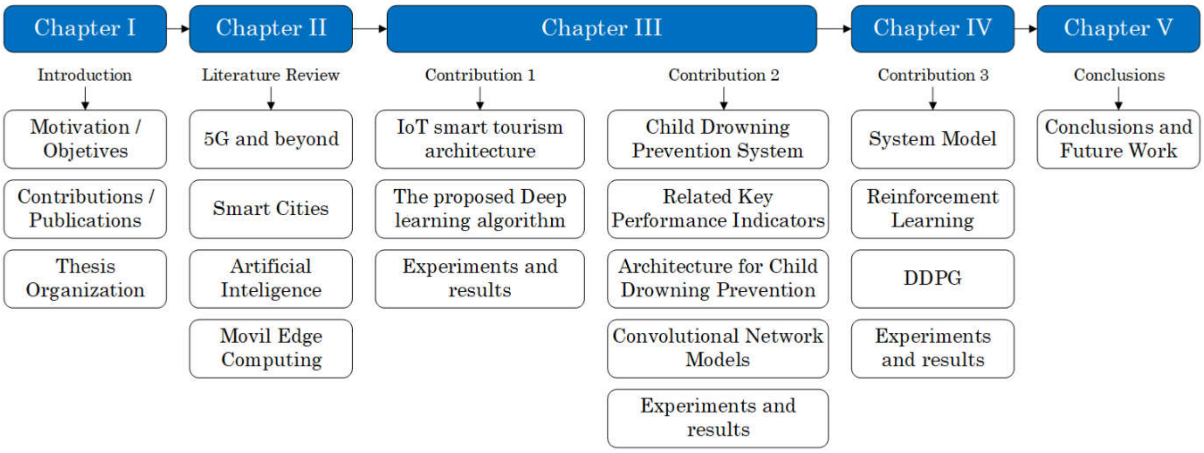
Convolutional Neural Networks: In this second section the current applications regarding drowning prevention and detection are discussed. The second contribution "5G and Beyond for Child Drowning Prevention in Swimming Pools" is presented in detail. A theoretical explanation is developed, and the implementation results are presented. Furthermore, a comparison with other convolutional algorithms is conducted to verify the accuracy of the results.

Chapter 4: Reinforcement learning, Edge computing, and 5G and beyond for smart cities.

This chapter presents a third contribution that integrates Reinforcement learning algorithms, in Edge computing and 5G scenarios and beyond. This section begins by defining the current state of research, followed by a detailed description of the development of the present contribution. This chapter also presents the results of the experiments, highlighting the improvement of underwater communications for smart coastal cities.

Chapter 5: Conclusions and future work.

This chapter presents the conclusions derived from this research. These conclusions are based on the results obtained throughout each of the chapters previously addressed. Furthermore, reflections and suggestions are included, oriented towards future works that will lead to the advancement of technologies and applications for smart cities and coastal cities, with the main purpose of improving the quality of life of their residents.


Chapter I
Introduction
Motivation / Objetives
Contributions / Publications
Thesis Organization

Chapter II
Literature Review
5G and beyond
Smart Cities
Artificial Inteligence
Movil Edge Computing

Chapter III
Contribution 1
IoT smart tourism architecture
The proposed Deep learning algorithm
Experiments and results
Contribution 2
Child Drowning Prevention System
Related Key Performance Indicators
Architecture for Child Drowning Prevention
Convolutional Network Models
Experiments and results

Chapter IV
Contribution 3
System Model
Reinforcement Learning
DDPG
Experiments and results

Chapter V
Conclusions
Conclusions and Future Work

Figure 1.2. Thesis organization.

CHAPTER 2

2 BACKGROUND AND STATE OF THE ART

2.1 5G and Beyond

The fifth generation of mobile technology, commonly referred to as 5G, has emerged as a technological breakthrough of singular relevance in the contemporary digital era and is a huge collective effort to specify, standardize, design, manufacture, and deploy the next generation of cellular networks [23]. In an environment characterized by an increasing reliance on wireless connectivity and a wide proliferation of smart devices, the deployment of 5G represents a significant evolution of mobile communications. This new technology promises not only exceptionally fast data transmission speed but also remarkably reduced latency, massive device connectivity capability, and the enablement of new emerging applications that integrate communication between sensors, machines, or different devices; most importantly, it can support machine-critical communications with instantaneous action and ultra-high reliability [7], these features become fundamental to the economic and social sector as depicted in Figure 2.1.

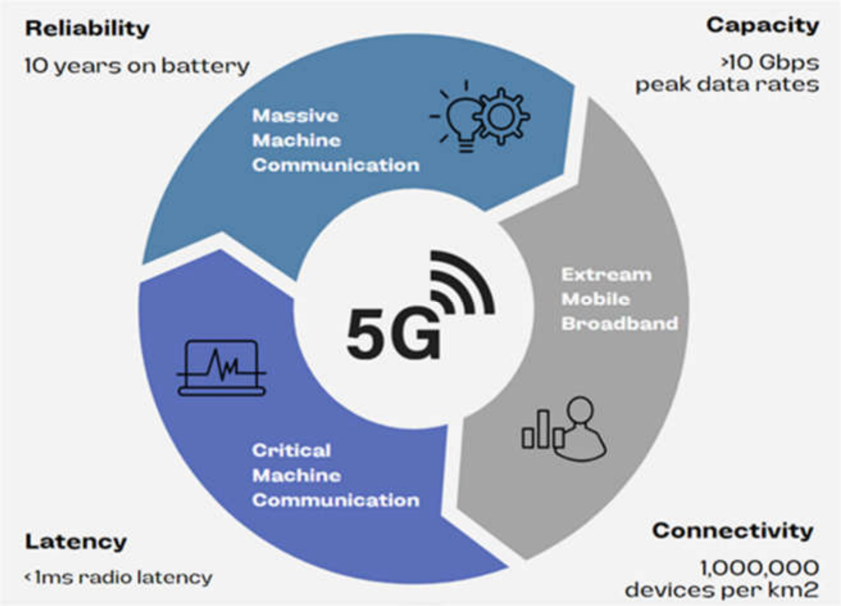


Figure 2.1. 5G key characteristics.

In the next section, the technological foundations of 5G, its architecture, and applications in different fields will be discussed. Ethical, regulatory, and security considerations related to its implementation will also be addressed. This analysis aims to summarize the transformational impact of 5G technology on society and build a foundation for understanding the sixth generation of mobile technology, known as 6G. This new generation promises even faster, more reliable, and more intelligent connectivity, which will not only change the way we communicate but also the way we interact with our environment. In this framework, 5G encourages us to explore the future of communications, where the possibilities are endless, and technology continues to challenge the limits of what is possible.

2.1.1 5G Technology

5G technology is based on many technological foundations and architectural principles that differentiate it from its predecessors (2G, 3G, and 4G) (see Table 2.1). These foundations are essential to understanding the evolution of 5G and its outstanding capabilities. The following are the underlying fundamentals that are the basis of 5G technology.

Table 2.1. Different generations of mobile communications [24].

Cellular Generations	Standards	Applications	Multiple Access Techniques	Physical Resources	Duplex Methods	Switching Techniques
1G	AMPS, NMT, TACS	Voice calls with analog signals	FDMA	Frequency	FDD	CS
2G	GSM, IS-54, IS-95	Voice services with digital signals, SMS. From 2.5G, email and web-browsing.	TDMA	Time slots	FDD	CS, 2.5G – CS & PS
3G	WCDMA, CDMA2000	Mobile TV, video telephony, and video conference	CDMA	Time slots/ PN codes	FDD/ TDD	CS & PS
4G	LTE	High data rate applications, e.g. HD TV, cloud computing, video gaming, etc.	OFDMA	Time/ Frequency	FDD/ TDD	PS
5G	NR	IoT, massive broadband, smart city, VR, AR	OFDMA	Time/ Frequency	FDD/ TDD	PS

Broader frequency spectrum. The frequency spectrum in 5G technology is greatly diverse, comprised of three key categories: low-frequency bands (Sub-6 GHz) that provide wide coverage and efficient signal propagation, Mid-frequency Bands (Mid-Band) that balance speed and signal coverage, and millimeter Waves (mmWave) that are highlighted by their capability to provide extremely high data rates that can reach about 10 Gbit/s and even increase with full-duplex capability [25]. Collectively, these categories enable advanced wireless connectivity that meets a wide range of needs and applications. In particular, mmWave offers significant advantages by providing exceptionally wide bandwidth and ultra-fast data rates. Although limited in range and subject to physical obstacles, they are ideal for high data-density applications and dense urban environments [26]. By reducing the network congestion in urban areas, mmWave not only improves Quality of Service (QoS) but also fosters new business opportunities, such as private 5G networks for enterprises and the Industrial Internet of Things (IIoT), driving innovation and digital transformation in diverse industries.

Advanced Massive MIMO technology. massive Multiple Input, Multiple Output (MIMO) or mMIMO in 5G technology is a continuation of MIMO technology and represents a fundamental innovation in wireless communications. It is characterized by the deployment of a large array of antennas on both base stations and user devices, enabling the simultaneous transmission and reception of multiple data streams. This technology improves network capacity and spectral efficiency by enabling targeted beamforming to specific devices, reducing interference, and improving the quality of service [27].

Reduced latency. In 5G technology reduced latency is a crucial advancement that enables virtually instantaneous communication between devices, this has been achieved through the deployment of more efficient network architectures, advanced signal processing technologies, and the use of massive MIMO antennas, resulting in faster and more efficient data transmission between devices and base stations [28]. The advantages are significant, as reduced latency enables real-time applications, such as telemedicine and mission-critical applications (see Table 2.2), improving quality of life and safety. In addition, it enables the interconnection of devices on the Internet of Things driving automation and efficiency in various sectors, from industry to entertainment.

Table 2.2. Mission-critical services that demand different latency and data rate requirements [29].

Use Case	Latency	Data rate	Remarks
Factory Automation	0.25 – 10 ms	1 Mbps	- Generally, factor automation applications require small data rates for motion and remote control. - Applications such as machine tools operation may allow latency as low as 0.25 ms.
Intelligent Transport Systems (ITS)	10 – 100 ms	0 – 700 Mbps	- Road safety of ITS requires latency on the order of 10 ms. - Applications such as virtual mirrors require data rates on the order of 700 Mbps.
Robotics and Telepresence	1 ms	100 Mbps	- Touching an object by palm may require latency down to 1 ms. - VR haptic feedback requires data rates on the order of 100 Mbps.
Virtual Reality (VR)	1 ms	1 Gbps	- Hi-resolution 360° VR requires high rates on the order of 1 Gbps while allowing a latency of 1 ms.
Health Care	1 – 10 ms	100 Mbps	- Tele-diagnostic, telesurgery, and tele-rehabilitation may require latency on the order of 1 ms with a data rate of 100 Mbps.
Serious Gaming	1 ms	1 Gbps	- Immersive entertainment and human interaction with high-quality visualization may require a latency of 1 ms and data rates of 1 Gbps for high performance.

Smart Grid	1 – 20 ms	10 – 1500 Kbps	- Dynamic activation and deactivation in a smart grid require latency on the order of 1 ms. - Cases such as wide area situational awareness require data rates on the order of 1500 Kbps.
Education and Culture	5 – 10 ms	1 Gbps	- Tactile Internet-enabled multi-modal human-machine interface may require latency as low as 5 ms. - Hi-resolution 360° and haptic VR may require data rates as high as 1 Gbps.

Dynamic radio topology. It plays a critical role in 5G-enabled mass connectivity by providing efficient management of network resources (power and signaling), real-time adaptation to the changing device population, and latency optimization. This ensures reliable and efficient connectivity in environments with a large number of connecting devices [30].

Network virtualization. 5G technology provides a flexible and manageable network infrastructure. This is made possible through Network Functions Virtualization (NFV) and Software-Defined Networks (SDNs). On the one hand, NFV facilitates the building of logical networks on a shared physical infrastructure, thereby enabling the dynamic allocation of resources according to the changing needs of diverse applications in 5G [31]. On the other hand, SDN focuses specifically on the management and control of the radio layer in wireless networks and enables centralized and programmable network management, which is crucial for rapidly adapting to evolving service and application demands [32]. Both technologies enable more efficient and agile network orchestration, ensuring optimal performance and exceptional scalability, which are key to the success of 5G in an increasingly connected and service-diverse world.

Boosting innovation. Considering that not all use cases in 5G technology require identical levels of performance and functionality, this technology must move away from a monolithic framework based on the most demanding requirements, since this approach would result in prohibitive costs. Therefore, 5G must inherently incorporate flexibility and scalability into its capabilities, which will enable it to encompass a wide variety of use cases and promote innovation through multiple business models and strategic alliances.

Leveraging the benefits of virtualized and programmable networks, 5G functions need to be designed in a modular approach to enable on-demand deployment according to specific needs. Therefore, 5G is envisioned as a network characterized by polymorphism, driven by advanced and enhanced radio access technologies, network functions that can be flexibly deployed, and comprehensive end-to-end orchestration [30].

Security and privacy. 5G security and privacy become critically important, especially in a virtualized context. With NFV, network functions are executed in virtualized environments, which increases flexibility and operational efficiency; however, this virtualization also introduces new security challenges. Ensuring security in 5G virtualization is essential to protect against potential vulnerabilities and cyber threats that could affect both service integrity and user privacy [33]. Therefore, security in 5G is not only important for network integrity and data protection but is also essential to support confidence in virtualization and ensure the continuity of critical services and ongoing innovation. [34].

Energy efficiency. Addressing energy efficiency is essential in 5G technology due to the potential significant energy consumption that its implementation involves. This requires strategies that include the adoption of energy-efficient equipment and components, intelligent resource management, implementation of energy-saving techniques, network functions virtualization, optimization of the radio layer, and renewable energy sources [35].

2.1.1.1 Overall 5G System Architecture

The 5G system architecture stands as a highly sophisticated and complex technological framework that provides the infrastructure for fifth-generation mobile communications. This meticulously designed architecture is essential to enable the distinctive capabilities of 5G, including high data rates, low latency, and massive device connectivity as we mentioned above. In this framework, the description of the overall 5G system architecture is presented as a detailed analysis of its key components and intrinsic functionality. The 5G network architecture according to the Third Generation Partnership Project (3GPP) Release 16 is detailed below.

2.1.1.1.1 5G Access Network

The 5G access network, known as the Next Generation Radio Access Network (NG-RAN), is essential in fifth-generation communications. Its main purpose is to facilitate high-speed wireless connections between millions of wireless devices and the core network, enabling data transmission at a significantly higher speed than in 4G networks. Diversified use cases, as well as QoS requirements such as reliability, latency, data rate, security, and privacy, make NG-RAN key to the successful deployment and operation of 5G networks [36].

5G networks operate in a variety of frequencies, including millimeter wave and sub-6 GHz bands, which balance speed and range. To ensure optimal performance, they employ densely distributed base stations called Next-Generation Base Nodes (gNB, or gNodeB), with more antennas than 4G networks. In addition, massive MIMO technologies, which involve multiple antennas, enable simultaneous communication with multiple devices, improving network coverage and capacity [37].

5G network evolution implies the transition from 4G networks; therefore, the 5G architecture integrates two Radio Access technologies that are important in its development. On the one hand, there is the Evolved UMTS Terrestrial Radio Access Network (E-UTRAN), which is directly inherited from LTE technology, and on the other hand, the so-called 5G New Radio, these two access networks enable it to operate in both "non-standalone" mode (NSA) and "standalone" mode (SA), respectively (see Figure 2.2). In SA operation, the gNB is connected to the 5G core network (5GC); in NSA operation, NR and LTE are tightly integrated and connected to the existing 4G core network, leveraging dual connectivity to the terminal. Both NSA and SA architecture options are specified as part of the 3GPP Phase 1 5G standards. This upgrade will enable more efficient use of 5G capabilities and provide greater flexibility and efficiency in data transmission. 5G access networks are critical to enabling high-demand applications and real-time connections in the fifth-generation communications ecosystem [38].

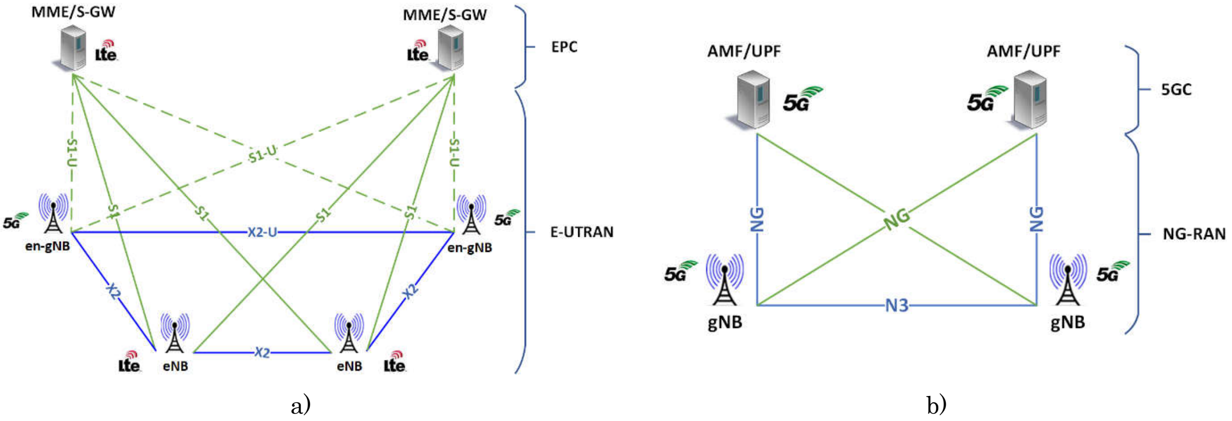


Figure 2.2. a) The NSA architecture, b) The SA architecture.

2.1.1.1.2 5G Network Core

The main components of the 5G network core (according to 3GPP) are essential elements that collaborate to manage and provide connectivity in the 5G network. These include the Access and Mobility Management Function (AMF) that controls device authentication and mobility, the Session Management Function (SMF) that establishes and manages communication sessions, and the User Plane Function (UPF) that handles user data transfer [39]. In addition, the Unified Data Management (UDM) stores user information and service profiles, and the Authentication Server Function (AUSF) authenticates users. Other components include the Network Slice Selection Function (NSSF) for managing custom network segments, the Policy Control Function (PCF) that controls QoS, the Network Exposure Function (NEF) that enables exposure of network capabilities to third-party applications, the Charging Function (CHF) for cache management, and the Unified Data Repository (UDR) for storing and managing user data. These components work together to support connectivity, security, service customization, and efficiency in the 5G network [40].

The modular and flexible architecture of the 5G network (see Figure 2.3) allows adaptation to a wide range of use cases and applications, making it a versatile platform for providing advanced and personalized communication services to users.

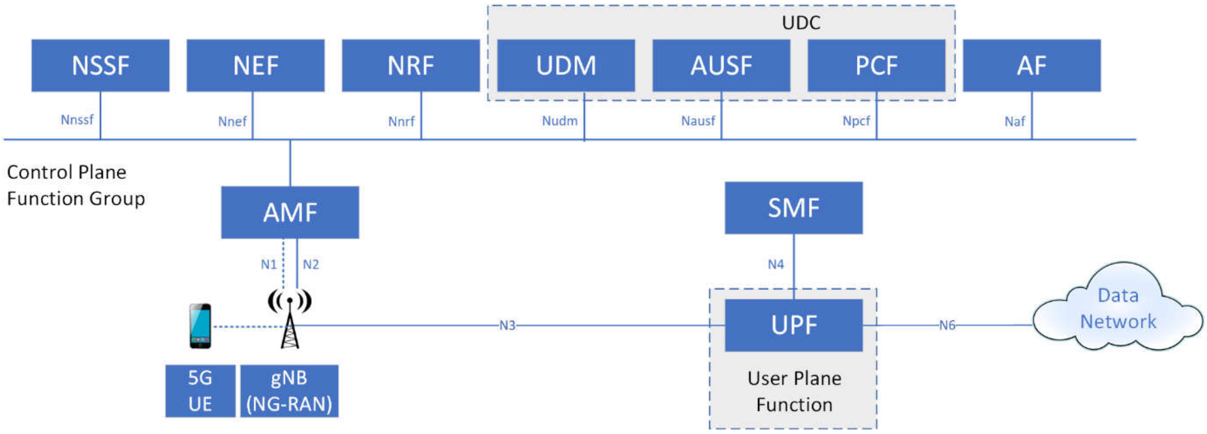


Figure 2.3. 5G network core architecture.

2.1.1.2 5G Technology Use Cases

As previously mentioned, 5G technology represents an important step in the evolution of mobile communications by providing a diverse ecosystem of applications and services. To address connectivity demands and understand its full potential, it is essential to explore the three fundamental categories of use cases presented by the 3GPP. Each of these categories addresses specific connectivity needs and leads to innovations in a wide range of areas such as healthcare, manufacturing, transportation, and entertainment [41](see Figure 2.4). Through this exploration, we will understand how 5G is revolutionizing communications and providing advanced solutions to meet the current and future demands of an increasingly interconnected society. Next, three use cases with their features and enabling services are described, namely Ultra-Reliable Low Latency Communications (URLLC), enhanced Mobile Broadband (eMBB), and massive Machine Type Communications (mMTC).

Ultra-Reliable Low Latency Communications is an essential component of the 5G network due to its capability to provide extremely fast connectivity and minimal latency. This enables a wide range of critical real-time applications, from remote surgery and industrial automation to connected transportation or Vehicle-to-Everything (V2X) mission-critical applications in areas such as healthcare and public safety [42]. URLLC is essential for improving safety, efficiency, and quality of life in diverse sectors, enabling near-instantaneous actions and decisions in a multitude of contexts, and is a major driver of the evolution of mobile communications in the age of 5G [43].

Enhanced Mobile Broadband in the 5G network is a key technical feature that provides high-speed, high-capacity connectivity. It is accomplished by using a wider frequency spectrum, advanced modulation technologies, and massive MIMO antennas, enabling data rates of several gigabits per second and extremely low latency. Moreover, eMBB incorporates techniques such as carrier aggregation and beamforming to optimize spectrum usage and improve signal quality [44]. This not only benefits end users with fast downloads and high-quality virtual reality experiences but also enables business and industry applications that require reliable, high-performance connectivity, driving innovation and digital transformation in diverse sectors.

Massive Machine Type Communications in the 5G network is characterized by the capability to efficiently connect a wide range of Internet of Things devices on a large scale. This is accomplished through advanced multiplexing and modulation techniques, enabling thousands of devices to transmit small data simultaneously in a shared frequency spectrum [45]. In addition, mMTC benefits from efficient medium access protocols, such as enhanced random access, which minimize data collisions in multi-device environments. Such efficiency is critical for applications that require real-time monitoring and control of IoT devices dispersed over large areas, such as industrial automation and resource management [46].

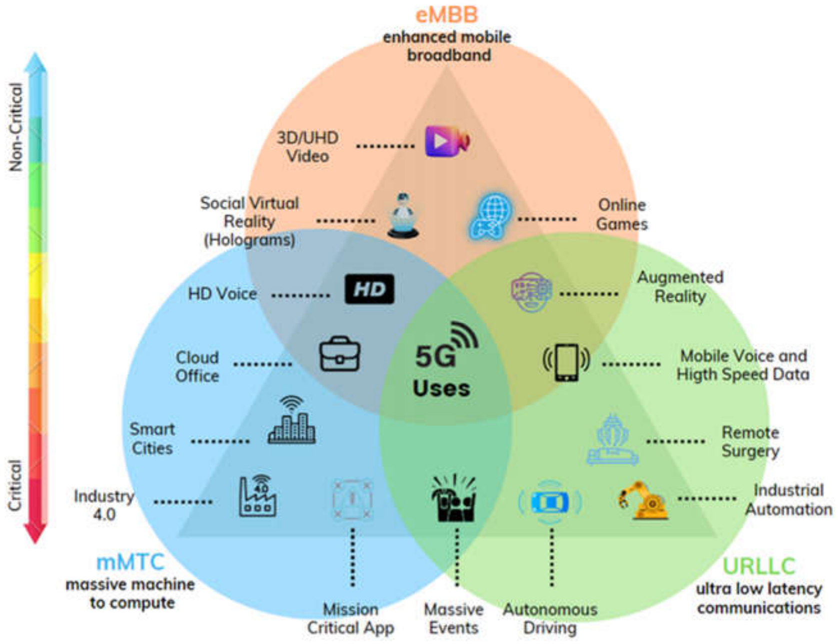


Figure 2.4. Three main 5G use cases with associated applications.

2.1.1.3 Network Slicing

The context of Network Slicing in the 5G technology represents a technical innovation of great relevance. As discussed previously in the 5G framework, Network Slicing involves the ability to create multiple independent and customized virtual networks within a shared physical network infrastructure. This network segmentation is based on the allocation of specific network resources, such as bandwidth, latency, and capacity, to each virtual "network slice," enabling the individual needs of various applications and services to be met [47](see Figure 2.5).

Resource Virtualization: Network Slicing relies on the virtualization of network resources. Physical network resources are abstracted and divided into virtual segments that are dynamically assigned to each network slice.

Traffic Isolation: Each network slice is completely independent and isolated from the others, ensuring that traffic and data from one slice do not interfere with or affect the others. This isolation is achieved through advanced configuration and security techniques [48].

Service Customization: Each slice is individually configured to meet the specific requirements of a particular application or service. For example, a slice may be optimized for IoT applications that require low latency, while another may be focused on high-quality video streaming [49].

Dynamic Orchestration: Network orchestration plays an essential role in Network Slicing. An orchestration system monitors and manages network resources dynamically, allocating and reallocating resources according to the changing needs of each slice in real-time.

As far as the application component is concerned, Network Slicing has a significant impact on several applications and scenarios. For example, in the field of massive IoT, the ability to create dedicated network slices enables efficient connectivity for a wide range of IoT devices, adapting to the particularities of traffic and energy efficiency of these devices. For applications that require high speed and quality, such as 4K and 8K resolution video streaming (eMBB), Network Slicing configures network slices with significant bandwidth and low latency, improving the user experience [50]. For ultra-reliable and low-latency communications, this approach provides network slices with minimal latency and high reliability, essential for critical real-time applications such as autonomous vehicle control and industrial automation [51]. Furthermore, the ability to establish customized 5G private networks provides organizations with an adaptable solution to meet their connectivity and security needs in enterprise environments.

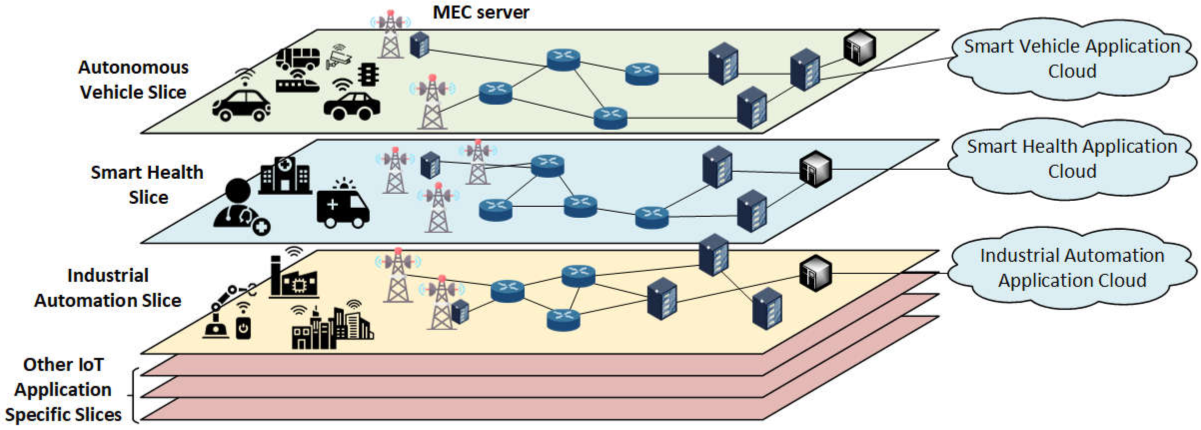

MEC server
Autonomous Vehicle Slice
Smart Vehicle Application Cloud
Smart Health Slice
Smart Health Application Cloud
Industrial Automation Slice
Industrial Automation Application Cloud
Other IoT Application Specific Slices

Figure 2.5. Network slicing.

2.1.1.4 Network Function Virtualization (NFV)

This technology is a fundamental element in the context of 5G technology. This innovation revolutionizes the way network functions are implemented and managed in telecommunications infrastructures. NFV stands as an essential enabling foundation that allows telecom operators and service providers to migrate network functions traditionally hosted on dedicated hardware to a virtualized environment. This means that tasks such as routing, firewalling, traffic management, and other network functions can run as software on virtualized servers instead of requiring dedicated physical devices.

This transformation has some key advantages in 5G deployment. Primarily, it facilitates rapid deployment and scalability of services, as network functions can be provisioned and scaled on demand without the need to purchase expensive new hardware [52]. It also promotes flexibility and customization, allowing networks to be tailored to various applications, from IoT to critical communications. Furthermore, it reduces operational costs by eliminating the requirement to maintain multiple physical devices and enabling centralized and efficient management of network resources.

2.1.1.5 Software Defined Networking (SDN)

This is a fundamental concept that has become even more relevant in the context of 5G technology. SDN is a network architecture that enables centralized and programmable management of network resources, rather than relying on static configurations and dedicated network devices. This technology is essential to achieve the flexibility, efficiency, and adaptability needed in 5G networks, which should be able to satisfy a wide variety of applications and services [52].

SDN enables more dynamic allocation of network resources to meet the changing bandwidth and latency demands of various applications, such as High-Definition (HD) video streaming, IoT, and real-time critical applications [53] (see Figure 2.6). In addition, SDN combined with NFV facilitates the implementation of Network Slicing, a key feature of 5G networks.

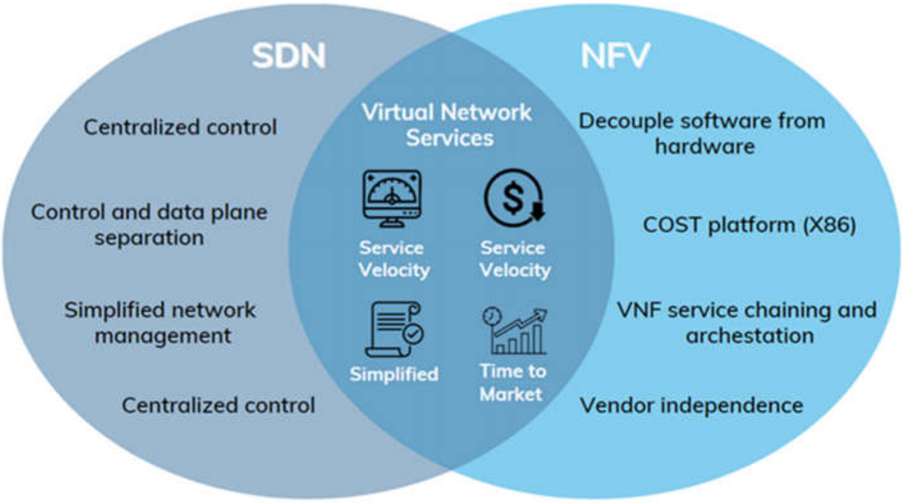


Figure 2.6. SDN vs NFV.

2.1.2 The Upcoming 6G Network

The next generation of mobile networks, 6G, is on the horizon for mobile communications technology. Although 5G technology is still being deployed and expanded, the technology community is already looking toward the future and possible features of 6G due to the challenges and shortcomings encountered during the actual deployment of 5G. As the main feature, it is estimated that in terms of speed, 6G will use a more advanced frequency spectrum that would increase data throughput 100 to 1000 times faster than 5G [8]. Offering even faster transmission speeds, near-zero latency, and an ability to support a variety of advanced applications, from hyper-realistic virtual reality to full automation, 6G promises to take connectivity to a new level [9].

This research effort is not just about speed and capacity; it is an exploration of how technology can merge with artificial intelligence, sustainability, security, and privacy to redefine the way we live, work, and communicate. From global collaboration to the incorporation of nanotechnology and advanced optics, 6G research is shaping the future of connectivity in ways we could only imagine. The next decade is expected to be dedicated to 6G networks (2030-3025) aiming to make "everything connected," in fact, several countries have already started programs for the development of a 6G ecosystem, including Finland with its flagship 6Genesis project [41]. As we immerse into this exciting world of possibilities, we prepare to discover the unimaginable and take mobile connectivity to a new dimension.

The future of 6G connectivity

6G connectivity is shaping up to be a revolution driven by artificial intelligence. Thorough integration of AI into 6G networks will result in smarter and more efficient connectivity, capable of adapting in real-time to traffic demands. It will enable mind-to-mind communication, as well as high-fidelity augmented and virtual reality experiences anywhere, anytime [54]. Moreover, holographic communication and visualization will become a reality through AI. This convergence of technology and global intelligence will make it possible to address challenges such as disaster management and remote medical care more effectively [55]. However, it is important to keep in mind that the advent of 6G is still in its early stages of development. Research and development in this area are ongoing, and the implementation of 6G on a global scale will take time. We expect 6G to play a pivotal role in enabling even more advanced applications and technologies as we move into this new era of connectivity.

2.2 Internet of Things

The IoT is a dynamic and highly versatile network that interconnects a huge variety of physical devices and everyday objects. These devices range from household appliances, security systems, and medical equipment, to autonomous vehicles. They are equipped with sensors, actuators, and communication technology that allow them to collect and share data without requiring direct human interaction. When these objects can perceive the environment and communicate, they can understand complexity and respond quickly [56]. In this context, fifth-generation communication technologies play a key role in the research for solutions to support millions of devices. 5G, along with techniques such as MIMO, and Non-Orthogonal Multiple Access (NOMA), become key enablers for various IoT applications [57]. Moreover, the IoT architecture enables anytime, anywhere interaction and communication, generating an unprecedented number of events and data (Big Data). This data can be sent to the cloud for further analysis to identify connections and patterns using artificial intelligence methods [58]. AI, which will be discussed later, is an experience-based system, where machine learning algorithms can learn what are the best actions to facilitate daily tasks or improve productivity without being explicitly programmed. Therefore, AI will be a fundamental enabler in enhancing the capabilities of IoT systems.

An essential component of the IoT is its technology platform. This platform consists of a comprehensive set of components, encompassing both software and hardware, carefully designed to enable and monitor the execution of IoT-related applications [59]. This infrastructure is essential to provide a wide range of tools and services to facilitate the creation, operation, and efficient management of IoT solutions. Some of the key components of this platform include devices and sensors, which are responsible for collecting data from physical environments, ranging from temperature and humidity sensors to Global Positioning System (GPS) tracking devices and cameras [60]. The IoT platform must provide robust and reliable connectivity for data transmission between devices and servers, which may include wireless technologies such as Wi-Fi, Bluetooth, 4G, 5G, and more.

The data collected requires processing to extract valuable information, which involves the use of processing servers capable of analyzing the data in real-time. To guarantee data integrity, secure storage must be provided, which may involve local servers or cloud storage systems. Security is critical since IoT involves the transmission of data, so it is important to implement measures that protect the privacy of information and prevent unauthorized access [61].

The platform must also enable efficient management of IoT devices, user interfaces, and applications to interact with other devices and access data. Additionally, the platform must integrate with existing systems and technologies and must be scalable to handle the potential growth of the number of devices and workload without compromising performance. Likewise, it can automate actions based on specific events and provide advanced data analytics, generating reports for valuable information. Compatibility with various platforms is crucial to ensure interoperability [62].

IoT is an adaptable technology with a wide range of applications in diverse fields. Some of the most prominent areas where IoT is used include the transformation of cities into Smart Cities, process automation in Industry 4.0, smart agriculture, healthcare with connected devices, smart homes, advanced transportation, and logistics systems, energy efficiency and smart grids, environmental monitoring, security and surveillance, personal wellness tracking, innovation in education, IoT in aquatic environments (IoUT) and more [63][64] (see Figure 2.7). IoT continues to expand as technology evolves, improving efficiency and decision-making in a wide variety of sectors and improving people's quality of life.

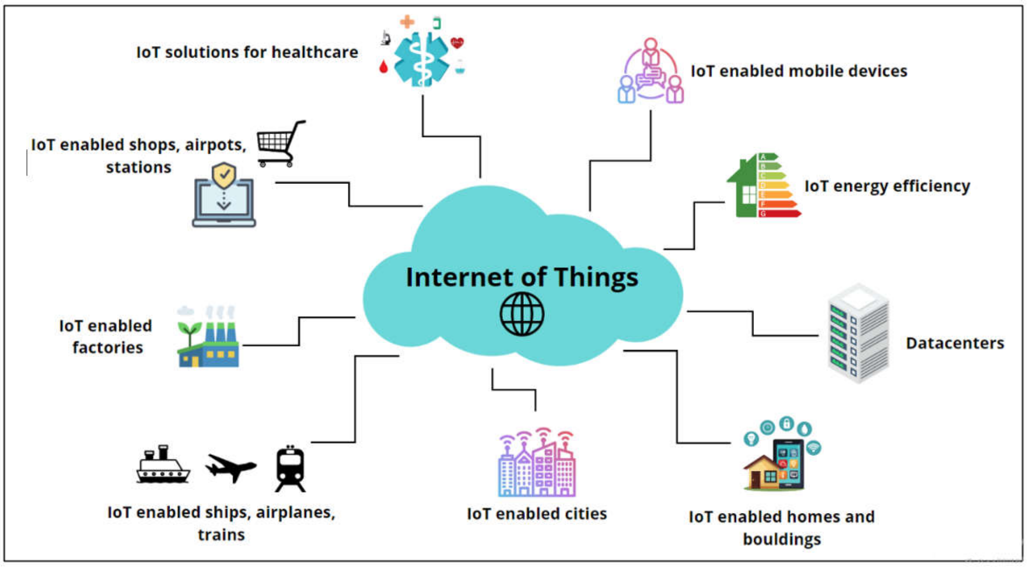

IoT solutions for healthcare
IoT enabled mobile devices
IoT enabled shops, airpots, stations
IoT energy efficiency
Internet of Things
IoT enabled factories
Datacenters
IoT enabled ships, airplanes, trains
IoT enabled cities
IoT enabled homes and bouldings

Figure 2.7. Internet of Things different application fields.

2.2.1.1 Smart Cities

Nowadays, there is no universal agreement on the definition of Smart Cities, which has led to a diversity of approaches in the literature. In this context, a definition describes a smart city as one that prioritizes the improvement of the Quality of Life (QoL) of its citizens [65] and the efficient management of its resources; especially through the use of information and communication technologies is proposed. ICT infrastructure is materialized as a smart network, which integrates several systems and services in different areas such as energy [59], transportation, public administration, security, health, and environment (see Figure 2.8). The overall purpose of this network is to provide a more efficient, sustainable, and convenient urban experience [66].

Smart cities incorporate sensors, IoT devices, advanced communication networks, and real-time data analysis systems to collect information and enable instant decisions that improve quality of life. Citizens interact with these systems through multiple devices, such as smartphones, vehicles, and connected homes. The integration of devices and data with physical infrastructure and urban services has the potential to not only reduce costs but also improve sustainability. IoT plays a key role in improving energy distribution, optimizing waste management, reducing traffic congestion, and even improving air quality [64].

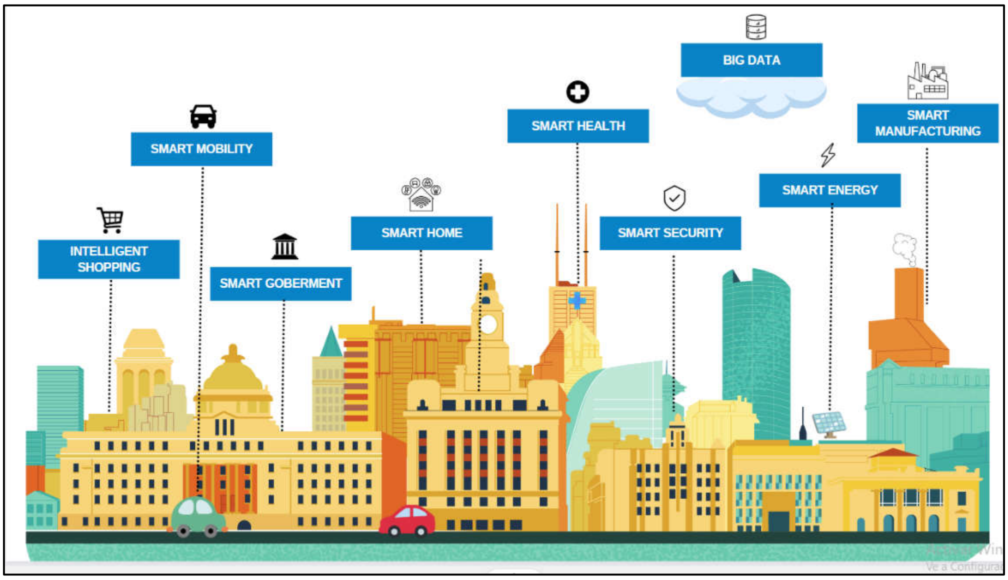


Figure 2.8. Smart cities and digital transformation through the Internet of Things.

The enabling technologies for the proper functioning of smart cities include the following:

Technology infrastructure: The smart city requires a solid ICT infrastructure; these technologies must be reliable and ubiquitous. IoT is quickly gaining popularity and is being used for different applications. However, wireless technologies are diverse, and their use must be considered depending on the application. Traffic type, distance, power consumption, and number of nodes are some of the considerations that must be taken into account when deciding how to transmit the collected data [65].

Citizen Participation: This process improves the quality of life by aligning technological projects with the real needs of the community, promoting sustainability through the integration of eco-friendly ideas. Additionally, it also strengthens the legitimacy and support of the population while fostering innovation and creativity. Moreover, citizen participation ensures transparency and accountability in decision-making and offers the flexibility to adapt to changing urban needs. Finally, it not only leads to more effective smart cities but also empowers the community and contributes to the creation of more livable and sustainable urban environments [67].

Systems integration and open data: A smart city involves the integration of multiple systems and services into a centralized platform. These systems need to be interconnected and share data efficiently to enable smart and coordinated management of city resources. Although the generated data is shared cautiously due to privacy concerns and security leaks, it is a crucial aspect of Smart Cities. The ability for all citizens to share information and combine it with contextual data for real-time analysis is a key skill [68]. Hence, informed decisions are made in real-time, and multiple sectors collaborate to achieve better sustainable outcomes. Some examples of cities that have successfully embraced open data initiatives include Amsterdam and

Copenhagen, which have implemented several real-time monitoring and management applications [69].

Security and privacy: In a smart city, data is collected and analyzed in real-time to optimize urban services. This data often includes sensitive personal information, such as citizen locations and preferences; highlighting the need to ensure the protection and ethical use of this data to prevent potential abuse. Advanced technological infrastructure, ranging from communication networks to sensors and surveillance cameras, can be vulnerable to cyber-attacks, making it crucial to implement robust security measures to protect both the data and the integrity of the city and its inhabitants.

In this context, the question arises of how to protect against hackers and cyberattacks, as well as prevent data theft [68]. The answer lies in the deployment of physical data vaults, strong authentication solutions, and effective identity management to ensure data is only shared with trusted and authorized users. In addition, it is essential to introduce appropriate regulations to address threats and potential gaps in the system, including the establishment of minimum-security requirements for connected devices. For smart cities to operate successfully, confidence in their security is critical. Moreover, all actors in the ecosystem, ranging from governments and businesses to technology and service providers, must assume responsibility for complying with basic security standards [70].

Flexible monetization schemes: In the context of the IoT and smart cities, the implementation of effective monetization schemes becomes a critical aspect to ensure the economic viability of the initiatives and services developed [71]. Software integration plays a central role in the structure of IoT solutions, enabling several actors of the ecosystem, including developers, integrators, and government authorities, to share the benefits of their contribution. In this regard, a high value and reward should be given to the intellectual property of each participant. [72].

Software subscription-based business models stand out as an option that allows each contributor to extract value from their contributions to the smart city ecosystem. However, fairness and transparency are essential elements in these schemes, and special attention must be paid to privacy and data security issues. Furthermore, to ensure citizen support and engagement, it is crucial to encourage active participation and provide clear communication around monetization models, involving the community in decision-making related to smart city services and infrastructure [73].

As urban areas continue to expand, smart city technology becomes a tool for improving sustainability and quality of life. Employing ubiquitous connectivity, focusing on open data, and implementing comprehensive security and software monetization solutions are key enablers to meet the evolving needs of smart cities and enhance the experience of all ecosystem stakeholders.

2.2.1.2 Internet of Underwater Things

The IoUT [74] refers to the application of IoT technology in aquatic environments, such as oceans, seas, lakes, and rivers. Similarly, to conventional IoT, IoUT uses sensors and connected devices to collect data and transmit information over communication networks, but the focus is underwater [75].

IoUT is important in underwater applications such as marine environmental monitoring, marine life observation, underwater exploration, aquatic resource management, disaster defense, and prediction, safety in aquatic activities, mission-critical applications, and scientific research in oceans and underwater environments [76]. Different components, including underwater sensors (e.g., Acoustic Doppler Current Profiler (ADCP)), autonomous underwater vehicles (AUVs), unmanned surface vehicles (USVs), surface buoys, and UAVs, acoustic communications, cellular, satellite or radio frequency links are integrated to achieve a complete communication ecosystem between devices located on the seafloor and servers in the cloud [77]. Figure 2.9 shows the general architecture of IoUT.

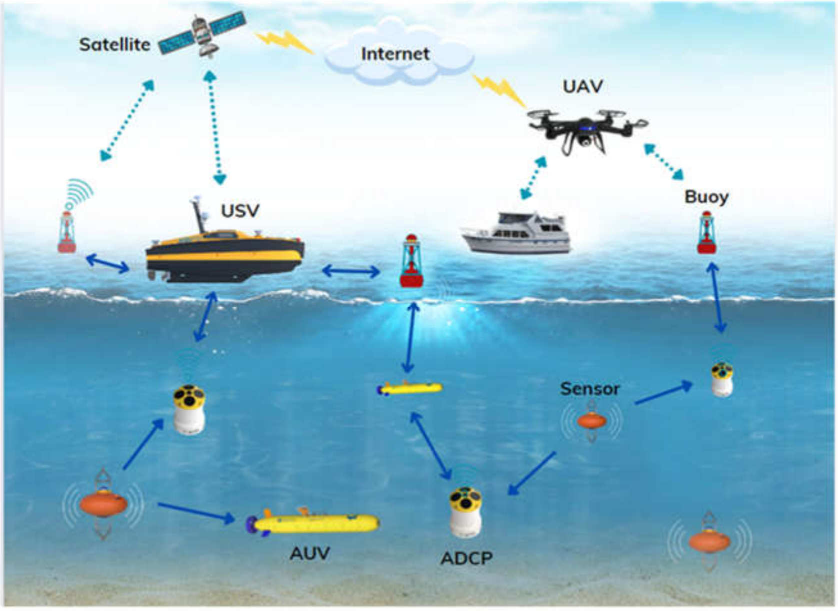


Figure 2.9. The network architecture of IoUT.

The pattern analysis of large volumes of data collected by the IoUT is an emerging field that faces specific challenges due to its aquatic environment. The combination of 5G networks and artificial intelligence plays a critical role in this context.

Approaches to analyze this data include:

Specific Sensors and Devices: The IoUT employs sensors and devices designed for underwater environments to collect data. This may include sensors for temperature, salinity, pressure, currents, sonar, cameras, and other specialized devices. The analysis starts from the collection of accurate data through these sensors [78].

Underwater Communications: 5G networks provide ultra-fast, low-latency underwater connectivity. This high speed and transmission capacity are essential for efficient data transmission from sensors and IoUT devices, enabling real-time communication in underwater environments but at short distances.

Data Processing Algorithms: Specific algorithms are developed for underwater data processing. These algorithms may include filtering techniques, error correction, interpolation, and transformations to guarantee data quality. The combination of 5G and AI enables real-time processing of data, this instantaneous processing is critical in underwater autonomous navigation applications, where AI must analyze sensor data and make autonomous decisions to control underwater systems safely and efficiently.

Modeling and Prediction: In IoUT data analysis, machine learning and AI approaches help in processing massive information and establishing the relationship between factors contributing to device failures. AI algorithms can discover complex patterns, detect anomalies, predict future events, and improve decision-making [75]. For instance, in technical applications such as marine life monitoring and underwater infrastructure inspection, AI plays a key role in identifying behaviors, classifying species, and assessing structural integrity.

Data Visualization and Presentation: Visual representation of underwater data is essential for researchers to understand patterns. 3D visualization tools and specialized software are used to display underwater data effectively (digital twins).

Resource Optimization: AI is also used to optimize resource management in IoUT applications, such as power management, data distribution, and resource allocation. This leads to more efficient use of resources and sustainable operation in underwater environments.

Data Security: In Underwater Sensor Networks (UWSNs), it is important due to the specific challenges of the environment. Issues of data integrity, confidentiality, protection against cyber-attacks, and location privacy must be addressed. Encryption techniques, digital signatures, advanced measures against cyber-attacks, secure key management, and network resilience are used to ensure security. These security features are essential to protect sensitive information and ensure the reliability of applications in challenging underwater environments [79]. IA also plays a key role in detecting and responding to underwater threats, including identifying suspicious activity in maritime environments and protecting critical underwater infrastructure.

These approaches enable researchers and professionals to analyze and understand the data collected by the IoUT, which in turn drives advances in marine exploration, environmental conservation, and other underwater applications.

2.2.1.2.1 Underwater Wireless Communications

Underwater Wireless Communications (UWCs) is a field of technology that focuses on data transmission in aquatic environments. They are highly relevant due to their wide range of applications and the resolution of critical challenges in aquatic and underwater environments. UWCs play a fundamental role in marine exploration and monitoring by enabling real-time data transmission from underwater sensors and devices for scientific research and environmental data collection [80]. They are also essential in the offshore energy industry, where they ensure connectivity for operations and monitoring, contributing to the safety and efficiency of these activities. On the other hand, it also plays an important role in communication for AUVs and Remotely Operated Vehicles (ROVs), facilitating cooperation, data collection, sending instructions, etc., in complex underwater environments [22]. In addition, UWCs are essential for emergency connectivity, such as in rescue and maritime security situations. Finally, with 5G networks expanding, UWCs become even more relevant in coastal areas as well as in the deployment of 5G applications in marine environments, ranging from ship and coastal platform connectivity to entertainment and tourism applications on the water. [81].

Underwater communications involve a variety of technologies and methods for transmitting data in an aquatic environment. Some of the more prominent approaches are included below.

Underwater Wireless Acoustic Communications

Underwater Wireless Acoustic Communications (UWACs) are efficient in transmitting signals underwater because sound is an effective communication medium in aquatic environments since it can transmit over certain distances with relatively little signal degradation. Underwater acoustic communications are crucial in applications including underwater exploration, real-time monitoring, and communication between AUVs [82]. Although UWAC is widely used, the underwater acoustic communication media is limited in several aspects, e.g., extremely slow propagation, acoustic interference from noise or marine life, limited range due to sound attenuation and scattering, significant latencies, vulnerability to changing environmental conditions, and limited bandwidth [83]. On the other hand, the efficiency of underwater acoustic communications is due to its ability to overcome certain obstacles and transmit signals through water in situations where other communication technologies may not be feasible.

Underwater wireless radio frequency communications

Underwater wireless radio frequency communications (UWRFC), this type of communication uses Radio Frequency (RF) waves for underwater communication and employs specific frequencies (between a few KHz and 1 GHz). Underwater electromagnetic communications are outstanding due to their noise immunity, efficient adaptation to changes in the environment (water/air), extremely tolerant to turbulence and water turbidity, and overcoming some of the limitations of acoustic communications regarding distance and transmission speed [84]. Although RF is efficient in terms of speed and latency, it should be noted that RF propagation is better in shallow water above tens of meters, known as buoyant RF underwater communication, so the propagation will be limited at greater depths due to signal attenuation [85].

Underwater communications research is crucial because it enables a broader understanding of the oceans, the exploration of underwater resources, and the interconnectivity of devices for different applications. Furthermore, its importance lies in the capacity to overcome communication challenges in an underwater environment and enable the transmission of critical data in real-time in this particular environment.

Underwater optical wireless communication

Underwater optical wireless communication (UOWC) refers to data transmission systems that use optical waves instead of traditional electromagnetic waves for information transmission. These systems use pulses of light, leveraging the optical properties of water, which can be generated by LED lights or underwater lasers. UWOCs offer many technical advantages such as high data rates, high bandwidth, high reliability, low installation/operating costs, and low latency [86].

Despite its multiple advantages compared to other underwater communication techniques, UOWC is limited to short range (typically up to several tens of meters). This is due to the high absorption of water in the optical frequency band and the high scattering of suspended particles. However, there is a blue-green wavelength optical window underwater with relatively low attenuation [87].

Therefore, extensive research is being carried out to increase the transfer of broadband optical signals over longer distances, with the benefit of different wavelengths [88]. Table 2.3 summarizes the underwater wireless communication technologies [88].

Table 2.3. Comparison between existing underwater wireless communication technologies.

Parameters	Acoustic	RF	Optical
Attenuation	Distance and frequency-dependent (0.1–4 dB/km)	Frequency and conductivity dependent (3.5–5 dB/m)	0.39 dB/m (ocean) 11 dB/m (turbid)
Speed	1500 ms^{-1}	2.3×10^8 ms^{-1}	2.3×10^8 ms^{-1}
Data Rate	kbps	Mbps	Gbps
Latency	High	Moderate	Low
Distance	more than 100 km	≤ 10 m	10–150 m (500 m potential)
Bandwidth	1 kHz–100 kHz	MHz	150 MHz
Frequency Band	10–15 kHz	30–300 MHz	5×10^{14} Hz
Transmission Power	10 W	mW–W	mW–W

2.3 Artificial Intelligence

It refers to the ability of machines to perform tasks that normally require human intelligence. These tasks concern learning, reasoning, problem-solving, perception, and understanding of natural language, and are based on algorithms and mathematical models. This enables machines to perform actions autonomously and improve their performance as they are exposed to more data and experiences. The applications of AI are diverse and can be found in fields such as virtual assistants, autonomous vehicles, medical diagnostics [89], and enterprise data analytics.

At the beginning of the new millennium, AI has experimented with a renaissance boosted by advances in machine learning algorithms and the growth in the processing capacity of computers. Nowadays, AI is constantly improving and has made significant advances in fields such as speech recognition, computer vision, natural language processing, and robotics [90]. These are only a few examples of applications that are transforming entire industries. In addition, deep learning (as part of machine learning) has revolutionized AI learning by enabling machines to analyze and understand even more complex data.

AI is a technology that has come a long way since its inception and has become an essential part of our daily lives. However, this steady growth also raises significant ethical concerns. While AI becomes more deeply integrated into society, it is critical to address issues such as privacy [91], fairness, transparency, and ethical decision-making in the design and use of systems based on AI [92]. The global community must work together to establish standards and guidelines to guide the responsible development of AI, ensuring that this technology benefits humanity. Figure 2.10 shows the subdivision of artificial intelligence into its different fields.

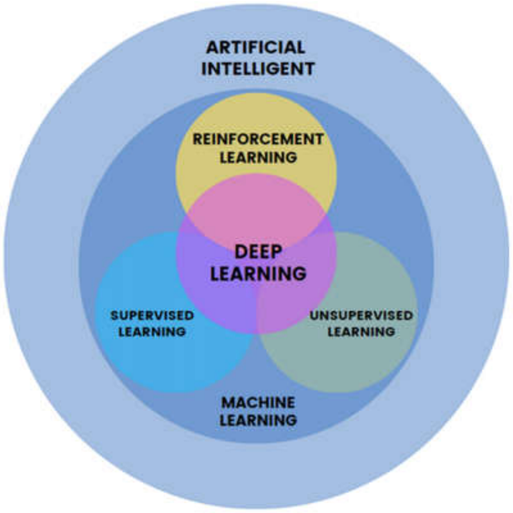


Figure 2.10. Overview of the relationship between artificial intelligence, machine learning, deep learning, and reinforcement learning.

2.3.1 Machine Learning

Machine learning is a part of artificial intelligence that focuses on developing algorithms and models that allow computers to automatically "learn" from data. The aim is to generate or improve predictions by automating statistical models, through the classification of data (instances) that are provided without the need to explicitly program rules, allowing the detection of different patterns or anomalies in large volumes of data. Thus, it is possible to predict future behaviors with the ability to improve autonomously over time without human intervention. This technique has revolutionized the approach of companies and organizations to decision-making and complex problem-solving [93]. The machine learning process can be divided into three main steps.

Data preparation: In this step, data is collected and prepared for model training. This involves cleaning data to remove useless or duplicate information, normalizing data to ensure that the data is presented uniformly, and splitting data for training, and test sets.

Model training: In this step, the training set is used to adjust the model parameters and improve its accuracy. To do so, the model iterates several times on the training data, adjusting the parameters after each iteration by backpropagation, to minimize the model prediction error.

Model evaluation and testing: Finally, the test set is used to evaluate the accuracy of the model. The model predictions are compared with the real responses of the test set and performance measures such as accuracy, error, and sensitivity are calculated.

Overall, the learning process is iterative and focuses on continuously improving the accuracy and efficiency of the model as new data is presented to it. With the advancement of technology and the growth in the amount of data available, machine learning is becoming an increasingly important and necessary tool for decision-making and complex problem-solving in a wide variety of fields [94].

In the machine learning field, there are several types of approaches, each designed to address distinct types of problems and tasks. ML classifies learning based on the level of supervision as supervised learning, unsupervised learning, semi-supervised learning, and reinforcement learning (detailed in section 2.3.3) [95]. Next, we will summarize each of these learning approaches.

2.3.1.1 Supervised Learning

Supervised learning is a fundamental technique in the field of machine learning. It is used to train models that can make accurate predictions based on a set of input data. This type of learning is based on a set of training instances that have their respective "identifications" or labels. These labels enable the algorithm to match their input value (features) with the desired or target output. During the training process, the weights of the model are adjusted until the output becomes the desired one [96]. This prior knowledge of the labels will help the algorithm to understand the data when it is processed for further prediction or decision-making, as shown in Figure 2.11.

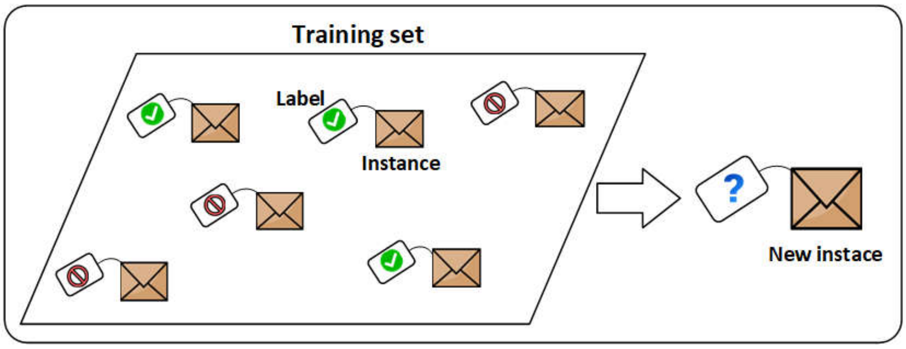


Figure 2.11. A labeled training set for supervised learning.

There are many supervised learning algorithms, including linear regression, logistic regression, decision trees, neural networks, and so on. Each algorithm has its strengths and weaknesses and is suitable for distinct types of problems. The expected outcome of classification algorithms is that the system can identify, through its input features, certain patterns that allow the algorithm to classify the information into distinct groups [97]. On the other hand, regression algorithms respond with a specific number or value associated with the input data (e.g., predicting the cost of a vehicle, given a set of input characteristics), as shown in Figure 2.12.

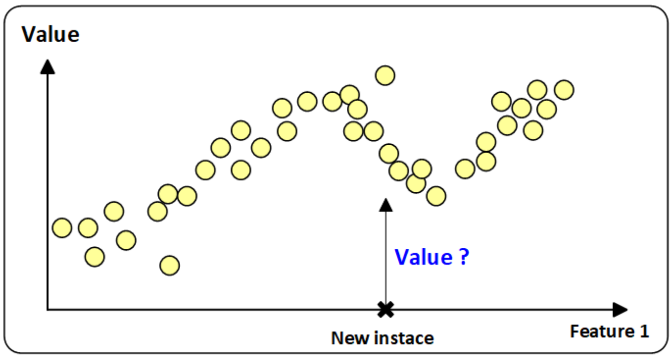


Figure 2.12. Logistic Regression.

2.3.1.2 Unsupervised Learning

Unsupervised learning is the paradigm that can obtain knowledge only from the input data without specifying the desired output. Unlike supervised learning, in unsupervised learning the algorithm is fed only with the input values (features), without the labels; the goal is for the algorithm to classify data and group them considering their similar characteristics. Furthermore, in unsupervised training, the goal is to discover unknown structures or patterns in the data. For instance, a clustering algorithm (Figure 2.13) can group data into different categories based on their similarity, while a dimensionality reduction algorithm can find important features in the data to represent them in a lower dimensional space [98].

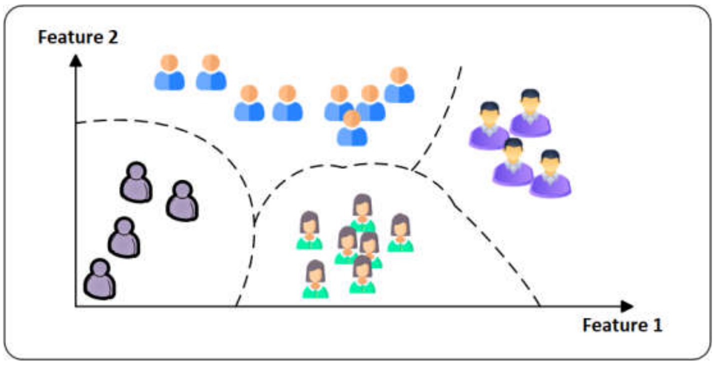


Figure 2.13. Clustering.

Unsupervised training is used in a wide variety of applications, such as customer segmentation in marketing, anomaly detection in security systems, genomic data exploration, and understanding weather and geophysical patterns [99].

2.3.1.3 Semi-supervised Learning

This learning method aims to leverage both labeled and unlabeled data to improve the performance of learning algorithms. Semi-supervised learning is based on the fact that in many cases, having large amounts of unlabeled data along with a limited set of labeled data may be more feasible than having a fully labeled data set [100].

Semi-supervised learning has proven useful in scenarios where labeled data collection is costly or difficult. This allows taking advantage of the information present in unlabeled data to improve the predictive performance and generalization capability of machine learning models [101].

Overall, semi-supervision focuses on the differential treatment of data points based on the presence or absence of associated labels; for labeled points, the algorithm employs a conventional supervision approach to fine-tuning the model weights; while for unlabeled points, the algorithm aims to minimize the difference in predictions among other similar training examples.

For a better understanding, consider the following dataset in Figure 2.14: A binary classification problem with one class for each group of data. Suppose we have only seven labeled data points and the remainder unlabeled.

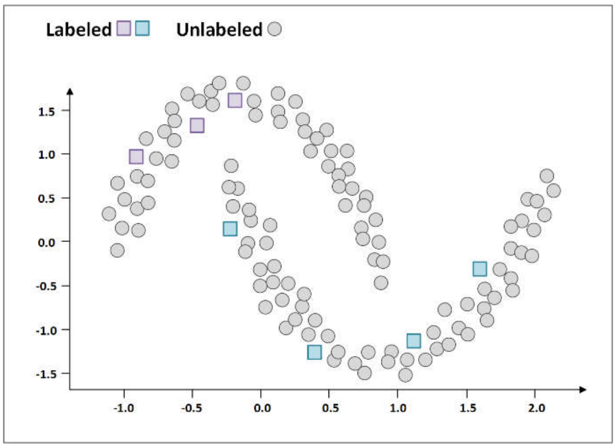
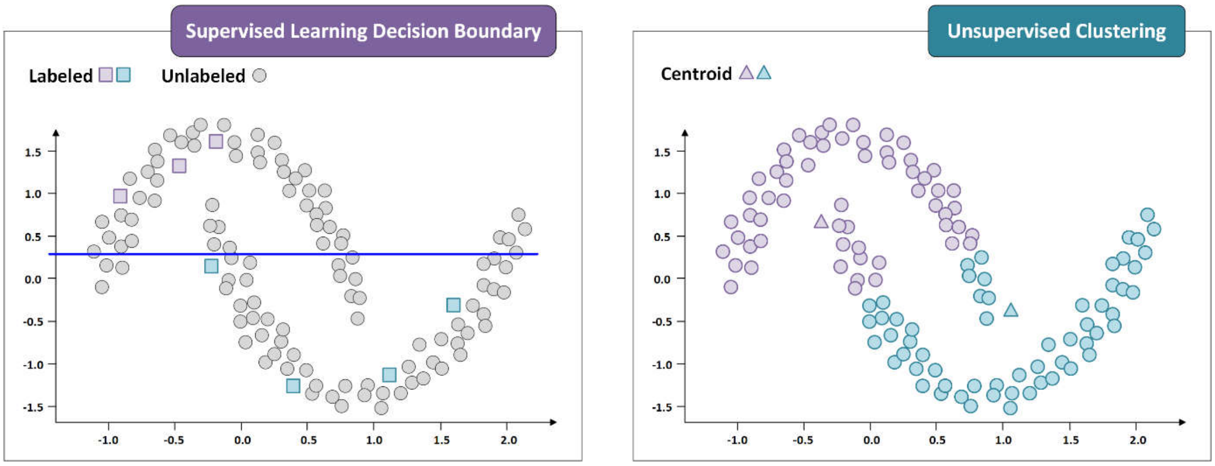


Figure 2.14. Dataset example for a binary classification problem.

On the one hand, supervised learning involves updating the model parameters to reduce the average difference between the predictions generated and the labels provided. However, in situations where labeled data are scarce, this process could result in the creation of a specific decision boundary that only fits the labeled points, without generalizing across the entire data distribution (see Figure 2.15a).



a) b)

Figure 2.15. a) Supervised Learning Decision Boundary, b) Unsupervised Clustering.

On the other hand, unsupervised learning is dedicated to cluster points based on the similarity of their features. However, the absence of labels to guide the training process can lead an unsupervised algorithm to identify clusters that do not correctly correspond to the true class distribution (see Figure 2.15b).

In scenarios where the availability of labeled data is limited, or when working in complex clustering environments, supervised and unsupervised techniques may face difficulties in achieving optimal results. However, in a semi-supervised context, labeled data points have a significant role by acting as reference markers. These points not only support the model predictions but also provide structure to the learning problem by determining the number of classes and the assignment of groups to each class. Unlabeled data, on the other hand, provide invaluable context. When exposing the model to this unlabeled information, a better understanding of the overall distribution of the data is achieved. This extended exposure to unlabeled data allows us to more accurately estimate the full shape of the distribution, which significantly improves the predictive ability of the model.

The strategic combination of labeled and unlabeled data in model training offers the possibility of developing more accurate and robust models. In this sense, the semi-supervised approach within our specific dataset has the potential to approximate the actual distribution, as illustrated in Figure 2.16.

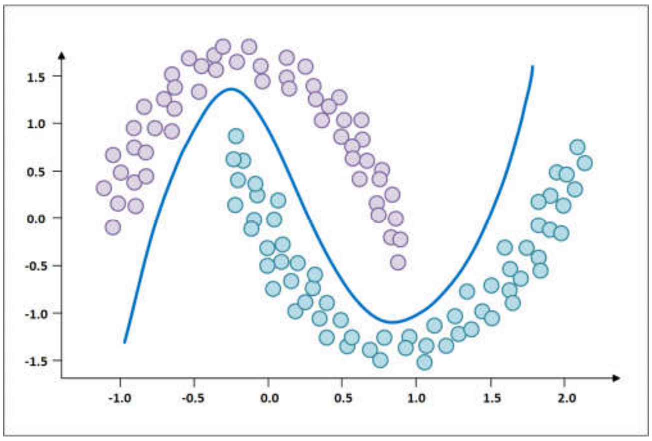


Figure 2.16. True class distribution for dataset example for a binary classification problem.

2.3.2 Deep Learning

Deep Learning, although it may seem a new theory, began to be investigated more or less at the beginning of the 1990s. Due to video games and the development of graphics cards that contain a series of matrix calculations, it has been possible to establish the basis for practically all technologies based on machine learning.

2.3.2.1 Deep Neural Networks (DNN)

DNNs are based on Artificial Neural Networks (ANN), which in turn are based on the functioning of the human brain, in particular on the connections and operations performed by neurons [102]. These networks are designed to simulate information processing in biological systems and are part of machine learning or artificial intelligence. DNNs are a type of artificial neural network composed of several layers of interconnected neurons. Each layer processes the data received from the previous layer and transforms it into a more abstract and complex representation. These networks are used to learn patterns in large data sets, and they are particularly effective for classification and prediction tasks. An explanation of the basic architecture of deep neural networks is presented below (see Figure 2.17).

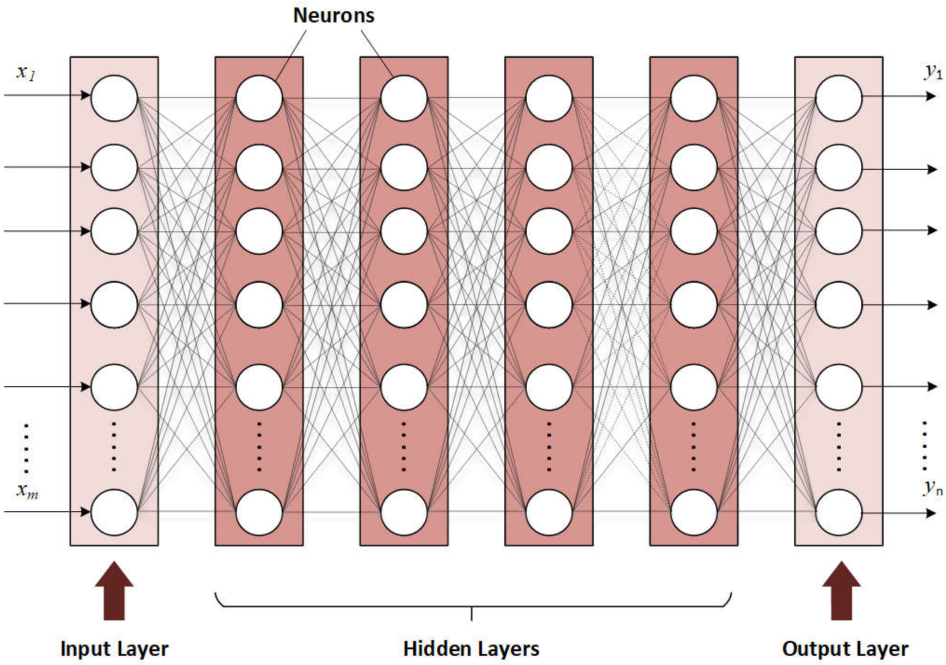


Figure 2.17. Deep neural network structure.

Input Layers: The initial layer receives the input data, which are generally numeric vectors representing features, such as pixels in the case of images.

Hidden Layers: These intermediate layers, also called hidden layers, perform nonlinear transformations of the input data. Each hidden layer is composed of multiple neurons, and the operations performed on these layers are based on activation functions, such as the Rectified Linear Unit (ReLU) function, which introduces nonlinearities into the model [103].

Output Layer: This last layer generates the predictions or results. The structure of this layer depends on the task being addressed. It may consist of a single neuron for binary classification problems or several neurons for multiclass classification or regression.

For the deep neural networks training process, we must consider two important steps:

Forward Pass: During this step, input data is propagated through the network from the input layer to the output layer. Each neuron computes a linear combination of the inputs, followed by the operation of an activation function. This is repeated layer by layer.

Backpropagation: In this step, the predicted output is compared with the real value (label or desired value), and the error is calculated. Then, the error is propagated backward through the network, and the weights of the neurons in each layer are adjusted to minimize the error. This process uses optimization algorithms, such as gradient descent, to update the network parameters [104].

Due to the aforementioned characteristics, DNNs are a powerful method for solving large-scale real-world problems [105]. DNNs are used in computer vision for image recognition, Natural

Language Processing (NLP) in text translation and analysis tasks, personalized recommendations in streaming and e-commerce platforms, creation of intelligent agents in games, robotics for perception and control of autonomous robots, bioinformatics in DNA and protein analysis, finance for financial data analysis, industrial automation, healthcare for medical diagnosis and patient monitoring [106], and for creative fields such as music and generative art. Their flexibility and ability to learn complex patterns make them a valuable tool in many fields, and they continue to transform many aspects of technology and everyday life.

2.3.2.2 Convolutional Neural Networks

Convolutional Neural Networks (CNNs) are a type of deep neural network architecture designed specifically for computer vision and image processing tasks. These networks have become a cornerstone in fields such as image classification, object detection, semantic segmentation, and many other applications related to image analysis. [107].

The distinguishing feature of CNNs is their ability to capture local patterns in images through the use of convolutional layers. Instead of connecting each neuron in the network to all neurons in the previous layer, as is done in fully connected neural networks, CNNs apply convolutional filters to local regions of the image. These filters are shifted across the entire image to extract local features. As the convolutional layers stack in the network, they become deeper and more complex, allowing them to detect higher-level features and abstract representations of the image.

In addition to the convolutional layers, the architecture of CNNs typically includes pooling layers, which reduce the spatial resolution of the extracted features, followed by activation layers to introduce nonlinearity, such as the ReLU function. Finally, fully connected layers are included for classification or regression, and an output layer produces the final prediction [108] (see Figure 2.18). The architecture of a CNN can vary in complexity. Popular architectures, such as AlexNet [109], VGG [110], ResNet [111], Inception [112], etc. have driven advances in the field of computer vision. These architectures can consist of multiple convolutional layers and are often pre-trained on massive datasets before fine-tuning the details of specific tasks [113].

CNNs have demonstrated high performance in a wide range of computer vision applications, such as object recognition, face detection, image segmentation, and others. In addition, they have been fundamental to the success of several artificial intelligence applications, such as autonomous vehicles, medical diagnostics, video analysis, and more. Their specific architecture and their ability to learn visual representations make them essential today for tackling complex computer vision problems [102].

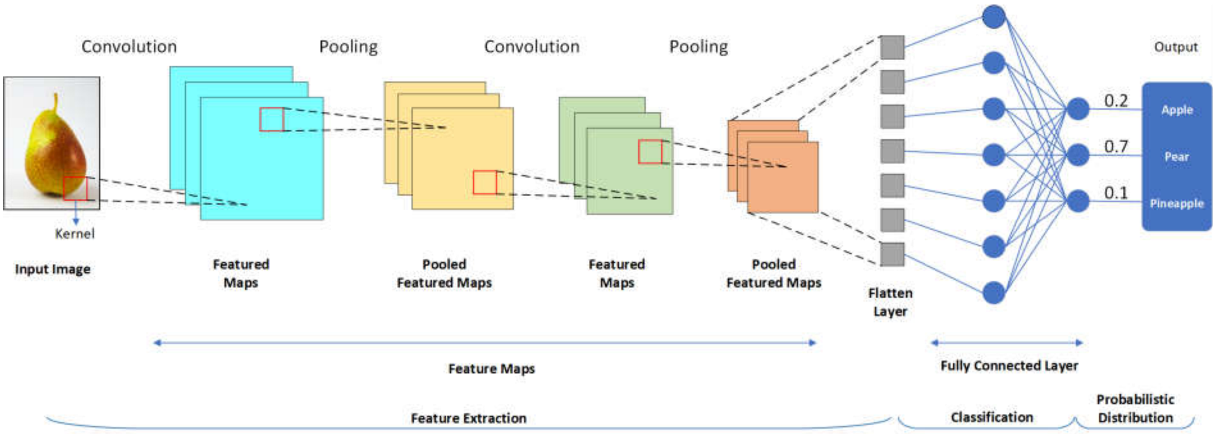


Figure 2.18. Convolutional neural network structure.

2.3.3 Reinforcement Learning

Reinforcement Learning (RL) is probably the most natural learning method that exists since it is based on the cause and effect generated by different situations. Human beings, for example, in the childhood stage, are confronted with situations that make them learn that something hurts them, and they do not do it again. When we talk about algorithms, the model is implemented in the form of an agent that must explore an unknown space and determine the actions to be carried out through trial and error. It will learn by itself through the rewards and penalties that it gets from its actions. The agent must develop the best possible strategy (policies) to obtain the greatest reward in time and effort. This learning can be combined with other types of learning, and it is very popular at the moment since, as mentioned above, the real world presents many of these scenarios [114].

Reinforcement learning is employed in a wide range of applications, such as robot control, gaming, resource management, online advertising, decision-making in finance, and so on. Some of the most common reinforcement learning algorithms include the Markov Decision Process (MDP), Q-learning, gradient policy, and value function approximation [115]. This type of learning is an exciting and evolving field of machine learning, with many practical real-life applications. As reinforcement learning becomes more sophisticated and is applied to a wide variety of problems, it has the potential to transform the way we interact with the world and make decisions in complex situations.

2.3.3.1 Markov Decision Process

It is a mathematical framework used in the field of reinforcement learning and sequential decision-making. This approach is fundamental for situations where an agent (i.e., a machine, robot, or any entity that can make decisions), must interact with an environment sequentially and dynamically. In MDP, the agent makes decisions in a stochastic environment, where the transition from one state to another and the rewards associated with each transition are uncertain. Each state has a reward associated with it, which is used to guide the agent toward optimal decision-making.

The agent's goal is to learn an optimal policy, i.e., a strategy that maximizes the expected reward in the long run. The agent's decisions are based on the information available in the current state and not on its history. This is known as the Markov property [116]. In the end, the agent will always try to maximize the reward through trial and error, and it will try different actions in the environment that will allow it to behave as we want it to. A discount factor can be applied to future rewards so that the agent values the immediate rewards more highly. The graphical representation of the MDP model is shown in Figure 2.19, where the key elements mentioned above (states, actions, rewards) are represented.

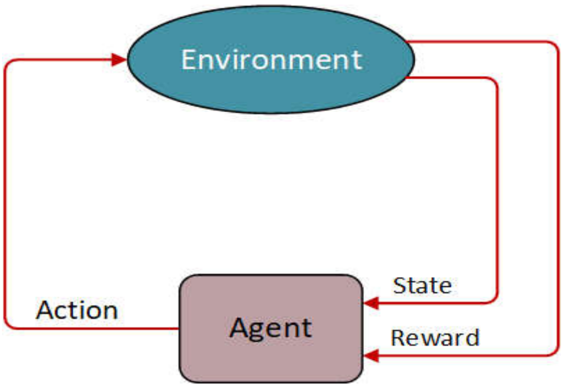


Figure 2.19. Graphical representation of the MDP model.

2.3.3.2 Q-Learning

Q-learning (Watkins, 1989) is a reinforcement learning algorithm used to learn an optimal policy in a Markov decision process. As mentioned earlier, in an MDP, an agent makes decisions in a stochastic environment and learns to maximize a long-term reward. *Q*-learning aims to achieve the agent make decisions based on its current knowledge and a function called "*Q*-Value". Which is defined as the sum of the immediate reward and the expected reward in the next state; weighted by a discount factor that reflects the relative importance of future rewards [117]. It means that it allows experiencing the consequences of actions without the need for them to construct maps of the domains.

The *Q*-learning process involves iterating through different states of the environment, selecting actions, and updating *Q*-values as the agent learns more about which actions lead to the highest rewards. This process is based on the Bellman equation, which relates the current *Q*-value to the *Q*-value of future states.

The *Q*-learning algorithm consists of the following steps:

Initialization: Initial *Q* values for all state-action pairs are set to zero.

Exploration and exploitation: The agent chooses the action in a specific state either by exploring (making random decisions) or exploiting (selecting the action with the highest Q value). Exploration is important for discovering new strategies.

Iteration: The process repeats as the agent explores more states and accumulates knowledge about the optimal policy.

Convergence: The Q-Learning algorithm ends when the Q-values converge to an optimal policy, which indicates the best action to take in each state to maximize future rewards [118].

Q-learning has been applied in a wide variety of fields, such as robotics, complex board games like GO, automatic control, robotic navigation, and recommender systems. It is especially useful in situations where the details of the environment are unknown, and the agent needs to learn an effective strategy through experience and interaction.

2.3.3.3 Deep Q neural networks

Deep Q Neural Networks (DQNs) are an advanced approach in the field of artificial intelligence, specifically in the field of reinforcement learning [119]. Reinforcement learning, as we discussed earlier, aims to achieve ideal behaviors as one interacts with a particular environment. DQNs are a significant evolution in this field and have demonstrated excellent results in complex tasks [120].

The fundamental basis of the DQN lies in the approximation of a function called "Q-function". This Q-function assigns a value to the quality of a given action in a particular state. In other words, it determines how beneficial an action is in a particular context. In reinforcement learning problems, the main challenge is to calculate this Q function accurately, since the space of states and actions can be vast and complicated [121]. DQNs use deep neural networks to approximate this Q function. These networks receive as input information about the current state of the environment and generate as output the Q value associated with each possible action. During the training process, DQNs adjust their weights and internal connections using techniques such as backpropagation to minimize the error in the Q-value predictions. The training aims to make the network learn to predict Q-values accurately, which will allow the agent to make optimal decisions in a particular environment.

DQNs have had a significant impact on a wide range of applications, from learning artificial intelligence agents to play video games to decision-making in robotics [122][123]. Their ability to handle complex problems and learn high-level representations of an environment makes them a valuable tool in the research and application of deep reinforcement learning.

2.3.3.4 Deep Deterministic Policy Gradient

Deep Deterministic Policy Gradient (DDPG) is an advanced reinforcement learning algorithm that addresses continuous control problems in complex and dynamic environments. The DDPG combines two fundamental concepts: DNN and deterministic policy.

DDPG is used in situations where an agent must make sequential decisions to maximize a reward over time in a particular environment. Put more simply, one can imagine an autonomous drone that needs to learn to fly safely or a robot that must learn to perform delicate and precise tasks in the real world. The DDPG is a solution to these types of challenges.

The role of Deep Neural Networks in DDPG is crucial. These networks are used to approximate two key components: the value function and the policy. The value function is an estimate of how good a certain action is in a particular state. It helps the agent evaluate actions and select those that maximize long-run rewards. In DDPG the policy is deterministic. The deterministic policy is a function that maps states directly to actions, rather than producing a probability distribution of actions, as in other reinforcement learning methods. A distinguishing feature of DDPG is its ability to handle continuous action spaces, where actions can be any value within a range, rather than discrete choices. This is critical in real-world applications, such as autonomous robotics or vehicle control, where precision in decision-making is required.

The DDPG training process is iterative. The agent interacts with the environment, collects data, and uses it to gradually update and improve both the policy and the value function. As training progresses, the agent becomes more competent in selecting actions that maximize its performance and obtain rewards. This approach has proven successful in a range of applications, such as robot control, drone navigation, and gaming, where precise and continuous control is required to achieve specific tasks.

2.4 Cloud Computing

Cloud Computing (CC) is a technology that has transformed the entire process of how organizations store, manage, and access computing resources. It involves providing computing services, such as storage, processing, and applications, over the Internet. This approach has revolutionized the information technology industry and has provided countless benefits to businesses and end users.

Fundamentally, cloud computing is based on the concept that computing resources can be virtualized and hosted on remote servers rather than depending on local hardware and software. It enables organizations to scale resources as needed, providing flexibility and efficiency in resource management [124]. The technical features of cloud computing involve the implementation of Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS) (see Figure 2.20). IaaS provides access to virtual computational resources, such as Microsoft Azure, Apple iCloud, and Google Drive. PaaS provides an environment for developing and running applications, i.e., Google App Engine. SaaS provides web-based applications accessible through browsers, with no need to install and run applications on computers, e.g. Google Apps, Cisco WebEx, and Salesforce [125].

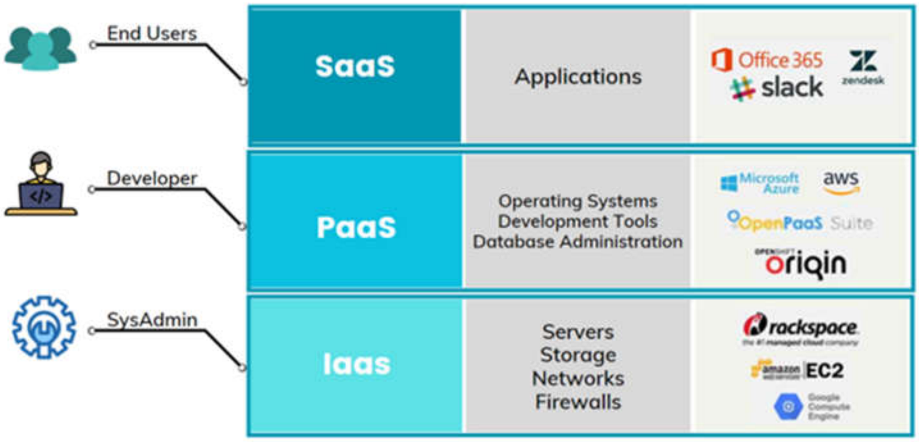


Figure 2.20. Cloud computing services: Iaas, PaaS, and Saas.

Cloud computing presents many important features, such as *auto-scalability* that allows the expansion and contraction of resources efficiently, avoiding unnecessary investments in hardware. *Global accessibility* facilitates remote work and collaboration, thus, cloud service providers take care of maintenance and upgrades, as well as automatic backups. *Virtualization and resource pooling* provide efficiency and flexibility to the cloud infrastructure. *Rapid elasticity* enables tools and features to be changed quickly, and so on [126]. These advantages have driven the widespread adoption of cloud computing in several organizations.

Although there are many advantages, there are significant challenges and considerations in cloud computing, such as data security and privacy, dependency on Internet connectivity, and cost management. Data security is a constant concern since data is stored outside the local environment. In addition, network latency and Internet connection availability can impact performance and accessibility. Cost management is crucial since excessive use of cloud resources can result in unexpected expenses [127][128].

The future of cloud computing is promising. It is expected to continue to evolve with advances in technologies such as quantum computing and artificial intelligence. The adoption of hybrid cloud and multi-cloud solutions is becoming more common, which will enable organizations to combine cloud resources from multiple vendors [129]. Moreover, automation and orchestration of cloud resources will be growth areas, which will further improve efficiency and scalability.

Cloud computing has consolidated its position as the predominant method for data management in the Internet of Things (IoT). In addition to cloud computing, fog, and edge computing strategies have seen extensive use. The latter is used to boost the speed and efficiency of data processing, bringing processing power and intelligence directly to IoT devices responsible for generating data (sensors) and executing actions on them (actuators). These computer technologies are characterized by their respective designs and objectives, although they often complement each other. A summary of each of these approaches is provided below.

2.4.1 Edge Computing

Edge Computing (EC) has experienced significant growth in recent years, driven by the demand for applications that require faster processing and low latency. Instead of relying on remote data centers or servers in the cloud, edge computing focuses on performing computation and data analysis tasks close to the data source [130]. This is essential for enabling more agile processing in applications ranging from the IoT to industrial automation systems, autonomous transportation, augmented reality, and mission-critical and real-time response applications [131].

By minimizing the need to transmit large volumes of data over the network, bandwidth utilization is optimized, and communication costs are reduced. It also strengthens data privacy and security by avoiding long transmissions and remote connections. In the context of 5G technology, edge computing becomes even more relevant. 5G networks offer exceptionally high speeds and low latency, which aligns perfectly with the needs of edge computing. When data processing is brought closer to the edge of the 5G network, faster connectivity and optimal performance are guaranteed in applications with immediate response requirements, such as autonomous vehicles and industrial monitoring [132]. In addition, edge computing has been applied in several areas, such as video surveillance systems, by processing the information received by the cameras before sending it to servers located in the cloud. It is also applied in the manufacturing industry to control machinery and in the healthcare sector to carry out real-time monitoring of patients [133].

The future of edge computing is certainly promising, with its continued expansion into fields such as the Internet of Things, artificial intelligence, and automation. However, there are challenges concerning management and maintenance, along with interoperability and security issues that need to be addressed for large-scale deployments [134].

2.4.2 Fog Computing

The term "Fog Computing" was introduced by Cisco Systems and is an extension of cloud computing that was developed to address processing and latency limitations in IoT applications and other distributed systems. Instead of relying on remote servers in the cloud, Fog Computing brings data processing closer to the source, using local servers and edge devices [135]. This technology has evolved to accommodate the growing demand for real-time and low-latency applications, such as autonomous vehicles, augmented reality, and mission-critical applications.

Fog Computing is a kind of bridge between edge computing and cloud computing (see Figure 2.21) since it allows processing closer to the edge but not as far away as the cloud. This feature allows for reduced latency and improved response speed. It also reduces the workload on the network and decreases the need to transmit large volumes of data to the cloud, which can result in bandwidth savings [136]. In addition, Fog Computing enables greater autonomy of devices since they can make decisions locally without relying on a constant connection to the cloud. In terms of applications, fog computing is being used in a variety of sectors, such as industrial automation, healthcare [137], autonomous transportation, and energy management.

Besides managing computation and networking, fog computing also deals with storage, control, and acceleration of data processing [138]. Even with the technological advancement and the advantages of fog computing, it also faces challenges, such as the management of dispersed edge servers, providing security and privacy of local data [139], and the selection of appropriate virtualization technology, since it is the main method for providing isolated environments [140].

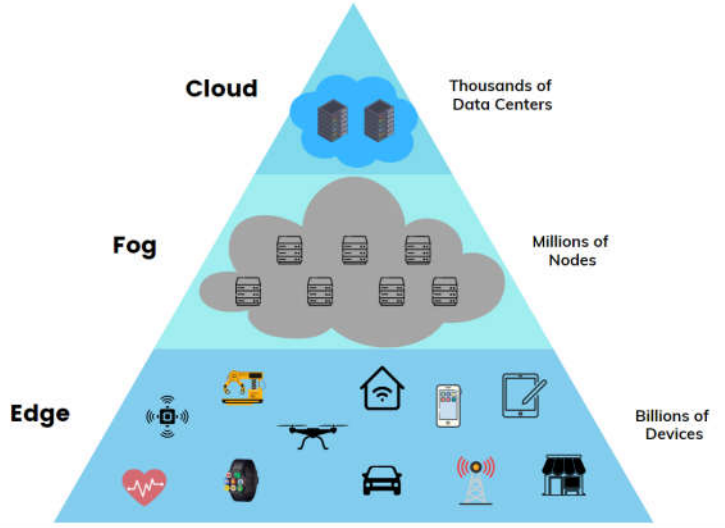


Figure 2.21. Hierarchical organization between cloud, fog, and edge computing.

2.4.3 Multi-Access Edge Computing

Multi-Access Edge Computing (MEC) is a specific implementation of edge computing designed for mobile networks and represents a new architecture concept that has become increasingly relevant in the wireless network environment, especially with the advent of 5G and the demand for URLLC applications [141]. MEC is integrated into the RAN infrastructure and allows processing and storage capacity to be driven closer to the data sources and applications, i.e., to the "edge".

MEC stands out for its ability to optimize applications that require low latency, such as cloud gaming, augmented realities, telemedicine, autonomous vehicles, etc. Applications adapt and operate efficiently in this edge environment, enhancing the user experience. Moreover, MEC integrates seamlessly with existing network infrastructure, and it communicates with network management systems to ensure efficient allocation of compute resources and bandwidth [142].

Although MEC presents many advantages in terms of latency and network efficiency, it also faces many technical challenges. On the one hand, the physical proximity of servers to the network edge can make them more vulnerable to security threats, requiring robust measures to protect data and ensure user privacy. Moreover, MEC must operate in a diversified network environment that includes technologies such as 4G, 5G, and Wi-Fi, which increases complexity due to the heterogeneity of technologies and the need to ensure interoperability [143]. On the other hand, the diversity of devices that MEC must support, ranging from smartphones to IoT sensors, adds complexity due to the different capabilities and requirements of these devices. Efficient coordination and management of computing resources at the network edge is a key challenge since decisions must be made about what applications must run in which location to optimize performance and minimize latency. Overcoming these and other challenges is essential to successfully deploying MEC and leveraging its potential in enabling low-latency, and high-throughput applications in mobile networks [144].

The successful implementation and deployment of MEC requires careful planning and collaboration between network operators, service providers, and equipment manufacturers. Communication standards and protocols must be considered to ensure interoperability and seamless integration with existing networks. NFV performs a key role in enabling flexibility and scalability of computing resources at the edge [145].

The future of MEC is promising, especially in the context of emerging technologies such as 5G and IoT. 5G is expected to provide higher network speed and capacity, which will enable more advanced MEC applications, such as autonomous vehicles, smart cities, and real-time streaming services [146]. Moreover, MEC will be combined with artificial intelligence and machine learning to enable autonomous decision-making and real-time analytics at the edge. The interconnection of IoT devices and the cloud with MEC will enable an entirely new ecosystem of applications and services.

CHAPTER 3

3 DEEP LEARNING, EDGE COMPUTING, AND 5G AND BEYOND FOR SMART CITIES.

This chapter is based on:

- Cepeda-Pacheco, Juan Carlos; Domingo, Mari Carmen. Deep learning and Internet of Things for tourist attraction recommendations in smart cities. Neural Computing and Applications, 2022, vol. 34, no 10, p. 7691-7709.
- Cepeda-Pacheco, Juan Carlos; Domingo, Mari Carmen. Deep learning and 5G and beyond for child drowning prevention in swimming pools. Sensors, 2022, vol. 22, no 19, p. 7684.

The massive growth of the Internet of Things supported by ML techniques has significantly transformed the paradigm of smart cities. The massive collection of data in cities by sensors and connected devices generates continuous streams of information; here is where ML techniques enter the scene to process and analyze these data efficiently. Furthermore, with the advantages provided by edge computing and 5G communications and beyond, valuable information can be quickly and timely provided for decision-making.

Regarding urban mobility, the successful integration of these technologies has led to more efficient transportation systems. Sensors located on streets and vehicles transmit data in real-time, enabling intelligent traffic management, route optimization, congestion reduction, etc. [147]. For resource management, systems are being used to monitor and control energy consumption in buildings, water distribution, and waste management [148]. Concerning culture and tourism, it has generated a significant impact in the field of smart tourism, redefining the way tourism services are managed, offered, and experienced [149]. The collection of data generated by IoT devices provides a solid basis for decision-making and efficiency improvement in the management of tourism resources, such as optimization of transportation routes, congestion control at tourism sites, and energy management [150]. Through ML, patterns can be identified, trends can be predicted and experiences can be customized more effectively.

In the field of public safety, surveillance cameras and sound detection sensors are integrated into IoT-based security systems. ML techniques can analyze images and sound in real-time to identify potential threats and alert authorities or citizens automatically [151]. In this context, it is necessary to analyze how the use of these technologies can influence the future and growth of smart cities.

For the development of this chapter, we have divided it into two sections: Deep Neural Networks and Convolutional Neural Networks. The following is a detailed description of each of them.

3.1 Deep Neural Networks

In this chapter, we will explore the role of deep neural networks in the context of smart cities, since they represent an innovative approach to address contemporary urban challenges through the use of advanced technologies. Neural networks, as a machine learning approach, offer a wide range of applications for improving the management and operation of smart cities, as their ability to analyze complex data, learn from patterns, and make real-time decisions contributes to the efficiency, quality, and sustainability of the cities in the future. As we advance in the era of urbanization, the use of neural networks or ML in smart city applications will continue to be a growing field with a significant impact on the quality of life of its inhabitants.

As mentioned above there are many areas for the application of neural networks in society, but in this section, we will focus on the approach of deep neural networks applied to art and smart tourism.

3.1.1 Current Status of Smart Tourism Research

In this section, the advances of both IoT in tourism and culture are reviewed, as well as the advances of existing tourism recommender systems and the differences with the proposed research proposal.

3.1.1.1 IoT in Tourism and Culture

Tourism has become a sector that contributes significantly to the economy of many cities. Cities have been increasingly competitive, aiming to expand their offerings and provide better experiences to attract more tourists.

Through the implementation of IoT and ICT, smart cities can develop new and interesting products for tourists [152] to provide a more immersive, richer, and personalized user experience [153]. Currently, there is research on IoT in the tourism field, such as geolocation, health care, or medical assistance [154] that allow the monitoring of tourists by following up on their state of health and promoting medical assistance if it is necessary, nature-based tourism activities, tourism services, and hospitality [155], and so on, aimed at boosting tourism development in cities, also assisted by the development of key technologies to build smart tourism destinations [156].

In the field of culture (museums/monuments) the deployment of IoT enables the development of monitoring and control systems for the protection and preservation of cultural heritage [157][158][159]. There are systems for early warning of disasters in any heritage monument [160]. Furthermore, as a solution option for indoor spaces, a platform has been developed that aims to create intelligent solutions and services to improve the interaction of visitors with cultural objects inside a museum [161]. To do so, the mobile application detects cultural objects equipped with Bluetooth Low Energy (BLE) sensor nodes. The cultural objects are ranked depending on user preferences and multimedia content related to the objects is recommended. Similarly, an indoor location-aware architecture uses image recognition and a localization technique based on BLE to provide the users with cultural content related to the observed artworks [162]. Both proposals [161][162] aim to improve the user's experience indoors (museum, art gallery, etc.). There are also implemented applications that propose a Deep learning (DL) methodology that enables the prediction of visitors' paths within a museum with two purposes: (1) to acquire more information about visitors' behaviors and (2) to predict the occupancy of the available rooms within a cultural space [163]. On the other hand, outdoor solutions include tourist recommendation systems.

3.1.1.2 Recommender Systems and Smart Tourism.

Recommender systems (RS) are models that let users find content or alternatives that may be of interest to him/her [164][165]. RS must be relevant, novel, or diverse to ensure that users receive precise information according to their requirements. RS have been deployed in entertainment, e-commerce, services, social media, etc. Recommender systems [166] suggest the most appropriate products or services for a particular user. A user's preference for an item is predicted based on information collected from different sources such as the user's inclination for particular products (songs, movies, etc.), the user's social information (ratings, followed, followers, etc.), his/her demographic features (age, gender, etc.), or the behavioral data from the Internet of Things (e.g., GPS, RFID tags, sensors, etc.). Collaborative filtering [167] is the most commonly implemented technique in recommender systems. It generates recommendations by identifying peer users with a preference/profile similar to the current user's. Collaborative filtering (CF) methods are classified as memory-based and model-based. Memory-based methods [166] usually use similarity metrics to obtain the distance between two users, or two items, based on each of their ratios. Model-based techniques incorporate machine learning to learn a model about a user. On the other hand, content-based recommendation systems suggest similar items based on a domain-specific notion of item content and a user profile. These algorithms try to recommend items similar to those that a user liked in the past or is examining in the present. In addition, context-based recommendation systems apply sensing and analysis of user context (location, user activity, etc.) to provide personalized recommendations [168]. When there is not enough information to make a recommendation that results in the “cold start problem”. The “cold start problem” refers to the entry of a new user into the recommender system. No recommendations can be made to this new user due to the lack of previous data [169].

Regarding tourism, some researchers have performed a systematic review covering all the major aspects of e-tourism management systems applied to smart tourism during the last few years. The techniques used by recommender systems are also classified. The most common are collaborative filtering, content, context, and hybrid models [170]. CF-based models employ user-user or item-item similarity to generate recommendations [171][172]. This is the most common method used by classic recommender systems, but it suffers from the cold start problem.

A content-based recommender system works with data that the user provides, either explicitly (rating) or implicitly (clicking on a link). The implicit preferences of the user for certain points of interest (PoI) such as restaurants, museums, and parks can be obtained from three main sources:

- Geotagged photos from social media [173][174][175][176][177][178][179].
- Location-based social networks [180][181].

Context-based models add information related to the user’s context (location, user activity, etc.) for its application to provide personalized recommendations [182]. For some research, user information is obtained from social networks, photo-sharing websites, and the location of the users without any personally identifiable information (PII) [173]-[181]. Employing machine learning in recommender methods can achieve higher predictive accuracy for online reviews (TripAdvisor) [183]. In addition, due to the significant achievements of deep learning in many fields such as natural language processing (NLP) and image processing, deep learning methods have been used for recommender systems [184], [185]. Cultural heritage recommendation systems [186][187][188][189] evaluate and classify the visitors’ behavior during a cultural event, provide information that matches the preferences of visitors, recommend artwork that could be of interest to the user, or even suggest routes. Furthermore, an architecture and conceptual framework for a hybrid tourism recommendation system based on big data and artificial intelligence has been presented [190]. As well as a hybrid recommendation system to combine the predictions of the content-based filtering (CB), collaborative filtering (CF) and demographic filtering (DF) approaches using a neural network model; the results compare with each of the traditional approaches individually, providing a better focus for recommending tourism sites in Taiwan [191].

3.1.1.3 Deep Learning.

Nowadays, different deep learning techniques have been created to facilitate the learning process in specific tasks. These techniques include:

- *The Multilayer Perceptron (MLP)* consists of a neural network with multiple hidden layers (deep neural network). The network is trained allowing the perceptron to adjust its parameters so the algorithm learns to achieve high accuracy for the task to be accomplished. This network is known for its wide range of applications in different areas of knowledge and is also considered one of the most versatile architectures thanks to its applicability.

- *An autoencoder (AE)* is an unsupervised neural network model trained to attempt to copy its input to its output. Traditionally, autoencoders were used for dimensionality reduction or feature learning [192].
- *The Convolutional Neural Network (CNN)* is a specialized kind of neural network for processing data that has a known, grid-like topology. Because its application is performed on two-dimensional arrays, this variation of a multilayer perceptron is very effective for computer vision tasks, such as image classification, segmentation, and other applications [193].
- *The Recurrent Neural Network (RNN)* is a neuronal network specialized in the processing of data sequences. It incorporates Long-Short-Term-Memory (LSTM) or Gated Recurrent Units (GRU) type layers that allow backpropagation through time; by connecting events that appear far apart in the input data since if the sequence is too long, the vanishing gradient problem may appear [194].
- The *Restricted Boltzmann Machine (RBM)* are two-layer surface neural network, that is the building block of deep-belief networks. The first layer of the RBM is called the visible or input layer, and the second is the hidden layer. The restriction in a constrained Boltzmann machine is that there is no communication between layers. Each node is a computational place that processes the input and starts making stochastic decisions about whether to transmit that input or not [195].
- The *Adversarial Network (AN)* is a type of convolutional neural network that generates images or songs, that are as realistic as possible, from a generator and a discriminator [196].
- *Deep reinforcement learning (DRL)* combines reinforcement learning and deep learning. RL addresses the problem of automatically learning optimal decisions over time, i.e. the problem of a computational agent learning to make decisions by trial and error. It consists of the following components: agents, environments, states, actions, and rewards. Deep reinforcement learning has been deployed in a diverse set of applications, including (but not limited to) robotics, video games, natural language processing, computer vision, education, transportation, finance, and healthcare [197].

From all the techniques mentioned above, for the present research, we will focus on deep neural networks.

3.1.2 Contribution: Internet of Things for Tourist Attraction Recommendations in Smart Cities.

Nowadays, when a person wants to take a trip to a specific city, the tourism companies offer package tours that include transport and accommodation. They can also provide additional services such as activities or outings during the holiday. These offers are adjusted to a heterogeneous group of people, although each tourist has different needs and characteristics. On the other hand, tourists who travel independently look for activities and attractions that best align with their interests. Therefore, before the trip, they need to search the Internet for possible options and their alternatives (opening hours of museums, ticket prices, weather forecasts, itineraries and transportation, etc.). In so doing, they find very extensive, dispersed, and disorganized information. This process represents long hours in front of a computer and causes an unnecessary loss of time. Due to the large number of attractions and activities that can be carried out while visiting a city, making a suitable itinerary can be a burdensome task for the tourist. This problem can be solved by recommender systems. Recently, deep learning-based recommender systems have also obtained very promising results [198],[199]. Furthermore, although IoT is considered a key concept in smart tourism, it is very rarely applied in smart tourism recommendation systems [170]. To fill this gap, an IoT-enabled deep learning-based recommendation system for tourist attractions is proposed in this section to enhance tourists' experience in a smart city. Travelers will enter the particular circumstances of a trip, such as traveling alone or with a companion, type of companion (partner, family with kids), traveling for professional or vacation purposes, and user side information (age of the traveler/s, hobbies, etc.) into the smart city app/website. This information is useful to improve the accuracy of the tourist attraction recommendations. For example, tourists who travel with children will avoid visiting many museums and will consider practicing outdoor activities, such as going to beaches or parks. Our proposed deep learning-based recommender system will process this personal set of input features to recommend the activities or attractions that best fit the tourist's profile. Furthermore, when the tourists are in the smart city, content-based information (already visited attractions) and context-related information (tourists' location, weather, time of day, etc.) are obtained in real-time using IoT devices. This information will allow our proposed deep learning-based tourist attraction recommender system to suggest additional tourism activities and/or attractions in real-time based on the tourist's own choices, the current time, and how far the new places are.

Therefore, we distinguish between two different cases (a) searching and planning activities before traveling and (b) looking for activities within the smart city.

In the first case (a), recommendations will be generated using model-based collaborative filtering and multi-label classification. In the second case (b), recommendations will be generated using a hybrid-based (model-based collaborative filtering, content, and context) approach and IoT context-related data (e.g., location, weather).

3.1.2.1 IoT smart tourism architecture and methodology

When the tourists are in the target smart city, content-based information (already visited attractions) and context-related information (tourists' location, weather, time of day, etc.) are obtained in real-time using the IoT. This information allows our proposed deep learning-based tourist attraction recommender system to suggest additional tourism activities and/or attractions in real-time based on the tourist's own choices, the current time, and the distance from the new places.

3.1.2.1.1 IoT smart tourism architecture

Our flexible IoT smart tourism architecture is shown in Figure 3.1. A three-layer proposal is presented as detailed below:

- *Device layer:* this layer is responsible for identifying objects and receiving information through sensors and monitoring the environment.
- *Fog layer:* this layer allows the transmission and processing of sensor data in a distributed manner for those services that are sensitive to latency.
- *Cloud layer:* this layer provides intelligent services that generate a global repository of relevant information, and provides recognition and learning patterns, which are fed from the other layers.

Next, we will go through the details of each layer.

3.1.2.1.1.1 Device layer

As shown in Figure 3.1, this layer is composed of physical devices such as sensors and actuators whose main function is to collect and process information. This broad set of intelligent IoT devices allows the system to monitor the user's movement within a smart city at any time, from any computer or mobile device. The collected information is securely stored in the cloud; analyzed and processed by our machine-learning algorithm on the edge servers, to provide user-specific recommendations.

The suggested IoT smart tourist devices are the *Global Positioning System, temperature sensors*, *RFID, sensors,* and *video cameras*. When tourists go sightseeing in the smart city, the attractions that they visit are registered using *GPS*. This way, we determine which recommendations the users acted upon, and recommend new similar attractions still based on their own preferences, their current location, and the current time of day.

One of the main elements to consider before a trip or sightseeing tour is the weather; through *temperature sensors,* the weather conditions are predicted to define possible routes and activities that can be carried out outdoors (e.g., parks) or indoors (e.g., museum visits) depending on the weather.

Tourists' behaviors can be captured more precisely by RFID, *video cameras,* and other sensors located in strategic places, such as stores, museums, churches, or entertainment places; this constant monitoring allows for updating the tourists' profile and improving future recommendations [200].

Furthermore, photographs uploaded on the web by tourists are a very powerful tool to obtain additional information about the user (granted that access is allowed by the user). They are processed through image recognition to obtain behavioral patterns and make even more appropriate recommendations.

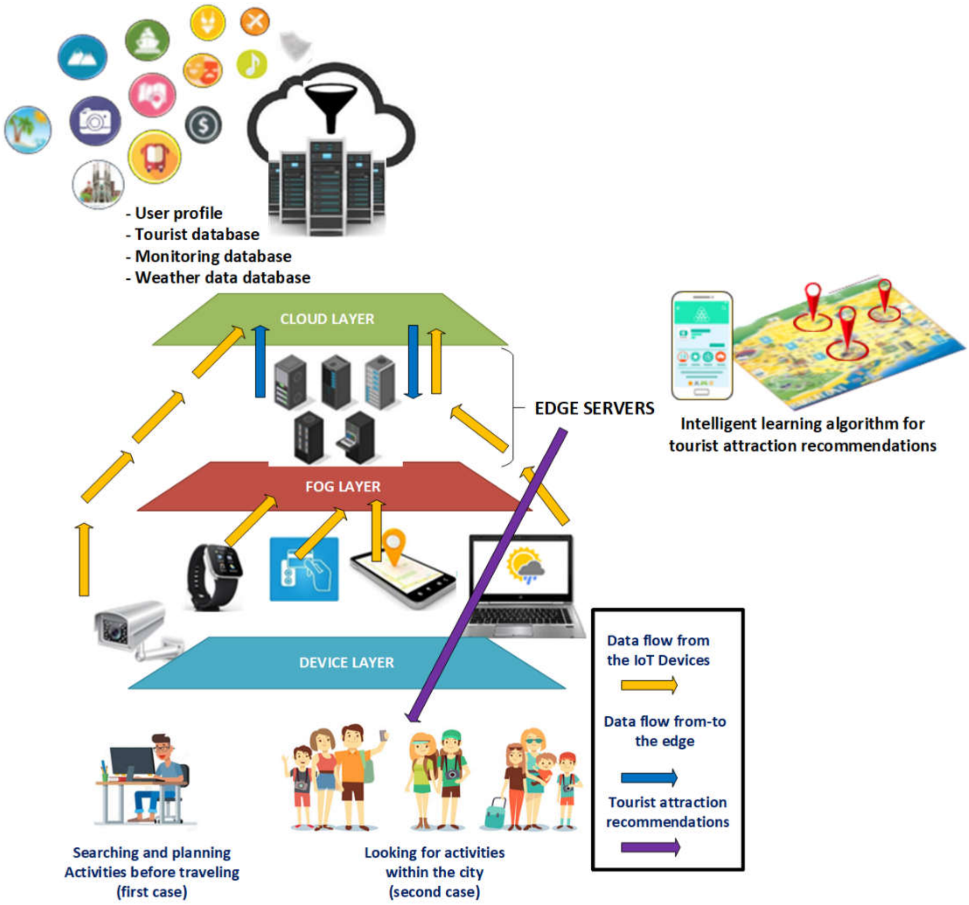

- User profile
- Tourist database
- Monitoring database
- Weather data database
CLOUD LAYER
EDGE SERVERS
Intelligent learning algorithm for tourist attraction recommendations
FOG LAYER
DEVICE LAYER
Data flow from the IoT Devices
Data flow from-to the edge
Tourist attraction recommendations
Searching and planning Activities before traveling (first case)
Looking for activities within the city (second case)

Figure 3.1. Schematic representation of the IoT-based smart tourism architecture.

3.1.2.1.1.2 Fog layer

The *fog layer* is shown in Figure 3.1 as the middle level in the current architecture; it offers real-time analysis and preprocessing services; afterward, the main information is transferred upwards to the servers in the cloud for storage and further treatment.

The sensors capture information from the *device layer* to send to the network, which can be congested with a huge amount of data. The *fog layer* should be understood as a part of the distributed architecture that expands between the cloud and the edge networks. This improves efficiency and reduces the volume of data transferred because critical tasks (such as computing, communication, storage, and decision-making) are distributed and closer to where the data is generated [201]. Therefore, the workload of the cloud and user devices is reduced. This is particularly important for time-sensitive IoT applications that require very low latency.

3.1.2.1.1.3 Cloud layer

This layer is responsible for the delivery of user-specific application services. It allows processes to be transferred to central servers distributed throughout the world, connected to the Internet through a high data throughput connection. It also supervises the applications of the Internet of Things, implementing decision-making processes based on Big Data analysis [202].

This layer receives all the data (which does not require immediate processing) generated by the different layers. It then carries out the functions of processing, analysis, and storage, becoming a global repository of relevant information to be used for automation or decision-making at a later time.

3.1.2.1.2 Proposed methodology

At this level, our machine-learning algorithm operates to recommend tourists in smart cities the best attractions/sights to visit according to their information profile and their IoT data. These suggestions are adjusted for each user to obtain a better experience when visiting a certain smart city. Since the raw data generated from sensors, GPS, etc. can be voluminous, rather than forwarding it to the cloud for processing, the idea is to do as much processing as possible in the fog layer using computing units. That way processed rather than raw data is forwarded to the cloud (or the devices in the device layer if necessary). A tourist database located in the cloud stores the dataset related to the tourist profile. A monitoring database located in the cloud stores the real-time location of tourists in the smart city and the places that they have already visited. The mobile device of the tourist monitors in real-time the tourist's location with the GPS and forwards this data to the monitoring database. Another database located in the cloud stores weather data. The weather sensors located in the device layer sense and forward the temperature to this database. The weather forecasts are also stored.

An app related to our proposed tourist attraction recommender system is installed on the mobile devices of the tourists. The proposed DNN with a multi-class classification algorithm, running on the edge servers will access the data of the different databases located in the cloud and use it as input data to return the most appropriate tourist attractions according to the user's profile. Figure 3.1 illustrates the different data flows. It can be observed that the raw data is transferred to the fog or the cloud. The machine learning algorithm returns an appropriate attraction recommendation to the device (mobile app) of the tourist based on predictions. In the following section, we will explain in detail our deep neural network algorithm tailored to our proposed approach.,

3.1.2.2 The proposed deep learning algorithm for smart city tourism

We have developed a tourist attraction recommendation system based on deep learning. We distinguish between two cases; the first one is when the tourist plans his/her trip and the second one is when the tourist is already in the smart city and is looking for new alternatives during his/her visit.

Figure 3.2 shows the block diagram of our proposal. A traveler will enter the particular circumstances of his/her trip (traveling alone or with a companion, type of companion such as a partner, family with kids, etc.) as well as user side information (age of the traveler/s, hobbies, etc.) into the smart city app/website. Our proposed deep learning algorithm will process this personal set of input features to predict and recommend the tourist activities or attractions that best fit his/her profile. For this purpose, the system must be trained with previously collected data, which consists of information about the most relevant tourist attractions in the city and the information provided by tourists who have previously visited the city. The information provided by tourists will be collected through surveys, which will contain information about the attributes of the tourists (tourists' profile) (dataset). Furthermore, for the second case, the system will pick up information (location, weather forecast, and already visited places) from the database collected by the IoT devices in real-time. These attributes will be the input data that will be normalized and processed by our deep-learning algorithm.

As previously described in section 3.1.1.3, there is a diversity of deep learning techniques. The use of different techniques for the same application provides better results in some cases than others. Therefore, the MLP technique is used in this research to achieve the proposed objective, which is to develop a tourist attraction recommender system. After having reviewed the different deep learning techniques and considering that we are doing supervised learning, we have decided to use the multilayer perceptron technique which consists of a neural network with multiple hidden layers (deep neural network). The network is trained by the perceptron adjusting its parameters. This way, the algorithm learns intending to achieve a high accuracy for the task to be accomplished. The output provides recommendations for tourist attractions according to the user's profile.

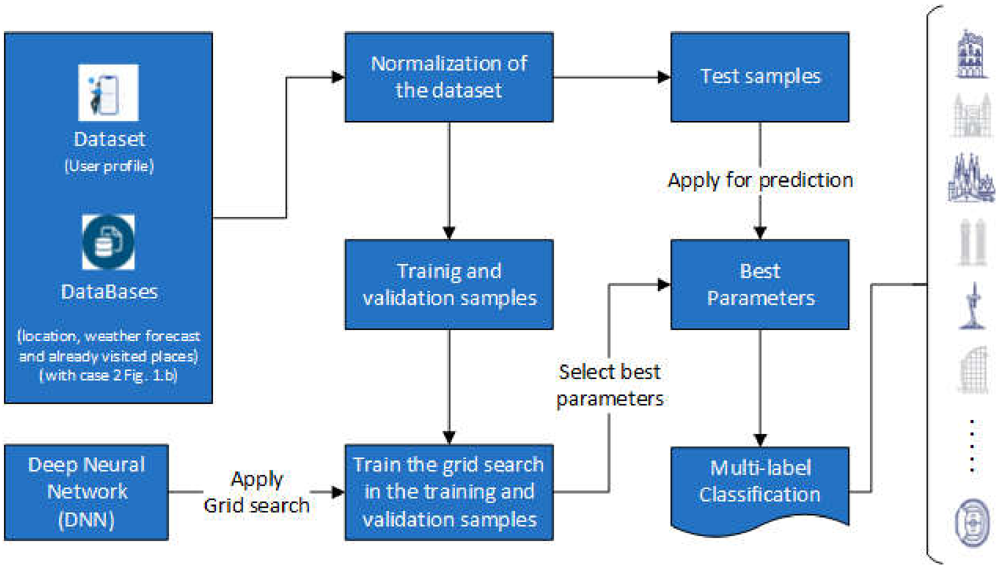

Dataset
(User profile)
DataBases
(location, weather forecast and already visited places)
(with case 2 Fig. 1.b)
Normalization of the dataset
Test samples
Apply for prediction
Trainig and validation samples
Best Parameters
Select best parameters
Deep Neural Network (DNN)
Apply Grid search
Train the grid search in the training and validation samples
Multi-label Classification

Figure 3.2. Block diagram and workflow of the proposed method.

3.1.2.2.1 Proposed DNN

The proposed deep neural network consists of several hidden layers with a required number of neurons. A vectorized representation of neurons in a hidden layer is given by

$$a^{[l]} = f(W^{[l]T}a^{[l-1]} + b^{[l]}) \tag{3.1}$$

where $a^{[l]}$ is a vector and each element represents a neuron in layer l, $W^{[l]}$ is the weight matrix in layer l such that $W^{[l]} \in R^{i \times j}$, where i is the number of nodes in the hidden layer l and j is the number of nodes in the previous layer (including the bias term $b^{[l]}$). Each neuron in the network is a nonlinear combination of inputs $a^{[l-1]}$ weighted by the parameters w_i. f is the activation function. The proposed model implements two activation functions for the hidden layers and the output layer.

The rectified linear unit (ReLU) has been used as activation function for the hidden layers:

$$f(x) = max(0, x) \tag{3.2}$$

The sigmoid function has been used as activation function for the output layer:

$$f(x) = \left(\frac{1}{1 + exp^{-x}}\right) \tag{3.3}$$

For predicting the different tourist sites, we consider four possible events, according to the number of days of the stay. Thus, when the tourist spends more days in the city, we can recommend a greater number of tourist attractions (see Table 3.2 in Section 3.1.2.3.3). For example, for tourists that stay one to three nights in the smart city, the system will recommend 8 out of 40 possible tourist attractions.

The output layer is composed of 40 neurons, one for each of the tourist sites that the system can recommend; the neurons are activated according to the number of recommendations made to each user depending on his/her profile (see Figure 3.3). The output of the last feed-forward layer is passed through a sigmoid activation function to scale each of its element values in the range [0,1]. The model is trained using binary cross-entropy as the loss function, which is defined as

$$L = \quad \frac{1}{T}\sum_{i=1}^{T}\sum_{j=1}^{U}(y_{ij}.log\left(\sigma\left(z_{ij}\right)\right) + (1 \quad y_{ij})\,log\,(1 \quad \sigma\left(z_{ij}\right)) \tag{3.4}$$

where σ denotes the sigmoid function, z_{ij} is the j^{th} element in z_i, and y_{ij} is the ground truth value for j^{th} tourist (label) and i^{th} tourist attraction. U is the number of tourists and T is the number of tourist attractions.

Gradient descent finds the minimum of an objective function by taking steps proportional to the negative of the gradient at the current point. A learning model estimates the weights by computing partial derivatives of the weight vector at each point and stopping when the minimum of the error function is reached.

$$w_{k+1} = w_k \quad \alpha\frac{\partial J(w,b)}{\partial w} \tag{3.5}$$

$$b_{k+1} = b_k \quad \alpha\frac{\partial J(w,b)}{\partial b} \tag{3.6}$$

where α is the learning rate, $J(w,b)$ is the cost function.

To compute the derivatives for a neural network, we apply the backpropagation technique to minimize the cost function using gradient descent. From (1), let $z^{[l]}$ represent the linear combination of weights and inputs in layer l, such that $z^{[l]} \in \mathrm{R}^{\mathrm{i}}$ where i is the number of nodes in layer l.

$$z^{[l]} = W^{[l]T}a^{[l-1]} + b^{[l]} \tag{3.7}$$

In addition, to fine-tune our system, each layer will be followed by Batch Normalization and Dropout. The selection of each parameter that composes this deep neural network will be explained later after the implementation of the grid search method, which is detailed in the next section.

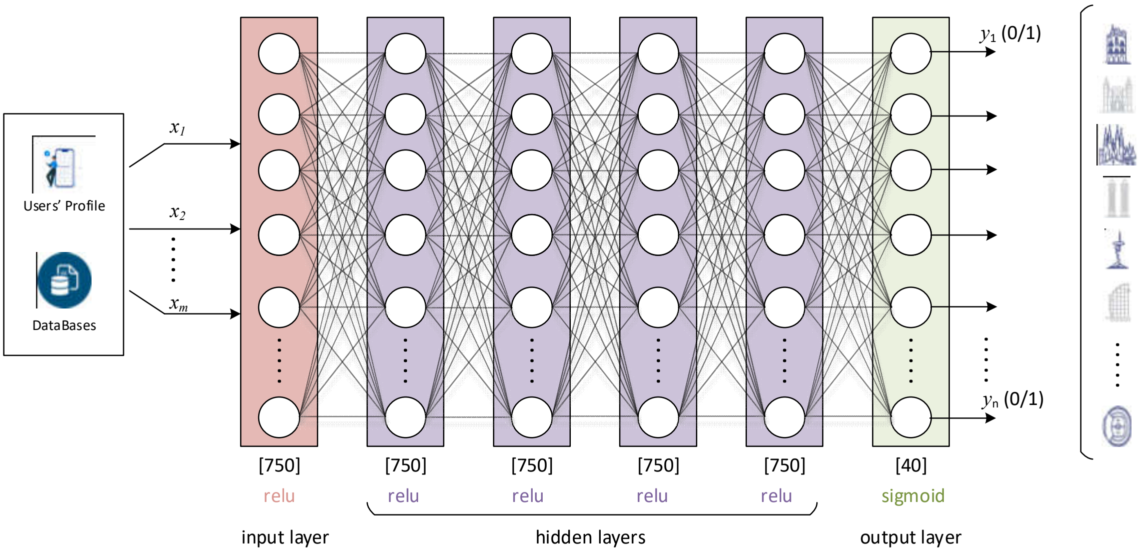


Figure 3.3. DNN structure created for the proposed recommender system with multi-label output.

3.1.2.3 Experiments and results

Next, we report the results of the experimental evaluation of the proposed tourist attraction recommendation system.

3.1.2.3.1 Smart city selection

We have selected Barcelona (Spain) as a reference for the implementation of our research since it is considered within the world ranking among the 20 most visited cities by foreign tourists (see Figure 3.4) [203] and among the 10 most visited cities in Europe (see Figure 3.5) [204].

Moreover, Barcelona is considered worldwide as a reference for smart cities in mobility, transport, urban planning, governance, technology, etc. [205][206]. Due to the great number of tourists that visit this city every year, we can count on a great variety of tourist profiles [207].

According to the data published [207], the profiles of tourists arriving at this tourist destination from 2014 to 2018 have the following characteristics: 40.44% of tourists are women, 59.56% are men, 65.98% travel for vacations, and 49.54% visit this city for the first time. The average age of visitors is 37 years, and 95% use various mobile applications to search for information on the Internet. In addition, the average stay is one week.

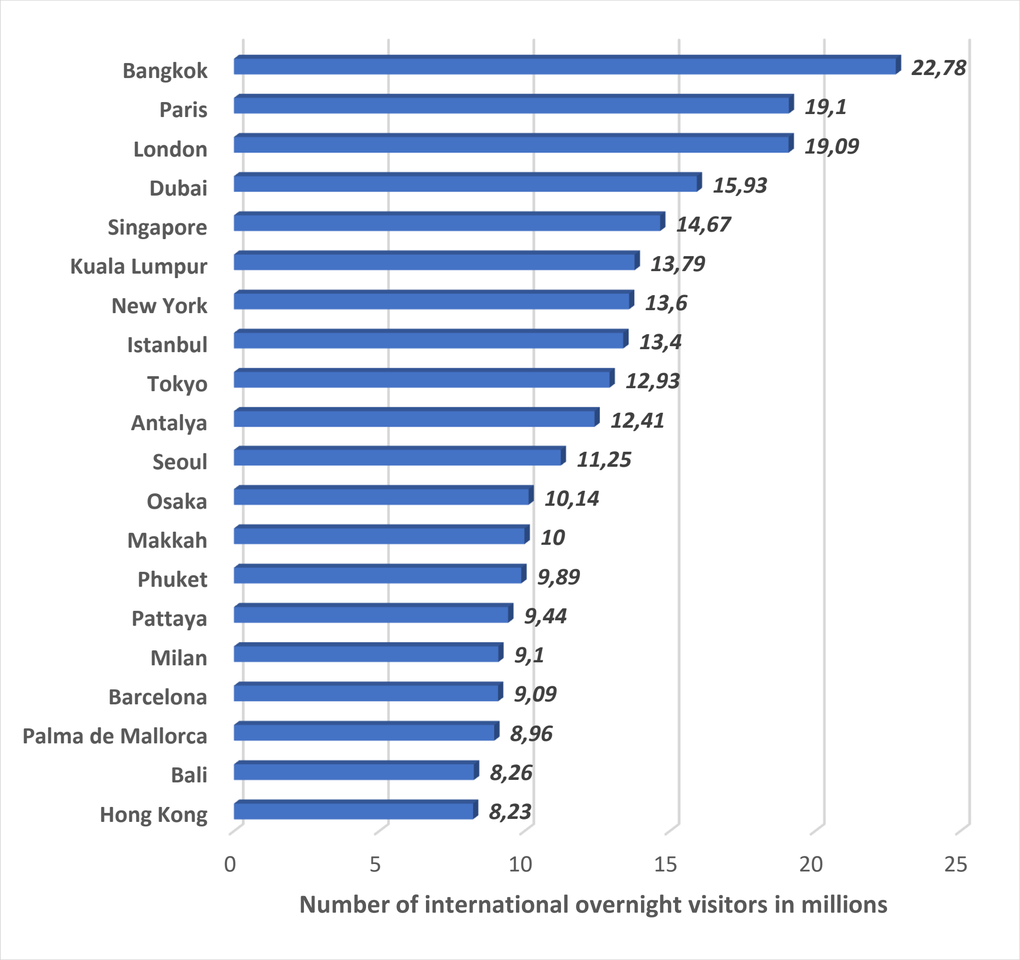


Figure 3.4. International overnight visitors in the most popular city destinations worldwide in 2018 (in millions).

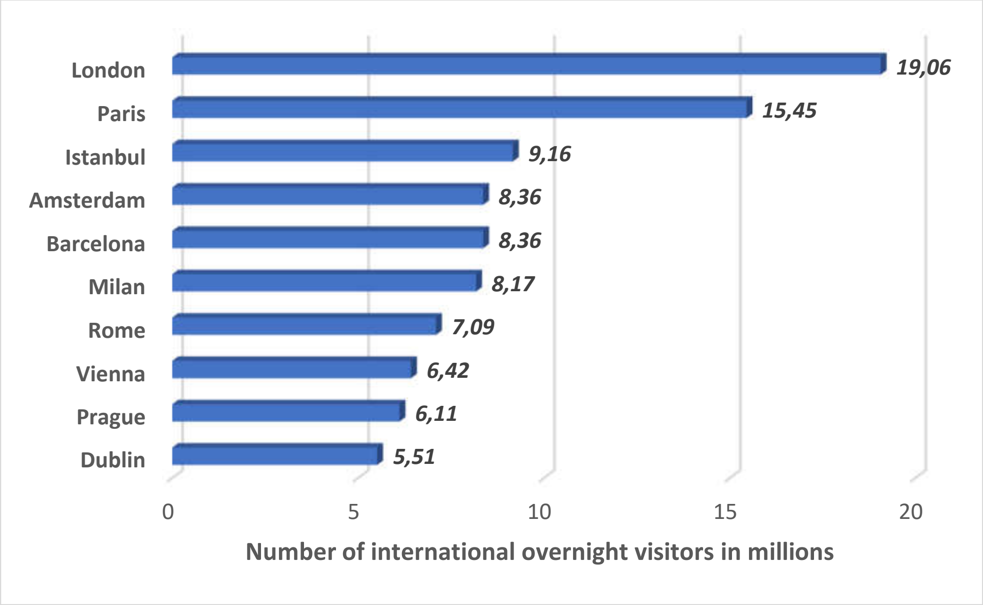


Figure 3.5. Number of international overnight visitors in the most popular European city destinations in 2016 (in millions).

The city of Barcelona has many and varied tourist attractions to meet the requirements and preferences of the visitors. Many of these visitors come to the city to change their daily routines, alone, in groups of friends, or with the family.

Furthermore, the city of Barcelona keeps accurate records of the most visited places by tourists in recent years (see Table 3.1) [208].

Table 3.1. Number of visitors to the tourist attractions in Barcelona for the period 2014-2018.

	2014	2015	2016	2017	2018
Expiratory temple of the Sagrada Familia	3.260.880	3.722.540	4.561.848	4.527.427	4.661.770
Park Güell	2.598.732	2.761.436	2.958.901	3.120.733	3.136.973
FC Barcelona Museum President Núñez	1.530.484	1.785.903	1.947.014	1.848.198	1.730.335
The Barcelona Aquarium	1.590.420	1.549.480	1.587.828	1.626.193	1.631.108
Poble Espanyol de Montjuïc	1.236.664	1.221.647	1.299.376	1.299.386	-
The Born Cultural Center	1.894.400	1.486.228	1.306.230	1.190.762	1.080.079
Casa Batlló	930.000	992.126	-	1.136.000	1.062.863
CosmoCaixa Barcelona	739.649	733.778	757.245	884.636	1.045.961
Picasso Museum	919.814	1.008.125	954.895	1.046.190	978.483
Palau Robert	810.000	715.000	827.957	865.776	976.276
Catalonia Foundation. Stone mine	932.356	990.112	1.207.087	972.508	934.524
National Art Museum of Catalonia (MNAC)	718.230	717.211	820.516	866.271	891.346
CaixaFòrum Barcelona	775.068	775.020	753.944	748.140	863.605
Montjuïc Castle	577.639	670.526	734.460	761.729	831.210
Barcelona History Museum	973.034	916.517	926.571	926.184	816.989
Barcelona Zoo	1.057.188	1.004.069	965.292	834.885	785.992

3.1.2.3.2 Input data

Next, we explain in detail our dataset as well as the IoT devices database.

3.1.2.3.2.1 User profile

Our dataset is a collection of data acquired through surveys designed to profile each tourist, to generate predictions and recommendations. These recommendations are based on the preferences of other tourists with similar profiles.

The survey was carried out in the municipality of Barcelona during the summer of 2019. 1000 surveys were conducted, which represent 1000 different profiles of tourists or groups of tourists. The interviews were done on paper or through digital media. The surveys were conducted close to tourist attractions or in their vicinity (museums, monuments, etc.), city access points (airport, train station, bus station, and cruise terminal), and hotels. For each of these places, the interviewees were chosen randomly throughout the day during working days, weekends, and holidays.

For the preparation of the survey, certain characteristics of the tourist's profile were considered; these attributes (X_i) were associated with the data samples and are grouped into the following categories:

a) *Members Attribute:* One attribute with five categorical data that contain the characteristics of the traveler, whether he/she travels alone, as a couple, in a family with or without children, or with a group of friends.

b) *Reason for the Trip Attribute:* One attribute with four categorical data denotes the characteristics that motivate the trip, which range from vacation to professional.

c) *Age Attributes:* Six attributes which group the ages of the members of the travel group.

d) *Activities Attributes:* Nine attributes with the different activities that can be carried out by the travel group during the stay. This makes it possible to identify the different activities associated with the respective ages of the travel group members.

e) *Sports Activities Attribute:* One attribute with three categorical data that identify the types of sports activities, ranging from the most traditional to those of an extreme nature.

f) *Number of Days of Stay Attribute:* One attribute with five categorical data that indicate the number of days available for visiting the city.

g) *Time of the Year / Dates of Stay Attribute:* The dataset also has one attribute with four categorical data that indicate the season of the year or month during the trip.

3.1.2.3.2.2 Database from IoT Devices

Once the tourist is already in the city of Barcelona, the system uses GPS to detect his/her location and distances from other landmarks (thirteen attributes), as well as sensors to detect the temperature (an attribute with six categorical data), weather forecast (one attribute with three categorical data) and hour (one attribute with twelve categorical data) to recommend places according to his/her location. Furthermore, the already visited tourist attractions will be detected by GPS. A database of these places will be maintained. After finishing the visit to a tourist attraction, the tourists will rate the attraction based on his/her preferences. This data is fed back into the system for future recommendations based on the tourist's preferences.

3.1.2.3.3 Output Data

We have grouped stays into four categories depending on the number of days of the stay. Thus, when the tourist spends more days in the city, we can recommend a greater number of tourist attractions to visit as shown in Table 3.2.

Table 3.2. Recommended tourist attractions based on the total visit duration.

Number of days to stay	Number of tourist attractions
One to three days	Eight sites
One week	Twelve sites
Two weeks	Sixteen sites
More than three weeks	Twenty-four sites

Our proposed DNN is trained for the tourist attraction classification task. Classification is the goal to learn a mapping from inputs x to outputs y, where $y \in \{1, \ldots, C\}$, with C being the number of classes. If $C > 2$ and the class labels are not mutually exclusive (e.g. somebody is classified as tall and strong), it is called multi-label classification [209]. The output of the proposed DNN generates a multi-label classification with the number of sites predicted according to the number of days of the stay. 40 tourist sites can be recommended according to the profile of each tourist.

3.1.2.3.4 Experimental Settings

For training, from the surveys, we get the user's profile information with his/her likes and preferences. We also use the information from the database of IoT devices, thus generating an attribute matrix of 1000 x 20 (1000 samples, 20 attributes) (X_i) for the first case and 1000 x 36 (1000 samples, 36 attributes) for the second case, also from the survey we get the information of the places he/she has visited and that have been most liked; considering the days of his/her visit getting a 1000 x 40 label matrix $(\mathrm{Y_i})$ with eight, twelve, sixteen, or twenty-four sites. The experiments were performed with 1000 samples, which were divided into two parts keeping ratio 7:2:1, for training, validation, and testing, respectively. The experiments were carried out on a Dell computer 2.5 GHz Intel (R) Core(TM) processor with 16 GB RAM. The algorithms were implemented in several Jupyter Notebooks in version 6.0.3 installed with the Anaconda programs suite, developed by Python. We have implemented the algorithms using the Python Keras library, and accuracy, loss, F1-score, recall, and precision have been selected during the training process as metrics to evaluate the performance of the algorithms.

3.1.2.3.5 Neural Network Modeling and Optimization

Machine Learning models have several parameters to adjust their behavior to reduce overfitting. The most common alternative is to use dropout, which has a parameter that must be tuned for optimal performance [210]. An alternative to reduce overfitting is to optimize the set of parameters through a process known as *grid search* and try to find the most appropriate combination that provides greater precision. The approach used by *grid search* is an exhaustive search by the brute force paradigm in which a list of values for different parameters is specified, and the computer evaluates the model's performance for each combination of these parameters to obtain the optimal set that gives us the highest performance [211]. The result is the values of the adjusted parameters such as the number of hidden layers, the number of neurons in each layer, the number of epochs, the size of the batch of the neural network, as well as the hyperparameters such as the dropout value. In this work, the grid search technique was used; after long training, it allowed us to obtain the optimal values for the correct performance of the algorithm. Which are represented graphically employing box and whisker plots.

3.1.2.3.5.1 Box and Whisker Plot

Box and whisker plots are graphical representations used to visualize the distribution of a data set and highlight its key characteristics. This type of graph provides information on the mean, median, quartiles, minimum, and maximum values, as well as outliers within a data set. It is plotted on a horizontally or vertically aligned rectangle.

This plot is composed of a rectangular box, where the longer sides show the interquartile range (IQR). This rectangle is divided by a vertical segment that indicates where the median is located. The ends of the box represent the first and third quartiles (Q1 and Q3, respectively), while the second quartile Q2 coincides with the median. The extended lines projecting from the box, called "whiskers", show the dispersion of the data beyond the quartiles. Outliers are represented as individual points outside the whiskers (see Figure 3.6).

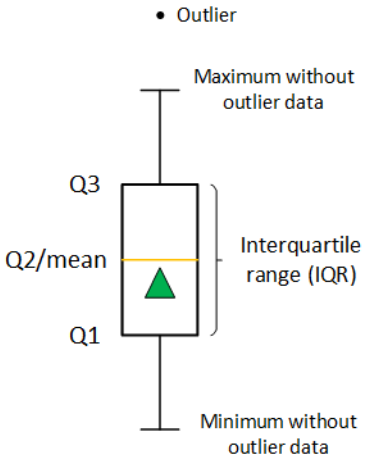

Outlier
Maximum without outlier data
Q3
Q2/mean
Interquartile range (IQR)
Q1
Minimum without outlier data

Figure 3.6. Box or whisker diagram representation.

In the figure, inside the box, the orange line corresponds to the mean representing 50% of the data, while the green triangle corresponds to the median, which is the midpoint that divides the data set into two equal parts.

This graphical representation is useful for comparing distributions of different data sets, identifying symmetry and concentration of the data, as well as detecting possible outliers that could influence the statistical analysis. Box plots offer a quick and visually effective way to summarize the dispersion and shape of data without detailing their full distribution.

3.1.2.3.5.2 Network Dimensions and Tuning

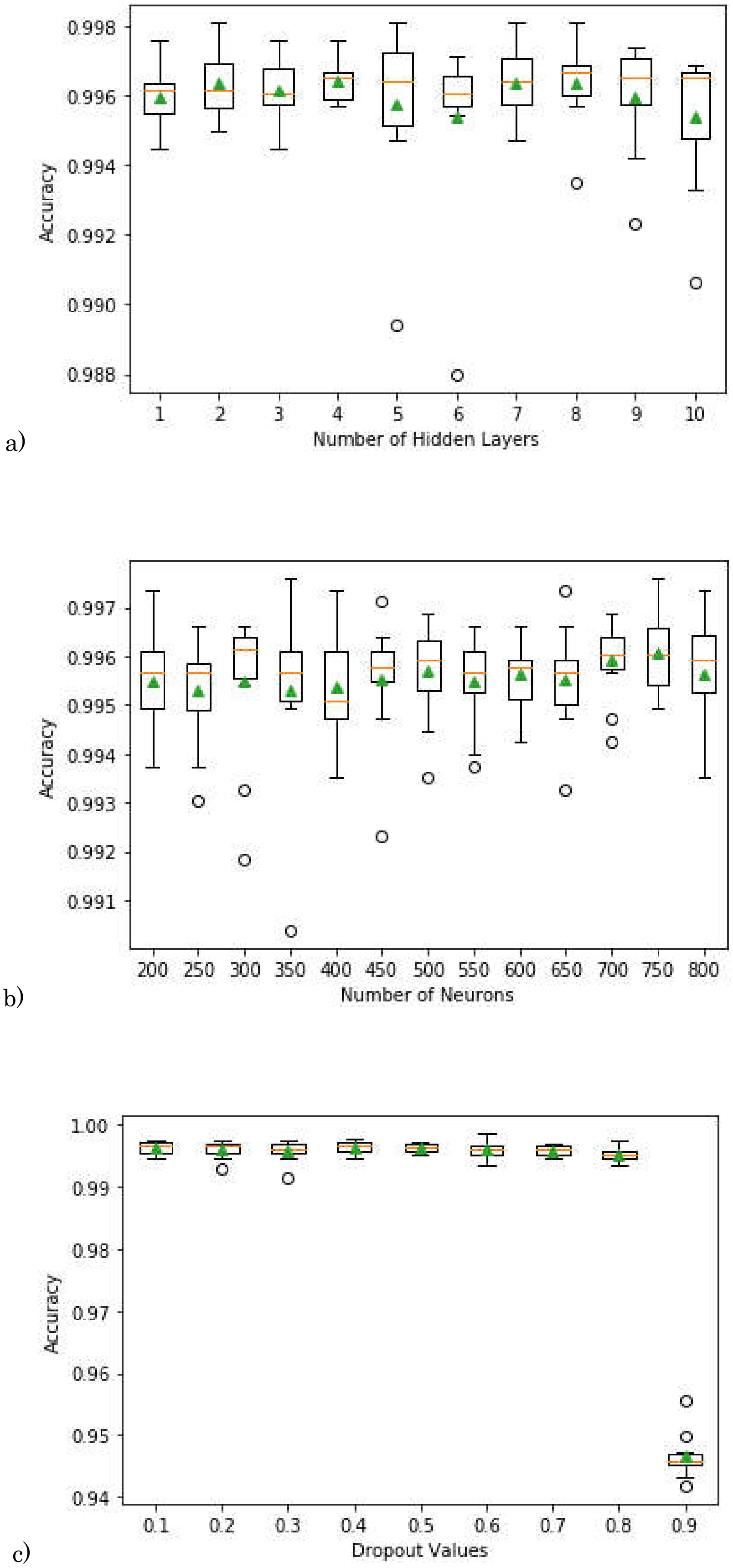
Two very important parameters for deploying a DNN are the depth and width of the network, i.e., the number of hidden layers and the number of neurons per hidden layer. On the one hand, the design of a shallow neural network can lead to the fact that in training it does not collect the correct information for an appropriate prediction. On the other hand, a very deep neural network may fall into overfitting in training, so the system learns the training data memorized, making it improper to make future predictions. Accordingly, different topologies of neural networks are tested with network depth varying from 1 to 10 hidden layers and from 200 to 800 neurons for each layer.

a) *Network depth*: Improvement in network performance is based on the depth of the network. In Figure 3.7a, box and whisker plots are shown; the performance of the DNN improves with a higher number of hidden layers. In Figure 3.7a, the average accuracy shows that the system obtains better performance when it reaches four hidden layers with an accuracy of 99.65%. Although in each case the system achieves a considerable value in most of the samples, we can notice the system with four hidden layers achieves better stability on training and it deteriorates when the number of hidden layers continues increasing meaning that the training may become inadequate.

b) *Network width*: This parameter measures the impact of the number of neurons per hidden layer on the performance of the overall system. Figure 3.7b shows the results with 4 hidden layers when the number of neurons varies between 200 and 800 neurons. We conclude that the performance of the neural network improves with a higher number of neurons per layer. The average accuracy shows that the system obtains a better performance when it reaches 750 neurons, achieving 1% higher performance than the case with 400 neurons.

c) *Network tuning*: To avoid overfitting we have considered the dropout technique. Figure 3.7c shows the most suitable value for an optimal performance of the neural network. The average accuracy shows that the system performs best when it reaches a value of 0.4 in a range of 0.1 and 0.9, reaching an optimal value of 99.8%. Note that the system achieves a remarkably high accuracy.

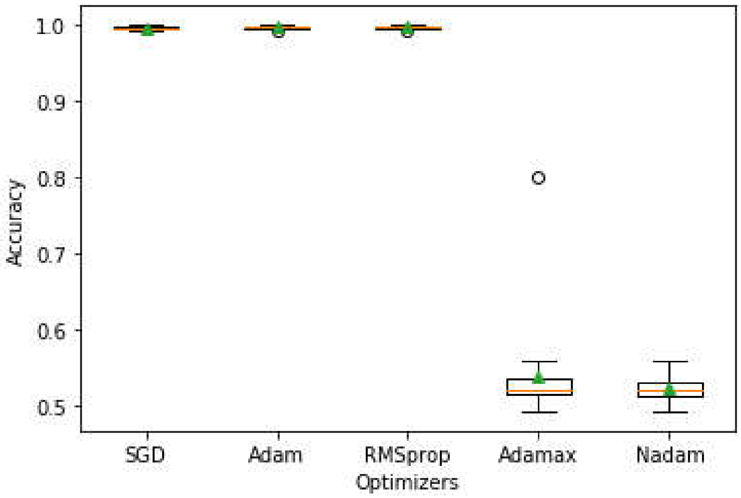
We have evaluated the Stochastic Gradient Descent (SGD), RMSprop, Adam, Adamax, and Nadam optimizers using the grid search technique. Grid search is a tuning technique that computes the optimal values for the hyperparameters. Figure 3.7d shows the box and whisker plot of the accuracies of each optimizer. The SGD optimizer achieves an accuracy of 99.55%, Adam 99.62%, RMSprop 99.61%, Adamax 53.70%, and Nadam 52.23%. In all cases, the learning rate is 0.001. The Adaptative Moment Estimation (Adam) optimizer has been selected to train the deep neural network because it obtains the best results. It minimizes the loss function and speeds up the training process.


0.998
0.996
0.994
0.992
0.990
0.988
Accuracy
1
2
3
4
5
6
7
8
9
10
Number of Hidden Layers
0.997
0.996
0.995
0.994
0.993
0.992
0.991
Accuracy
200
250
300
350
400
450
500
550
600
650
700
750
800
Number of Neurons
1.00
0.99
0.98
0.97
0.96
0.95
0.94
Accuracy
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
Dropout Values

a)

b)

c)



d)

Figure 3.7. DNN's performance. a) The box-and-whisker plot of accuracy versus the number of hidden layers. b) The box-and-whisker plot of accuracy versus the number of neurons per hidden layer. c) The box-and-whisker plot of accuracy versus dropout values. d) The box-and-whisker plot of accuracy versus most popular optimizers.

3.1.2.3.6 Training model

The deep neural network consists of 4 hidden layers, each with 750 neurons according to the above paragraph findings. The deep learning algorithm implementation was performed using the *Keras* library with the results provided by the grid search technique implementation. Table 3.3 summarizes the parameter values that should be adjusted for the optimal performance of the proposed algorithm. The Adaptive Moment Estimation (Adam) optimizer has been selected to minimize the loss function and speed up the training process.

Table 3.3. Best hyperparameter values found after the grid search process.

	Number of Hidden Layers	Number of Neurons / Hidden layer	Dropout	Learning rate	Optimizer
DNN	4	750	0.4	0.001	Adam

Adam optimizer (Algorithm 1) is one of the most popular gradient descent optimization algorithms because it is computationally efficient and has very little memory requirements. This method calculates the individual adaptive learning rate for each parameter from estimates of the first and second moments of the gradients.

Algorithm 1: Adam, our proposed algorithm for the training process.

```
1: Declare the parameters Objective function $f(\theta)$, hyperparameter learning rate $\alpha$, exponential decay rates $\beta_1$, $\beta_2$ for moment estimates, tolerance parameter $\epsilon > 0$ for numerical stability
```

```
2: Initialize first moment vector $m_0 = 0$, second moment vector $v_0 = 0$ and timestep $t = 0$
3: while $\theta_t$ has not converged do
    3.1 update timestep $t = t + 1$
    3.2 compute gradient of objective using $g_t = \nabla_\theta f_t(\theta_t \quad 1)$
    3.3 update first moment estimate and second moment estimate using eq. (3.8) and eq. (3.9), respectively.
    3.4 compute unbiased first and second moment estimate using eq. (3.10) and eq. (3.11), respectively.
    3.5 update objective parameters using eq. (3.12).
end while
4: return final parameter $\theta_t$
```

Adam algorithm first updates the exponential moving averages of the gradient (m_t) and the squared gradient (v_t) which is the estimates of the first and second moment. The hyperparameters β_1, $\beta_2 \in [0, 1)$ control the exponential decay rates of these moving averages as shown in the following equations:

$$m_t = \beta_1 m_{t-1} + (1 \quad \beta_1) g_t \tag{3.8}$$

$$v_t = \beta_2 v_{t-1} + (1 \quad \beta_2) g_t^2 \tag{3.9}$$

where g is the current gradient value of the error function for the neural network training.

Moving averages are initialized as 0. The moment estimates are biased around 0, especially during the initial timesteps. This initialization bias can easily be counteracted resulting in a bias-corrected estimate.

$$\widehat{m}_t = \frac{m_t}{1 \quad \beta_1^t} \tag{3.10}$$

$$\hat{v}_t = \frac{v_t}{1 \quad \beta_2^t} \tag{3.11}$$

Finally, we update the parameter as shown below

$$\theta_t = \theta_{t-1} \quad \frac{\alpha \widehat{m}_t}{\sqrt{\hat{v}_t} + \varepsilon} \tag{3.12}$$

We have used in our experiments for the Adam optimizer a learning rate $\alpha = 10^{-3}$ and two decay parameters β_1 = 0.9 and β_2 = 0.999 [212].

3.1.2.3.7 Experimental results

The recommendation of different tourist attractions is a very complex task because it must link the input attributes with the possible recommendations generated by the DNN algorithm. The two already mentioned stages or cases related to the smart city trip were tested, that is, (a) searching and planning activities before traveling and (b) looking for activities within the smart city of Barcelona.

To evaluate the effectiveness of this approach, the main indicators in the field of multi-label classification [213] were applied. Let D be a multi-label evaluation data set, consisting of $|D|$

multi-label examples $(X_i, Y_i), i = 1..|D|$. Let H be a multi-label classifier and $Z_i = H(X_i)$ be the set of labels predicted by H for example x_i.

$$Accuracy(H, D) = \frac{1}{|D|}\sum_{i=1}^{|D|}\frac{|Y_{Xi} \quad Z_{Xi}|}{|Y_{Xi} \quad Z_{Xi}\ |} \tag{3.13}$$

$$Precision(H, D) = \frac{1}{|D|}\sum_{i=1}^{|D|}\frac{|Y_{Xi} \quad Z_{Xi}\ |}{|Z_{Xi}|} \tag{3.14}$$

$$Recall(H, D) = \frac{1}{|D|}\sum_{i=1}^{|D|}\frac{|Y_{Xi} \quad Z_{Xi}\ |}{|Y_{Xi}|} \tag{3.15}$$

$$F1 \quad score(H, D) = 2\frac{\sum_i |Y_{Xi} \quad Z_{Xi}|}{2 \ \sum_i |Y_{Xi} \quad Z_{Xi}| + \sum_i |Y_{Xi} \quad Z_{Xi}| + \sum_i |Z_{Xi} \quad Y_{Xi}|} \tag{3.16}$$

3.1.2.3.7.1 First case

The first case refers to searching and planning before traveling. The tourist attraction recommender system suggests activities in a generic mode according to the data previously entered by the user based on his/her profile and particular circumstances of the trip, without taking into account distances, weather, or rain predictions.

For this first case, several tests have been performed to measure the accuracy of our model. Figure 3.8, and Figure 3.9 show the results for accuracy and loss, respectively, for training and testing data; our deep neural network on the training data reaches an accuracy, precision, recall, and F1-score of 99.6%, 99.9%, 99.9%, and 99.8%, respectively with the loss of 0.4%. For testing data, it reaches an accuracy, precision, recall, and F1-score of 99.7%, 99.9%, 99.9%, and 99.8%, respectively with the loss of 0.5%. Table 3.4 shows the accuracy, recall, and precision values during training and testing. We conclude that these results confirm the effectiveness of our proposed tourism recommendation system for this first case.

Table 3.4. Testing versus training accuracies, losses, F1 scores, recalls, and precisions for our DNN first case.

Data	Accuracy	Loss	F1 score	Recall	Precision
Training	0.996	0.004	0.998	0.999	0.999
Testing	0.997	0.005	0.998	0.999	0.999

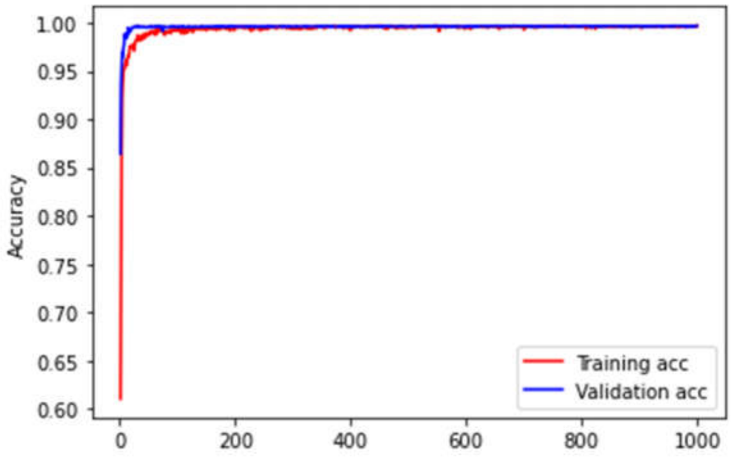


Figure 3.8. DNN model training and validation accuracies versus training epochs (first case).

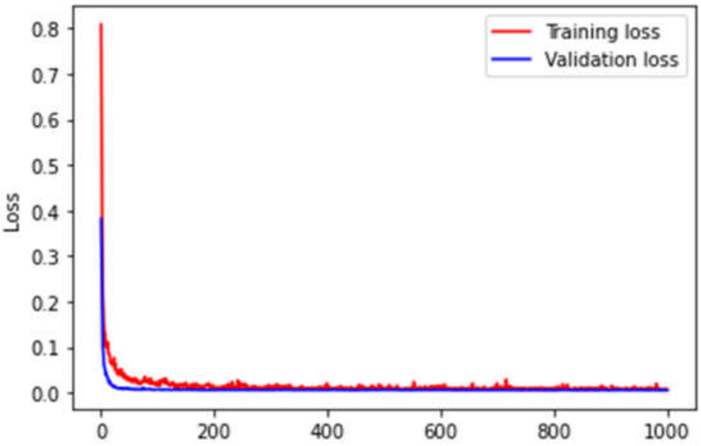


Figure 3.9. DNN model training and validation losses versus training epochs (first case).

3.1.2.3.7.2 Second case

In addition to the information previously entered by the user (user profile and particular circumstances of the trip), in this case, the tourism recommender system also uses information collected from various IoT devices: corresponding location, temperature, and weather forecasts. With all this information, the system recommends the tourist attractions that best suit the requirements of the user based on the weather forecast and the nearest locations.

For this second case, several tests were performed to measure the accuracy of our model. Figure 3.10, and Figure 3.11 show the results for accuracy and loss, respectively for training and testing data; after the tests were performed, our deep neural network on the training data reached an accuracy, precision, recall, and F1-score of 99.7%, 99.7%, 99.8%, and 99.8%, respectively with the loss of 0.1%. The testing data reaches an accuracy, precision, recall, and F1-score of 99.5%, 99.8%, 99.7%, and 99.8%, respectively with the loss of 3.7%. From Table 3.5, we conclude that these results confirm the effectiveness of our proposed tourism recommender system for our second case.

Table 3.5. Testing versus training accuracies, losses, F1 scores, recalls, and precisions for our DNN second case.

Data	Accuracy	Loss	F1-score	Recall	Precision
Training	0.997	0.001	0.998	0.998	0.997
Testing	0.995	0.037	0.998	0.997	0.998

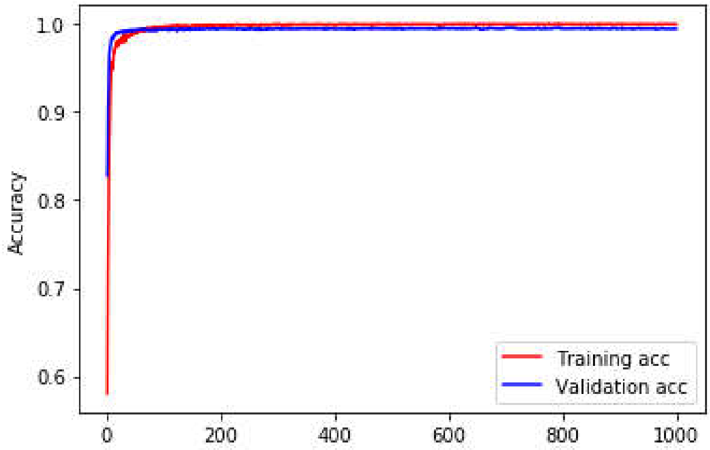


Figure 3.10. DNN model training and validation accuracies versus training epochs (second case).

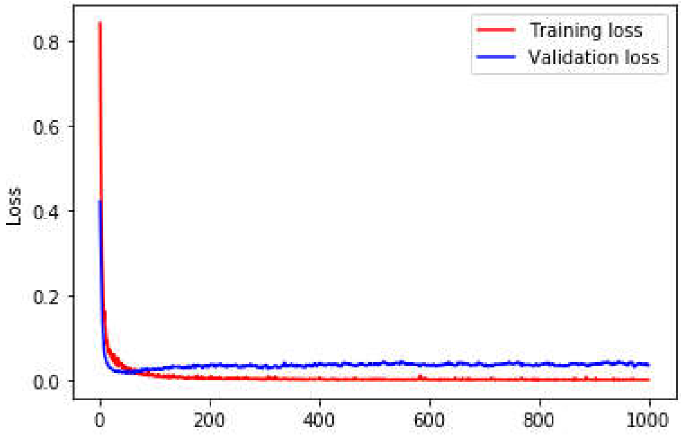


Figure 3.11. DNN model training and validation losses versus training epochs (second case).

3.1.2.3.8 Comparison to other models

Python libraries such as scikit-learn enable the deployment of many traditional models such as Support Vector Machines (SVM), k-nearest neighbors algorithm, random forest classifier, and so on. Figure 3.12 and Figure 3.13 show the classification algorithms and models available for implementation in different use cases [214]. For this research, we will focus on supporting multilabel classification.

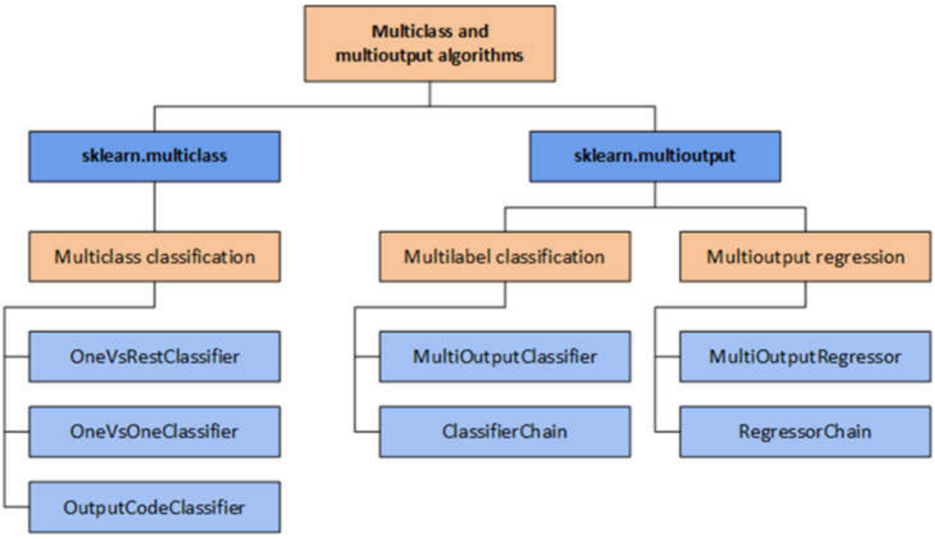
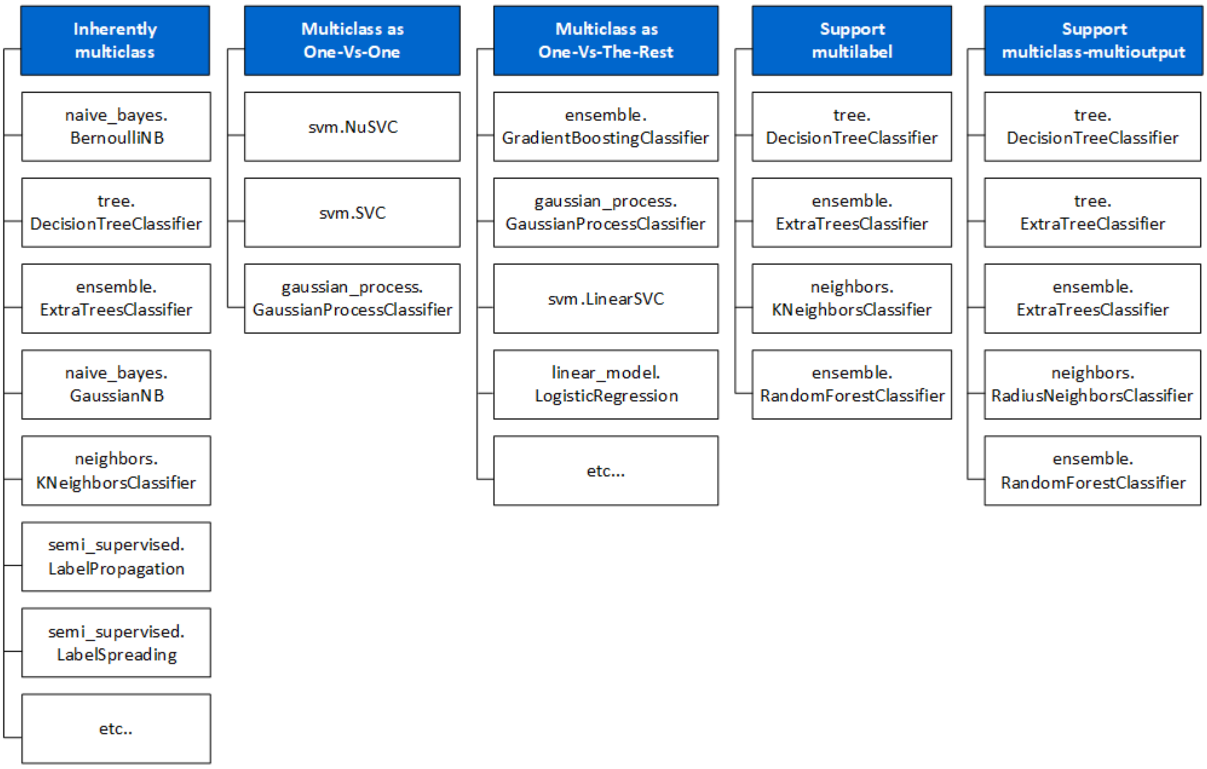


Figure 3.12. Classification scheme of multiclass and multioutput modules supported by scikit-learn.



Inherently multiclass	Multiclass as One-Vs-One	Multiclass as One-Vs-The-Rest	Support multilabel	Support multiclass-multioutput
naive_bayes. BernoulliNB	svm.NuSVC	ensemble. GradientBoostingClassifier	tree. DecisionTreeClassifier	tree. DecisionTreeClassifier
tree. DecisionTreeClassifier	svm.SVC	gaussian_process. GaussianProcessClassifier	ensemble. ExtraTreesClassifier	tree. ExtraTreeClassifier
ensemble. ExtraTreesClassifier	gaussian_process. GaussianProcessClassifier	svm.LinearSVC	neighbors. KNeighborsClassifier	ensemble. ExtraTreesClassifier
naive_bayes. GaussianNB		linear_model. LogisticRegression	ensemble. RandomForestClassifier	neighbors. RadiusNeighborsClassifier
neighbors. KNeighborsClassifier		etc...		ensemble. RandomForestClassifier
semi_supervised. LabelPropagation				
semi_supervised. LabelSpreading				
etc..				

Figure 3.13. Overview of scikit-learn estimators that integrate multi-learning, organized by strategy.

For our analysis, we compared our proposal with the models provided by the scikit-learn library that support multi-label classification. These classifiers are decision tree, extra tree, k-Nearest Neighbor (kNN), and random forest. We have compared our proposal with all of these classifiers. Tests were also performed for the two proposed cases.

We also employ cross-validation as a statistical method for the evaluation and comparison of learning algorithms. Table 3.6 shows the results achieved by our deep neural network model and the traditional models implemented for the first case.

Table 3.6. Comparison of accuracy, F1-score, recall, and precision for our proposed DNN and other traditional machine learning algorithms (first case).

	Accuracy	F1-score	Recall	Precision
Our Neural Network	0.997	0.998	0.999	0.999
Decision tree	0.892	0.92	0.90	0.98
Extra tree	0.891	0.91	0.91	0.90
K-nearest neighbor	0.905	0.92	0.89	0.89
Random forest	0.384	0.52	0.40	0.62

The analysis shows that our proposed DNN model achieves the highest accuracy, precision, recall, and F1-score of 99.7%, 99.9%, 99.9%, and 99.8%, respectively, compared to traditional machine learning models. We can also observe that from the traditional models, the k-nearest neighbor achieves the highest accuracy, precision, recall, and F1-score of 90.5%, 89%, 89%, and 92%, respectively, followed by decision tree with accuracy, precision, recall and F1-score of 89.2%, 98%, 90%, and 92%, respectively.

Table 3.7 shows a summary of the scores achieved by our deep neural network model and each of the traditional models implemented for the second case.

Table 3.7. Comparison of accuracy, F1-score, recall, and precision for our proposed DNN and other traditional machine learning algorithms (second case).

	Accuracy	F1-score	Recall	Precision
Our Neural Network	0.995	0.998	0.997	0.998
Decision tree	0.909	0.91	0.88	0.87
Extra tree	0.886	0.89	0.87	0.85
K-nearest neighbor	0.584	0.60	0.61	0.56
Random forest	0.370	0.69	0.65	0.68

The analysis shows that our proposed DNN model in this second case, achieves the highest accuracy, precision, recall, and F1-score of 99.5%, 99.8%, 99.7%, and 99,8%, respectively, compared to traditional machine learning models. We can also observe that from the traditional models, decision tree achieves the highest accuracy, precision, recall, and F1-score of 90.9%, 87%, 88%, and 91%, respectively, followed by the extra tree with accuracy, precision, recall and F1-score of 88.6%, 85%, 87% and 89%, respectively.

In both cases our algorithm has a better performance, so with these values, we can appreciate that our algorithm can perform future predictions adequately.

3.1.2.3.8.1 Discussion

The proposal has been carried out in the Barcelona city, considering the most emblematic places of the city. All the information collected for the realization of this proposal is based on real data (location, temperature, weather forecasts, and profile of each user). The implementation of our model using the Keras library allows incorporating and adjusting more parameters (batch normalization, dropout, etc.). We have obtained better results in comparison with other traditional models provided by the scikit-learn library. In addition, due to the implementation of the grid search technique, we can get the most suitable configuration of these parameters for the optimal performance of the system. The results show that our algorithm can make future predictions to obtain optimal recommendations according to the information provided by the tourist profile as well as the information provided by the IoT devices.

3.1.2.4 Conclusions

This research proposal aims to analyze the impact of using a deep neural network (DNN) topology in the context of tourist attraction recommendations involving multi-label classification. Two specific use cases are addressed: a) searching and planning activities before traveling and b) looking for activities within the smart city.

The results achieved in this research indicate that the performance of the algorithm is significantly improved as the depth of the neural network is increased. To select the DNN topology used, a grid search technique was employed which led to the choice of a configuration with four hidden layers, each composed of seven hundred and fifty neurons. Also, a dropout value of 0.4 was used to mitigate overfitting during the neural network training process.

In addition, a comparison was made between our DNN classifier and traditional models. This comparison yielded significant results in both use cases. In the first case, the highest performance metrics were obtained, with 99.7% accuracy, 99.9% precision, 99.9% recall, and 99.8% F1 score. In the second case, our DNN classifier also achieved the highest accuracy, precision, recall, and F1 score, with values of 99.5%, 99.8%, 99.7%, and 99.8%, respectively.

These results conclusively indicate that the application of a deep neural network topology, such as the one described in this research, represents an effective strategy for performance improvement in the field of tourist attraction recommendations with multi-label classification.

3.2 Convolutional Neural Networks

The CNN development represents a class of deep learning models that have demonstrated high efficiency in a wide range of computer vision tasks. These networks, inspired by the organization of neurons in the biological visual cortex, have found significant applications in the context of smart cities. In this context, they have been widely used to address complex challenges related to traffic management, safety, air quality, and infrastructure. Within the field of object detection and traffic surveillance, CNNs are used to detect vehicles, pedestrians, and traffic signs in real-time, aiming to improve traffic management, incident monitoring, and more efficient route planning. In the video analytics and public safety sector, CNNs are used in surveillance systems to identify suspicious activities or anomalous events, improving public safety in urban areas. In addition, CNNs are employed in the inspection of urban infrastructure, such as bridges and roads, to identify early signs of wear and tear or damage, enabling preventive maintenance and the possibility of reducing long-term costs. Since there are many applications of CNNs in the cities, they play a key role in the transformation of cities into smarter and more sustainable environments. The ability to efficiently process visual data and extract information makes them powerful tools to address urbanization challenges.

In this section, we will rely on the approach of convolutional neural networks applied to the detection of distractions for caregivers in an aquatic distraction environment.

3.2.1 Current status of Drowning Prevention Research

In this section, the advances of convolutional neural networks and their uses for image processing are reviewed. In addition, some applications existing in both surveillance and security in recreational aquatic environments are presented.

3.2.1.1 IoT and CNN in Drowning Prevention.

Currently, monitoring and supervision at swimming pools or aquatic recreation locations have drawn the attention of the research community [215]. Therefore, several types of research have been developed particularly for drowning prevention and early detection of possible drowning [216][217]. The integration of IoT (surveillance cameras), and different image processing techniques enable the provision of appropriate and automated solutions for swimming pool safety.

Some proposed drowning detection systems [218][219][220] employ underwater cameras to detect motionless drowned victims sunk at the bottom of the pool using techniques such as background extraction [220], which consists of detecting the moving objects by identifying the difference between the current frame and a reference frame, often called a 'background-image' or 'background model'; however, these systems are limited to victims that have sunk to the bottom of the pool, thus wasting valuable time as they are unable to detect victims before them drowning.

Other proposed methods consist of overhead cameras mounted around the pool (such as our proposed system) [221][222][223]; these systems consist of two main parts: a vision component that can detect and track swimmers and an event-inference (water crisis) module that analyzes swimmer observation sequences for possible drowning behavior signals. Several studies have been carried out regarding the detection of swimmers based on overhead cameras [224][225]. This task is still challenging owing to disturbances at the water's surface (e.g., water exhibits random homogeneous blob movements, which could be easily misidentified as foreground objects) [226][227]. In addition, lightning and color variations over time due to ambient brightness even further complicate automated monitoring based on video surveillance. Several works apply background subtraction to solve the swimmer detection problem [220][226][227]. Likewise, a real-time detection method for constant monitoring of swimmers at an outdoor swimming pool is proposed [227]. A background subtraction scheme is introduced, where the background has been modeled as a composition of homogeneous region processes. Furthermore, to solve the foreground (swimmer) detection problem, a devised thresholding scheme has been proposed to attain a good trade-off between maximizing target detection while minimizing background noises. In addition, to enhance the visibility of the foreground (swimmer), a pre-processing filtering scheme able to classify each pixel of a current frame into different pixel types has been proposed; this way, appropriate filtering actions such as color compensation can be applied when necessary. On the other hand, a background subtraction scheme based on motion and intensity information has been developed to identify swimmers in each video frame [226]. Image pixels are classified according to motion as random/stationary, ripple, and swimming. A motion map is developed through the computation of dense optical flow that characterizes the motion contents of image pixels over a short sequence of video frames rather than a single image. Intensity information has been modeled using a block-based mixture of Gaussians (MoG). However, these systems ([226][227]) only specify how to detect a swimmer; they do not specify how to detect if he/she is drowning.

Current improvements in computing power have enabled the use of deep learning algorithms for human detection and other computer-vision-related problems. Most state-of-the-art object detectors use deep learning algorithms (CNNs) to extract features from input images (or videos) and perform classification and localization, respectively [228]. Another solution has been proposed for monitoring, detection, and classification of fallen objects in a swimming pool [215]; the system uses a single image to detect objects and classify them into three types: humans, animals, and objects. In addition, a system for detecting the drowning of swimmers [229] is presented, by recognizing in real-time the posture of the swimmers in the pool and being able to determine if the person is drowning. Moreover, there is a method to detect swimmers in low-quality videos using two convolutional neural networks (YOLOv2 and Tiny-YOLO) [230]. Furthermore, a real-time vision system to detect drowning incidents using overhead cameras at an outdoor swimming pool is presented [231]. The system uses a model comprising data fusion and hidden Markov modeling to learn of drowning events early. They focus on (1) foreground swimmer silhouette extraction and (2) behavior recognition during a drowning event. Currently, the development of wearable devices has become a very common practice. It has allowed researchers to develop sensor systems to monitor the physiological signals of high-performance swimming athletes [232][233], to detect pre-drowning symptoms and alert rescue staff [234], and supervise children. Wearable sensor systems for infants can perceive external threats such as falls or drowning; and may be useful as an aid in the detection of possible drowning.

3.2.1.2 Convolutional Network Models

Convolutional neural networks were created out of the need to be able to process images effectively and efficiently; nowadays, they are also used for speech recognition. However, their strength is in image processing. Next, we describe the CNNs used in our research.

VGG model: This architecture was proposed by Karen Simonyan and Andrew Zisserman [235]; it was the winner of the ImageNet Large-Scale Visual Recognition Challenge 2012 (ILSVRC12). It was designed with 16 hidden layers in VGG-16 and 19 hidden layers in VGG-19 versions. The architecture processes input images of size 224 × 224 pixels with three channels for color images (RGB). The image is passed through five convolutional blocks (Figure 3.14). In VGG-19, the first two blocks incorporate two convolutional layers, and the remainder incorporate four convolutional layers. Each convolutional layer uses 3 × 3 filters and rectified linear unit (ReLU) as an activation function; the convolutional blocks also incorporate maxpooling layers to reduce image size and prevent overfitting problems; the upper layers are composed of two full-connected layers with 4096 neurons each, at the top, one output layer for image classification into 1000 different categories.

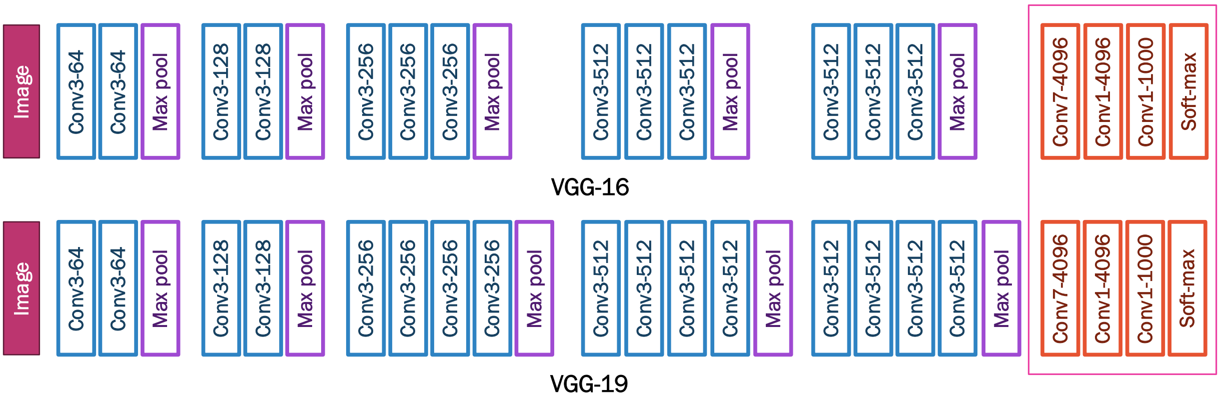


Figure 3.14. VGG-16 and VGG-19 architecture.

ResNet model: It is a type of advanced convolutional neural network; this model was proposed by Kaiming He in his 2016 document [236]. The ResNet-50 version consists of 50 layers. This model is based on the idea of residual and identity blocks that use skip connections (shortcut) (Figure 3.15), where the input is passed to a deeper layer. In other words, the simple deep convolutional neural network is inspired by VGG with 3 × 3 filters and a ReLU activation function, which is modified to become a residual network by adding skip connections to define residual blocks. On the top, the architecture contains a fully connected output layer with a softmax activation function for classification. Figure 3.16 shows the general configuration of the residual network architecture, including ResNet-50, ResNet-101, and ResNet-152.

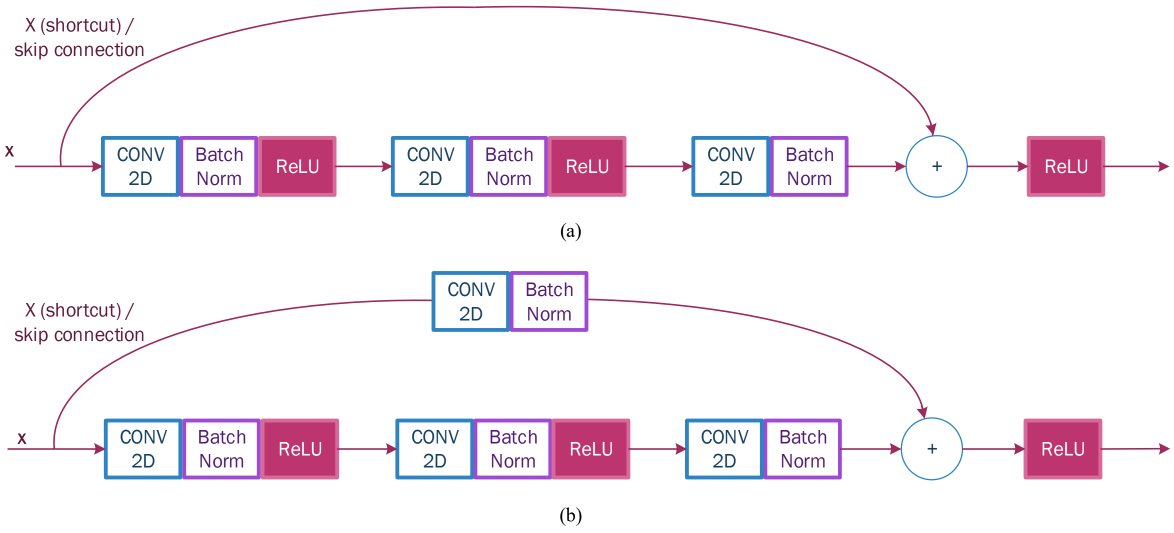


Figure 3.15. (a) ResNet identity block and (b) ResNet convolutional block.

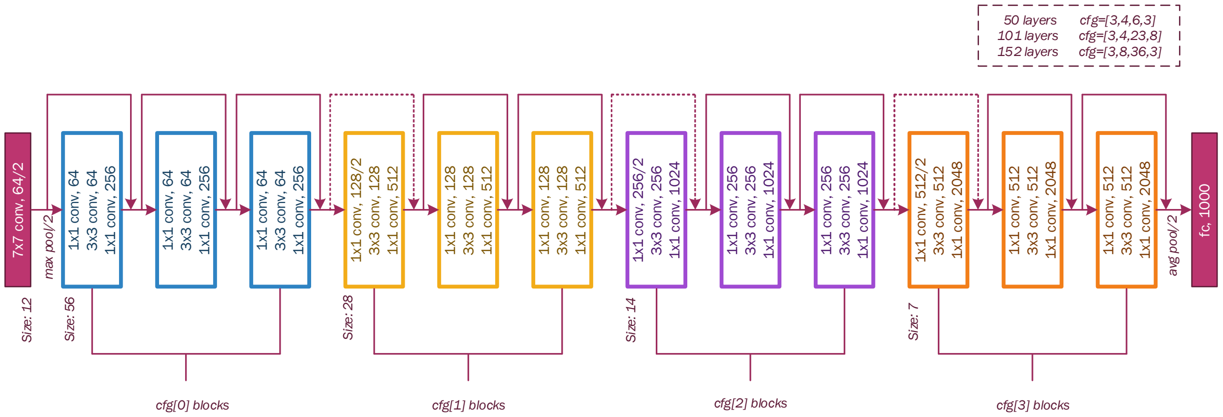


Figure 3.16. Configuration of residual network architecture, including ResNet-50, ResNet-101, and ResNet-152.

Inception-v3 model: This convolutional neural network was developed by Google. The first version of inception, called "GoogLeNet", was presented in the ImageNet Large-Scale Visual Recognition Challenge 2014 (ILSVRC14) [237]. This first version of the architecture is made up of 22 layers including convolutional, pooling, and a characteristic layer called inception; the latter is a type of convolutional layer, but it is characterized by using only 1 × 1, 3 × 3, and 5 × 5 filters simultaneously (Inception blocks) (Figure 3.17); this way, the number of parameters to calculate is greatly reduced. This was achieved with what Google called bottlenecks, which were convolutional layers with 1 × 1 filters to reduce the complexity of the network. Google also includes auxiliary classifiers with the intention of facilitating the propagation of the gradients backward and reducing the associated cost. Therefore, reducing the number of parameters and complexity resulted in a more powerful network.

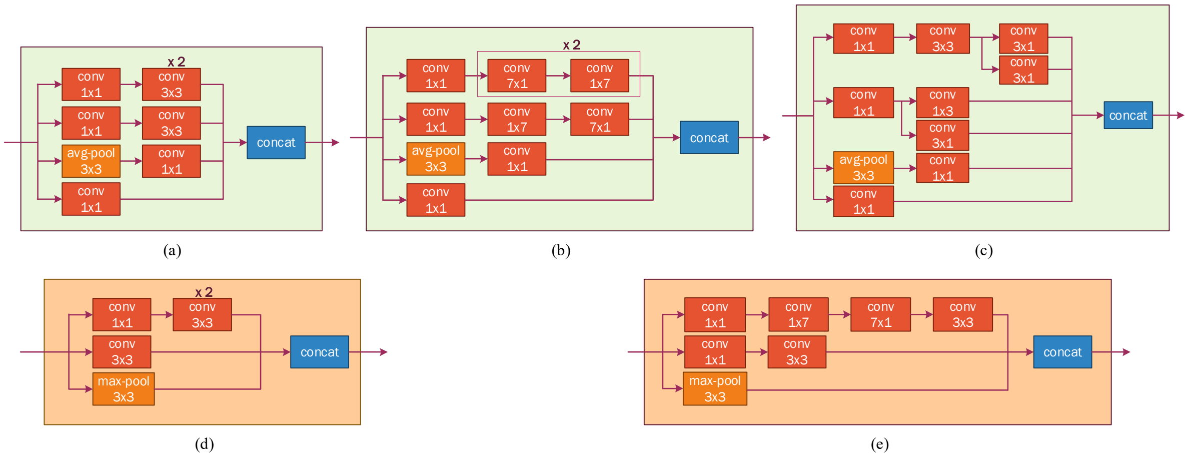


Figure 3.17. (a) Inception-A block, (b) inception-B block, (c) inception-C block, (d) reduction-A block, and (e) reduction-B.

Figure 3.18 shows the inception and reduction blocks that were set for the third version of this architecture.

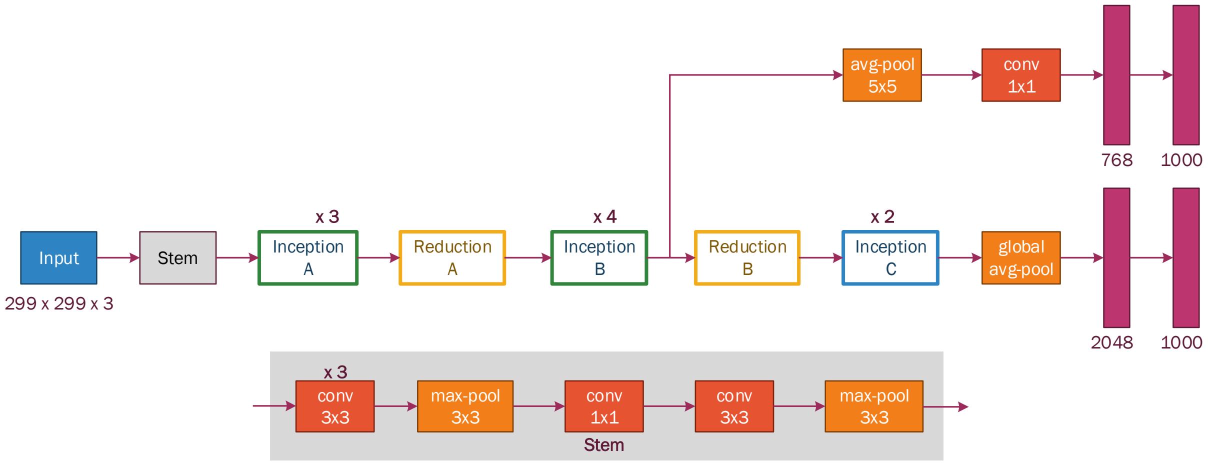


Figure 3.18. Inception-v3 architecture.

3.2.2 Contribution: The Proposed 5G and Beyond Child Drowning Prevention System

Drowning is a major health issue worldwide. The World Health Organization's global report on drowning states that the highest rates of drowning deaths occur among children aged 1–4 years, followed by children aged 5–9 years. Young children can drown silently in as little as 25 s, even in the shallow end or in a baby pool. The report also identifies that the main risk factor for children drowning is the lack of or inadequate supervision. The same report identifies the absence of or inadequate supervision as a key risk factor for the drowning of children [238]. Another report [239] from the Royal Life Saving Society Australia (RSLA, Sydney, Australia) linked distracted parents to 77.8% of drownings in children aged 5–9 years in public and commercial pools between 1 July 2005 and 30 June 2015. In the cases of drowning without supervision, the parent or caregiver of the child was missing, or physically near the child but distracted (talking to another adult or attending to another child in his/her care). Furthermore, the German Lifeguard Association (DLRG, Bad Nenndorf, Germany) (the biggest organization of its kind in the world) reported that more than 300 people died in Germany during 2018 (from the beginning of the year through the summer) and associated the growing number of children drowning to their parents' obsession with mobile phones [240]. In addition, Royal Life Saving Australia reported that, between 2002 and 2017, 447 children under the age of four drowned. Roughly 5% of those deaths were a direct result of a failure to supervise owing to the use of electronic devices (smartphones, tablets, laptops, and so on) [241].

To solve the problem of inadequate child supervision, in this work, we propose a novel 5G and beyond child drowning prevention system based on deep learning that detects and classifies distractions of inattentive parents or caregivers. It can be deployed in indoor swimming pools or outdoor locations such as beaches or aquatic recreation locations aided by unmanned aerial vehicles (drones). The system detects distracted parents/caregivers in charge of a minor (not swimmer detection as in [230]) and alerts them to concentrate on the supervision task. A 5G network slicing architecture for child drowning prevention has also been introduced.

3.2.2.1 Proposed Methodology

In the proposed scenario, families need to register when they arrive at the swimming pool. A facial image of each family member is acquired to recognize them. The swimming pool database registers the age of each child and links the photos of the children with their parents and/or other family member/s. The family decides who is going to be the primary caregiver that is going to watch the children and be responsible for their safety inside the swimming pool and a pager is given to him/her. This task can be shared between the parents (or other family members 18 years or older) simultaneously, which means that none of them should be distracted. It is also possible that there is only one primary caregiver during a certain time slot and another during the next time slot (e.g., the father is the primary caregiver from 15:00 to 17:00 and the mother from 17:00 to 19:00).

After all of these decisions are made using the swimming pool app, the family can access the swimming pool area. The proposed 5G and beyond child drowning prevention system is shown in Figure 3.19.

If the primary caregiver decides to supervise the children outside of the pool, a specific seat will be assigned to him/her close to the swimming pool. This guarantees that he/she will have a good sight of the swimming pool to supervise the children. In addition, a video camera will be directly facing him/her to detect distractions. The cameras are strategically located at an optimal distance in a way that does not obstruct people. In the case of multiple primary caregivers, the same or multiple video cameras can be facing them. Real-time video will be transmitted to the command center. Distractions of primary caregivers will be detected using a deep learning algorithm.

If the primary caregiver decides to supervise the child inside the pool, different video cameras mounted surrounding the pool will detect him/her using computer vision. For this purpose, a high-quality monitoring system is required that consists of video cameras with multiple high-end lenses that can zoom and steer around to detect critical details. The video cameras need to coordinate with each other to be able to track the primary caregivers at any time to detect possible distractions. The video cameras will identify the primary caregiver from different perspectives inside the pool. Automated analysis of the video footage will be carried out. A caregiver can be considered ‘distracted’ if the convolutional neural network analyzes the images from all of the video cameras that are simultaneously capturing his/her behavior and he/she is characterized as being ‘distracted’ by most of them. That is, the images of the parents/caregivers are not combined; instead, the images from each camera are classified into a category. It is decided if the parent/caregiver is distracted or not by analyzing which category is repeated the most.

When a distraction event is detected, an alert will warn the primary caregiver so that he/she can focus on active child supervision. We assume that alerts will be sent immediately if the kids to supervise are 5 years old or under. For kids that can swim (usually older than 5 years), parents will be alerted if the convolutional neural network detects continuous distracted behavior for more than 10 seconds because drowning accidents happen very quickly. Alert messages can be sent to a pager. The pager lights up or vibrates in case the caregiver is distracted. Alert messages can also be heard through the swimming pool speakers located in the closest vicinity of the caregiver. Furthermore, lifeguards will also get these notification messages and act accordingly. This information will be, for example, useful if certain caregivers are notified several times; in this case, lifeguards can supervise the associated children much closer and talk to the parents/caregivers or take other necessary steps if no change in their attitude is observed.

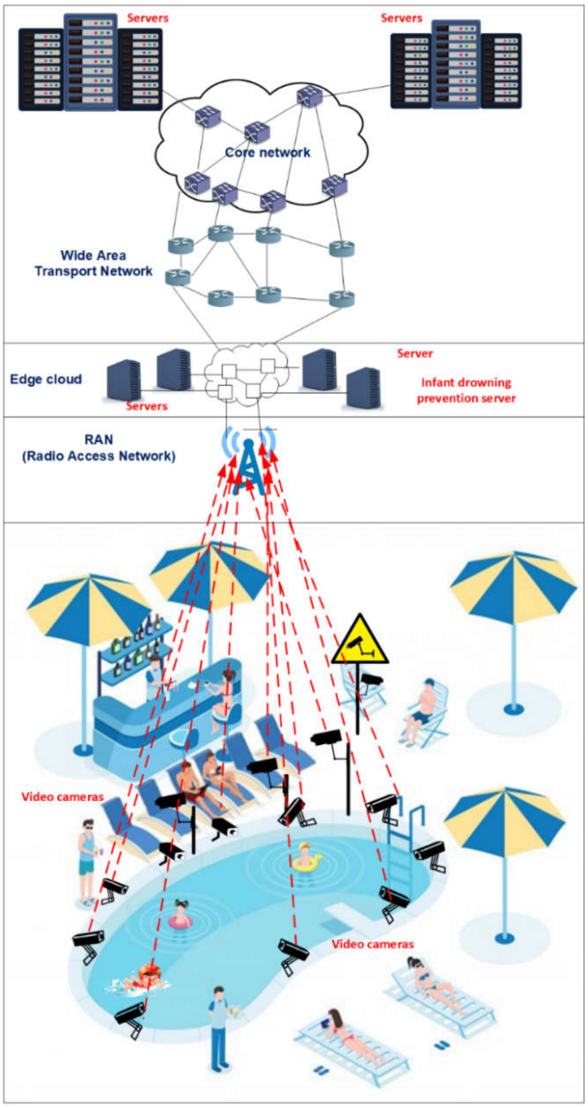


Figure 3.19. Proposed 5G-enabled child drowning prevention system.

3.2.2.2 Related Key Performance Indicators

The proposed 5G-enabled child drowning prevention system can be identified as a Mission-Critical Communications (MCC) service because it requires real-time and reliable communications for a large number of users, as well as strong security and pre-emption handling [242]. Table 3.8 summarizes the major Key Performance Indicators (KPIs) for child drowning prevention. The end-to-end latency can be measured as the time interval required to send the packages from a source to a destination, measured at the application level.

Mission critical: A quality or characteristic of a communication activity, application, service, or device that requires low setup and transfer latency, high availability, and reliability, the ability to handle large numbers of users and devices, strong security, and priority and pre-emption handling.

It would be possible for our use case to connect to the nearest edge server via Wi-Fi 7 (802.11be) because this standard will support a maximum throughput of at least 30 Gbps. Features operating at both the MAC (medium access control) layer and the physical layer (PHY) such as multi-access point coordinated beamforming, time-sensitive networking, and multi-link operation will bring Wi-Fi 7 latency performance into the sub-10 ms realm. These characteristics would be enough to support our high-throughput low-latency child drowning prevention use case. However, the Institute of Electrical and Electronics Engineers (IEEE) task group announced draft 2.0 of 802.11be, and the final version will be released in 2024.

IEEE 802.11ax (Wi-Fi 6) received final approval from the IEEE Standards Board on 1 February 2021. This standard offers a theoretical speed of up to 9.6 Gbps and 10 ms latency. Wi-Fi 6 does not perform well in large-scale outdoor coverage scenarios and cannot meet the ultra-low latency requirements (<10 ms).

It has been shown in [243] that Wi-Fi 6 can achieve ultra-reliable low latency performance (i.e., <1 ms packet latency at 99.999% reliability) only when optimized and operating in a low load up to 0.16 bps/Hz that is not appropriate for our use case.

On the other hand, 5G can reach up to 10 Gbps (only slightly higher than Wi-Fi 6), but this technology has been designed to address the requirements of ultra-reliable and low-latency communications. URLLC has stringent requirements for capabilities such as latency, reliability, and availability. Some use cases include wireless control of industrial manufacturing or production processes, remote medical surgery, and transportation safety. It has been demonstrated in [243] that 5G NR-FDD (Frequency Division Duplex) has superior URLLC performance and meets the sub-ms delay requirement at >5× higher load than Wi-Fi 6.

Therefore, 5G is the appropriate technology for our use case thanks to its better latencies. The proposed system requires that real-time video is backhauled from the video cameras to the command center for remote control and analysis. The number of video cameras will vary depending on the size of the swimming pool. Moreover, 5G can be deployed in indoor swimming pools or even in outdoor locations such as beaches or aquatic recreation locations that extend several kilometers. In these cases where so many video images need to be processed as quickly and efficiently as possible, a 5G network is required to provide sufficiently high uplink data throughput and transmission reliability as well as sufficiently low latency. The short end-to-end latency will enable alert messages to be sent as fast as possible if necessary as drowning happens quickly. Reliability is critical to detecting incidents, which means that performance should not be compromised irrespective of the channel conditions.

Table 3.8. Main KPIs for child drowning prevention.

	End-to-End Latency	**Data Rate (Uplink/Downlink)**	**Reliability**
5G-enabled child drowning prevention system	20 ms	40 Mbit/s for one video camera/1 Mbps for remote control	99.999%

3.2.2.3 5G Service-Based Architecture

Next, the 5G system architecture of the non-roaming case is illustrated in Figure 3.20 [244]. The user plane (UP) and control plane (CP) are decoupled to obtain scalable and flexible deployments. Whereas the CP is used for network signaling, the UP carries only user traffic.

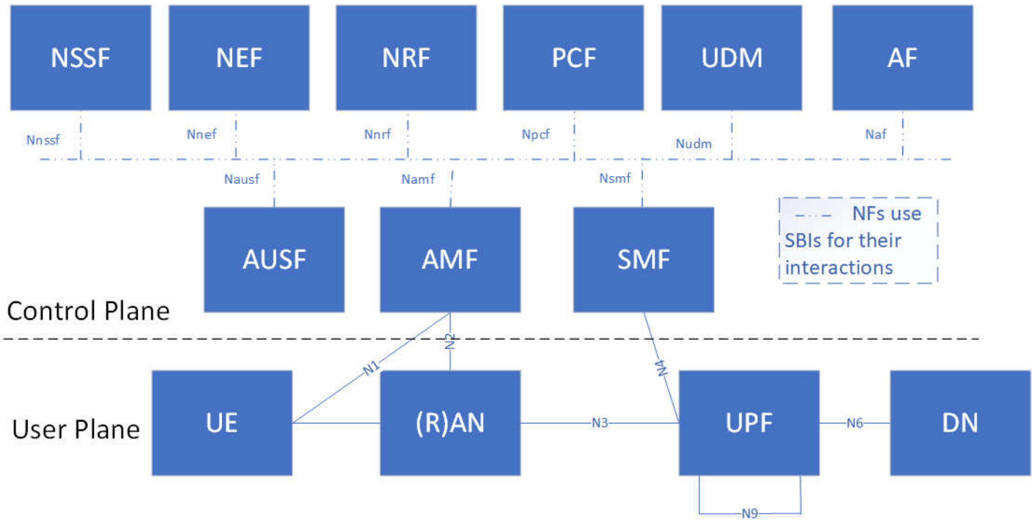


Figure 3.20. Service-based representation of the 5G non-roaming system architecture [244].

The user equipment (UE) in the user plane is connected to either the radio access network (RAN) or a non-3GPP access network (e.g., wireless local area network, WLAN) as well as to the access and mobility management function (AMF).

Next, we explain the network functions (NFs) of the 5G core network (see the upper part of the figure):

- Access and mobility management function (AMF): it is responsible for UE registration, reachability, and mobility.
- Session management function (SMF): it offers UE IP address allocation and management, policy enforcement and quality of service, user plane function (UPF) selection, and control.
- User plane function (UPF): it is the anchor point for intra and inter-radio access technology (RAT) mobility, packet routing, and forwarding.
- Policy control function (PCF): it integrates a policy framework for network slicing.
- Application function (AF): it is responsible for different services provided after the interaction with the core network.
- User data management (UDM): it is responsible for subscriptions and many services related to users.
- Authentication server function (AUSF): it performs the UE authentication service.
- Network slice selection function (NSSF): it offers an optimal selection of network instances serving the users.

- Network exposure function (NEF): it collects, stores, and exposes the services and capabilities provided by 3GPP NFs in a secure manner.
- NF repository function (NFR): it maintains and provides the deployed NF instances; it also supports the service discovery function.

3.2.2.4 A 5G Network Slicing Architecture for Child Drowning Prevention

Network slicing refers to the division of a physical network into multiple logical networks (network slices), so that each logical network can provide specific network characteristics for a particular use case. Network slicing provides services across multiple network segments and different administrative domains. A 5G slice can combine resources that belong to different infrastructure providers [245]. Network slicing is the best way for network operators to build and manage a network that meets the requirements from a wide range of users. Network slicing provides service flexibility and the ability to deliver services faster with high security, isolation, and according to the QoS requirements of the different applications. This way, network operators can manage their network resources efficiently and provide differentiated and scalable services.

Slices are isolated from each other, which means that faults or errors in one slice do not affect the proper functioning of another slice.

Next, we introduce the main design elements of our proposed 5G network slicing architecture for child drowning prevention (see Figure 3.21).

It is divided into three layers plus an additional management and orchestration layer, whose basic functionalities are summarized as follows:

Infrastructure layer: It refers to all of the parts of the physical network because slices should be end-to-end. This layer includes the IoT networks, telecommunication networks, satellites, edge computing technologies, and the cloud. It provides the allocation of virtual or physical resources such as computing, storage, network, or radio.

We assume that all network devices are SDN-enabled switches managed by SDN controllers that can program their routing tables.

The 5G core is generally divided into the 'core-user plane' in charge of bearer delivery and the 'core-control plane (CP)' in charge of control functions. The core-control plane will stay in the central cloud (NFV), but the 'core-user plane (UP)' will be distributed to its tens of edge nodes nationwide and be installed in edge clouds (NFV). Security, reliability, and latency will be critical for a 5G slice supporting the child drowning prevention case. For such a slice, all of the necessary (and potentially dedicated) network functions should be instantiated at the edge node. We consider that all the 5G core functions/units (UP) should be in the edge cloud close to the users. MEC drastically reduces the latency between network nodes and remote servers in the cloud [246] because video processing servers are placed right where the core functions/units are located. This way, we can minimize the transmission delay to match the requirements of our delay-critical slice for such an MCC application. Furthermore, machine learning is crucial in supporting MCC by enabling a local decision-making process at the edge servers [247].

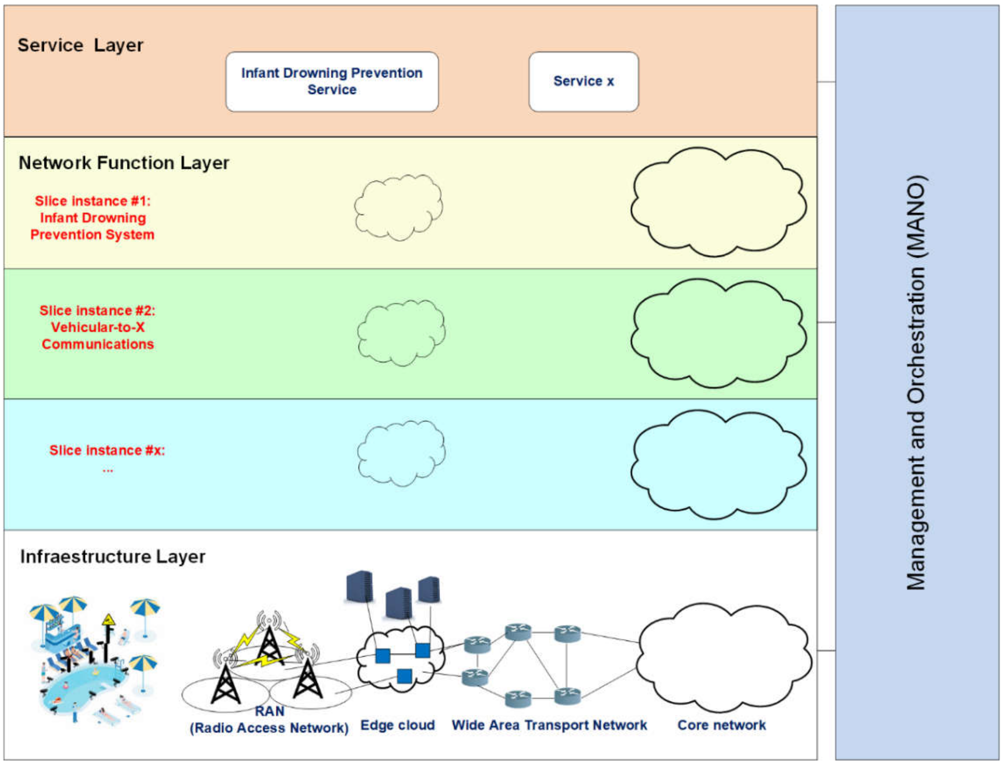

Service Layer
Infant Drowning Prevention Service
Service x
Network Function Layer
Slice instance #1: Infant Drowning Prevention System
Slice instance #2: Vehicular-to-X Communications
Slice instance #x: ...
Infraestructure Layer
RAN (Radio Access Network)
Edge cloud
Wide Area Transport Network
Core network
Management and Orchestration (MANO)

Figure 3.21. Network slicing architecture for child drowning prevention.

Network function layer: It encapsulates all of the operations related to the configuration and life cycle management of the network functions that offer an end-to-end service. Network function virtualization [248] and software-defined networking [249] are two fundamental technologies to configure virtual network resources. NFV decouples specific network functions from dedicated and expensive hardware platforms. This technology can provide software building blocks named VNFs (virtualized network functions) for the data plane that can be connected and chained according to the service type. SDN technology enables the separation of the control plane from the data plane to offer flexible resource management.

Service layer: This layer provides a unified vision of the service requirements. Each service is represented by a service instance, which embeds all of the network characteristics that satisfy the SLA (service level agreement) requirements such as throughput or latency. A network slice instance (NSI) is a managed entity created by an operator's network with a lifecycle independent of the lifecycle of the service instance(s) [250]. An NSI provides the network characteristics required by a service instance. It is also possible that an NSI is shared across multiple service instances of a network operator.

Based on the main KPIs (see Section 3.2.2.2) and functional requirements of our use case, child drowning prevention, we propose that the drowning prevention slice has ultra-reliable and low-latency communications (URLLC) requirements. URLLC use cases (such as mission-critical applications) have stringent latency, reliability, and availability requirements.

Management and Orchestration (MANO): It is the framework for the management and orchestration of all network resources (computing, networking, storage, and virtual machines) in the cloud. It comprises three functional blocks: NFV orchestrator (NFVO), VNF manager (VNFM), and virtualized infrastructure manager (VIM). NFVO performs on-boarding of new network service and VNF packages, network service lifecycle management, and resource management. VNFM manages the lifecycle of VNF instances. VIM controls and manages the lifecycle of virtual resources as requested by the NFVO in an NFV infrastructure (NFVI) domain.

3.2.2.5 Experiments and Results

3.2.2.5.1 Dataset

The dataset is a collection of 38,000 images generated by us in the summer of 2019. The location of the video recording was the facilities of the Fontsanta swimming pool, located at Carrer del Marquès de Monistrol, 30, 08970 in Sant Joan Despí, Barcelona—Spain. Five primary caregivers (people in charge of the children) were involved in the development of these experiments. They were recorded on video, doing different activities (one video for each action related to each of the different categories) both inside and outside the water. The images captured from each video correspond to a specific category, so the images have been identified and labeled manually for each category. The capture was made taking into account that only the participants appear in the video to protect the privacy and confidentiality of other people who are at the swimming pool. The videos were recorded with high-resolution smart mobile devices (1920 × 1880), although the images are preprocessed according to the input data requirements of each model (224 × 224). The images were finally collected and classified into seven (7) categories:

- *I_distracted:* In the water distracted.
- *I_watching:* In the water watching the children.
- *O_distracted:* Out of the water distracted.
- *O_talk_cell:* Out of the water talking on a cell phone.
- *O_reading:* Out of the water reading a book.
- *O_chatting:* Out of the water chatting on a cell phone.
- *O_watching:* Out of the water watching the children.

To achieve a great performance during the training process with our own dataset, the videos were not shot from a single angle. Instead, they were shot from different angles, covering all potential perspectives of a caregiver. Furthermore, because the swimming pool is located outdoors, the varying lighting conditions throughout the day provide a richer dataset.

3.2.2.5.2 Experimental Settings

The dataset consists of approximately 38,000 images; it was split into two parts, keeping a ratio of 8:2, i.e., around 30,000 images for training and 8000 for testing. In addition, data augmentation was used to expand the training set and obtain better generalization. Data augmentation is a technique that expands our original training dataset virtually, through a random series of transformations from the original image, resulting in new plausible-looking images, to obtain a larger number of images for training. In computer vision, this technique became a standard for regularization, as well as to improve accuracy, generalization, and control of overfitting in CNNs. For this research, the techniques chosen are as follows: rescale = 1./255, rotation_range = 2, shear_range = 0.2, zoom_range = 0.2, and horizontal_flip = True.

We have selected the images from a different subject for testing purposes in order not to contaminate the testing set. Figure 3.22 and Figure 3.23 show a set of images of each category with their training and testing labels.

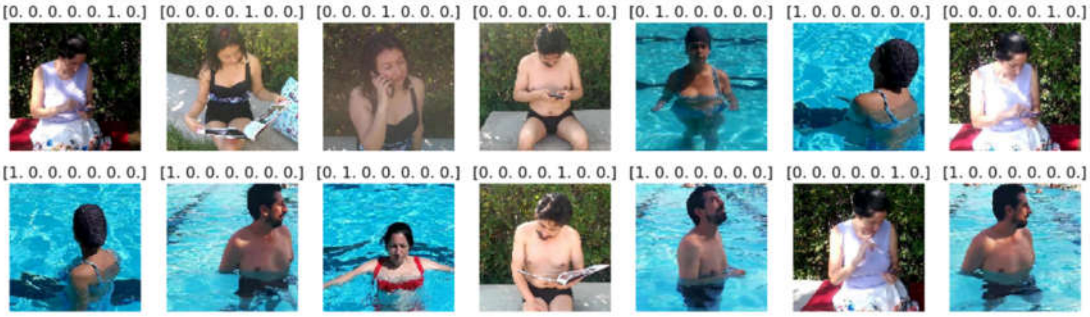


Figure 3.22. Image set of each category with their respective training labels.

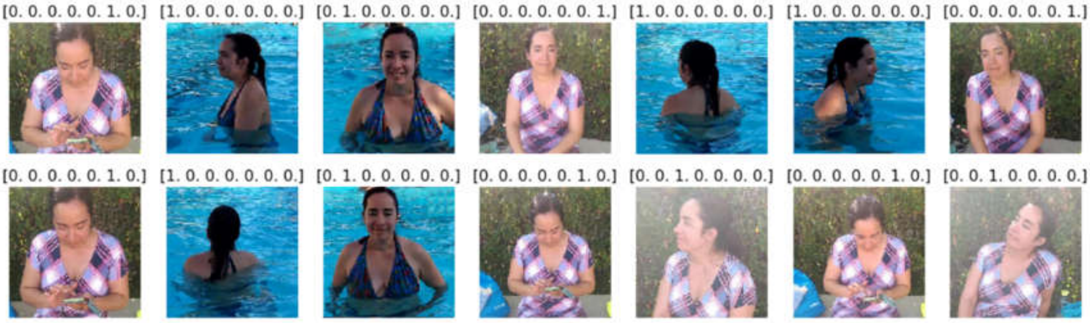


Figure 3.23. Image set of each category with their respective testing labels.

The algorithms were implemented in several Jupyter Notebooks in version 6.0.3 installed with the Anaconda programs suite, developed in Python. The experiments were carried out on a Lenovo computer 2.9 GHz Intel (R) Xeon (R) processor with 72 GB RAM, without GPU.

We implemented three different algorithms using the preset models from the Python Keras library; each one was specifically adapted to obtain optimal results after each training. The transfer learning technique was used (further details will be provided in Section 3.2.2.5.4) to take advantage of the pre-trained weights. Early stopping and dropout were implemented as techniques to avoid overfitting to achieve an improvement of the generalization capacity. Accuracy was selected during the training process as a metric to evaluate the performance of each algorithm.

The setup of each model to be used is detailed below.

3.2.2.5.3 Convolutional Neural Network Architectures

In this work, experiments were performed to evaluate the proposed approach with three different CNN architectures: VGG-19, ResNet-50, and Inception-v3. Table 3.9 presents a summary of the configuration for each model. For all experiments, we used an image size of 224 × 224 × 3 and a batch size of 64.

3.2.2.5.3.1 VGG-19

We implemented the VGG-19 version because it has a greater number of layers (deeper network) compared with the VGG-16 version mentioned above. It is made up of a 224 × 224 × 3 input layer, five convolutional blocks with kernel 3 × 3, ReLU activation function, without padding, and a maxpooling layer after each block followed by a flattened layer and two additional blocks; each additional block consists of a fully connected dense layer with 4092 neurons, a BatchNormalization layer, and a dropout layer. The last layer is a dense layer with a softmax activation function that contains seven neurons to classify our categories.

Table 3.9. Architectures of the three CNN models.

Input	VGG-19 Image	ResNet-50 Image	Inception-v3 Image
Convolutional part	conv3-64 conv3-64 max pooling layer conv3-128 conv3-128 max pooling layer conv3-256 conv3-256 conv3-256 conv3-256 max-pooling layer conv3-512 conv3-512 conv3-512 conv3-512 max-pooling layer conv3-512 conv3-512 conv3-512 conv3-512 max-pooling layer	conv7-64, s = 2 max pooling layer [conv1-64; conv3-64; conv1-256]–[conv1-64] 2 blocks of [conv1-64; conv3-64; conv1-256] [conv3-128, s = 2; conv1-128; conv1-512]–[conv3-128, s = 2] 3 blocks of [conv1-128; conv3-128; conv1-512] [conv1-256, s = 2; conv3-256 conv1-1024]–[conv1-256, s = 2] 5 blocks of [conv1-256 conv3-256 conv1-1024] [conv1-512, s = 2; conv3-512; conv1-2048]–[conv1-512, s = 2] 2 blocks of [conv1-512 conv3-512 conv1-2048] global_average-pooling layer	Conv3-32, s = 2 Conv3-32 Conv3-64 max pooling layer Conv3-80 Conv3-192, s = 2 max pooling layer Inception A-256 Inception A-288 Inception A-288 Reduction A-768 Inception B-768 Inception B-768 Inception B-768 Inception B-768 Reduction B-1280 Inception C-2048 Inception C-2048 global_average-pooling layer

MLP classifier	FC layer-4096 FC layer-4096 FC layer-07	FC layer-2048 FC layer-2048 FC layer-07	FC layer-2048 FC layer-2048 FC layer-07

3.2.2.5.3.2 ResNet-50

This model contains an input layer of 224 × 224 × 3, fifty convolutional blocks with their respective skip connections, followed by a global average pooling layer. At the top of the model, we have added two additional blocks; each block consists of a fully connected dense layer with 2048 neurons, a BatchNormalization layer, and a dropout layer. The last layer is a dense layer with a softmax activation function that contains seven neurons for our classification.

3.2.2.5.3.3 Inception-v3

This model is composed of a 224 × 224 × 3 input layer, two convolutional blocks of three and two layers, followed by a maxpooling layer after each block. The central part consists of several types of inception and reduction blocks, along with a global_average-pooling layer. At the top of the model, we added two additional blocks; each block consists of a dense layer fully connected with 2048 neurons, a BatchNormalization layer, and a dropout layer. The last layer is a dense layer with a softmax activation function that contains seven neurons for our classification.

3.2.2.5.4 Training

The dataset consists of approximately 38,000 images (N records); it was split into two parts, keeping a ratio of 8:2, i.e., around 30,000 images for the training set (n records) and 8000 for the testing set (N-n records). For the training, we applied cross-validation. Cross-validation is a technique commonly used to validate machine learning models and estimate the performance of the model trained on unseen data. The most robust and widely used method of cross-validation is K iterations or K-fold cross-validation. This method consists of splitting the training dataset into K subsets (see Figure 3.24). During iterations, each of the subsets is used as validation data or testing folds, and the rest (K-1) as training data or training folds. The cross-validation process is performed repeatedly for K iterations, with each of the subsets of validation data. The arithmetic average of the results of each iteration is finally performed to obtain a single result. This method is highly efficient as we evaluate it from K combinations of training and validation data, but it still has a disadvantage, that is, computationally, it is slow. However, the choice of the number of iterations depends on how large the dataset is. Cross-validation is most commonly used with K values ranging from 5 to 10. If the model (estimator) is a classifier and the target variable (*y*) is binary or multiclass (as in this research), the StratifiedKfold technique is used by default. This approach introduces stratified folds, i.e., by keeping the proportion of samples from each class in all folds. Therefore, the data from the training and testing folds are distributed equally. It is useful when unbalanced datasets are used. To evaluate the results, we used several metrics that are very common in machine learning applications for classification problems.

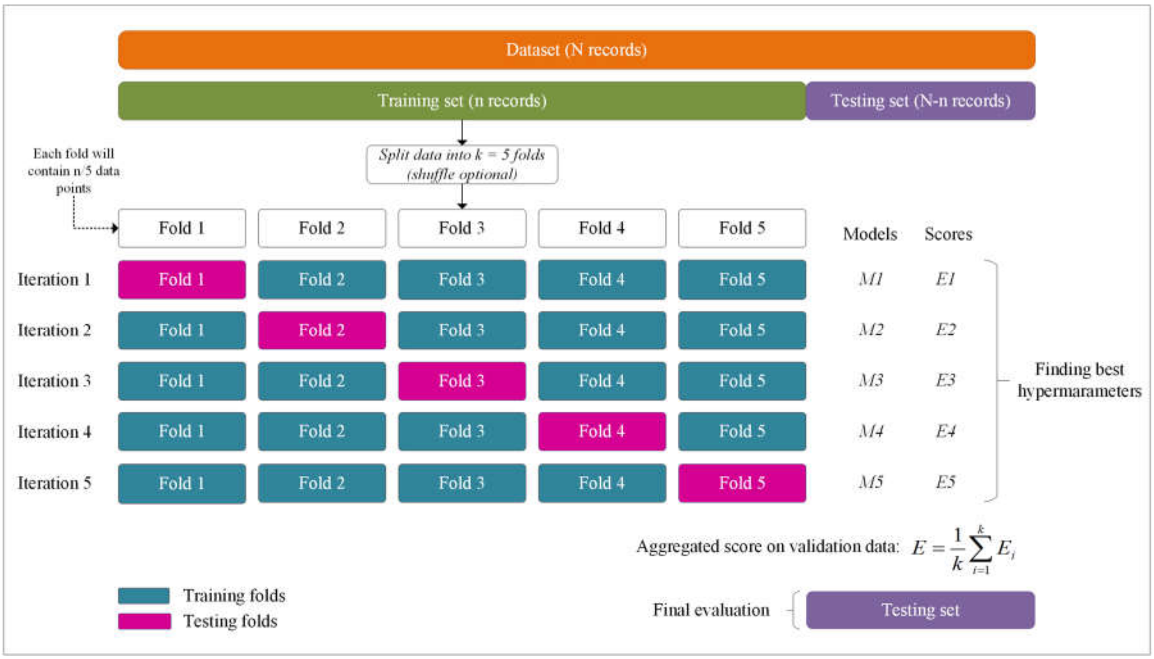


Figure 3.24. Use of each fold in the cross-validation process (fivefold representation).

3.2.2.5.4.1 Loss or Cost Function

A loss function is employed to optimize a machine learning algorithm. Several different cost functions can be used. Each of them penalizes errors differently. The loss function most commonly used in deep neural networks for classification problems is cross-entropy. In this research, we employed categorical cross-entropy. Categorical cross-entropy is a loss function that is used in multi-class classification tasks, where a sample can be considered to belong only to a specific category with a probability of 1 and to other categories with a probability of 0, and the model must decide which category each one belongs to.

3.2.2.5.4.2 Transfer Learning and Early Stopping

A model can be trained from scratch when it is not very large or when the necessary computational capacity for its execution is available. On the other hand, it is possible to take advantage of the benefits of pre-established models and use them in new models. This technique is known as transfer learning; this means that it allows us to transfer learning from a pre-trained model such as VGG-19, ResNet-50, Inception-v3, and so on (pre-trained models for 1000 objects' classification) and apply it to new classification algorithms. Furthermore, it is possible to unfreeze some pre-trained layers by adapting the model (fine-tuning) to re-train them along with the new fully connected layers; this method implies increasing the training time to avoid overfitting problems and to obtain optimal performance from the algorithm.

A popular technique to overcome overfitting is early stopping. For this purpose, at each iteration, the training set is divided into training and validation folds. The training folds are used to train the model and the validation folds are used as validation data at each iteration. In each training of the model, the validation folds help us to verify the accuracy of the model at the end of each epoch. Therefore, as soon as the test error starts to increase, the training is stopped.

3.2.2.5.5 Evaluation Metrics

To evaluate the results, we used several metrics that are very common in machine learning applications for classification problems.

3.2.2.5.5.1 Accuracy

It is defined as the number of predictions made correctly by the model of the total number of records.

$$accuracy = \frac{TP + TN}{TP + FP + FN + TN} \tag{3.17}$$

where TP represents true positives, TN represents true negatives, FP represents false positives, and FN represents false negatives.

3.2.2.5.5.2 Precision

We evaluate our data for its performance of "positive" predictions.

$$precision = \frac{TP}{TP + FP} \tag{3.18}$$

3.2.2.5.5.3 Recall (Sensitivity) (True Positive Rate)

It is calculated as the number of correct positive predictions divided by the total number of positives.

$$recall \ = \frac{TP}{TP + FN} \tag{3.18}$$

3.2.2.5.5.4 Specificity (True Negative Rate)

It is calculated as the number of correct negative predictions divided by the total number of negatives.

$$specificity = \frac{TN}{TN + FP} \tag{3.19}$$

3.2.2.5.5.5 F1 Score

It is the weighted average of precision and sensitivity. Therefore, this score takes into account both false positives and false negatives.

$$F1\, score \ = 2 \times \frac{(precision \times recall)}{(precision + recall)} \tag{3.20}$$

3.2.2.5.5.6 Loss

Loss is the value that reflects the sum of errors in our model. It indicates whether the model is performing well (high value) or not (low value); on the other hand, the accuracy can be defined as the number of correct predictions divided by the number of total predictions.

Therefore, if we analyze these two metrics together (loss and accuracy) (see Table 3.10), we can deduce more information about the model performance. If loss and accuracy are low, it implies that the model makes small errors in most of the data. However, if both are high, it makes large errors in some of the data. Low accuracy but high loss would mean that the model makes large errors in most of the data. However, if the accuracy is high and the loss is low, then the model makes small errors in only some of the data, which would be the ideal case.

Table 3.10. Analysis of both loss and accuracy metrics together.

	Low Loss	High Loss
Low Accuracy	A lot of small errors	A lot of big errors
High Accuracy	A few small errors	A few big errors

3.2.2.5.6 Experimental Results

After training with different configurations in the upper layers of each model, the following results were obtained.

3.2.2.5.6.1 Loss and Accuracy

For training, cross-validation was performed; therefore, the early stopping technique was used to avoid overfitting (as mentioned above); thus, training is stopped once it has reached the maximum accuracy value. Furthermore, the checkpoint was used to save the weights of the trained model when a new maximum value arises and we can load it in the future. Table 3.11 shows a summary of the accuracy and loss for the training and testing of each model. We can see that, for training, all models achieve an accuracy above 99% and ResNet-50 achieves a higher loss value compared with the other two models. Furthermore, for testing, ResNet-50 achieves the highest accuracy, but also the largest loss of 98% and 0.3203, respectively. VGG-19 achieves an accuracy of 94% and the lowest loss of 0.0039 and, finally, Inception-v3 achieves an accuracy of 90% and a loss of 0.0364. Based on the accuracy, ResNet-50 has developed much better performance compared with the other trained models.

Table 3.11. Accuracy and loss for VGG-19, ResNet-50, and Inception-v3 model.

Models	Training		Testing	
	Accuracy	Loss	Accuracy	Loss
VGG-19	0.9987	0.0056	0.9445	0.0039
ResNet-50	0.9973	0.0110	0.9803	0.3203
Inception-v3	0.9993	0.0019	0.9044	0.0364

Table 3.12 shows the accuracy achieved by each model with each of the classification categories (seven), evidencing the performance in more detail. VGG-19 achieves an accuracy of 100% for the I_watching and O_reading categories, an average accuracy of 97.42% for the remaining categories, and a lower value of 72.73% for the O_chatting category. Similarly, ResNet-50 achieves an accuracy of 100% for the I_watching and O_talk_cell categories and the worst result for the O_distracted category, with an accuracy of 95.4%. On the other hand, Inception-v3 achieves a high accuracy of 98.68% for the I_distracted category and a lower accuracy of 66.6% for the O_talk_cell category.

As this research work focuses on parental distraction detection for child drowning prevention, the "In the water watching the children" (I_watching) and "Out of the water watching the children" (O_watching) categories are the most relevant ones to detect if parents/caregivers are really supervising their children. All of the other categories just represent that the caregivers are distracted and should be warned. For I_watching, the VGG-19 and ResNet-50 models achieve an accuracy of 100% and Inception-v3 achieves an accuracy of 96.83%. Likewise, for O_watching, the VGG-19 and ResNet-50 models achieve an accuracy of 99.61% and Inception-v3 achieves an accuracy of 84.42% (Table 3.11).

Table 3.12. Accuracy of each model with each category.

Parent Status	VGG-19 Accuracy (%)	ResNet-50 Accuracy (%)	Inception-v3 Accuracy (%)	Total Samples
I_distracted	98.99	97.75	98.68	1291
I_watching	100	100	96.83	883
O_distracted	92.32	95.4	95.2	1458
O_talk_cell	98.75	100	66.6	1036
O_reading	100	97.85	94.29	1069
O_chatting	72.73	97.97	90.91	935
O_watching	99.61	99.61	84.42	507

3.2.2.5.6.2 Precision, Recall, and F1-Score

Accuracy should not be considered as a single metric for measuring model performance when using an unbalanced data set, as it counts the number of correct predictions regardless of the type of category, leaning towards the majority of categories. In other words, from a dataset of 100 cases where 95 belong to the category "a" and five to category "b"; if only all the cases in the first category are correctly predicted, an accuracy of 95% would be obtained. This value is misleading because 95% refers only to the correctly predicted values of one category (50% of the total predictions).

Because our data are unbalanced, we also consider other metrics such as recall, precision, specificity, and F1-score to evaluate our results. Table 3.13 shows the values obtained in every category based on the above-mentioned metrics for VGG-19. F1-score is the harmonic mean of precision and recall and it takes into account both false positives and false negatives. The VGG-19 model performs well because it achieves an accuracy between 96% and 99% for most categories and a smaller accuracy of 84% for the O_reading category. We can also observe that, for the most relevant categories (I_watching and O_watching), this model reaches an F1-score of 98%, demonstrating good performance in training.

Table 3.13. Evaluation metrics of the VGG-19 model.

Category	Precision	Recall	F1-Score	Total Samples
I_distracted	0.99	0.99	0.99	1291
I_watching	0.98	1.00	0.98	883
O_distracted	0.96	0.92	0.96	1458
O_talk_cell	0.99	0.99	0.99	1036
O_reading	0.87	1.00	0.87	1069
O_chatting	0.84	0.73	0.84	935
O_watching	0.98	1.00	0.98	507

Table 3.14 shows a summary of the already mentioned metrics in every category for the ResNet-50 model. It achieves an F1-score between 97% and 99% for all categories. It should be pointed out that this model reaches an F1-score of 98% and 99% for the most relevant categories (I_watching and O_watching), which is the best performance of the three models.

Table 3.14. Evaluation metrics of the ResNet-50 model.

Category	Precision	Recall	F1-Score	Total Samples
I_distracted	1.00	0.98	0.99	1291
I_watching	0.97	1.00	0.98	883
O_distracted	0.99	0.95	0.97	1458
O_talk_cell	0.95	1.00	0.98	1036
O_reading	1.00	0.98	0.99	1069
O_chatting	0.96	0.98	0.97	935
O_watching	0.98	1.00	0.99	507

Finally, Table 3.15 shows a summary of the already mentioned metrics in every category for the Inception-v3 model. This model achieves an F1-score between 91% and 98% for most categories and a minimum F1-score of 79% for the O_talk_cell category. In this case, the Inception-v3 model achieves an F1-score of 98% for the I_watching category, but the lowest F1-score of 84% for the *O_watching* category (most relevant categories).

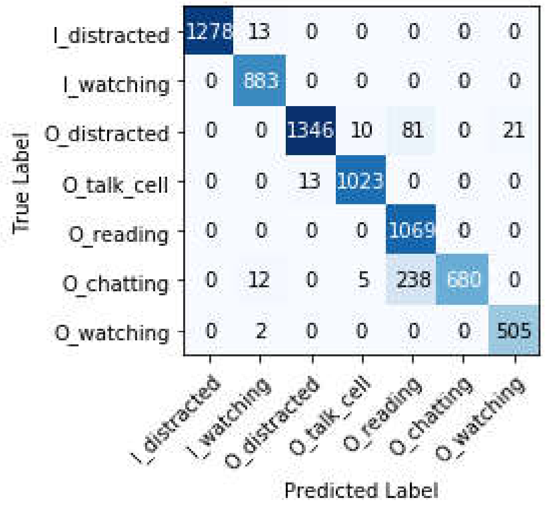
Table 3.15. Evaluation metrics of the Inception-v3 model.

Category	Precision	Recall	F1-Score	Total Samples
I_distracted	0.98	0.99	0.98	1291
I_watching	0.98	0.97	0.98	883
O_distracted	0.75	0.95	0.84	1458
O_talk_cell	0.98	0.67	0.79	1036
O_reading	0.98	0.94	0.96	1069
O_chatting	0.92	0.91	0.91	935
O_watching	0.84	0.84	0.84	507

According to this, we conclude that the ResNet-50 model shows excellent performance for this classification problem, reaching F1-scores of 98% and 99% in the I_watching and O_watching categories, respectively (see Table 3.14). However, the VGG-19 model with a value of 98% in the mentioned categories shows a solid performance as well (see Table 3.13).

3.2.2.5.6.3 Confusion Matrix, False Positive Rate, and False Negative Rate

Figure 3.25, Figure 3.26, and Figure 3.27 show the confusion matrices for each model. The main diagonal shows the number of matches found for each category between the true labels (columns) and the predicted labels (rows).



	I_distracted	I_watching	O_distracted	O_talk_cell	O_reading	O_chatting	O_watching
I_distracted	1278	13	0	0	0	0	0
I_watching	0	883	0	0	0	0	0
O_distracted	0	0	1346	10	81	0	21
O_talk_cell	0	0	13	1023	0	0	0
O_reading	0	0	0	0	1069	0	0
O_chatting	0	12	0	5	238	680	0
O_watching	0	2	0	0	0	0	505

Figure 3.25. Confusion matrix VGG-19.

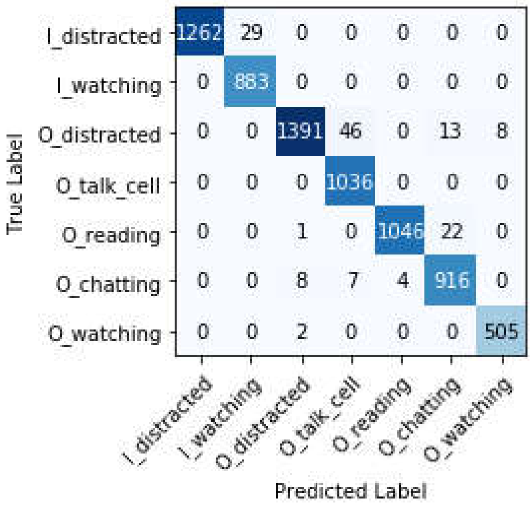


Figure 3.26. Confusion matrix ResNet-50.

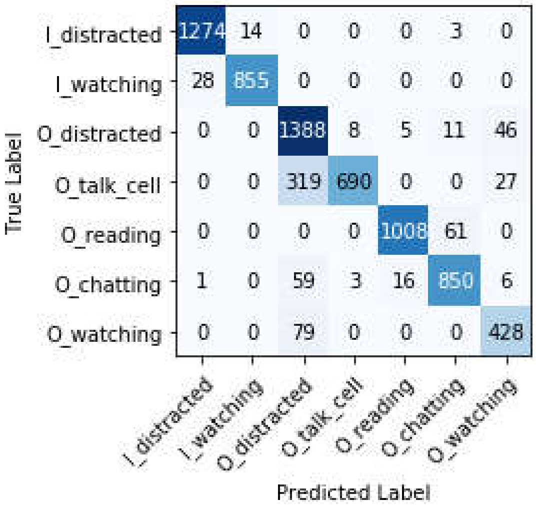


Figure 3.27. Confusion matrix Inception-v3.

All categories are well predicted. Considering the most relevant categories "In the water watching the children" (I_watching) and "Out of the water watching the children" (O_watching) mentioned above, it is possible to have some wrong predictions, which means that, in some cases, certain distractions have not been detected. The three models sometimes classify distracted behaviors of caregivers as 'watching the children' (false positives). These cases represent a risk for children's safety, but fortunately, do not occur often compared with the true positive values for these categories. Inception-v3 obtains fewer false positives for I_watching, with 14 versus 27 and 29 cases for VGG-19 and ResNet-50, respectively. ResNet-50 obtains fewer false positives for O_watching, with 8 versus 21 and 79 cases for VGG-19 and Inception-v3, respectively. We define the false positive rate as subtracting 1 from the specificity or as dividing false positives by the sum of false positives and true negatives. The false-positive rate for I_watching and the three models VGG-19, ResNet-50, and Inception-v3 is 0.43%, 0.46%, and 0.22%, respectively. The false-positive rate for O_watching and the three models (VGG-19, ResNet-50, and Inception-v3) is

0.31%, 0.12%, and 1.18%, respectively. In terms of the false-positive rate, we observe that the obtained values are always very small; VGG-19 and ResNet-50 perform a little worse than Inception-v3 for I_watching. ResNet-50 shows clearly the best results for O_watching.

Furthermore, the three models sometimes classify "watching the children" as distracted behaviors (false negatives). These cases do not pose any risk but could be annoying for caregivers who are warned to supervise the children when they actually were doing so. ResNet-50 and VGG-19 do not obtain any false negatives for I_watching versus 28 cases for Inception-v3. ResNet-50 and VGG-19 obtain fewer false negatives for O_watching, with 2 cases each, versus 79 cases for Inception-v3. If we also consider the false-negative rate for the most relevant categories (we define the false-negative rate as subtracting one from recall), we can see that, for I_watching and the two models VGG-19 and ResNet-50, it is 0% and, for Inception-v3, it is 3.17%. The false-negative rate for O_watching and the two models VGG-19 and ResNet-50 is 0.39% and, for Inception-v3, it is 15.58%. The false-negative rates obtained are very small (except for the O_watching category for Inception-v3). These results show that, for VGG-19 and ResNet-50, the child drowning prevention system works correctly with a minimal error rate versus Inception-v3.

3.2.2.6 Conclusions

In this work, a novel 5G and beyond child drowning prevention system that detects distracted parents or caregivers and alerts them to focus on active child supervision in swimming pools was developed. For this purpose, we evaluated and implemented three well-known CNN models: ResNet-50, VGG-19, and Inception-v3, to process and classify images. The proposed deep CNN models have revealed that they can be used to automatically detect (based on images) possible distractions of a caregiver who is supervising a child and generate alerts to warn them.

The proposed child drowning prevention system can successfully perform a seven-class classification with very high accuracies of 98% for ResNet-50, 94% for VGG-19, and 90% for Inception-v3. VGG-19 and ResNet-50 achieve the same high performance in the most relevant categories I_watching and O_watching, with accuracies of 100% and 99.61%, respectively. For I_watching, the three models achieve an F1-score of 98%. For O_watching, they reach a F1-score of 98%, 99%, and 84% for VGG-19, ResNet-50, and Inception-V3, respectively. In terms of false-positive rate, the obtained values are always very small; VGG-19 and ResNet-50 perform a little worse than Inception-v3 for I_watching. ResNet-50 shows the best results for O_watching. The false-negative rates obtained are also very small (except for the O_watching category for Inception-v3). VGG-19 and ResNet-50 models perform quite well with a minimal false-negative rate versus Inception-v3 for I_watching and O_watching of 0% and 0.39%, respectively. ResNet-50, compared with the other models performs a better classification for most categories. According to the results reached in this research, the proposed system was tested in a swimming pool, but we think it could also be implemented even in swimming lakes or beaches to avoid possible child drowning.

On the other hand, special attention must be paid to security/privacy. Although no doubt distracted parent detection can save lives, associated privacy and security issues need to be analyzed to make our child drowning system socially acceptable. These issues include access rights to data (video images), storage of data, security of data transfer, data analysis rights, and governing policies. The proposed child drowning prevention system may be vulnerable to a variety of active and passive security attacks (such as eavesdropping) with disastrous consequences (especially if unauthorized parties access underage images). For this reason, security and privacy risks should be minimized by applying existing technical solutions such as encryption, authentication mechanisms, cryptographic access control during data collection and transmission, encryption message digests, and hashing to assure the integrity of data during data storage and processing. In addition, further work is also required to maintain the security and confidentiality of data by introducing advanced encryption-based techniques. All of these security and privacy challenges must be addressed so that the proposed child drowning prevention system comes out as a promising way to increase swimming pool safety.

We can define the total reaction time as the time elapsing from an observation (image), its transmission to the edge server, the image processing for activity recognition, and the transmission of an alert (if necessary) based on the observation ($D = D_{UE} + D_{Uplink} + D_{processing} + D_{Downlink}$). In future work, we would like to run the entire system (processing of the images with the neural network and transmission using 5G) in real-time. The expected response time for our child drowning prevention system would be around twenty milliseconds (see Table 3.8). Neural networks have an infinitesimal response time once the weights and the topology have been defined [251]. Further, 5G has been designed to address the requirements of ultra-reliable and low-latency communications. URLLC has stringent requirements for capabilities such as latency, reliability, and availability. Some use cases include wireless control of industrial manufacturing or production processes, remote medical surgery, and transportation safety. Therefore, 5G is the appropriate technology for our use case.

CHAPTER 4

4 REINFORCEMENT LEARNING, EDGE COMPUTING, AND 5G AND BEYOND FOR SMART CITIES.

This chapter is based on:

- Cepeda-Pacheco, Juan Carlos; Domingo, Mari Carmen. Reinforcement Learning and Mobile Edge Computing for 6G-Based Underwater Wireless Networks. (submitted for publication).

Reinforcement learning is a machine learning approach where an agent learns to make sequential decisions by interacting with an environment. It relies on the concept of maximizing a reward signal over time through the performance of actions. This approach resembles the mode by which humans learn since it involves a trial-and-error process. This technique has proven its effectiveness in training systems to make optimal autonomous decisions in dynamic and complex situations.

Smart cities, with their focus on efficiency, sustainability, and improving the quality of life of their inhabitants, benefit greatly from the integration of IoT, reinforcement learning, 5G and beyond, and edge computing. The aim is to facilitate the connectivity of devices and sensors in urban environments, collecting real-time data on traffic, waste management, and energy consumption, among other aspects. Reinforcement learning can use IoT data collected in smart cities to improve decision-making and actions of autonomous systems, such as optimizing public transport routes, efficient energy management in buildings, or coordinating traffic lights to reduce congestion.

In the context of underwater communications, reinforcement learning can play a significant role. Underwater data transmission presents unique challenges due to signal attenuation, interference, and limited bandwidth capacity. Here, reinforcement learning could be applied to optimize the efficiency of underwater communications [252]. For instance, a reinforcement learning agent could dynamically adapt transmission parameters, such as modulation, signal power, or transmission paths, based on changing conditions in the underwater environment. This could improve the reliability and speed of communications by adapting to variations in temperature, salinity, and other factors in the aquatic environment.

4.1 Current Status of Data Collection and Reinforcement Learning in Underwater Applications

In this section, we review the advances in reinforcement learning for collecting data and their use cases for applications in the underwater environment.

4.1.1 Data Collection in Underwater Environments.

Data collection is highly indispensable since it allows the collection of information from the seabed to develop applications for the discovery and control of the underwater environment To collect data underwater, various technologies and methods tailored to underwater conditions are used. The sensors and devices used must be robust and able to resist water pressure, as well as be reliable and precise in hostile environments. AUVs are a crucial tool in this context. These devices can carry out data collection missions autonomously, using sensors such as sonar, cameras, and other specialized instruments to gather information about the seafloor topography.

Different strategies or methods have been implemented for data collection from underwater sensor nodes. On the one hand, in the basic ones (traditional multi-hop data collection) data is relayed from one sensor node to another until it reaches the sink node at the sea surface. This strategy presents several drawbacks such as the time needed for the information to reach the control center and the high energy consumption. On the other hand, more complex strategies require that other aerial/underwater devices (UAVs, AUVs) are involved in data collection; in many cases, they are based on mobile edge computing or use reinforcement learning techniques to efficiently ensure the optimal collection of information from the underwater environment [253][254]. Next, we summarize them.

A. AUV-assisted Data Collection

Different research works have focused on AUV-assisted data collection schemes for the IoUT. In [255] Hybrid Data Collection Scheme (HDCS), which takes into account both real-time data collection and Energy Efficiency (EE), is presented. In [256], based on the energy limitation of underwater devices and the high demand for data collection, an AUV-assisted underwater acoustic sensor network is presented to optimize the energy consumption problem and network performance. In [257], a heterogeneous underwater data collection scheme is presented to optimize the peak Age of Information (AoI) and improve the energy efficiency of the aquatic device nodes. To reduce the energy consumption of resource-constrained devices, the use of aquatic mobile devices (AUVs) is introduced; thereby, the transmission distance between the sensor nodes and the AUV, as well as between the AUV and the sink nodes is reduced. However, since the main goal is only to collect data, it will take time until the information reaches the cloud-based servers and is processed.

Furthermore, some research explores frameworks for enabling edge computing in underwater environments using AUVs.

B. Mobile Edge Computing (MEC) data collection

In [258], a data collection scheme based on an underwater mobile edge element (an AUV) is proposed, and a target selection algorithm is designed to compute the mobility path of the AUV for data collection in a stable 3D environment. In this approach, an AUV is deployed to visit all target nodes and collect data using Magnetic Induction (MI) communication. Here, the AUV acts as a mobile edge platform that processes and stores a large amount of data to be sent to the sink node, and then the sink node sends the data to the cloud. In [259], a service-driven intelligent ocean convergence platform using software-defined networking and edge computing is presented. Similarly, as discussed above, these studies are based on data acquisition, but instead of sending the data to the cloud directly, it is sent to an edge server.

4.1.2 Reinforcement Learning for Underwater Environments

The deployment of reinforcement learning algorithms has become a very useful tool in the development of applications that enable the control, localization, and resource allocation of underwater devices, as well as the discovery of the marine environment. Next, we will mention some of these applications.

In [260] the researchers develop an approach for adaptive control applications of AUVs based on an actor-critical target-oriented DRL architecture. Similarly, in [261] a *Q*_learning-based system for the autonomous control of a bionic underwater robot propelled by undulating fins is presented. Also, in [262] a modified Q-learning-based control approach for AUVS for obstacle avoidance is presented. RL is also used in underwater environments for image collection and its application for image enhancement [263]. In addition, in [264][15] researchers propose a path-planning approach for intelligent underwater vehicles that aims to create digital twins and sensor data. This method consists of mapping the real ocean environment into a virtual digital space. To achieve this, a path planning algorithm based on reinforcement learning (based on Double DQN (DDQN) and Double Dueling DQN (DDDQN)) has been developed, and a detailed exploration of the optimal parameters of the network structure used in the process is carried out.

In the field of underwater localization, in [265], the researchers developed an RL-based localization algorithm to estimate the locations of AUVs and active and passive sensor nodes where a value iteration procedure is performed to solve the Internet of Underwater Things device localization problem. Regarding resource allocation, in addition, in [266] the authors present a network architecture that integrates space-air-ground-sea, considering the requirements of edge computing and cloud computing, by implementing RL (Deep Q-learning) techniques for the intelligent allocation of joint resources. To do so, the information collected using ships and buoys is sent to the MEC server assisted by UAV or Low Earth Orbit (LEO) satellite. In [267] a Multi-Level Underwater Computing (MTUC) framework composed of IoUT devices, AUVs, and surface stations is proposed to intrinsically combine computing, communications, and storage resources.

The successful application of reinforcement learning in underwater communications could contribute significantly to the development of more robust and efficient systems for data transmission in underwater environments. However, continued research and development are required to adapt and optimize reinforcement learning algorithms to the complex conditions and constraints of underwater communications.

4.1.3 Reinforcement Learning-based Methods

In this section, we will briefly review the most common methods based on reinforcement learning.

A. Q-learning

Q-learning is a method proposed by Watkins [268][269] to solve the Markov Decision Process with incomplete information. In other words, *Q*-learning is a reinforcement learning technique where an agent is in a state s and to change to state s', it makes use of the action a that is executed in an environment from which it receives certain information and a reward r. The information that the agent gathers from the environment $Q(s,a)$ or also called *Q*-values, is used to learn the optimal policy π in a Markov decision process [120] to achieve the highest possible reward.

Q in *Q*-learning represents the quality whereby the model finds its next action by improving the quality in each state. For this purpose, the *Q*-Learning algorithm uses the Bellman equation. This equation is used to learn the *Q*-values.

$$Q(s,a) = r + [\gamma max_{a'} Q(s',a')] \tag{4.1}$$

where $\gamma \in \{0,1\}$ is the discount rate that helps to balance the effect of the next rewards on the new values.

The model stores all values for each pair of state actions in a table, namely, the Q table. This table is initialized to zero since it does not include any prior knowledge.

From the Algorithm 2 code, we highlight:

$$Q(s,a) \leftarrow Q(s,a) + \alpha[r + \gamma max_a Q(s',a) \quad Q(s,a)] \tag{4.2}$$

Where α is the learning rate, which is a constant that determines the weight to be added in the Q table.

As a limitation, it's worth noting that *Q*-learning is suitable when small state spaces exist.

```
Algorithm 2 Q-learn algorithm
Initialize $Q(s,a), for\ all\ s \in S^{+}, a \in A(s)$e,  arbitrarily, and except that $Q(terminal, ) = 0$
Repeat (for each episode):
  Initialize $s$
  Repeat (for each step of the episode):
    Choose $a$ from $s$ using policy derived from $Q$ (e.g., ε-greedy)
    Take action $a$, observe $r, s'$
    $Q(s,a) \leftarrow Q(s,a) + \alpha[r + \gamma max_{a} Q(s',a) \quad Q(s,a)]$
    $s \leftarrow s'$
  until s is terminal
end for
```

B.DQN

When there are large state spaces, deep *Q*-learning can help the model directly update the *Q*-table with the appropriate values and perform the tasks more efficiently. Deep *Q*-learning enables the use of the *Q*-Learning strategy by integrating artificial neural networks: Neural Networks (NNs), Deep Neural Networks, and Convolutional Neural Networks. A neural network will help the agent choose the state by receiving inputs [270]. These inputs are the states of the environment. After receiving the input, the neural network will estimate the Q value. The agent will make decisions based on these Q values, for this purpose we should similarly use the Bellman equation for DQN:

$$Q(s,a;\theta) = r + [\gamma max_{\,a'} Q(s',a';\ \theta')] \tag{4.3}$$

We can then, train the neural network and compute the loss or cost function by comparing the target value and the output model. This is possible once we choose the target value $\hat{y}$. Where $\hat{y} = r + \gamma\, max_{\,a_{t+1}}\, Q(s',a';Q')$ (see Algorithm 3)

$$L(\theta) = E[(r + \gamma\, max_{\,a'}\, Q(s',a';\theta') \quad Q(s,a;\theta))^{2}] \tag{4.4}$$

```
Algorithm 3 Deep Q-learning with Experience Delay
Initialize replay memory $D$ to capacity $N$
Initialize action-value function $Q$ with random weights
for episode = 1, $M$ do
  Initialize sequence $s_1 = \{x_1\}$ and preprocessed sequenced $\phi_1 = \phi(s_1)$
  for $t$ = 1, $T$ do
    With probability $\epsilon$ select random action $a_t$
    otherwise, select $a_t = max_a\, Q\ (\phi_1\,(s_t), a; \theta)$
    Execute action $a_t$ in emulator and observe reward $r_t$ and image $x_{t+1}$
    Set $s_{t+1} = s_t, a_t, x_{t+1}$ and preprocess $\phi_{t+1} = \phi_1(s_{t+1})$
    Store transition $(\phi_t, a_t r_t, \phi_{t+1})$ in $D$
    Sample random minibatch of transitions $(\phi_j, a_j, r_j, \phi_{j+1})$ from $D$
  Set $y_j = \begin{cases} r_j & \text{for terminal } \phi_{j+1} \\ r_j + \gamma\, max_{\,a'}\, Q(\phi_{j+1}, a'; \theta) & \text{otherwise} \end{cases}$
    Perform a gradient descent step on $(y_j \quad Q(\phi_j, a_j; \theta))^2$
```

```
        with respect to the network parameters θ
    end for
end for
```

C.DDPG

Q learning and DQN perform well with discrete action spaces, if we discretize continuous action spaces, we could end up with too many action spaces and convergence would be difficult to achieve. The deep deterministic policy gradient algorithm is an extension of the Deep *Q*-learning algorithm when the domain of the action set is continuous [271]. DDPG is a model-free off-policy actor-critic algorithm that combines Deterministic Policy Gradient (DPG) [272] with DQN.

DDPG uses two networks: the Actor-network and the Critic network (see Figure 4.1). The Actor network is represented by $\mu(s|\theta^{\mu})$, which takes a state as input and the action as output, where θ^{μ} are the weights of the Actor-network. Similarly, the Critical network is represented by $Q(s,a|\theta^{Q})$, which takes as input a state and the action; afterward, it returns the value *Q*, where θ^{Q} are the weights of the Critic network.

Similarly, a target network is defined for the actor-network and the Critic network with the same structure: for the actor-network $\mu(s|\theta^{\mu'})$ and for the Critic network $Q\left(s,a|\theta^{Q'}\right)$, where $\theta^{\mu'}$ and $\theta^{Q'}$ are the weights of the Actor and Critic target networks respectively.

The algorithm (Algorithm 4) first selects an action produced by the actor-network, to which the exploration noise N is added such that, $a_t = \mu(s_t|\theta^{\mu}) + N_t$. This action is evaluated in a state s_t, receiving a reward r_t and is placed in the next state s_{t+1}. This transition information (s_t, a_t, r_t, s_{t+1}) is stored in a repetition buffer.

After several iterations, a mini-batch of *N* transitions (s_i, a_i, r_i, s_{i+1}) are sampled from the repetition buffer and the network is trained. We then calculate the target Q value $y_i = r_i + \gamma Q'\left(s_{i+1}, \mu'(s_{i+1}|\theta^{\mu'})|\theta^{Q'}\right)$. We update the weights of our Critic networks with the gradients calculated from the loss *L*.

$$L = \frac{1}{N}\sum_{i}(y_i \quad Q(s_i, a_i|\theta^{Q}))^2 \tag{4.5}$$

Similarly, the actor-network policy is updated using the gradient of the sampled policy:

$$\nabla_{\theta\mu} J \approx \frac{1}{N}\sum_{i} \nabla_{a} Q(s,a|\theta^{Q})|_{s=s_i,a=\mu(s_i)} \nabla_{\theta\mu}\mu(s|\theta^{\mu})\,|_{s_i} \tag{4.6}$$

Next, the weights of the Actor and Critic networks in the target network are also updated slowly, which promotes greater stability; this is referred to as a soft replacement:

$$\theta^{Q'} \leftarrow \tau\theta^{Q} + (1 \quad \tau)\theta^{Q'} \tag{4.7}$$

$$\theta^{\mu'} \leftarrow \tau\theta^{\mu} + (1 \quad \tau)\theta^{\mu'} \tag{4.8}$$

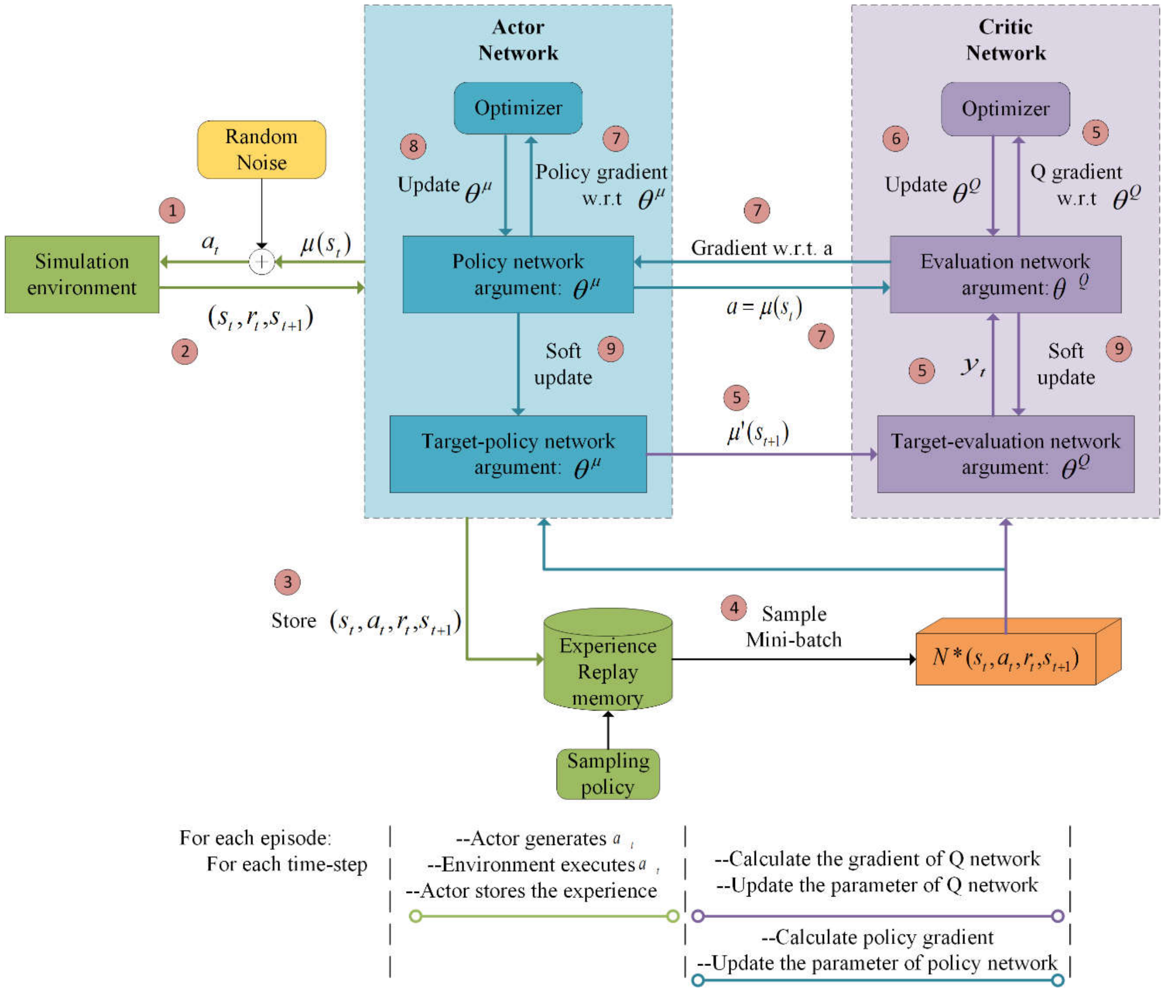

Actor Network
Critic Network
Optimizer
Optimizer
Random Noise
8 Update θ^{μ}
7 Policy gradient w.r.t θ^{μ}
6 Update θ^{Q}
5 Q gradient w.r.t θ^{Q}
1
a_t
$\mu(s_t)$
Simulation environment
Policy network argument: θ^{μ}
7 Gradient w.r.t. a
Evaluation network argument: θ^{Q}
$a = \mu(s_t)$
7
(s_t, r_t, s_{t+1})
2
Soft update 9
5 y_t
Soft update 9
5
$\mu'(s_{t+1})$
Target-policy network argument: θ^{μ}
Target-evaluation network argument: θ^{Q}
3 Store (s_t, a_t, r_t, s_{t+1})
4 Sample Mini-batch
Experience Replay memory
$N*(s_t, a_t, r_t, s_{t+1})$
Sampling policy
For each episode:
For each time-step
--Actor generates a_t
--Environment executes a_t
--Actor stores the experience
--Calculate the gradient of Q network
--Update the parameter of Q network
--Calculate policy gradient
--Update the parameter of policy network

Figure 4.1. Deep Deterministic Policy Gradient (DDPG) algorithm structure.

Algorithm 4 DDPG algorithm

```
Randomly initialize critic network $Q(s,a|\theta^Q)$ and actor $\mu(s|\theta^\mu)$ with weights $\theta^Q$ and $\theta^\mu$.
Initialize target network $Q'$ and $\mu'$ with weights $\theta^{Q'} \leftarrow \theta^Q, \theta^{\mu'} \leftarrow \theta^\mu$
Initialize replay buffer $R$
for episode = 1, $M$ do
    Initialize a random process $N$ for action exploration.
    Receive initial observation state $s_1$
    for $t = 1, T$ do
        Select action $a_t = \mu(s_t|\theta^\mu) + N_t$ according to the current policy and exploration noise.
        Execute action $a_t$ and observe reward $r_t$ and observe the new state $s_{t+1}$
        Store transition $(s_t, a_t, r_t, s_{t+1})$ in $R$
        Sample a random minibatch of $N$ transitions $(s_i, a_i, r_i, s_{i+1})$ from $R$
        Set $y_i = r_i + \gamma Q'(s_{i+1}, \mu'(s_{i+1}|\theta^{\mu'})|\theta^{Q'})$
        Update critic by minimizing the loss: $L = \frac{1}{N}\sum_i (y_i \quad Q(s_i, a_i|\theta^Q))^2$
        Update the actor policy using the sampled policy gradient:
```

```
        $\nabla_{\theta^\mu} J \approx \frac{1}{N}\sum_i \nabla_a Q(s,a|\theta^Q)|_{s=s_i,a=\mu(s_i)} \nabla_{\theta^\mu}\mu(s|\theta^\mu)|_{s_i}$
      Update the target networks:
            $\theta^{Q'} \leftarrow \tau\theta^Q + (1 \quad \tau)\theta^{Q'}$
            $\theta^{\mu'} \leftarrow \tau\theta^\mu + (1 \quad \tau)\theta^{\mu'}$
    end for
end for
```

4.2 Contribution: Reinforcement Learning and Mobile Edge Computing for 6G-based Underwater Wireless Networks

6G-based ground-air-underwater networks represent a groundbreaking paradigm shift in wireless communication. These networks aim to provide seamless connectivity across terrestrial, aerial, and aquatic domains, enabling unprecedented data exchange and interaction between sensors, devices, and vehicles operating in these diverse environments This fosters the creation of innovative applications. Concerning the aquatic domain, the Internet of Underwater Things (IoUT) [74] can be defined as a worldwide network of smart interconnected underwater objects with a digital entity. These devices sense, interpret, and react to the environment due to the combination of the Internet, powerful tracking technologies, and embedded sensors.

Data is sent from the sensors to the surface for computation and processing. Underwater devices use different communication techniques to transmit information [273], with acoustic communication being the most commonly used. Data transmission directly from underwater sensor nodes to the surface sink is very energy-intensive. Autonomous Underwater Vehicles (AUVs) can help reduce communication distances between sensors and sink nodes by collecting sensor data. The sink nodes, likewise, send the collected information to ships or unmanned aerial vehicles (in the aerial domain) via radio frequency waves [274] or via satellite communications. Finally, the data is sent to the ground stations in the terrestrial domain, where it is stored in the servers (cloud computing). At this point, the data is processed, and depending on the type of application the result must be sent back, as soon as possible, to the underwater devices. In this process, latency is the most detrimental factor for applications that have real-time or mission-critical constraints such as large-scale sensing data fusion, navigation systems, real-time sensing data fusion, and more [275]. To cope with this requirement, Mobile Edge Computing (MEC) has been developed. It is implemented in devices close to the local devices [276]. These nearby devices should be equipped with cloud-like computing resources, providing computational services, therefore enabling high reliability, scalability, and low latency in underwater networks.

Most prior works ignore the computing capability provided by AUVs in the underwater medium, and only a few studies consider AUVs as edge computing nodes able to execute tasks [253]. In this research, we distinguish between “local AUVs” and a “MEC AUV”. “Local AUVs” may have limited processing capabilities. The ‘MEC AUV’, on the other hand, is specialized to perform computational tasks efficiently. AUV-enabled MEC systems involving IoUT devices, cluster-heads, local AUVs, and MEC AUVs have not been studied.

In this work, we envision an innovative AUV-enabled MEC system where cluster-heads that collect data from IoUT devices offload their associated computing tasks to local AUVs. These AUVs are strategically placed (1) execute these tasks fully locally, (2) execute them partially locally and offload the rest or (3) offload them fully to a more resourceful AUV (MEC AUV).

Despite the advantages of AUV-assisted MEC, several challenges in network deployment and operation should be considered. Firstly, it is challenging how much computation an MEC-enabled AUV should allocate to each offloaded task from a local AUV considering the limited number of onboard resources. Furthermore, it is also challenging how to control each of the AUVs' trajectories (diving direction and speed) considering that each local AUV must serve cluster-heads on its way and an MEC-enabled AUV has to serve different local AUVs in the different collection points. In addition, it is challenging how to determine the optimal route for the AUVs taking into account the effect of ocean currents on their trajectory.

Inspired by the challenges mentioned above, we propose an algorithm based on deep reinforcement learning, Deep Deterministic Policy Gradient (DDPG), that enables, first to minimize the energy consumption and delay of all tasks, by jointly optimizing the task offloading strategy, resource allocation, and AUV trajectory. An efficient trajectory optimization, task offloading, and resource allocation model is formulated as a non-convex optimization problem that aims to minimize the weighted sum of the service delay of all local AUVs (task offloading delay and computation delay) and the AUV energy consumption (transmission energy and computation energy).

4.2.1 System Model

In this section, we define the scenario for data processing (real-time response for mission-critical applications) and data collection (cloud storage) tasks. Afterward, we analyze the reinforcement learning algorithm used for the intelligent offloading of tasks to the edge device. For clarity, we summarize all the following notations and their definitions in Table 4.1.

A. Network Architecture

Figure 4.2 shows the proposed network architecture. Along the seabed, several sensor nodes (IoUT nodes fixed at the seafloor) are randomly deployed in a 3D $(L\ x\ W\ x\ H)$ cartesian coordinate system. They are grouped based on their location, and then a cluster head (CH) node $p \in \{1,2,\dots,P\}$ is chosen among them to avoid excessive energy consumption when each node sends its collected data individually. Two types of $AUVs$, namely a set of $AUVs_{LOCAL}$ denoted as $K = \{1,2,\dots,K\}$ and a AUV_{MEC} are employed for this scenario. Each AUV_{LOCAL} is configured to collect data from the cluster heads; an AUV_{MEC} can receive data from several $AUVs_{LOCAL}$ and send it to the sink node located on the sea surface.

Moreover, for complete and uninterrupted coverage, each AUV_{LOCAL} moves in a two-dimensional (2D) horizontal plane, to collect data from the cluster heads in a specific area and at a certain depth, without surfacing. All $AUVs_{LOCAL}$ are deployed at the same depth and close to the CHs. On the other hand, the AUV_{MEC} moves also in a two-dimensional (2D) horizontal plane but at a different depth, between the $AUVs_{LOCAL}$ and the sink node, to serve all the $\mathrm{AUVs}_{\mathrm{LOCAL}}$. The AUV_{MEC} uses collection points also called strategic points denoted as $i \in I = \{1,2,\dots,I\}$; i.e. the AUV_{MEC} moves within the horizontal plane traversing each of these strategic points to receive the information from the $AUVs_{LOCAL}$ and communicate with the sink node. Therefore, this process results in energy savings reflected in a subsea operation for a much longer time. For all $AUVs_{LOCAL}$, we assume that the communication is based on Orthogonal Frequency Division Multiplexing Access (OFDMA) [277], so mutual interference between them is not taken into account.

We propose to bring computing resources closer to IoUT devices. Therefore, we consider that the AUV_{MEC} provides mobile edge computing in addition to the data collection service. IoUT devices are organized into clusters and one node is selected as cluster head in each cluster. The IoUT devices sense data and send it to a cluster head. Each cluster head performs data aggregation and generates a task related to this data. We assume that each jth cluster head generates one computation-intensive task in the tth time slot. This means that T tasks are generated for each cluster head, and we have $t \in T = \{1,2,\dots,T\}$.. If a task is related to data processing, the cluster head node sends the input data from the task to the nearest AUV_{LOCAL}. The $\mathrm{AUV}_{\mathrm{LOCAL}}$ uses a reinforcement learning algorithm to decide whether the task should be processed fully locally, fully offloaded, or partially offloaded to the AUV_{MEC}. This approach aims to achieve energy savings in data processing and reduce the overall delay in task execution. We assume that the $\mathrm{AUV}_{\mathrm{MEC}}$ has enough powerful resources to process the data and return the results to the AUV_{LOCAL}, which means there is no need to send the data to a central server for cloud computing. Therefore, the system performance is improved, and the latency is reduced by processing the information at the edge. This is essential for mission-critical applications with strict reliability and latency requirements [278]. On the other hand, if the task is related to data collection, the $\mathrm{AUVs}_{\mathrm{LOCAL}}$ send the data to a $\mathrm{AUV}_{\mathrm{MEC}}$. The AUV_{MEC} sends the data to the sink nodes located on the sea surface, which forwards the data through the unmanned aerial vehicle (UAV) to the Ground Base Station (GBS) for storage in the cloud servers. The cloud-computing-based data processing is useful in non-mission-critical IoUT applications [278].

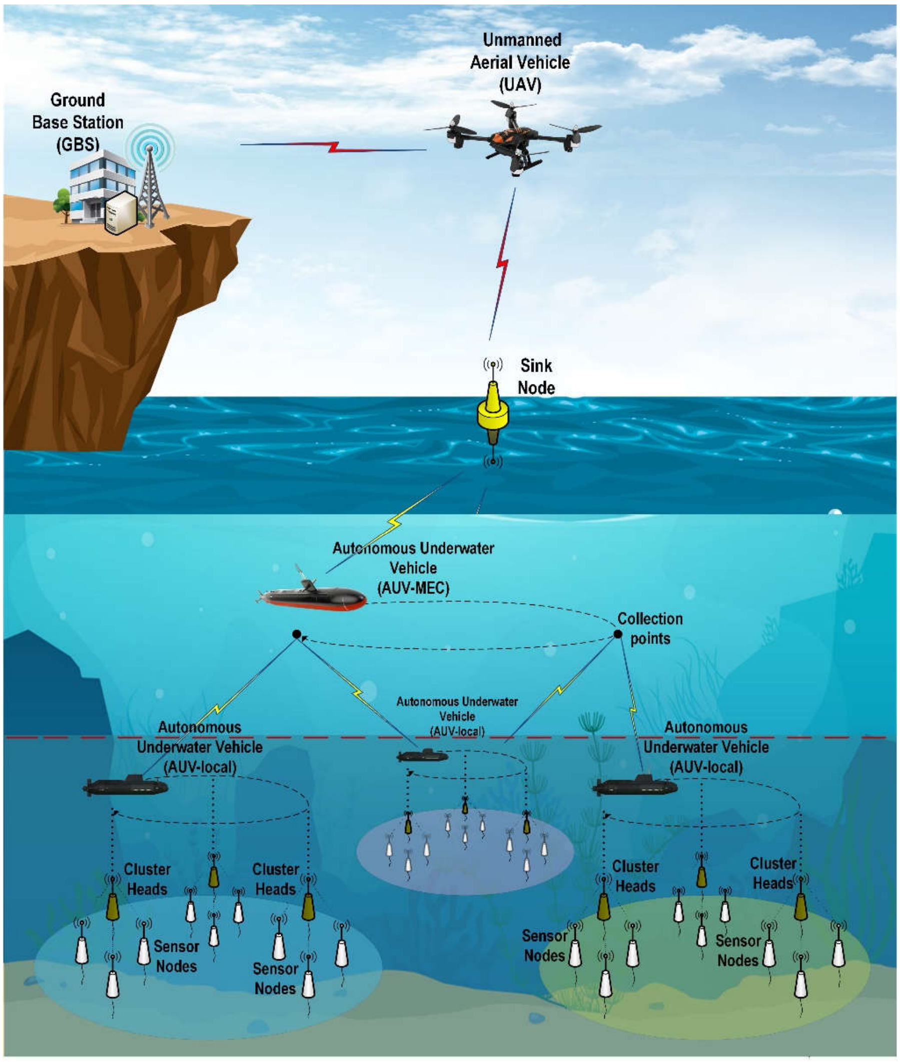


Figure 4.2. Proposed network architecture.

Table 4.1. Main notation list of this section.

Symbol	Definition
$L\ x\ W\ x\ H$	3D cartesian coordinate system.
p, P, P	Index, number, set of cluster head nodes.
k, K, K	Index, number, set of local AUVs (AUV$_{\mathrm{LOCAL}}$).
j, R_k, R_k	Index, number, set of cluster head nodes only associated with AUV$_{\mathrm{LOCAL}}$ k.
i, I, I	Index, number, set of collection/strategic points.
t, T, T	Index, number, set of time slots
AUV_{LOC}	A local AUV
AUV_{MEC}	A Mobile Edge Computing AUV
R_k	The total number of cluster heads associated by AUV$_{\mathrm{LOCAL}}$ k.
$M_{k,j}$	The task that may be executed fully locally by the AUV$_{\mathrm{LOCAL}}$, fully offloaded to the AUV$_{\mathrm{MEC}}$, or partially offloading between AUV$_{\mathrm{LOCAL}}$ and AUV$_{\mathrm{MEC}}$
$L_{k,j}$	The size of computation input data.
$C_{k,j}$	The total number of CPU cycles required to accomplish the computation task $M_{k,j}$.
α_k	The offloading decision {0, 1}
(l, f)	Distance or transmission range, Signal frequency.
$SL(f), A(l,f)\ , N(f), DI$	Source level, Transmission loss, Noise level, Directivity index.
$N_t(f), N_s(f), N_w(f), N_{th}(f)$	Ambient noise: turbulence, shipping, waves, and thermal noise, respectively
$k, \alpha(f)$	The spreading factor and absorption coefficient.
s	The shipping activity factor (0,1).
w	Wind speed

I_T^{SW}, I_T^{DW}	The transmission intensity for shallow (depth lower than 100 m) and deep water, respectively.
P_T, z	The transmission power, and depth.
v_{mec}, v_{local}, v_{mec}^{max}, v_{local}^{max}	AUV$_{\mathrm{MEC}}$, AUV$_{\mathrm{LOCAL}}$ speed, and maximum speed, respectively
V_t, ai	The shortest path planned by the AUVs, and the angle to the x-axis, respectively
V, θ^h	AUV speed and angle between V and the x-axis, respectively
Uc, $ai1$	Ocean current speed, and angle between Uc and x-axis, respectively
V_d, V_n	Rectangular components of the vector V to the vector V$_{\mathrm{t}}$
Uc_d, Uc_n	Rectangular components of the vector Uc to the vector V$_{\mathrm{t}}$
V_L	The magnitude of the resulting AUV desired trajectory
θ_{mec}^h, θ_k^h	AUV$_{\mathrm{MEC}}$, AUV$_{\mathrm{LOCAL}}$ velocity vector angle, respectively
$d_{mec,i}^h$, $d_{k,j}^h$	AUV$_{\mathrm{MEC}}$-to-strategic point, and AUV$_{\mathrm{LOCAL}}$-to-Cluster head horizontal distances, respectively.
$[X,Y,Z]$, X^{max}, Y^{max}	AUV$_{\mathrm{MEC}}$ position and maximum limit, respectively
$[x_k, y_k, z_k]$, x_k^{min}, y_k^{min}, x_k^{max}, y_k^{max}	AUV$_{\mathrm{LOCAL}}$ position, minimum, and maximum limits, respectively
$[x_i, y_i, z_i]$	Coordinates of the collection points
$[x_j, y_j, z_j]$	Coordinates of the cluster heads
$D_{tx}^{ch,auvloc}$, $D_{tx}^{auvloc,auvmec}$	Cluster head-to-AUV$_{\mathrm{LOCAL}}$, and AUV$_{\mathrm{LOCAL}}$-to-AUV$_{\mathrm{MEC}}$ transmission delays, respectively.
$r_{ch,auvloc}$, $r_{auvloc,auvmec}$	Cluster head-to-AUV$_{\mathrm{LOCAL}}$, and AUV$_{\mathrm{LOCAL}}$-to-AUV$_{\mathrm{MEC}}$ data rates, respectively.
$D_{prop}^{ch,auvloc}$, $D_{prop}^{auvloc,auvmec}$	Cluster head-to-AUV$_{\mathrm{LOCAL}}$, and AUV$_{\mathrm{LOCAL}}$-to-AUV$_{\mathrm{MEC}}$ propagation delays, respectively.
$d_{k,j}^e$, $d_{mec,k}^e$	Cluster head-to-AUV$_{\mathrm{LOCAL}}$, and AUV$_{\mathrm{LOCAL}}$-to-AUV$_{\mathrm{MEC}}$ transmission distances, respectively.
v_{prop}	The nominal speed of sound underwater.
D_{AUV_L}, D_{AUV_M}	AUV$_{\mathrm{LOCAL}}$, and AUV$_{\mathrm{MEC}}$ execution delays, respectively.
f_{AUV_L}, F	AUV$_{\mathrm{LOCAL}}$, and AUV$_{\mathrm{MEC}}$ computing capacities, respectively.
f_{kj}^m	The specific portion of F allocated to AUV$_{\mathrm{LOCAL}}$ k, $f_{kj}^m \in [0,1]$
$E_{k,j}^L$, $E_{k,j}^O$, $E_{k,j}^M$	AUV$_{\mathrm{LOCAL}}$, Offloading and AUV$_{\mathrm{MEC}}$ energy consumption, respectively
$P_{tx}^{auvloc,auvmec}$	The data transmission power of the AUV$_{\mathrm{LOCAL}}$ k
D_{rx}	AUV$_{\mathrm{MEC}}$-to-AUV$_{\mathrm{LOCAL}}$ transmission delay
L_m	Size of the calculated result
r_{rx}	The download data rate of AUV$_{\mathrm{LOCAL}}$
$D_{k,j}^L$	Total time to complete a task $M_{k,j}$ locally
$D_{k,j}^O$	Total time to complete a task $M_{k,j}$ using computational offloading
$D_{k,j}$	Total time to complete the task $D_{k,j}$
$E_{k,j}$	Overall energy consumption

B. Task Model

We consider that in our proposed scenario the $AUVs_{LOCAL}$ and cluster heads are denoted as $K = \{1,2,\dots,K\}$ and $P = \{1,2,\dots,P\}$, respectively. The total number of cluster heads served by $AUV_{LOCAL}\ k$, $k \in \{1,2,\dots,K\}$, can be represented by R_k only associated by AUV$_{\mathrm{LOCAL}}$ k denoted as $j \in R_k = \{1,\dots,R_k\}$, i.e., each cluster head j sends task requests to their respective $AUV_{LOCAL}\ k$. The computation-intensive task is denoted by $M_{k,j}$ $\{L_{k,j}, C_{k,j}\}$, where $L_{k,j}$ refers to the size of computation input data, $C_{k,j}$ denotes the total number of CPU cycles required to accomplish the computation task $M_{k,j}$. We assume that the AUV_{LOCAL} has the capability to decide whether the task $M_{k,j}$ 1) can be executed locally in the edge-computing-enabled AUV_{LOCAL}, 2) be offloaded completely to the AUV-enabled MEC server AUV_{MEC}, or 3) be partially executed on both AUV_{LOCAL} and AUV_{MEC}. We denote the offloading decision by a continuous variable $\alpha_{k,j}(t) \in \{0,1\}, \forall\ k\ \in K, \forall\ j\ \in R_k, \forall\ t\ \in T$, where $\alpha_{k,j}(t) = 0$ represents that in the slot t, $AUV_{LOCAL}\ k$ executes the task fully locally,

$\alpha_{k,j}(t) = 1$ represents that AUV_{LOCAL} k fully offloads the task to the AUV_{MEC}, and $0 < \alpha_{k,j}(t) < 1$ represents the rate of partially offloading of the task.

C. Communication Model

We consider two communications interfaces, i.e., cluster head-to-AUV_{LOCAL} and AUV_{LOCAL}-to-AUV_{MEC}. Next, the data rate for each interface is analyzed. For this purpose, the underwater acoustic channel is introduced.

1) Underwater acoustic channel

The narrow-band *SNR* of an emitted underwater signal at the receiver can be expressed by the passive sonar equation [279]:

$$SNR\ (l,f) = SL(f) \quad A(l,f) \quad N(f) + DI \geq DT \tag{4.9}$$

where *DT* has been defined as the detection threshold, $SL(f)$ is the source level, $A(l,f)$ is the transmission loss, $N(f)$ is the noise level and DI is the directivity index. Since we assume an omnidirectional source, DI is equal to zero.

The attenuation, transmission loss, or path loss over a transmission range l for a frequency f (in kHz for underwater communications) can be given by [280].

$$10\, log\, A(l,f) = k \quad 10\, log\, l + l \quad 10\, log\, \alpha(f) \tag{4.10}$$

where k is the spreading factor that defines the geometry of the propagation ($k = 1$ for shallow water (cylindrical spreading) and $k = 2$ for deep water (spherical spreading) and $k = 1.5$ for practical spreading) and $\alpha(f)$ is the absorption coefficient.

The absorption coefficient can be calculated using Thorp's expression where $\alpha(f)$ is expressed in dB/km and f in kHz [281]. For frequencies above a few hundred Hz it is given as:

$$10\, log\, \alpha(f) = \frac{0.11\, f^2}{1 + f^2} + \frac{44\, f^2}{4100 + f^2} + 2.75 \times 10^{-4}\, f^2 + 0.003 \tag{4.11}$$

and for lower frequencies it is given by:

$$10\, log\, \alpha(f) = 0.002 + \frac{0.11\, f^2}{1 + f^2} + 0.011\, f^2 \tag{4.12}$$

The ambient noise can be modeled by four basic sources [282]

$$N(f) = N_t(f) + N_s(f) + N_w(f) + N_{th}(f) \tag{4.13}$$

where the ambient noise due to turbulence is $N_t(f)$, the noise due to shipping is $N_s(f)$, the noise due to wind is $N_w(f)$ and thermal noise is $N_{th}(f)$. Those are described by the following equations.

$$10\, log\, N_t(f) = 17 \quad 30\, log(f) \tag{4.14}$$

$$10\, log\, N_s(f) = 40 + 20(s \quad 0.5) + 26\, log(f) \quad 60\, log(f + 0.03) \tag{4.15}$$

where s is the shipping activity factor whose value ranges between 0 and 1 for low and high activity, respectively.

$$10\, log\, N_w(f) = 50 + 7.5w^{1/2} + 20\, log(f) \quad 40\, log(f + 0.4) \tag{4.16}$$

where w is the wind speed in m/s.

$$10\, log\, N_{th}(f) = \quad 15 + 20\, log(f) \tag{4.17}$$

The dominant noises according to the operation frequencies are turbulence noise for f < 10 Hz, distant shipping for f = 10–100 Hz, surface motion caused by wind-driven waves for f = 100 Hz–100 kHz (operating region used by the majority of the acoustic systems) and thermal noise for f > 100 kHz. The noise level generally decreases with increasing frequency.

The source level *SL* is related to the intensity I_T as $SL = 10\, log\left(\frac{I_T}{0.67\times10^{-18}}\right)$. The relation between the intensity I_T, and the transmission power of the transceiver, P_T, is expressed as

$$I_T^{SW} = \frac{P_T}{2\pi.z},\ I_T^{DW} = \frac{P_T}{4\pi} \tag{4.18}$$

for shallow and deep water, respectively, where P_T is given in watts and z is the depth in meters.

If we consider the frequency-dependent part of the narrow-band SNR $\gamma(l, f)$ =1/($A(l, f)N(f)$), it has been shown in [282] that for each transmission distance, there exists an optimal frequency $f_0(l)$ for which the SNR is maximized. It is assumed that the noise is Gaussian and the channel is time-invariant for some interval of time [282]. The total bandwidth can be divided into many narrow sub-bands, adding their individual capacities. The ith sub-band is centered around the frequency $f_i, i = 1,2, \ldots$ and it has a width $\quad f$. The maximum data rate supported by an underwater acoustic channel for a given source power and source/receiver (that is, the channel capacity) can be obtained as [282]

$$r_{tx} = \sum\nolimits_i \quad f\, log_2\left(1 + \frac{SL(l, f)}{A(l, f)N(f)}\right) = B\, log_2\left[1 + \frac{I_T\gamma(l, f)}{1\mu Pa}\right] \tag{4.19}$$

where $1\mu Pa = 0.67 \times 10^{-18}$.

Therefore, the data rate of the cluster head-to-AUV_{LOCAL} acoustic link denoted by $r_{ch,auvloc}$ and the data rate of the AUV_{LOCAL}-to-AUV_{MEC} acoustic link denoted by $r_{auvloc,auvmec}$ are lower than or equal to the channel capacity of the corresponding Additive White Gaussian Noise (AWGN) channel.

D. Computing Model

1) The Velocity Synthesis Approach

Accounting for the effect of ocean currents in the underwater environment, $AUVs$ ($AUVs_{LOCAL}$ or AUV_{MEC}) may deviate from their trajectories, especially where the current is opposite to the AUV's direction of movement. In our approach, we consider that the $AUVs$ move to reach their target points, using a planned trajectory. We consider the straight line joining the AUV's initial position and the target point coordinate as the shortest path; the objective will be to control the AUV's movements along the shortest trajectories towards the desired targets. To solve this problem, we present a simple synthesis velocity algorithm (see Figure 4.3) that decomposes the ocean current velocity and the AUV velocity to make the summed velocity point directly to the desired target.

The lower left point in the figure represents the AUV and the higher point represents the target point). The vector V_t indicates the shortest path planned by the AUV. The angle between V_t and the x-axis is ai. The vector V indicates the speed of the AUV (v_{mec}, $\mathrm{v_{local}}$), which can be adjusted according to the system requirements. The angle between V and the x-axis is θ^h (θ^h_{mec}, θ^h_k). The vector Uc represents the ocean current velocity, and the angle between Uc and the x-axis is $ai1$.

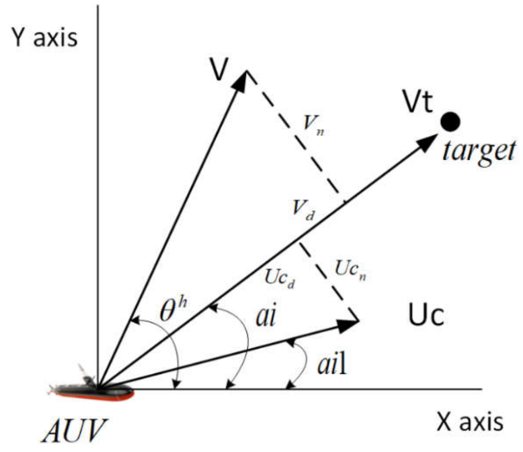


Figure 4.3. Velocity synthesis algorithm.

The main challenge is to find out $\theta^h$to ensure that the AUV moves along V_t. The component of the vehicle velocity that assists the motion along the desired path vector V_t has a magnitude $V_d = V \quad cos(\theta^h \quad ai)$. The component of the vehicle velocity that is perpendicular to the desired trajectory is denoted by $V_n = V \quad sin(\theta^h \quad ai)$. Similarly, the component of the ocean current that assists motion along the desired trajectory is $Uc_d = Uc \quad cos(ai \quad ai1)$, its component that is perpendicular to the desired trajectory is $Uc_n = Uc \quad sin(ai \quad ai1)$. To keep the AUV in the planned direction requires V_n to cancel Uc_n, which can be described as the following equation [283]:

$$V \quad sin(\theta^h \quad ai) = Uc \quad sin(ai \quad ai1) \tag{4.20}$$

From the above equation, it can be calculated

$$\theta^h = arcsin\left(Uc \quad \frac{sin(ai \quad ai1)}{V}\right) + ai \tag{4.21}$$

The speed synthesis algorithm implementation is based on the precondition,

$$|Uc| < |V| \tag{4.22}$$

$$V_L = V_d + Uc_d \tag{4.23}$$

Summarizing, by combining equations (10) to (13), θ^h y V_L can be calculated, where V_L is the magnitude of the resulting AUV velocity along the desired trajectory.

2) AUVs trajectory.

As mentioned above, we assume that all $AUVs_{LOCAL}$ are at the same depth, while AUV_{MEC} is at a different depth; both move within a range of coverage that spans the horizontal plane. Their motions are based on the direction of motion (ai), the initial AUV velocity (V), and the ocean current velocity (Uc) using the velocity synthesis approach. We also assume that in the *t*-th time slot the AUV_{LOCAL} k moves to serve the cluster head j in the direction of its target. The AUV_{MEC} moves from the starting point to the strategic location points to serve the $AUVs_{LOCAL}$.

The horizontal distance between the AUV_{MEC} to a strategic point is given by

$$d_{mec,i}^h(t) = \sqrt{\left(X(t) \quad x_i(t)\right)^2 + \left(Y(t) \quad y_i(t)\right)^2} \tag{4.24}$$

where $[X(t), Y(t), Z]$ and $[x_i(t), y_i(t), z_i]$ denote the coordinates of the AUV_{MEC} and a strategic point, respectively; here $Z = z_i = 0$. Similarly, the horizontal distance from AUV_{LOCAL} k to the cluster heads j is given by

$$d_{k,j}^h(t) = \sqrt{\left(x_k(t) \quad x_j(t)\right)^2 + \left(y_k(t) \quad y_j(t)\right)^2} \tag{4.25}$$

where $[x_k(t), y_k(t), z_k]$ and $\left[x_j(t), y_j(t), z_j\right]$ represent the coordinates of the AUV_{LOCAL} k and cluster head j (to visit), respectively; here $z_k = z_j = 0$.

For security reasons, each AUV can only move in a rectangular area to avoid possible collisions; whose maximum lengths are denoted for the AUV_{MEC} as

$$0 \leq X(t) \leq X^{max}, \forall\ t \in T \tag{4.26}$$

$$0 \leq Y(t) \leq Y^{max}, \forall\ t \in T \tag{4.27}$$

and the minimum and maximum lengths for the AUV_{LOCAL} k are

$$x_k^{min} \leq x_k(t) \leq x_k^{max}, \forall\ k \in K, t \in T \tag{4.28}$$

$$y_k^{min} \leq y_k(t) \leq y_k^{max}, \forall\ k \in K, t \in T \tag{4.29}$$

Therefore, the $AUVs_{LOCAL}$ move along a path from the nearest cluster head to the last one, guaranteeing that each cluster head is covered and served only once.

To get the trajectory of the AUV_{MEC}, we consider the displacement at each time interval t, where $X(t) = X(0) + \sum_{l=1}^{t} d_{mec,i}^h(l)\, cos\left(\theta_{mec}^h(l)\right)$, $Y(t) = Y(0) + \sum_{l=1}^{t} d_{mec,i}^h\, (l)\, sin(\theta_{mec}^h(l))$ here $[X(0), Y(0), Z]$ is the initial coordinate of the AUV_{MEC}; as well as for the AUV_{LOCAL} k trajectory, where $x_k(t) =$

$x_k(0) + \sum_{l=1}^{t} d_{kj}^{h}\,(l)\,cos\left(\theta_k^h(l)\right)$, $y_k(t) = y_k(0) + \sum_{l=1}^{t} d_{k,j}^{h}\,(l)\,sin(\theta_k^h(l))$ here $[x_k(0), y_k(0), z_k]$ is the initial coordinate of the AUV_{LOCAL} k.

3) Cluster Head Data Collection

We assume that the AUV_{MEC} and the AUV_{LOCAL} have reached the desired positions to maintain coverage between the AUV_{LOCAL} and its cluster heads as well as between the AUV_{MEC} and the AUV_{LOCAL}.

Next, each cluster head $j \in R_k = \{1, \dots, R_k\}$, first offloads the input data of a task $M_{k,j}$ to the *k*-th AUV_{LOCAL}. The required times are the transmission delay $D_{tx}^{ch,auvloc}(t)$ and the propagation delay $D_{prop}^{ch,auvloc}(t)$ at time slot t. They are given by

$$D_{tx}^{ch,auvloc}(t) = \frac{L_{k,j}(t)}{r_{ch,auvloc}} \tag{4.30}$$

where $r_{ch,auv}$ stands for the uplink rate of $CH\ j$ in the underwater medium.

To get the propagation delay $D_{prop}^{ch,auvloc}(t)$ we first calculate the Euclidian distance between the AUV_{LOCAL} k, and the cluster head *j*.

$$d_{k,j}^{e}(t) = \sqrt{\left(x_k(t) \quad x_j(t)\right)^2 + \left(y_k(t) \quad y_j(t)\right)^2 + \left(z_k(t) \quad z_j(t)\right)^2} \tag{4.31}$$

$$D_{prop}^{ch,auvloc}(t) = \frac{d_{k,j}^{e}(t)}{v_{prop}} \tag{4.32}$$

where v_{prop}=1500 m/s is the nominal speed of sound underwater.

Afterward, if the task $M_{k,j}$ is a computation task, the k-th AUV_{LOCAL} has to 1) execute the task $M_{k,j}$, locally in the edge-computing-enabled $\mathrm{AUV_{LOCAL}}$, 2) offload it completely to the AUV-enabled MEC server AUV_{MEC}, or 3) partially execute it on both AUV_{LOCAL} and AUV_{MEC}.

4) Local or Partially Local Computing Model

When the AUV_{LOCAL} chooses to execute the whole task $M_{k,j}$ locally, $\alpha_{k,j}(t) = 0$. Otherwise, if partially offloading is selected, $\alpha_{k,j}\,L_{k,j}(t)$ represents the input data volume offloaded to the $\mathrm{AUV_{MEC}}$ and $\left(1 \quad \alpha_{k,j}\right) L_{k,j}(t)$ represents the input data volume left for local computation and the total local or partially local execution delay of the AUV_{LOCAL} k at time slot t would be represented by:

$$D_{AUV_L}(t) = \frac{\left(1 \quad \alpha_{k,j}\right) L_{k,j}(t)\, C_{k,j}(t)}{f_{AUV_L}} \tag{4.33}$$

where $L_{k,j}(t)$ indicates the size of the computing task, $C_{k,j}(t)$ indicates the CPU cycles required to process each unit byte, and f_{AUV_L} indicates the computing capacity of the AUV_{LOCAL}. Also, the power required by each of the $AUVs_{LOCAL}$ to execute the task locally can be evaluated as follows:

$$E^{L}_{k,j}(t) = \mu(f_{AUV_L})^{\lambda} D_{AUV_L}(t) \tag{4.34}$$

where μ is a constant that depends on the average switched capacitance and the average activity factor, while λ is a constant typically set to 3 [284].

5) Offloading or Partially Offloading Computing Model

It is considered that the $AUV_{LOCAL}\ k$ chooses to execute the task $M_{k,j}$ by offloading computing completely ($\alpha_{k,j}(t) = 1$) or partially ($0 < \alpha_{k,j}(t) < 1$). The complete offloading approach is divided into three steps. Firstly, $AUV_{LOCAL}\ k$ needs to upload sufficient input data (i.e., program codes and parameters) to AUV_{MEC} in the underwater medium. Then the AUV_{MEC} allocates part of the computational resource and executes the computing task. Finally, the AUV_{MEC} returns the execution results to $AUV_{LOCAL}k$. Based on the steps above, the time required for the first step of offloading computing is the transmission delay and the propagation delay. The transmission delay $D^{auvloc,auvmec}_{tx}(t)$ can be represented by:

$$D^{auvloc,auvmec}_{tx}(t) = \frac{\alpha_{k,j}\ L_{k,j}(t)}{r_{auvloc,auvmec}} \tag{4.35}$$

where $r_{auvloc,auvmec}$ stands for the uplink rate of $AUV_{LOCAL}\ k$ in the underwater medium.

The Euclidian distance between the $AUV_{LOCAL}\ k$ and the AUV_{MEC} is represented as

$$d^{e}_{mec,k}(t) = \sqrt{\left(X(t) \quad x_k(t)\right)^2 + \left(Y(t) \quad y_k(t)\right)^2 + \left(Z(t) \quad z_k(t)\right)^2} \tag{4.36}$$

The propagation delay can be given by

$$D^{auvloc,auvmec}_{prop}(t) = \frac{d^{e}_{mec,k}(t)}{v_{prop}} \tag{4.37}$$

where v_{prop}=1500 m/s is the nominal speed of sound underwater.

The overall energy required by the $AUV_{LOCAL}\ k$ to transmit to the AUV_{MEC} in the time slot (t) is given by:

$$E^{O}_{k,j}(t) = P^{auvloc,auvmec}_{tx} D^{auvloc,auvmec}_{tx}(t) \tag{4.38}$$

where $P^{auvloc,auvmec}_{tx}$ is the data transmission power of the $AUV_{LOCAL}\ k$. For the second step of offloading computing, the time delay that it takes the AUV_{MEC} to process the downloaded task is represented by:

$$D_{AUV_M}(t) = \frac{\alpha_{k,j}\ L_{k,j}(t)\ C_{k,j}(t)}{f^{m}_{kj}\ F} \tag{4.39}$$

where F represents the allocated computational resources (computational capacity of the $\mathrm{AUV_{MEC}}$), and $f^{m}_{kj} \in \{0,1\}$ is the specific portion of F allocated to $AUV_{LOCAL}\ k$. Similarly, the power consumption required by the AUV_{MEC} to execute the remaining task offloaded can be evaluated as:

$$E_{k,j}^{M}(t) = \mu(f_{kj}^{m}F)^{\lambda}D_{AUV_M}(t) \tag{4.40}$$

where μ and λ are constants explained above.

For the last step of offloading computing, the time delay of receiving the processed result can be expressed as follows:

$$D_{rx}(t) = \frac{L_m(j)}{r_{rx}} \tag{4.41}$$

where $L_m(j)$ indicates the size of the computed result and r_{rx} indicates the download data rate of AUV_{LOCAL}. But based on [267], the download duration of the results from the AUV_{MEC} is usually ignored, because the result size is considerably smaller than the input data size of $L_{k,j}$. Consequently, the delay and propagation delay of this step are neglected in the rest of this work.

6) Task Completion Time and Energy Consumption

The total time to complete a task $M_{k,j}$ locally is expressed as

$$D_{k,j}^{L}(t) = D_{tx}^{ch,auvloc}(t) + D_{prop}^{ch,auvloc}(t) + D_{AUV_L}(t) \tag{4.42}$$

The total time to complete a task $M_{k,j}$ using computational offloading is expressed as

$$D_{k,j}^{O}(t) = D_{tx}^{ch,auvloc}(t) + D_{prop}^{ch,auvloc}(t) + D_{tx}^{auvloc,auvmec}(t) + D_{prop}^{auvloc,auvmec}(t) + D_{AUV_M}(t) \tag{4.43}$$

To sum up, the time to complete the task $D_{k,j}$ is given by

$$D_{k,j}(t) = \begin{array}{lll} D_{k,j}^{L}(t), & \alpha_{k,j} = 0; & local\ execution \\ D_{k,j}^{O}(t), & \alpha_{k,j} = 1; & offloading \\ max\left(D_{k,j}^{L}(t), D_{k,j}^{O}(t)\right), & 0 < \alpha_{k,j} < 1; & Partial\ Offloading \end{array} \tag{4.44}$$

And the overall energy consumption $E_{k,j}(t)$ is

$$E_{k,j}(t) = \begin{cases} E_{k,j}^{L}(t), & \alpha_{k,j} = 0; \quad local\ execution \\ E_{k,j}^{O}(t), & \alpha_{k,j} = 1; \quad offloading \\ E_{k,j}^{L}(t) + E_{k,j}^{O}(t) + E_{k,j}^{M}(t), & 0 < \alpha_{k,j} < 1; \quad Partial\ Offloading \end{cases} \tag{4.45}$$

E.Problem formulation

In this work, we consider jointly optimizing the trajectory of $AUVs_{LOCAL}$ and the AUV_{MEC} , the task offloading strategy of the $AUVs_{LOCAL}$ and computing resource allocation on the AUV_{MEC} to minimize the total delay to complete the tasks and energy consumption. The trajectory of the $AUVs_{LOCAL}$ and the AUV_{MEC} is defined as $U = \{\theta_{mec}^{h}, \theta_{k}^{h}, d_{mec,k}^{h}(t), d_{k,j}^{h}(t), \forall\, k \in K, \forall\, j \in R_k, \forall\, t \in T\}$, the offloading strategy is defined as $A = \{\alpha_{k,j}(t), \forall\, k \in K, \forall\, j \in R_k, \forall\, t \in T\}$, and the computing resource allocation vector is defined as $F = \{f_{kj}^{m}(t), \forall\, k \in K, \forall\, j \in R_k, \forall\, t \in T\}$. Therefore, the problem for joint optimization of the trajectory *U*, the offloading strategy of tasks *A*, and the resource allocation F can be formulated as

$$\min_{U,A,F} \sum_{t\in T} \left[(1 \quad \omega) \sum_{k\in K} \sum_{j\in R} D_{k,j}(t) \; + \omega \sum_{k\in K} \sum_{j\in R} E_{k,j}(t) \right] \tag{4.46}$$

s. t.

$$0 \leq \theta_{mec}^{h}(t) \leq 2\pi, \forall\, k \in K, t \in T \tag{4.46a}$$

$$0 \leq \theta_{k}^{h}(t) \leq 2\pi, \forall\, k \in K, t \in T \tag{4.46b}$$

$$0 \leq X_{mec}(t) \leq X_{mec}^{max}, \forall\, t \in T \tag{4.46c}$$

$$0 \leq Y_{mec}(t) \leq Y_{mec}^{max}, \forall\, t \in T \tag{4.46d}$$

$$x_{k}^{min} \leq x_{k}(t) \leq x_{k}^{max}, \forall\, k \in K, t \in T \tag{4.46e}$$

$$y_{k}^{min} \leq y_{k}(t) \leq y_{k}^{max}, \forall\, k \in K, t \in T \tag{4.46f}$$

$$f_{kj}^{m} = 0\,;\; if\ \alpha_{k,j}(t) = 0 \tag{4.46g}$$

$$\textstyle\sum_{1}^{k} f_{kj}^{m} \leq 1\,;\;\; \alpha_{k,j}(t) \neq 0 \tag{4.46h}$$

where ω represents the weights of the delay and the energy consumption of the task $L_{k,j}(t)$. The weights satisfy $0 \leq \omega \leq 1$.

The constraints (4.46a, 4.46b) guarantee the horizontal direction of motion for the AUV_{MEC} and the AUV_{LOCAL}, respectively. The restrictions (4.46c-4.46f) indicate the maximum area of coverage for the $AUVs$. The constraint (4.46g) denotes that if the task is executed locally, no computational resources will be allocated on the MEC server. The constraint (4.46h) guarantees that the resource allocation assigned by the AUV_{MEC} does not exceed the maximum available.

An efficient trajectory optimization, task offloading, and resource allocation model is formulated as a nonlinear problem. Its objective is to reduce the time and energy cost of $AUVs_{LOCAL}$. This model presents a challenging nonlinear problem due to its non-convex nature and integer programming constraints, rendering it NP-hard and impractical for exact solution derivation, chiefly due to its high-dimensional complexity. Consequently, instead of using traditional optimization approaches, reinforcement learning-based methods are used to obtain the near-optimal solutions for the variable parameters efficiently. For this purpose, we employ a deep deterministic policy gradient algorithm to solve it.

4.2.2 DDPG Algorithm Methodology Problem Solution

In this section, we propose a DDPG algorithm for MEC-based underwater networks that allows the joint optimization of the AUV trajectory, the task offloading strategy, and the allocation of computational resources in a continuous action space; to minimize both the total delay for the computation of tasks as well as the energy consumption in the system.

There are three key elements in the reinforcement learning method, namely, state, action, and reward. These are detailed below:

1) *State* $s(t)$: $s(t) = \{[X(t), Y(t), Z], [x_k(t), y_k(t), z_k], L_{k,j}\ \forall k \in K\}$; $s(t)$ is the set of coordinates of all $AUVs$.

2) *Action* $c(t)$: $c(t)$ is the set of the actions of all $AUVs$, it includes the velocity $v_{mec}(t)$ for AUV_{MEC}, velocity $v_{local}(t)$ for $AUVs_{LOCAL}$, the offloading strategy $\alpha_{k,j}(t)$, and the computing resource allocation vector f_{kj}^{m}. Hence, the action set can be defined as $c(t) = [v_{mec}(t), v_{local(1)}(t), \dots, v_{local(k)}(t), \alpha_{1j}(t), \dots, \alpha_{kj}(t), f_{1j}^{m}(t), \dots, f_{kj}^{m}(t)]$.

3) *Reward* $z(t)$: $z(t)$ is defined as the overall minimum delay for the entire process in each time slot is defined as:

$$\min_{U,A,F} \sum_{t \in T} \left[(1 \quad \omega) \sum_{k \in K} \sum_{j \in R} D_{k,j}(t) \ + \omega \sum_{k \in K} \sum_{j \in R} E_{k,j}(t) \right] \tag{4.47}$$

4.2.3 Experiments and Results

In this section, simulations are conducted to verify the effectiveness and evaluate the design of the proposed algorithm. First, we describe the environment and simulation parameters employed during the experiments. Afterward, we present a discussion of the obtained results, comparing the proposed algorithm with other baseline schemes.

All algorithms are evaluated with simulations implemented on several Jupyter Notebooks in version 6.0.3 installed with the Anaconda software suite, developed in Python 3.7. The experiments were performed on a Lenovo computer with Intel (R) Xeon (R) 2.9 GHz processor and 72 GB RAM; furthermore, NumPy, Matplotlib, and TensorFlow libraries are used to develop RL algorithms.

4.2.3.1 Simulation Setting

In the proposed scheme, we consider a total coverage area of $100x100\ m^2$ (LxW) on the seabed. It is divided into four equal quadrants of $50x50\ m^2$ each. Several IoUT devices or sensor nodes are randomly distributed over the total area, strategically clustered into groups and each group is led by a CH. Each CH in each quadrant will be responsible for collecting information from the other sensor nodes. There are 4 $AUVs_{LOCAL}$ (one for each quarter) that collect the information/tasks from the CHs for processing. Tasks collected by $AUVs_{LOCAL}$ can be processed fully locally, fully offloaded to the AUV_{MEC} server or partially in both $AUVs_{LOCAL}$ and AUV_{MEC}. In addition, an AUV_{MEC} that is responsible for serving all $AUVs_{LOCAL}$. We have considered a fixed depth of 20 m, 60 m, and 95 m for the AUV_{MEC}, $AUVs_{LOCAL}$, and CHs, respectively. Furthermore, each training epoch is divided into 60 time slots. In each time slot, a task with computation requirements is generated at each

CH. We also assume that the $AUVs_{LOCAL}$ and AUV_{MEC} have a maximum speed of 2 m/s and 5 m/s respectively. The transmission bandwidth is set to $B = 2\ KHz$. The maximum CPU frequency of the MEC server AUV_{MEC} and $AUVs_{LOCAL}$ is $F = 1.5x10^9\ Hz$, and $f_{AUV_L} = 6x10^7 Hz$, respectively. The transmission power is denoted by $P_T = 1x10^{-3}\ W$. Moreover, it should be noted that the computational download data size R_n is set in Kbytes and the number of cycles is set in Megacycles/second. Other detailed simulation parameters are summarized in Table 4.2.

To evaluate the performance of the proposed algorithm and for comparison purposes, we describe the following benchmark approaches below:

- *Offloading of all tasks to the* AUV$_{\mathrm{MEC}}$ *(Offloading):* The AUV_{MEC} provides computing resources to the $AUVs_{LOCAL}$ at a designated location. Each of the $AUVs_{LOCAL}$ offloads all its computing tasks to be processed remotely.
- *Execution of all tasks locally (Locally):* All computer tasks of the $AUVs_{LOCAL}$ are executed locally without offloading to the AUV_{MEC}.
- *Deep Deterministic Policy Gradient (DDPG):* We set the parameters for the proposed DDPG algorithm to obtain optimal system performance values.
- *Actor-Critic (AC):* We implemented the continuous action space-based RL algorithm for the computational offloading problem to compare and evaluate the performance of the proposed DDPG algorithm.

Table 4.2. Simulation Parameters.

Parameter	Default value	Parameter	Default value
$(L\ x\ W)_{MEC}$	$100x100\ m^2$	$C\ \ anel\ Parameters(s, w, k)$	$(0.5, 0, 1)$
$(L\ x\ W)_{LOCAL}$	$50x50\ m^2$	$Number\ AUV_{MEC}$	1
$Dept\ \ of\ AUV_{MEC}$	$20\ m$	$Number\ AUV_{LOCAL}$	4
$Dept\ \ of\ AUVs_{LOCAL}$	$60\ m$	$Number\ CHs$	4
$Dept\ \ of\ CHs$	$\geq 90\ m$	$Number\ of\ time\ slots\ (t)$	60
f_{AUV_L}	$6x10^7 Hz$	f	$20\ KHz$
F	$1.5x10^9\ Hz$	B	$2\ KHz$
v_{mec}^{max}	$5\ m/s$	η	0.2
v_{local}^{max}	$2\ m/s$	P_T	$1x10^{-3}\ W$
$L_{k,j}$	$[10^5, 3x10^5]\ bit$	ω	0.7
$C_{k,j}$	$1200\ cycles/bit$		

4.2.3.2 Simulation Results

We trained the deep neural networks of the proposed models over a total of 2000 iterations/episode. The configuration parameters of the DDPG algorithm are as follows: two hidden layers with 400 and 300 fully connected neurons for both the actor and the critic network, soft update coefficient $\tau = 0.001$, reward discount factor $\gamma = 0.9$, optimizer = Adam Optimizer and learning rate $lr = 0.001$ and 0.002 for actor and critical network, respectively, variance $\sigma = 0.05$. The parameters of the Actor-Critic algorithm are fundamentally the same as DDPG, except that each hidden layer is configured with 500 and 400 neurons, respectively.

Figure 4.4 shows the performance comparison between AC and DDPG reinforcement learning algorithms. The abscissa denotes the number of iterations of the main loop. The ordinate is the episodic reward, which is the total reward obtained by the system in an episode. The final convergence of the DDPG algorithm shows that it performs better based on the cumulative reward compared to the AC joint optimization algorithm. Although both algorithms perform favorably, the DDPG algorithm reaches convergence after around 250 iterations and remains more stable. This means that the total system delay and energy consumption decrease as the number of iterations increases. Compared with its counterpart, this shows the efficiency that our proposed RL algorithm.

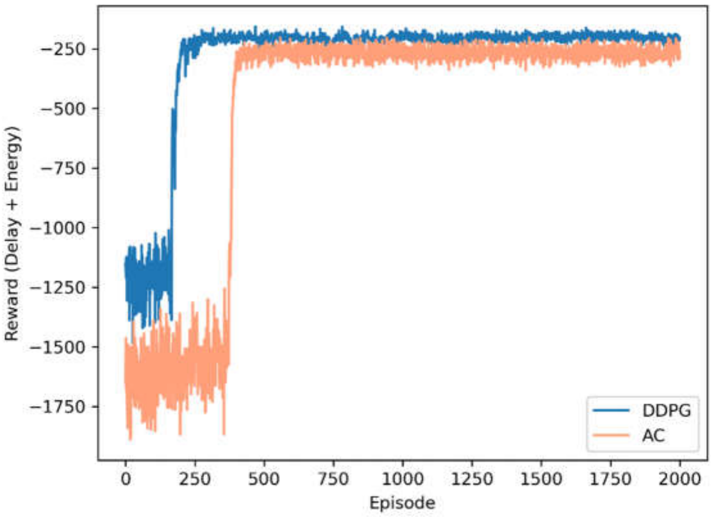


Figure 4.4. The total accumulated reward for the episode.

In Figure 4.5 and Figure 4.6, we compare the influence of hyperparameters for DDPG. Figure 4.5 shows the convergence performance of the proposed algorithm (DDPG) with different batch sizes. The figure shows an enlarged picture of the convergence performance for each batch size. We observe that the DDPG algorithm has similar convergence performance for different batch sizes and only becomes more stable during the training process, for batch size 64.

When the batch size is 512 it converges at the same time; however, it does not remain stable and gets worse as the number of epochs increases. When the batch size is 256, although it may appear to converge optimally, we can see that throughout training it does not remain stable. When the batch size is 128, the algorithm converges nicely but does not remain stable as the training progresses. When the batch size is 64, an optimal convergence is obtained around 250 epochs, and it remains even more stable than the previous ones as training progresses.

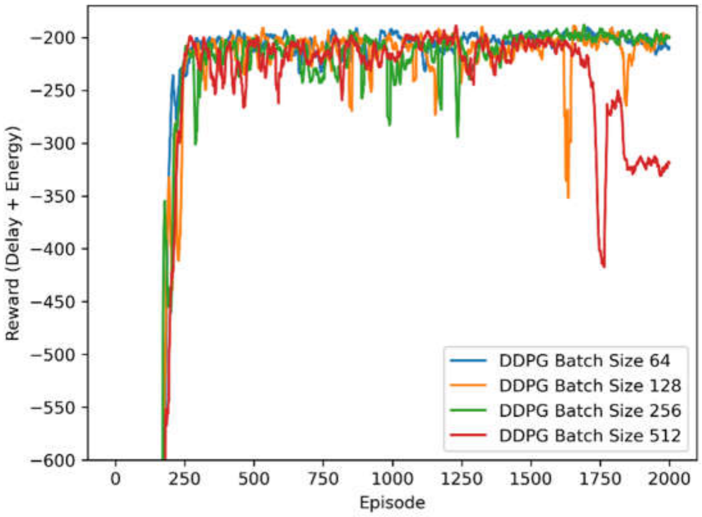


Figure 4.5. Convergence performance of the DDPG algorithm with different batch sizes.

Figure 4.6 shows the convergence performance of the proposed algorithm with different learning rate values for both the actor-network (*lr-actor*) and the critic network (*lr-critic*). We assume that the learning rate of the actor-network and the critic network are different. It is clear from the figure that when the value of the learning rate is higher (*lr-actor* = 0.01 and *lr-critic*=0.02), the algorithm does not converge. Similarly, when the learning rate value is very small (*lr-actor* = 0.0001 and *lr-critic* = 0.0002), the algorithm seems to reach convergence, but during training, it is getting worse. Therefore, with the learning rate values *lr-actor* = 0.001 and *lr-critic* = 0.002 the convergence performance of the algorithm is optimal; it converges faster and is more stable during the whole training process.

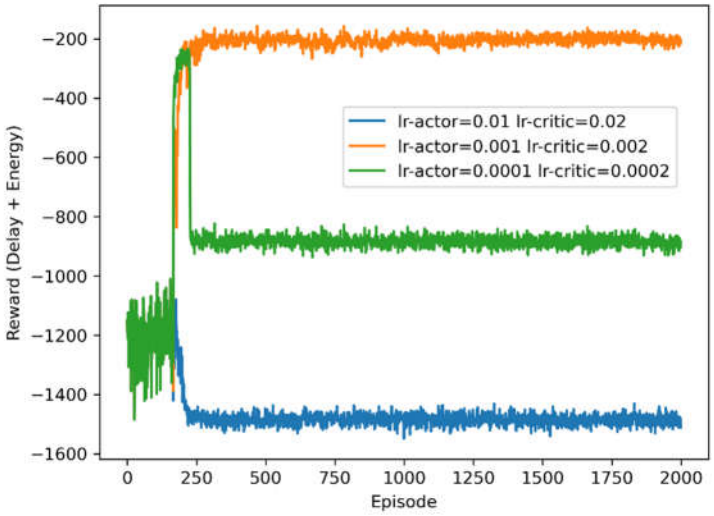


Figure 4.6. Convergence performance of the DDPG algorithm with different values of learning rates.

After the training and convergence of the algorithms, in our proposal, we study the impact of the data size on the total delay and energy consumption by comparing the performance of these algorithms with different data sizes. Figure 4.7 shows the total cumulative reward for DDPG, AC, Locally, and Offloading strategies. From the figure, we can notice that the total cumulative reward of the proposed algorithms can reach a near-optimal result, which means the reasonable application of the computational offloading policy can substantially reduce the total overhead of the $AUVs$ when tasks are partially executed. Furthermore, the AC algorithm's performance is considerably good compared to the DDPG algorithm, because the two algorithms explore a continuous action space and take a precise action that ultimately leads to the acquisition of the optimal offloading strategy and significantly reduces the latency and energy consumption. If $\mathrm{AUVs}_{\mathrm{LOCAL}}$ execute the tasks locally without offloading (Locally strategy), they cannot use the computing resources of the entire system and they consume more energy and accelerate their battery exhaustion. Hence, as the data size increases, the delay and power consumption of each $\mathrm{AUV}_{\mathrm{LOCAL}}$ also increases, leading to a degradation in system performance. Both the delay and energy consumption obtained by the DDPG algorithm are significantly lower than AC and Locally as the data size increases, which indicates the better performance of our proposed algorithm. In general, the total delay and energy of all schemes grow as the data size increases. Therefore, the optimal offloading strategy used by each $\mathrm{AUVs}_{\mathrm{LOCAL}}$ is very important to make the best use of $AUVs_{LOCAL}$ and AUV_{MEC} computing resources.

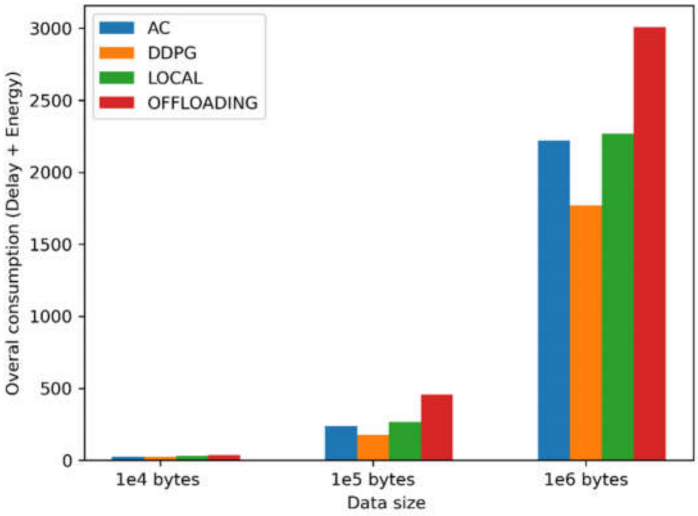


Figure 4.7. Comparison of total cumulative reward benefit and task data size (AUV's workload).

Figure 4.8 and Figure 4.9 show the optimal trajectory of the $AUVs$ for data collection from the CHs as well as for offloading from the $AUVs_{LOCAL}$ to the AUV_{MEC}. For each figure, we randomly selected the location of the cluster heads and chose a different value for the velocity vector and the ocean current direction (Uc). In the figures, the gray dots represent the cluster heads located on the seafloor in each of the quadrants. The lines with several colors represent the trajectory of each of the $AUVs_{LOCAL}$ and the black line represents the AUV_{MEC} trajectory. The trajectories of the $AUVs_{LOCAL}$ and the AUV_{MEC} that minimize the delay time and energy consumption have been selected. It is worth noting that, regardless of the location points of the cluster heads, the algorithm will be in charge of guiding the $AUVs_{LOCAL}$ to the target point (CH), starting from the closest one and covering each of the cluster heads only one at a time. Once the cycle is finished, the last point becomes the new starting point by searching again for the closest one to follow its new trajectory (thus getting randomness in serving each of the cluster heads). Similarly, the algorithm is in charge of guiding the AUV_{MEC} from its starting point to the strategic coverage points to serve the $AUVs_{LOCAL}$ located in each quadrant.

In Figure 4.8, the speed of the ocean current $Uc = 0.5$ m/s and an angle of $ai1 =$ 45° has been considered. Figure 4.8 shows in red the angle θ^h and the velocity vector V that each AUV must have to reach the desired objective. On the other hand, in Figure 4.9, a speed of the ocean current Uc = 0.8 m/s and an angle of $ai1$ = 45° have been considered instead. We can observe that the AUVs are able to choose their trajectories optimally and reach their destinations irrespective of the starting cluster head locations, the ocean current speeds, and the $ai1$ angles.

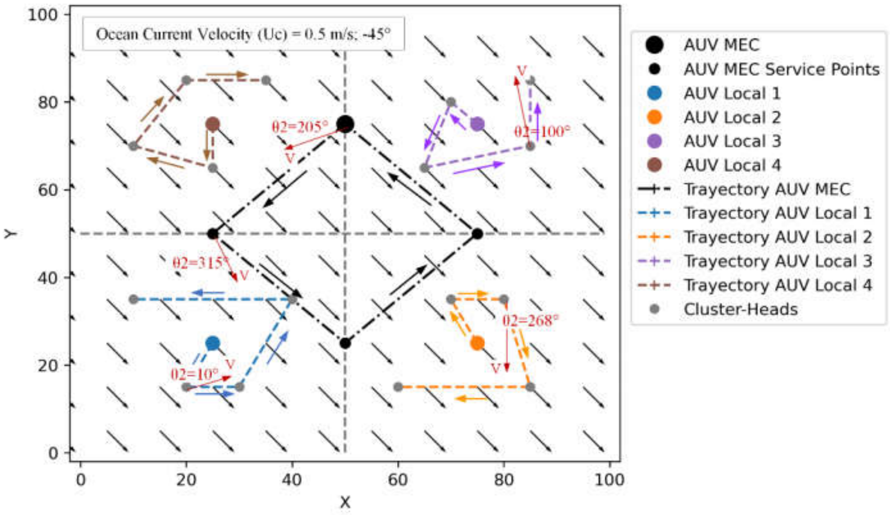


Figure 4.8. AUVs trajectory planning with UC=0.5 m/s; -45°.

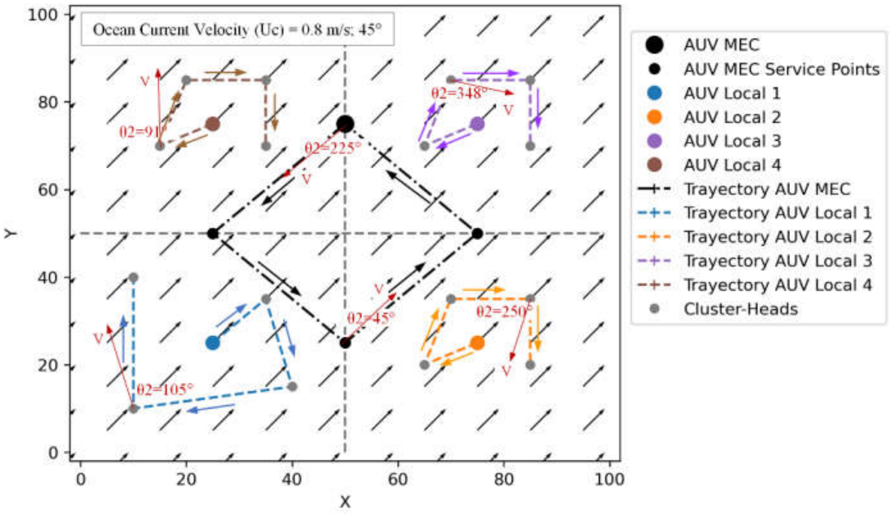


Figure 4.9. AUVs trajectory planning with UC=0.8 m/s; 45°.

4.2.4 Conclusions

In this work, a novel AUV-enabled MEC system has been introduced. A joint optimization algorithm to solve the offloading strategy, resource allocation, and trajectory selection of both $AUVs_{LOCAL}$ and AUV_{MEC} has been proposed. The deep reinforcement learning-based approach DDPG has been presented to optimize AUV trajectories, task offloading, and resource allocation for improved underwater communication in terms of energy efficiency and delay minimization. We have described the training process of DDPG and AC algorithms. We have compared DDPG with other strategies (Locally, Offloading, and AC) Simulation results show that DDPG and AC converge well, but the DDPG algorithm performs considerably better- in terms of cumulative reward and stability. The convergence performance of the DDPG algorithm is studied with different batch sizes and learning rates. Batch size 64 is found to be optimal for convergence and stability. As data size increases, the DDPG algorithm demonstrates significantly lower delay and energy consumption compared to AC and the other strategies. In AUV-enabled MEC systems, some challenges remain. As future work, we plan to investigate the MEC systems assisted by multiple AUVs functioning as a swarm of server mobile edge computing devices. This approach will enable us to achieve extended coverage and explore interference and offload selection among them.

CHAPTER 5

5 CONCLUSIONS AND FUTURE WORK

5.1 Conclusions

In this research work, we have tried to boost the implementation of IoT-based applications and the optimization (improvement) of underwater communications, through artificial intelligence, edge computing, and 5G networks and beyond, contributing to the technological development in smart cities and ocean/sea. To achieve this purpose two different specific objectives have been addressed: 1) Improving the efficiency of applications in smart cities and 2) Improving the efficiency of underwater communications in smart coastal cities both through the use of artificial intelligence, edge computing, and 5G and beyond.

To achieve these objectives, the existing literature on 5G networks and beyond (architecture and services), smart cities (enabling technologies), and artificial intelligence (applications and use cases) has been analyzed. A compilation of technical documentation was carried out to obtain an updated view of the various technologies that facilitate the development of applications based on 5G technology. This initiative is aimed at generating new and innovative alternatives in critical areas such as tourism, security, improved underwater communications, and marine exploration, to drive development that effectively responds to the needs of citizens in smart urban environments and the vast oceanic territory.

As a result of this study, the first contribution has emerged. It consists of the analysis, design, and implementation of a tourist attraction recommendation system based on a deep learning algorithm for smart cities, aiming to reduce the time that it may take a user to search for possible places to visit and to improve how recommendations of tourist attractions are performed in a given city. Simulations show that our proposed multi-label deep learning classifier outperforms other models (decision tree, extra tree, k-next neighbor, and random forest) and can successfully recommend tourist attractions for both cases (a) searching and planning activities before traveling and (b) searching activities within the smart city providing real-time recommendations using IoT contextual information (location and weather forecast) once the tourist is searching for activities within the smart city.

From this foundation and with the desire to continue generating innovative strategies and solutions, the second contribution arises driven by surveillance and security, which consists of a distraction detection system for the prevention of drowning in aquatic places, developed in a 5G and beyond network environment. To do so, we proposed an approach of surveillance cameras that capture images of people in charge of minors in swimming pools or beaches; and through an ML algorithm (convolutional neural networks) classify the type of distraction that a person in charge of a minor may incur. The simulations show that our proposal successfully performs the classification of distractions with very high accuracy and through that the decision-making to alert the person who is distracted and focus on the care of the child.

The resulting results not only have demonstrated effectiveness in the specific areas addressed but have also demonstrated the ability to integrate emerging technologies. The combination of technologies such as 6G, artificial intelligence, and edge computing offers ample scope for addressing challenges in both urban and maritime environments. The contributions presented not only represent significant advances in their respective areas but also lay the groundwork for future research and development in smart city construction and optimization of underwater communications, thus reinforcing the transformative potential of artificial intelligence, edge computing, and advanced wireless networks in these domains.

From the research work carried out, the following particular conclusions can be drawn:

- Our study highlights that the choice of a deep neural network topology is key to the performance of an IoT and deep learning-based tourist attraction recommendation system. Our thesis experiments have demonstrated that deeper networks improve the efficiency of a deep neural network algorithm, and the grid search methodology can identify an optimal DNN topology.
- Our study also demonstrates the effectiveness of deep learning as a key tool for advanced visual data processing. The ability of deep learning algorithms to discern complex patterns in real-time images proves to be an essential component for the accurate identification of risky situations, such as a distracted caregiver not noticing a child in danger in a swimming pool. This approach, supported by a comprehensive and diverse data set, proves to be a promising strategy for minimizing false positives and ensuring reliable detection. Furthermore, the integration of 5G and beyond technology emerges as a key enabler for the operational efficiency of the proposed prevention systems. The data transmission capability, coupled with lower latency, provides a robust foundation for the effective implementation of preventive strategies. Instantaneous communication between devices and the ability to make dynamic adjustments based on real-time information are crucial attributes that strengthen the responsiveness and adaptability of the designed infrastructure.
- The application of reinforcement learning algorithms is essential to improve QoS parameters in underwater networks. Resource management in multi-tier subsea environments ("local AUVs" and an "MEC AUV") is addressed comprehensively. The implementation of advanced resource management techniques contributes to a more efficient utilization of computational capacity, maximizing the performance of distributed

underwater systems. The proposed approach succeeds in mitigating the computational complexity associated with decision-making in dynamic and changing underwater environments. Proactive consideration of environmental information enables effective simplification of decision-making processes, improving system robustness.

5.2 Future Work

The results obtained in this study not only validate the effectiveness in the specific areas addressed but also highlight the ability to merge continuously evolving technologies. The future of applications based on reinforcement learning, 6G and the development of smart cities is shaping up towards a horizon of unprecedented technological innovation and significant progress in the interconnection of intelligent systems.

However, certain aspects remain to be studied, which either maintain or open up new lines of research in the field of emerging technologies. Next, we present certain important areas for future work that are derived from the research conducted:

Personalization of Tourism Recommendations in Smart Cities.

- Due to advances in ultra-fast connectivity (5G and 6G), there is a need to research and develop more immersive and personalized tourism experiences. By implementing Mixed Reality (AR and VR) technologies. This approach involves the creation of applications that merge previous advances in tourism recommendations with AR to provide real-time contextual information about destinations and the use of VR experiences that could take tourism recommendations to a whole new level, allowing users realistic and exciting visits.
- Explore the use of techniques and analyze patterns associated with emotions present in stored data (emotion recognition) through analysis of images, data, text, or natural language processing to tailor recommendations according to the mood of tourists, providing more personalized experiences.
- Collaborate with travel platforms and social networks to enrich the information available and provide more accurate recommendations based on experiences shared by other travelers.

Development of Intelligent Systems for Drowning Prevention:

- First, research on transfer learning techniques is suggested to adapt the Deep Learning model to different pool environments, considering the analysis of new variables such as size, shape, and lighting conditions.
- Another interesting approach is the incorporation of multi-modal sensing, introducing additional sensors, underwater sound detection systems, or more advanced vision technologies to improve the detection of risky situations, such as the ability to identify

multiple people in the water simultaneously or assess water quality to predict dangerous conditions through advanced sensors that monitor parameters such as turbidity, chemical concentration or temperature in a pool or aquatic area.

- Exploring approaches to address privacy concerns when using surveillance technologies, such as anonymizing data and incorporating ethical measures into system design.
- Finally, it is proposed to consider expanding research into broader applications, such as aquatic safety in natural environments, beaches, or other aquatic locations, by deploying drones to enable image collection in outdoor areas, leveraging the combination of Deep Learning and advanced connectivity technologies to address broader concerns in the prevention of child drowning.

Optimization of Underwater Communication Systems:

- Extend the research to a wider variety of underwater operational scenarios, considering different depths, water temperatures, and geographical conditions. This will allow evaluation of the robustness and adaptability of reinforcement learning models and the effectiveness of moving edge computing in diverse underwater contexts.
- Encourage interdisciplinary collaboration with experts in oceanography, marine biology, and environmental sciences to fully exploit the capabilities of the developed technology in practical applications, such as underwater environmental monitoring and scientific research.
- Consider the implementation of recurrent neural networks (RNN) to address the sequential nature of underwater data, enabling more accurate modeling of temporal patterns. The application of RNNs could improve the predictive and adaptive capabilities of reinforcement learning models in changing underwater contexts.
- Investigate the integration of blockchain technologies to improve the security and integrity of subsea transmissions. The application of smart contracts and distributed logs could provide an additional layer of security and traceability in subsea communications.
- Develop adaptive algorithms for dynamic spectrum management in underwater environments. Real-time optimization of frequency allocation using adaptive algorithms will help improve bandwidth efficiency and reduce interference by adapting to changing subsea channel conditions.

BIBLIOGRAPHY

[1] N. United Nations, Department of Economic and Social Affairs, “World population projected to reach 9.8 billion in 2050, and 11.2 billion in 2100,” Un.Org. Accessed: Apr. 20, 2023. [Online]. Available: https://www.un.org/development/desa/en/news/population/world-population-prospects-2017.html

[2] P. D. United Nations, Department of Economic and Social Affairs, *World Urbanization Prospects: The 2018 Revision*, vol. 12. New York: United Nations.: (ST/ESA/SER.A/420), 2019.

[3] A. Zanella, N. Bui, A. Castellani, L. Vangelista, and M. Zorzi, “Internet of things for smart cities,” *IEEE Internet Things J*, vol. 1, no. 1, pp. 22–32, 2014, doi: 10.1109/JIOT.2014.2306328.

[4] ITU International Telecommunication Union Internet Reports, “The Internet of Things - Executive Summary.” Accessed: Jun. 07, 2021. [Online]. Available: http://www.itu.int/publ/S-POL-IR.IT-2005/eS

[5] M. E. E. Alahi *et al.*, “Integration of IoT-Enabled Technologies and Artificial Intelligence (AI) for Smart City Scenario: Recent Advancements and Future Trends,” *Sensors*, vol. 23, no. 11, 2023, doi: 10.3390/s23115206.

[6] B. Foubert and N. Mitton, “Long-range wireless radio technologies: A survey,” *Future Internet*, vol. 12, no. 1, 2020, doi: 10.3390/fi12010013.

[7] S. Yrjölä and A. Jette, “Spectrum for the Industrial Internet of Things - Industry Needs, Barriers and Recommended New Models,” no. November, p. 7, 2018.

[8] A. S. Cacciapuoti, K. Sankhe, M. Caleffi, and K. R. Chowdhury, “Beyond 5G: THz-Based Medium Access Protocol for Mobile Heterogeneous Networks,” *IEEE Communications Magazine*, vol. 56, no. 6, pp. 110–115, 2018, doi: 10.1109/MCOM.2018.1700924.

[9] K. David and H. Berndt, “6G Vision and requirements: Is there any need for beyond 5g?,” *IEEE Vehicular Technology Magazine*, vol. 13, no. 3, pp. 72–80, 2018, doi: 10.1109/MVT.2018.2848498.

[10] M. Mashaly, “Connecting the twins: A review on digital twin technology & its networking requirements,” *Procedia Comput Sci*, vol. 184, pp. 299–305, 2021, doi: 10.1016/j.procs.2021.03.039.

[11] J. Wu, Y. Yang, X. U. N. Cheng, H. Zuo, and Z. Cheng, "The Development of Digital Twin Technology Review," *Proceedings - 2020 Chinese Automation Congress, CAC 2020*, pp. 4901–4906, 2020, doi: 10.1109/CAC51589.2020.9327756.

[12] B. Dash and P. Sharma, "Role of Artificial Intelligence in Smart Cities for Information Gathering and Dissemination (A Review)," *Academic Journal of Research and Scientific Publishing*, vol. 4, no. 39, pp. 58–75, 2022, doi: 10.52132/ajrsp.e.2022.39.4.

[13] J. S. Alagha, M. Seyam, M. A. Md Said, and Y. Mogheir, "Integrating an artificial intelligence approach with k-means clustering to model groundwater salinity: the case ofGaza coastal aquifer (Palestine)," *Hydrogeol J*, vol. 25, no. 8, pp. 2347–2361, 2017, doi: 10.1007/s10040-017-1658-1.

[14] R. Amuthakkannan, K. Vijayalakshmi, S. Al Araimi, and M. Ali Saud Al Tobi, "A Review to do Fishermen Boat Automation with Artificial Intelligence for Sustainable Fishing Experience Ensuring Safety, Security, Navigation and Sharing Information for Omani Fishermen," *J Mar Sci Eng*, vol. 11, no. 3, 2023, doi: 10.3390/jmse11030630.

[15] J. M. Zheng, K. W. Chan, and I. Gibson, "Virtual reality," *IEEE Potentials*, vol. 17, no. 2, pp. 20–23, 1998, doi: 10.1109/45.666641.

[16] Y. Chen, Q. Wang, H. Chen, X. Song, H. Tang, and M. Tian, "An overview of augmented reality technology," *J Phys Conf Ser*, vol. 1237, no. 2, 2019, doi: 10.1088/1742-6596/1237/2/022082.

[17] A. Ali, Y. Ming, S. Chakraborty, and S. Iram, "A comprehensive survey on real-time applications of WSN," *Future Internet*, vol. 9, no. 4, 2017, doi: 10.3390/fi9040077.

[18] L. U. Khan, I. Yaqoob, N. H. Tran, S. M. A. Kazmi, T. N. Dang, and C. S. Hong, "Edge-Computing-Enabled Smart Cities: A Comprehensive Survey," *IEEE Internet Things J*, vol. 7, no. 10, pp. 10200–10232, 2020, doi: 10.1109/JIOT.2020.2987070.

[19] N. Hassan, S. Gillani, E. Ahmed, I. Yaqoob, and M. Imran, "The Role of Edge Computing in Internet of Things," *IEEE Communications Magazine*, vol. 56, no. 11, pp. 110–115, 2018, doi: 10.1109/MCOM.2018.1700906.

[20] Y. Liu, C. Yang, L. Jiang, S. Xie, and Y. Zhang, "Intelligent Edge Computing for IoT-Based Energy Management in Smart Cities," *IEEE Netw*, vol. 33, no. 2, pp. 111–117, 2019, doi: 10.1109/MNET.2019.1800254.

[21] K. M. Awan, P. A. Shah, K. Iqbal, S. Gillani, W. Ahmad, and Y. Nam, "Underwater Wireless Sensor Networks: A Review of Recent Issues and Challenges," *Wirel Commun Mob Comput*, vol. 2019, 2019, doi: 10.1155/2019/6470359.

[22] K. Y. Islam, I. Ahmad, D. Habibi, and A. Waqar, “A survey on energy efficiency in underwater wireless communications,” *Journal of Network and Computer Applications*, vol. 198, no. May 2021, p. 103295, 2022, doi: 10.1016/j.jnca.2021.103295.

[23] J. Navarro-Ortiz, P. Romero-Diaz, S. Sendra, P. Ameigeiras, J. J. Ramos-Munoz, and J. M. Lopez-Soler, “A Survey on 5G Usage Scenarios and Traffic Models,” *IEEE Communications Surveys and Tutorials*, vol. 22, no. 2, pp. 905–929, 2020, doi: 10.1109/COMST.2020.2971781.

[24] A. F. M. S. Shah, “A Survey from 1G to 5G Including the Advent of 6G: Architectures, Multiple Access Techniques, and Emerging Technologies,” *2022 IEEE 12th Annual Computing and Communication Workshop and Conference, CCWC 2022*, pp. 1117–1123, 2022, doi: 10.1109/CCWC54503.2022.9720781.

[25] T. S. Rappaport, Y. Xing, G. R. MacCartney, A. F. Molisch, E. Mellios, and J. Zhang, “Overview of Millimeter Wave Communications for Fifth-Generation (5G) Wireless Networks-With a Focus on Propagation Models,” *IEEE Trans Antennas Propag*, vol. 65, no. 12, pp. 6213–6230, 2017, doi: 10.1109/TAP.2017.2734243.

[26] B. K. J. Al-Shammari, I. Hburi, H. R. Idan, and H. F. Khazaal, “An Overview of mmWave Communications for 5G,” *International Conference on Communication and Information Technology, ICICT 2021*, pp. 133–139, 2021, doi: 10.1109/ICICT52195.2021.9568459.

[27] R. Chataut and R. Akl, “Massive MIMO systems for 5G and beyond networks—overview, recent trends, challenges, and future research direction,” *Sensors (Switzerland)*, vol. 20, no. 10, pp. 1–35, 2020, doi: 10.3390/s20102753.

[28] M. Erel-Özçevik and B. Canberk, “Road to 5g reduced-latency: A software defined handover model for eMBB services,” *IEEE Trans Veh Technol*, vol. 68, no. 8, pp. 8133–8144, 2019, doi: 10.1109/TVT.2019.2925393.

[29] I. Parvez, A. Rahmati, I. Guvenc, A. I. Sarwat, and H. Dai, “A survey on low latency towards 5G: RAN, core network and caching solutions,” *IEEE Communications Surveys and Tutorials*, vol. 20, no. 4, pp. 3098–3130, Oct. 2018, doi: 10.1109/COMST.2018.2841349.

[30] M. Iwamura, “NGMN view on 5G architecture,” *IEEE Vehicular Technology Conference*, vol. 2015, pp. 1–5, 2015, doi: 10.1109/VTCSpring.2015.7145953.

[31] R. Abhishek, D. Tipper, and D. Medhi, “Network Virtualization and Survivability of 5G Networks,” *Journal of Network and Systems Management*, vol. 28, no. 4, pp. 923–952, 2020, doi: 10.1007/s10922-020-09541-0.

[32] J. Yao, Z. Han, M. Sohail, and L. Wang, “A robust security architecture for SDN-based 5G networks,” *Future Internet*, vol. 11, no. 4, pp. 1–14, 2019, doi: 10.3390/FI11040085.

[33] D. V. Bernardo and B. B. Chua, "Introduction and analysis of SDN and NFV security architecture (SN-SECA)," *Proceedings - International Conference on Advanced Information Networking and Applications, AINA*, vol. 2015-April, pp. 796–801, 2015, doi: 10.1109/AINA.2015.270.

[34] A. Dutta and E. Hammad, "5G Security Challenges and Opportunities: A System Approach," *2020 IEEE 3rd 5G World Forum, 5GWF 2020 - Conference Proceedings*, pp. 109–114, 2020, doi: 10.1109/5GWF49715.2020.9221122.

[35] M. Usama and M. Erol-Kantarci, "A survey on recent trends and open issues in energy efficiency of 5G," *Sensors (Switzerland)*, vol. 19, no. 14, 2019, doi: 10.3390/s19143126.

[36] T. O. Olwal, K. Djouani, and A. M. Kurien, "A Survey of Resource Management Toward 5G Radio Access Networks," *IEEE Communications Surveys and Tutorials*, vol. 18, no. 3, pp. 1656–1686, 2016, doi: 10.1109/COMST.2016.2550765.

[37] A. Shaikh and M. J. Kaur, "Comprehensive Survey of Massive MIMO for 5G Communications," *2019 Advances in Science and Engineering Technology International Conferences, ASET 2019*, pp. 1–5, 2019, doi: 10.1109/ICASET.2019.8714426.

[38] B. Bertenyi, R. Burbidge, G. Masini, S. Sirotkin, and Y. Gao, "NG Radio Access Network (NG-RAN)," *Journal of ICT Standardization*, vol. 6, no. 1, pp. 59–76, 2018, doi: 10.13052/jicts2245-800x.614.

[39] S. Rommer, P. Hedman, M. Olsson, L. Frid, S. Sultana, and C. Mulligan, *5G Core Networks: Powering Digitalization*. Elsevier, 2020. doi: 10.1016/C2018-0-01335-3.

[40] H. Kim, "5G core network security issues and attack classification from network protocol perspective," *Journal of Internet Services and Information Security*, vol. 10, no. 2, pp. 1–15, 2020, doi: 10.22667/JISIS.2020.05.31.001.

[41] A. Dogra, R. K. Jha, and S. Jain, "A Survey on beyond 5G Network with the Advent of 6G: Architecture and Emerging Technologies," *IEEE Access*, vol. 9, pp. 67512–67547, 2021, doi: 10.1109/ACCESS.2020.3031234.

[42] B. S. Khan, S. Jangsher, A. Ahmed, and A. Al-Dweik, "URLLC and eMBB in 5G Industrial IoT: A Survey," *IEEE Open Journal of the Communications Society*, vol. 3, no. May, pp. 1134–1163, 2022, doi: 10.1109/OJCOMS.2022.3189013.

[43] M. E. Haque, F. Tariq, M. R. A. Khandaker, K. K. Wong, and Y. Zhang, "A Survey of Scheduling in 5G URLLC and Outlook for Emerging 6G Systems," *IEEE Access*, vol. 11, no. March, pp. 34372–34396, 2023, doi: 10.1109/ACCESS.2023.3264592.

[44] N. U. Ginige, K. B. Shashika Manosha, N. Rajatheva, and M. Latva-Aho, "Admission Control in 5G Networks for the Coexistence of eMBB-URLLC Users," *IEEE Vehicular*

Technology Conference, vol. 2020-May, pp. 1–6, 2020, doi: 10.1109/VTC2020-Spring48590.2020.9129141.

[45] S. R. Pokhrel, J. DIng, J. Park, O. S. Park, and J. Choi, "Towards Enabling Critical mMTC: A Review of URLLC within mMTC," *IEEE Access*, vol. 8, pp. 131796–131813, 2020, doi: 10.1109/ACCESS.2020.3010271.

[46] T. A. Bruza Alves and T. Abrão, "Improving Random Access with NOMA in mMTC XL-MIMO," in *2023 IEEE 97th Vehicular Technology Conference (VTC2023-Spring)*, Florence, Italy: IEEE, Jun. 2023, pp. 1–5. doi: 10.1109/VTC2023-Spring57618.2023.10199336.

[47] S. Wijethilaka and M. Liyanage, "Survey on Network Slicing for Internet of Things Realization in 5G Networks," *IEEE Communications Surveys and Tutorials*, vol. 23, no. 2, pp. 957–994, 2021, doi: 10.1109/COMST.2021.3067807.

[48] N. Huin *et al.*, "Hard-isolation for Network Slicing," *INFOCOM 2019 - IEEE Conference on Computer Communications Workshops, INFOCOM WKSHPS 2019*, pp. 955–956, 2019, doi: 10.1109/INFCOMW.2019.8845282.

[49] NGMN Alliance, "5G White Paper 1," *Next Generation Mobile Networks (NGMN).*, pp. 1–125, 2015.

[50] Y. Liu, B. Clerckx, and P. Popovski, "Network Slicing for eMBB, URLLC, and mMTC: An Uplink Rate-Splitting Multiple Access Approach," *IEEE Trans Wirel Commun*, vol. PP, no. 8, p. 1, 2023, doi: 10.1109/TWC.2023.3295804.

[51] J. Mei, X. Wang, and K. Zheng, "Intelligent network slicing for V2X services toward 5G," *IEEE Netw*, vol. 33, no. 6, pp. 196–204, 2019, doi: 10.1109/MNET.001.1800528.

[52] W. Zhuang, Q. Ye, F. Lyu, N. Cheng, and J. Ren, "SDN/NFV-Empowered Future IoV with Enhanced Communication, Computing, and Caching," *Proceedings of the IEEE*, vol. 108, no. 2, pp. 274–291, 2020, doi: 10.1109/JPROC.2019.2951169.

[53] H. G. Abreha, S. Member, and H. Chougrani, "Topology-Aware Dynamic VNF Mapping and Scheduling in SDN / NFV-Enabled Satellite Edge Networks for Mission-Critical Applications Topology-Aware Dynamic VNF Mapping and Scheduling in SDN / NFV-Enabled Satellite Edge Networks for Mission-Critical Applicatio," 2023, doi: 10.36227/techrxiv.23703234.v1.

[54] F. Nawaz, J. Ibrahim, M. Junaid, S. Kousar, T. Parveen, and M. A. Ali, "A review of vision and challenges of 6G technology," *International Journal of Advanced Computer Science and Applications*, vol. 11, no. 2, pp. 643–649, 2020, doi: 10.14569/ijacsa.2020.0110281.

[55] 6G Flagship, *Key Drivers and Research challenges for Ubiquitous wireless Intelligence.*, no. September. 2019.

[56] S. Madakam, R. Ramaswamy, and S. Tripathi, “Internet of Things (IoT): A Literature Review,” *Journal of Computer and Communications*, vol. 03, no. 05, pp. 164–173, 2015, doi: 10.4236/jcc.2015.35021.

[57] Z. Ding, L. Dai, and H. V. Poor, “MIMO-NOMA Design for Small Packet Transmission in the Internet of Things,” *IEEE Access*, vol. 4, pp. 1393–1405, 2016, doi: 10.1109/ACCESS.2016.2551040.

[58] N. N. Misra, Y. Dixit, A. Al-Mallahi, M. S. Bhullar, R. Upadhyay, and A. Martynenko, “IoT, Big Data, and Artificial Intelligence in Agriculture and Food Industry,” *IEEE Internet Things J*, vol. 9, no. 9, pp. 6305–6324, 2022, doi: 10.1109/JIOT.2020.2998584.

[59] N. H. Motlagh, M. Mohammadrezaei, J. Hunt, and B. Zakeri, “Internet of things (IoT) and the energy sector,” *Energies (Basel)*, vol. 13, no. 2, pp. 1–27, 2020, doi: 10.3390/en13020494.

[60] R. Mahmoud, T. Yousuf, F. Aloul, and I. Zualkernan, “Internet of things (IoT) security: Current status, challenges and prospective measures,” *2015 10th International Conference for Internet Technology and Secured Transactions, ICITST 2015*, pp. 336–341, 2016, doi: 10.1109/ICITST.2015.7412116.

[61] L. Xiao, X. Wan, X. Lu, Y. Zhang, and D. Wu, “IoT Security Techniques Based on Machine Learning: How Do IoT Devices Use AI to Enhance Security?,” *IEEE Signal Process Mag*, vol. 35, no. 5, pp. 41–49, 2018, doi: 10.1109/MSP.2018.2825478.

[62] M. Jia, A. Komeily, Y. Wang, and R. S. Srinivasan, “Adopting Internet of Things for the development of smart buildings: A review of enabling technologies and applications,” *Autom Constr*, vol. 101, no. February, pp. 111–126, 2019, doi: 10.1016/j.autcon.2019.01.023.

[63] F. Meneghello, M. Calore, D. Zucchetto, M. Polese, and A. Zanella, “IoT: Internet of Threats? A Survey of Practical Security Vulnerabilities in Real IoT Devices,” *IEEE Internet Things J*, vol. 6, no. 5, pp. 8182–8201, 2019, doi: 10.1109/JIOT.2019.2935189.

[64] F. Shrouf, J. Ordieres, and G. Miragliotta, “Smart factories in Industry 4.0: A review of the concept and of energy management approached in production based on the Internet of Things paradigm,” *IEEE International Conference on Industrial Engineering and Engineering Management*, vol. 2015-Janua, pp. 697–701, 2014, doi: 10.1109/IEEM.2014.7058728.

[65] L. Garcia, J. M. Jiménez, M. Taha, and J. Lloret, “Wireless Technologies for IoT in Smart Cities,” *Network Protocols and Algorithms*, vol. 10, no. 1, p. 23, 2018, doi: 10.5296/npa.v10i1.12798.

[66] C. S. Lai *et al.*, "A Review of Technical Standards for Smart Cities," *Clean Technologies*, vol. 2, no. 3, pp. 290–310, 2020, doi: 10.3390/cleantechnol2030019.

[67] A. Simonofski, T. Vallé, E. Serral, and Y. Wautelet, "Investigating context factors in citizen participation strategies: A comparative analysis of Swedish and Belgian smart cities," *Int J Inf Manage*, vol. 56, no. February 2019, p. 102011, 2021, doi: 10.1016/j.ijinfomgt.2019.09.007.

[68] F. Al-Turjman, H. Zahmatkesh, and R. Shahroze, "An overview of security and privacy in smart cities' IoT communications," *Transactions on Emerging Telecommunications Technologies*, vol. 33, no. 3, pp. 1–19, 2022, doi: 10.1002/ett.3677.

[69] L. Penco, E. Ivaldi, and A. Ciacci, "Entrepreneurial ecosystem and well-being in European smart cities: a comparative perspective," *TQM Journal*, vol. 33, no. 7, pp. 318–350, 2021, doi: 10.1108/TQM-04-2021-0097.

[70] E. Ismagilova, L. Hughes, N. P. Rana, and Y. K. Dwivedi, "Security, Privacy and Risks Within Smart Cities: Literature Review and Development of a Smart City Interaction Framework," *Information Systems Frontiers*, vol. 24, no. 2, pp. 393–414, 2022, doi: 10.1007/s10796-020-10044-1.

[71] A. Puliafito, G. Tricomi, A. Zafeiropoulos, and S. Papavassiliou, "Smart cities of the future as cyber physical systems: Challenges and enabling technologies," *Sensors*, vol. 21, no. 10, pp. 1–25, 2021, doi: 10.3390/s21103349.

[72] A. K. Kar, V. Ilavarasan, M. P. Gupta, M. Janssen, and R. Kothari, "Moving beyond Smart Cities: Digital Nations for Social Innovation & Sustainability," *Information Systems Frontiers*, vol. 21, no. 3, pp. 495–501, 2019, doi: 10.1007/s10796-019-09930-0.

[73] P. T. I. Lam and W. Yang, "Factors influencing the consideration of Public-Private Partnerships (PPP) for smart city projects: Evidence from Hong Kong," *Cities*, vol. 99, no. December 2019, p. 102606, 2020, doi: 10.1016/j.cities.2020.102606.

[74] M. C. Domingo, "An overview of the internet of underwater things," *Journal of Network and Computer Applications*, vol. 35, no. 6, pp. 1879–1890, 2012, doi: 10.1016/j.jnca.2012.07.012.

[75] N. Victor *et al.*, "Federated Learning for IoUT: Concepts, Applications, Challenges and Opportunities," *Engineering*, vol. 8, pp. 33–41, 2022, doi: https://doi.org/10.48550/arXiv.2207.13976.

[76] N. Krishnaraj, M. Elhoseny, M. Thenmozhi, M. M. Selim, and K. Shankar, "Deep learning model for real-time image compression in Internet of Underwater Things (IoUT)," *J Real Time Image Process*, vol. 17, no. 6, pp. 2097–2111, 2020, doi: 10.1007/s11554-019-00879-6.

[77] Q. Zhao, Z. Peng, and X. Hong, “A named data networking architecture implementation to internet of underwater things,” *ACM International Conference Proceeding Series*, no. i, 2019, doi: 10.1145/3366486.3366506.

[78] M. C. Vuran, A. Salam, R. Wong, and S. Irmak, “Internet of underground things: Sensing and communications on the field for precision agriculture,” *IEEE World Forum on Internet of Things, WF-IoT 2018 - Proceedings*, vol. 2018-Janua, pp. 586–591, 2018, doi: 10.1109/WF-IoT.2018.8355096.

[79] A. G. Yisa, T. Dargahi, S. Belguith, and M. Hammoudeh, “Security challenges of Internet of Underwater Things: A systematic literature review,” *Transactions on Emerging Telecommunications Technologies*, vol. 32, no. 3, pp. 1–16, 2021, doi: 10.1002/ett.4203.

[80] A. V. Gadagkar and B. R. Chandavarkar, “A Comprehensive Review on Wireless Technologies and Their Issues with Underwater Communications,” *2021 12th International Conference on Computing Communication and Networking Technologies, ICCCNT 2021*, pp. 1–6, 2021, doi: 10.1109/ICCCNT51525.2021.9579572.

[81] M. F. Ali, D. N. K. Jayakody, Y. A. Chursin, S. Affes, and S. Dmitry, *Recent Advances and Future Directions on Underwater Wireless Communications*, vol. 27, no. 5. Springer Netherlands, 2020. doi: 10.1007/s11831-019-09354-8.

[82] N. Tang, Q. Zeng, D. Luo, Q. Xu, and H. Hu, “Research on Development and Application of Underwater Acoustic Communication System,” *J Phys Conf Ser*, vol. 1617, no. 1, 2020, doi: 10.1088/1742-6596/1617/1/012036.

[83] O. Onasami, D. Adesina, and L. Qian, “Underwater Acoustic Communication Channel Modeling using Deep Learning,” *WUWNet 2021 - 15th ACM International Conference on Underwater Networks and Systems*, 2021, doi: 10.1145/3491315.3491323.

[84] P. S. S. P. Ganesh and H. Venkataraman, “RF-based multihop wireless communication for shallow underwater environment,” *2019 International Conference on Wireless Communications, Signal Processing and Networking, WiSPNET 2019*, vol. 1, pp. 222–228, 2019, doi: 10.1109/WiSPNET45539.2019.9032810.

[85] M. Furqan Ali, D. K. Nalin Jayakody, T. D. Ponnimbaduge Perera, K. Srinivasan, A. Sharma, and I. Krikidis, “Underwater Communications: Recent Advances,” in *ETIC 2019*, Samdrup Jongkhar, Bhutan, 2019, pp. 97–102.

[86] P. K. Sajmath, R. V. Ravi, and K. K. A. Majeed, “Underwater Wireless Optical Communication Systems: A Survey,” in *2020 7th International Conference on Smart Structures and Systems (ICSSS)*, Chennai, India: IEEE, Jul. 2020, pp. 1–7. doi: 10.1109/ICSSS49621.2020.9202150.

[87] H. Kaushal and G. Kaddoum, "Underwater Optical Wireless Communication," *IEEE Access*, vol. 4, pp. 1518–1547, 2016, doi: 10.1109/ACCESS.2016.2552538.

[88] G. S. Spagnolo, L. Cozzella, and F. Leccese, "Underwater optical wireless communications: Overview," *Sensors (Switzerland)*, vol. 20, no. 8, 2020, doi: 10.3390/s20082261.

[89] P. Hamet and J. Tremblay, "Artificial intelligence in medicine," *Metabolism*, vol. 69, pp. S36–S40, 2017, doi: 10.1016/j.metabol.2017.01.011.

[90] X. Li and Y. Shi, "Computer vision imaging based on artificial intelligence," *Proceedings - 2018 International Conference on Virtual Reality and Intelligent Systems, ICVRIS 2018*, pp. 22–25, 2018, doi: 10.1109/ICVRIS.2018.00014.

[91] J. hua Li, "Cyber security meets artificial intelligence: a survey," *Frontiers of Information Technology and Electronic Engineering*, vol. 19, no. 12, pp. 1462–1474, 2018, doi: 10.1631/FITEE.1800573.

[92] A. Jean, "A brief history of artificial intelligence," *Medecine/Sciences*, vol. 36, no. 11, pp. 1059–1067, 2020, doi: 10.1051/medsci/2020189.

[93] Z.-H. Zhou, *Machine Learning*. Singapore: Springer Nature Singapore, 2021. doi: 10.1007/978-981-15-1967-3.

[94] P. P. Shinde and S. Shah, "A Review of Machine Learning and Deep Learning Applications," *Proceedings - 2018 4th International Conference on Computing, Communication Control and Automation, ICCUBEA 2018*, pp. 1–6, 2018, doi: 10.1109/ICCUBEA.2018.8697857.

[95] A. Geron, *Hands-on Machine Learning with Sckit-Learn Keras & Tensorflow*. O'Reilly Media, 2019.

[96] Mahesh Batta, "Machine Learning Algorithms - A Review," *International Journal of Science and Research (IJSR)*, vol. 9, no. 1, pp. 381–386, 2020, doi: 10.21275/ART20203995.

[97] L. Zhao, Y. Chen, and D. W. Schaffner, "Comparison of Logistic Regression and Linear Regression in Modeling Percentage Data," *Appl Environ Microbiol*, vol. 67, no. 5, pp. 2129–2135, 2001, doi: 10.1128/AEM.67.5.2129-2135.2001.

[98] Z. Ghahramani, "Unsupervised Learning," in *Machine Learning*, 2004, pp. 72–112. doi: 10.1007/978-3-540-28650-9_5.

[99] G. Pu, L. Wang, J. Shen, and F. Dong, "A hybrid unsupervised clustering-based anomaly detection method," *Tsinghua Sci Technol*, vol. 26, no. 2, pp. 146–153, 2021, doi: 10.26599/TST.2019.9010051.

[100] M. F. A. Hady and F. Schwenker, "Semi-supervised Learning," in *Intelligent Systems Reference Library*, vol. 49, 2013, pp. 215–239. doi: 10.1007/978-3-642-36657-4_7.

[101] Y. C A Padmanabha Reddy, P. Viswanath, and B. Eswara Reddy, "Semi-supervised learning: a brief review," *International Journal of Engineering & Technology*, vol. 7, no. 1.8, p. 81, Feb. 2018, doi: 10.14419/ijet.v7i1.8.9977.

[102] R. M. Cichy and D. Kaiser, "Deep Neural Networks as Scientific Models," *Trends Cogn Sci*, vol. 23, no. 4, pp. 305–317, 2019, doi: 10.1016/j.tics.2019.01.009.

[103] Y. Sun, X. Huang, D. Kroening, J. Sharp, M. Hill, and R. Ashmore, "Testing Deep Neural Networks," *ArXiv e-prints*, pp. 1–28, Mar. 2018, [Online]. Available: http://arxiv.org/abs/1803.04792

[104] M. Mathieu, M. Henaff, and Y. LeCun, "Fast training of convolutional networks through FFTS," *2nd International Conference on Learning Representations, ICLR 2014 - Conference Track Proceedings*, pp. 1–9, 2014.

[105] T. Barnett, S. Jain, U. Andra, and T. Khurana, "Global Internet Growth and Trends Source: Cisco VNI Global IP Traffic Forecast," no. December 2018, pp. 2017–2022, 2018.

[106] H. Y. Xiong *et al.*, "The human splicing code reveals new insights into the genetic determinants of disease," *Science (1979)*, vol. 347, no. 6218, Jan. 2015, doi: 10.1126/science.1254806.

[107] Z. Li, F. Liu, W. Yang, S. Peng, and J. Zhou, "A Survey of Convolutional Neural Networks: Analysis, Applications, and Prospects," *IEEE Trans Neural Netw Learn Syst*, vol. 33, no. 12, pp. 6999–7019, 2022, doi: 10.1109/TNNLS.2021.3084827.

[108] A. Saxena, "An Introduction to Convolutional Neural Networks," *Int J Res Appl Sci Eng Technol*, vol. 10, no. 12, pp. 943–947, 2022, doi: 10.22214/ijraset.2022.47789.

[109] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," in *Advances in Neural Information Processing Systems 25*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds., Curran Associates, Inc., 2012, pp. 1097–1105. [Online]. Available: http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf

[110] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, 2015, pp. 1–14.

[111] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778. doi: 10.1109/CVPR.2016.90.

[112] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the Inception Architecture for Computer Vision," in *Proceedings of the IEEE Computer Society*

Conference on Computer Vision and Pattern Recognition, 2016, pp. 2818–2826. doi: 10.1109/CVPR.2016.308.

[113] J. Murphy, “An Overview of Convolutional Neural Network Architectures for Deep Learning,” *Microway Inc*, pp. 1–22, 2016.

[114] S. Levine, A. Kumar, G. Tucker, and J. Fu, “Offline Reinforcement Learning: Tutorial, Review, and Perspectives on Open Problems,” pp. 1–43, 2020.

[115] Deepanshu Mehta, “State-of-the-Art Reinforcement Learning Algorithms,” *International Journal of Engineering Research and*, vol. V8, no. 12, pp. 717–722, 2020, doi: 10.17577/ijertv8is120332.

[116] M. Van Otterlo and M. Wiering, “Reinforcement learning and markov decision processes,” *Adaptation, Learning, and Optimization*, vol. 12, pp. 3–42, 2012, doi: 10.1007/978-3-642-27645-3_1.

[117] M. Guo, Y. Liu, and J. Malec, “A New Q-Learning Algorithm Based on the Metropolis Criterion,” *IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics)*, vol. 34, no. 5, pp. 2140–2143, Oct. 2004, doi: 10.1109/TSMCB.2004.832154.

[118] B. Jang, M. Kim, G. Harerimana, and J. W. Kim, “Q-Learning Algorithms: A Comprehensive Classification and Applications,” *IEEE Access*, vol. 7, pp. 133653–133667, 2019, doi: 10.1109/ACCESS.2019.2941229.

[119] G. V. de la Cruz, Y. Du, and M. E. Taylor, “Pre-training Neural Networks with Human Demonstrations for Deep Reinforcement Learning,” *ALA 2018 - Adaptive Learning Agents - Workshop at the Federated AI Meeting 2018*, Sep. 2017.

[120] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.

[121] K. Arulkumaran, M. P. Deisenroth, M. Brundage, and A. A. Bharath, “Deep Reinforcement Learning: A Brief Survey,” *IEEE Signal Process Mag*, vol. 34, no. 6, pp. 26–38, Nov. 2017, doi: 10.1109/MSP.2017.2743240.

[122] M. G. Bellemare, Y. Naddaf, J. Veness, and M. Bowling, “The Arcade Learning Environment: An evaluation platform for general agents,” *Journal of Artificial Intelligence Research*, vol. 47, pp. 253–279, 2013, doi: 10.1613/jair.3912.

[123] S. Jafari, S. Hoseinzadeh, and A. Sohani, “Deep Q-Value Neural Network (DQN) Reinforcement Learning for the Techno-Economic Optimization of a Solar-Driven Nanofluid-Assisted Desalination Technology,” *Water (Switzerland)*, vol. 14, no. 14, 2022, doi: 10.3390/w14142254.

[124] W. Kim, “Cloud computing: Today and Tomorrow,” *Journal of Object Technology*, vol. 8, no. 1, pp. 65–72, 2009, doi: 10.5381/jot.2009.8.1.c4.

[125] H. Shukur, S. Zeebaree, R. Zebari, D. Zeebaree, O. Ahmed, and A. Salih, “Cloud Computing Virtualization of Resources Allocation for Distributed Systems,” *Journal of Applied Science and Technology Trends*, vol. 1, no. 3, pp. 98–105, 2020, doi: 10.38094/jastt1331.

[126] C. Mustafa Mohammed and S. R. M Zeebaree, “Sufficient Comparison Among Cloud Computing Services: IaaS, PaaS, and SaaS: A Review,” *International Journal of Science and Business*, vol. 5, no. 2, pp. 17–30, 2021, doi: 10.5281/zenodo.4450129.

[127] T. Dillon, C. Wu, and E. Chang, “Cloud computing: Issues and challenges,” *Proceedings - International Conference on Advanced Information Networking and Applications, AINA*, pp. 27–33, 2010, doi: 10.1109/AINA.2010.187.

[128] R. Velumadhava Rao and K. Selvamani, “Data security challenges and its solutions in cloud computing,” *Procedia Comput Sci*, vol. 48, no. C, pp. 204–209, 2015, doi: 10.1016/j.procs.2015.04.171.

[129] O. Tomarchio, D. Calcaterra, and G. Di Modica, “Cloud resource orchestration in the multi-cloud landscape: a systematic review of existing frameworks,” *Journal of Cloud Computing*, vol. 9, no. 1, 2020, doi: 10.1186/s13677-020-00194-7.

[130] Q. Luo, S. Hu, C. Li, G. Li, and W. Shi, “Resource Scheduling in Edge Computing: A Survey,” *IEEE Communications Surveys and Tutorials*, vol. 23, no. 4, pp. 2131–2165, 2021, doi: 10.1109/COMST.2021.3106401.

[131] K. Jain and S. Mohapatra, “Taxonomy of edge computing: Challenges, opportunities, and data reduction methods,” *EAI/Springer Innovations in Communication and Computing*, pp. 51–69, 2019, doi: 10.1007/978-3-319-99061-3_4.

[132] A. Santoyo-Gonzalez and C. Cervello-Pastor, “Network-Aware Placement Optimization for Edge Computing Infrastructure under 5G,” *IEEE Access*, vol. 8, pp. 56015–56028, 2020, doi: 10.1109/ACCESS.2020.2982241.

[133] Y. Siriwardhana, P. Porambage, M. Liyanage, and M. Ylianttila, “A Survey on Mobile Augmented Reality with 5G Mobile Edge Computing: Architectures, Applications, and Technical Aspects,” *IEEE Communications Surveys and Tutorials*, vol. 23, no. 2, pp. 1160–1192, 2021, doi: 10.1109/COMST.2021.3061981.

[134] S. Garg, K. Kaur, G. Kaddoum, P. Garigipati, and G. S. Aujla, “Security in IoT-Driven Mobile Edge Computing: New Paradigms, Challenges, and Opportunities,” *IEEE Netw*, vol. 35, no. 5, pp. 298–305, Sep. 2021, doi: 10.1109/MNET.211.2000526.

[135] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, “Fog computing and its role in the internet of things,” *MCC’12 - Proceedings of the 1st ACM Mobile Cloud Computing Workshop*, pp. 13–15, 2012, doi: 10.1145/2342509.2342513.

[136] M. Firdhous, O. Ghazali, and S. Hassan, "Fog Computing: Will it be the Future of Cloud Computing? | SDIWC Organization - Academia.edu," *3rd International Conference on Informatics & Applications (ICIA2014)*, no. October, pp. 8–15, 2014.

[137] A. Kumari, S. Tanwar, S. Tyagi, and N. Kumar, "Fog computing for Healthcare 4.0 environment: Opportunities and challenges," *Computers and Electrical Engineering*, vol. 72, pp. 1–13, 2018, doi: 10.1016/j.compeleceng.2018.08.015.

[138] M. Iorga, L. Feldman, R. Barton, M. J. Martin, N. Goren, and C. Mahmoudi, "Fog computing conceptual model," Gaithersburg, MD, Mar. 2018. doi: 10.6028/NIST.SP.500-325.

[139] I. Stojmenovic, S. Wen, X. Huang, and H. Luan, "An overview of Fog computing and its security issues," *Concurr Comput*, vol. 28, no. 10, pp. 2991–3005, Jul. 2016, doi: 10.1002/cpe.3485.

[140] S. Yi, Z. Hao, Z. Qin, and Q. Li, "Fog computing: Platform and applications," *Proceedings - 3rd Workshop on Hot Topics in Web Systems and Technologies, HotWeb 2015*, pp. 73–78, 2016, doi: 10.1109/HotWeb.2015.22.

[141] K. Kumaran and E. Sasikala, "Learning based Latency Minimization Techniques in Mobile Edge Computing (MEC) systems: A Comprehensive Survey," *2021 International Conference on System, Computation, Automation and Networking, ICSCAN 2021*, pp. 1–6, 2021, doi: 10.1109/ICSCAN53069.2021.9526410.

[142] Y. Jararweh, A. Doulat, O. Alqudah, E. Ahmed, M. Al-Ayyoub, and E. Benkhelifa, "The future of mobile cloud computing: Integrating cloudlets and Mobile Edge Computing," *2016 23rd International Conference on Telecommunications, ICT 2016*, pp. 1–5, 2016, doi: 10.1109/ICT.2016.7500486.

[143] X. Sun and N. Ansari, "EdgeIoT: Mobile Edge Computing for the Internet of Things," *IEEE Communications Magazine*, vol. 54, no. 12, pp. 22–29, 2016, doi: 10.1109/MCOM.2016.1600492CM.

[144] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "A Survey on Mobile Edge Computing: The Communication Perspective," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 4, pp. 2322–2358, 2017, doi: 10.1109/COMST.2017.2745201.

[145] N. Abbas, Y. Zhang, A. Taherkordi, and T. Skeie, "Mobile Edge Computing: A Survey," *IEEE Internet Things J*, vol. 5, no. 1, pp. 450–465, 2018, doi: 10.1109/JIOT.2017.2750180.

[146] P. Ranaweera, A. Jurcut, and M. Liyanage, "MEC-enabled 5G Use Cases: A Survey on Security Vulnerabilities and Countermeasures," *ACM Comput Surv*, vol. 54, no. 9, 2022, doi: 10.1145/3474552.

[147] F. Zantalis, G. Koulouras, S. Karabetsos, and D. Kandris, "A review of machine learning and IoT in smart transportation," *Future Internet*, vol. 11, no. 4, pp. 1–23, 2019, doi: 10.3390/FI11040094.

[148] A. Salehi-Amiri, N. Akbapour, M. Hajiaghaei-Keshteli, Y. Gajpal, and A. Jabbarzadeh, "Designing an effective two-stage, sustainable, and IoT based waste management system," *Renewable and Sustainable Energy Reviews*, vol. 157, no. January, p. 112031, 2022, doi: 10.1016/j.rser.2021.112031.

[149] A. Verma, V. K. Shukla, and R. Sharma, "Convergence of IOT in tourism industry: A pragmatic analysis," *J Phys Conf Ser*, vol. 1714, no. 1, 2021, doi: 10.1088/1742-6596/1714/1/012037.

[150] S. Tiwari, J. Rosak-Szyrocka, and J. Żywiołek, "Internet of Things as a Sustainable Energy Management Solution at Tourism Destinations in India," *Energies (Basel)*, vol. 15, no. 7, 2022, doi: 10.3390/en15072433.

[151] W. Cao *et al.*, "CNN-based intelligent safety surveillance in green IoT applications," *China Communications*, vol. 18, no. 1, pp. 108–119, 2021, doi: 10.23919/JCC.2021.01.010.

[152] W. P. den Beemt and R. Smith, "Smart tourism tools: linking technology to the touristic resources of a city," in *Smart Tourism Congress Barcelona*, 2016, pp. 1–12. [Online]. Available: http://elib.hcmussh.edu.vn/handle/HCMUSSH/136114

[153] M. Hu, E. Pantano, and N. Stylos, "How does Internet of Things (IOT) affect travel experience?," *Tourism marketing in Western Europe*, pp. 9–25, 2021, doi: 10.1079/9781789248753.0001.

[154] E. Balandina, S. Balandin, Y. Koucheryavy, and D. Mouromtsev, "IoT Use Cases in Healthcare and Tourism," *Proceedings - 17th IEEE Conference on Business Informatics, CBI 2015*, vol. 2, pp. 37–44, 2015, doi: 10.1109/CBI.2015.16.

[155] T. Car, L. Pilepić Stifanich, and M. Šimunić, "Internet of Things (Iot) in Tourism and Hospitality: Opportunities and Challenges," in *ToSEE – Tourism in Southern and Eastern Europe, Vol. 5*, Opatija, Croatia, May 2019, pp. 163–173. doi: 10.20867/tosee.05.42.

[156] P. Popova, M. Petrova, V. Popov, and K. Marinova, "Internet of Things as a Key Technology for Developing Smart Tourism Destinations," *2022 IEEE 9th International Conference on Problems of Infocommunications, Science and Technology (PIC S&T)*, pp. 151–156, 2023, doi: 10.1109/picst57299.2022.10238665.

[157] E. M. A. González, E. Municio, M. N. Alemán, and J. M. Marquez-Barja, "Cultural Heritage and Internet of Things," *ACM International Conference Proceeding Series*, pp. 248–251, 2020, doi: 10.1145/3411170.3411267.

[158] Y. P. Wang, X. Dai, J. J. Jung, and C. Choi, “Performance analysis of smart cultural heritage protection oriented wireless networks,” *Future Generation Computer Systems*, vol. 81, pp. 593–600, 2018, doi: 10.1016/j.future.2017.04.007.

[159] A. Perles *et al.*, “An energy-efficient internet of things (IoT) architecture for preventive conservation of cultural heritage,” *Future Generation Computer Systems*, vol. 81, pp. 566–581, 2018, doi: 10.1016/j.future.2017.06.030.

[160] V. G. Spasova, B. G. Georgiev, P. D. Stefanov, and B. P. Stoyanov, “Prototype of Smart Monument with IoT-based System of Early Warning,” *IOP Conf Ser Mater Sci Eng*, vol. 1031, no. 1, pp. 1–9, 2021, doi: 10.1088/1757-899X/1031/1/012126.

[161] F. Piccialli and A. Chianese, “A location-based IoT platform supporting the cultural heritage domain,” *Concurr Comput*, vol. 29, no. 11, 2017, doi: 10.1002/cpe.4091.

[162] S. Alletto *et al.*, “An Indoor Location-Aware System for an IoT-Based Smart Museum,” *IEEEE Internet of Things*, vol. 3, no. 2, pp. 244–253, 2016.

[163] F. Piccialli, F. Giampaolo, G. Casolla, V. S. Di Cola, and K. Li, “A Deep Learning approach for Path Prediction in a Location-based IoT system,” *Pervasive Mob Comput*, vol. 66, p. 101210, 2020, doi: 10.1016/j.pmcj.2020.101210.

[164] F. Ricci, B. Shapira, and L. Rokach, *Recommender systems handbook, Second edition*. 2015. doi: 10.1007/978-1-4899-7637-6.

[165] C. C. Aggarwal, “An Introduction to Recommender Systems,” *Recommender Systems*, pp. 1–28, 2016, doi: 10.1007/978-3-319-29659-3_1.

[166] J. Bobadilla, F. Ortega, A. Hernando, and A. Gutiérrez, “Recommender systems survey,” *Knowl Based Syst*, vol. 46, pp. 109–132, 2013, doi: 10.1016/j.knosys.2013.03.012.

[167] F. O. Isinkaye, Y. O. Folajimi, and B. A. Ojokoh, “Recommendation systems: Principles, methods and evaluation,” *Egyptian Informatics Journal*, vol. 16, no. 3, pp. 261–273, 2015, doi: 10.1016/j.eij.2015.06.005.

[168] M. Unger, A. Tuzhilin, and A. Livne, “Context-Aware Recommendations Based on Deep Learning Frameworks,” *ACM Trans Manag Inf Syst*, vol. 11, no. 2, 2020, doi: 10.1145/3386243.

[169] V. A. Rohani, Z. M. Kasirun, S. Kumar, and S. Shamshirband, “An effective recommender algorithm for cold-start problem in academic social networks,” *Math Probl Eng*, vol. 2014, 2014, doi: 10.1155/2014/123726.

[170] R. A. Hamid *et al.*, “How smart is e-tourism? A systematic review of smart tourism recommendation system applying data management,” *Comput Sci Rev*, vol. 39, p. 100337, 2021, doi: 10.1016/j.cosrev.2020.100337.

[171] N. E. I. Karabadji, S. Beldjoudi, H. Seridi, S. Aridhi, and W. Dhifli, "Improving memory-based user collaborative filtering with evolutionary multi-objective optimization," *Expert Syst Appl*, vol. 98, pp. 153–165, 2018, doi: 10.1016/j.eswa.2018.01.015.

[172] N. Ranjbar Kermany and S. H. Alizadeh, "A hybrid multi-criteria recommender system using ontology and neuro-fuzzy techniques," *Electron Commer Res Appl*, vol. 21, pp. 50–64, 2017, doi: 10.1016/j.elerap.2016.12.005.

[173] M. Sertkan, J. Neidhardt, and H. Werthner, "From Pictures to Travel Characteristics: Deep Learning-Based Profiling of Tourists and Tourism Destinations," in *Information and Communication Technologies in Tourism 2020*, J. Neidhardt and W. Wörndl, Eds., Cham: Springer International Publishing, 2020, pp. 142–153.

[174] T. Kurashima, T. Iwata, G. Irie, and K. Fujimura, "Travel route recommendation using geotags in photo sharing sites," *International Conference on Information and Knowledge Management, Proceedings*, pp. 579–588, 2010, doi: 10.1145/1871437.1871513.

[175] Z. Xu, L. Chen, and G. Chen, "Topic based context-aware travel recommendation method exploiting geotagged photos," *Neurocomputing*, vol. 155, pp. 99–107, 2015, doi: 10.1016/j.neucom.2014.12.043.

[176] I. R. Brilhante, J. A. Macedo, F. M. Nardini, R. Perego, and C. Renso, "On planning sightseeing tours with TripBuilder," *Inf Process Manag*, vol. 51, no. 2, pp. 1–15, 2015, doi: 10.1016/j.ipm.2014.10.003.

[177] G. Cai, K. Lee, and I. Lee, "Itinerary recommender system with semantic trajectory pattern mining from geo-tagged photos," *Expert Syst Appl*, vol. 94, pp. 32–40, 2018, doi: 10.1016/j.eswa.2017.10.049.

[178] K. H. Lim, "Recommending tours and places-of-interest based on user interests from geo-tagged photos," *Proceedings of the ACM SIGMOD International Conference on Management of Data*, vol. 31-May-201, pp. 33–38, 2015, doi: 10.1145/2744680.2744693.

[179] K. H. Lim, J. Chan, C. Leckie, and S. Karunasekera, "Personalized trip recommendation for tourists based on user interests, points of interest visit durations and visit recency," *Knowl Inf Syst*, vol. 54, no. 2, pp. 375–406, 2018, doi: 10.1007/s10115-017-1056-y.

[180] Y. Liu, T. A. N. Pham, G. Cong, and Q. Yuan, "An experimental evaluation of pointofinterest recommendation in location-based social networks," *Proceedings of the VLDB Endowment*, vol. 10, no. 10, pp. 1010–1021, 2017, doi: 10.14778/3115404.3115407.

[181] K. Kesorn, W. Juraphanthong, and A. Salaiwarakul, "Personalized Attraction Recommendation System for Tourists Through Check-In Data," *IEEE Access*, vol. 5, pp. 26703–26721, 2017, doi: 10.1109/ACCESS.2017.2778293.

[182] I. Ben Sassi, S. Mellouli, and S. Ben Yahia, “Context-aware recommender systems in mobile environment: On the road of future research,” *Inf Syst*, vol. 72, pp. 27–61, 2017, doi: 10.1016/j.is.2017.09.001.

[183] J. Shen, C. Deng, and X. Gao, “Attraction recommendation: Towards personalized tourism via collective intelligence,” *Neurocomputing*, vol. 173, pp. 789–798, 2016, doi: 10.1016/j.neucom.2015.08.030.

[184] A. Da’u and N. Salim, “Recommendation system based on deep learning methods: a systematic review and new directions,” *Artif Intell Rev*, vol. 53, no. 4, pp. 2709–2748, Apr. 2020, doi: 10.1007/s10462-019-09744-1.

[185] S. Zhang, L. Yao, A. Sun, and Y. Tay, “Deep Learning Based Recommender System: A Survey and New Perspectives,” *ACM Comput. Surv.*, vol. 52, no. 1, 2019, doi: 10.1145/3285029.

[186] I. Bartolini *et al.*, “Recommending multimedia visiting paths in cultural heritage applications,” *Multimed Tools Appl*, vol. 75, no. 7, pp. 3813–3842, 2016, doi: 10.1007/s11042-014-2062-7.

[187] M. Hong, J. J. Jung, F. Piccialli, and A. Chianese, “Social recommendation service for cultural heritage,” *Pers Ubiquitous Comput*, vol. 21, no. 2, pp. 191–201, 2017, doi: 10.1007/s00779-016-0985-x.

[188] A. Chianese, F. Marulli, F. Piccialli, P. Benedusi, and J. E. Jung, “An associative engines based approach supporting collaborative analytics in the Internet of cultural things,” *Future Generation Computer Systems*, vol. 66, pp. 187–198, 2017, doi: 10.1016/j.future.2016.04.015.

[189] S. Cuomo, P. De Michele, F. Piccialli, A. Galletti, and J. E. Jung, “IoT-based collaborative reputation system for associating visitors and artworks in a cultural scenario,” *Expert Syst Appl*, vol. 79, pp. 101–111, 2017, doi: 10.1016/j.eswa.2017.02.034.

[190] K. Al Fararni, F. Nafis, B. Aghoutane, A. Yahyaouy, J. Riffi, and A. Sabri, “Hybrid recommender system for tourism based on big data and AI: A conceptual framework,” *Big Data Mining and Analytics*, vol. 4, no. 1, pp. 47–55, 2021, doi: 10.26599/BDMA.2020.9020015.

[191] M. Indriana and C.-S. Hwang, “Applying Neural Network Model to Hybrid Tourist Attraction Recommendations,” *Jurnal ULTIMATICS*, vol. 6, no. 2, pp. 63–69, 2014, doi: 10.31937/ti.v6i2.339.

[192] M. Chen, Z. Xu, K. Q. Weinberger, and F. Sha, “Marginalized denoising autoencoders for domain adaptation,” *Proceedings of the 29th International Conference on Machine Learning, ICML 2012*, vol. 1, pp. 767–774, 2012.

[193] M. Woźniak, J. Siłka, and M. Wieczorek, "Deep neural network correlation learning mechanism for CT brain tumor detection," *Neural Comput Appl*, 2021, doi: 10.1007/s00521-021-05841-x.

[194] M. Wozniak, M. Wieczorek, J. Silka, and D. Polap, "Body Pose Prediction Based on Motion Sensor Data and Recurrent Neural Network," *IEEE Trans Industr Inform*, vol. 17, no. 3, pp. 2101–2111, 2021, doi: 10.1109/TII.2020.3015934.

[195] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press. http://www.deeplearningbook.org., 2016.

[196] I. J. Goodfellow *et al.*, "Generative Adversarial Networks," *Advances in neural information processing systems*, vol. 27, Jun. 2014, Accessed: Aug. 14, 2021. [Online]. Available: http://arxiv.org/abs/1406.2661

[197] Y. Li, "Deep Reinforcement Learning," *CoRR, abs/1810.06339,* Oct. 2018, Accessed: Aug. 14, 2021. [Online]. Available: http://arxiv.org/abs/1810.06339

[198] N. Sivaramakrishnan, V. Subramaniyaswamy, A. Viloria, V. Vijayakumar, and N. Senthilselvan, "A deep learning-based hybrid model for recommendation generation and ranking," *Neural Comput Appl*, vol. 4, 2020, doi: 10.1007/s00521-020-04844-4.

[199] Q. Shambour, "A deep learning based algorithm for multi-criteria recommender systems," *Knowl Based Syst*, vol. 211, p. 106545, 2021, doi: 10.1016/j.knosys.2020.106545.

[200] T. Nakahara and K. Yada, "Analyzing consumers' shopping behavior using RFID data and pattern mining," *Adv Data Anal Classif*, vol. 6, no. 4, pp. 355–365, 2012, doi: 10.1007/s11634-012-0117-z.

[201] H. F. Atlam, R. J. Walters, and G. B. Wills, "Fog computing and the internet of things: A review," *Big Data and Cognitive Computing*, vol. 2, no. 2, pp. 1–18, 2018, doi: 10.3390/bdcc2020010.

[202] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of Things (IoT): A vision, architectural elements, and future directions," *Future Generation Computer Systems*, vol. 29, no. 7, pp. 1645–1660, 2013, doi: 10.1016/j.future.2013.01.010.

[203] Statista, "Leading city destinations worldwide in 2018, by number of overnight visitors." Accessed: Jun. 07, 2021. [Online]. Available: https://www.statista.com/statistics/310355/overnight-visitors-to-top-city-destinations-worldwide/

[204] Statista, "Number of international overnight visitors in the most popular European city destinations in 2016." Accessed: Jun. 07, 2021. [Online]. Available: https://es.statista.com/estadisticas/487720/turistas-internacionales-en-los-principales-destinos-europeos/

[205] Eden Strategy Institute and ONG&ONG, *Smart city governments 1*. 2018.

[206] IESE Business School, *IESE Cities in Motion Index*. 2020.

[207] Barcelona municipality, "Tourists and overnight stays." Accessed: Jun. 07, 2021. [Online]. Available: https://ajuntament.barcelona.cat/estadistica/castella/index.htm

[208] Barcelona municipality, "Other tourist information." Accessed: Jun. 07, 2021. [Online]. Available: https://www.bcn.cat/estadistica/angles/dades/anuari/cap13/C1306010.htm

[209] A. H. Jackson, *Machine learning: A Probabilistic Perspective*, vol. 5, no. 2. 1988. doi: 10.1111/j.1468-0394.1988.tb00341.x.

[210] N. Srivastava, H. Geoffrey, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A Simple Way to Prevent Neural Networks from Overfittin," *Journal of Machine Learning Research*, 2014, doi: 10.1016/0370-2693(93)90272-J.

[211] P. Liashchynskyi and P. Liashchynskyi, "Grid Search, Random Search, Genetic Algorithm: A Big Comparison for NAS," no. 2017, pp. 1–11, 2019.

[212] D. P. Kingma and J. L. Ba, "Adam: A method for stochastic optimization," *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, pp. 1–15, 2015.

[213] G. Tsoumakas, I. Katakis, and I. Vlahavas, "A Review of Multi-Label Classification Methods," *Proceedings of the 2nd ADBIS Workshop on Data Mining and Knowledge Discovery (ADMKD 2006)*, pp. 99–109, 2006.

[214] Scikit-learn and M. L. in Python, "Multiclass and multioutput algorithms." Accessed: Jun. 07, 2021. [Online]. Available: https://scikit-learn.org/stable/modules/multiclass.html#multioutputclassifier

[215] A. Alotaibi, "Automated and intelligent system for monitoring swimming pool safety based on the IoT and transfer learning," *Electronics (Switzerland)*, vol. 9, no. 12, pp. 1–13, 2020, doi: 10.3390/electronics9122082.

[216] S. Jalalifar *et al.*, "A Smart Multi-Sensor Device to Detect Distress in Swimmers," *Sensors*, vol. 22, no. 3, pp. 1–15, 2022, doi: 10.3390/s22031059.

[217] C. Burnay, D. I. Anderson, C. Button, R. Cordovil, and A. E. Peden, "Infant Drowning Prevention: Insights from a New Ecological Psychology Approach," *Int J Environ Res Public Health*, vol. 19, no. 8, 2022, doi: 10.3390/ijerph19084567.

[218] J. Meniere, "System for monitoring a swimming pool to prevent drowning accidents," U. S. Patent 6,133,838, Oct. 17, 2000 [Online]. Available: https://patents.google.com/patent/US6133838A/en

[219] E. Menoud, “Alarm and monitoring device for the presumption of bodies in danger in a swimming pool,” U.S. Patent 5,886,630, Mar. 23, 1999 [Online]. Available: https://patents.google.com/patent/US5886630A/en

[220] C. Zhang, X. Li, and F. Lei, “A Novel Camera-Based Drowning Detection Algorithm,” in *Advances in Image and Graphics Technologies: 10th Chinese Conference, IGTA*, vol. 525, T. Tan, Q. Ruan, S. Wang, H. Ma, and K. Di, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2015, pp. 224–233. doi: 10.1007/978-3-662-47791-5_26.

[221] W. Lu and Y.-P. Tan, “A camera-based system for early detection of drowning incidents,” in *Proceedings. International Conference on Image Processing*, 2002, pp. 445–448. doi: 10.1109/ICIP.2002.1039001.

[222] W. Lu and Y. P. Tan, “A vision-based approach to early detection of drowning incidents in swimming pools,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 14, no. 2, pp. 159–178, 2004, doi: 10.1109/TCSVT.2003.821980.

[223] A. H. Kam, W. Lu, and W. Y. Yau, “A video-based drowning detection system,” *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 2353, pp. 297–311, 2002, doi: 10.1007/3-540-47979-1_20.

[224] B. Victor, Z. He, S. Morgan, and D. Miniutti, “Continuous Video to Simple Signals for Swimming Stroke Detection with Convolutional Neural Networks,” *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, vol. 2017-July, pp. 122–131, 2017, doi: 10.1109/CVPRW.2017.21.

[225] H. L. Eng, K. A. Toh, A. H. Kam, J. Wang, and W. Y. Yau, “An automatic drowning detection surveillance system for challenging outdoor pool environments,” *Proceedings of the IEEE International Conference on Computer Vision*, vol. 1, pp. 532–539, 2003, doi: 10.1109/iccv.2003.1238393.

[226] K. L. Chan, “Detection of swimmer using dense optical flow motion map and intensity information,” *Mach Vis Appl*, vol. 24, no. 1, pp. 75–101, 2013, doi: 10.1007/s00138-012-0419-3.

[227] H. L. Eng, J. Wang, A. H. K. Siew Wah, and W. Y. Yau, “Robust human detection within a highly dynamic aquatic environment in real time,” *IEEE Transactions on Image Processing*, vol. 15, no. 6, pp. 1583–1600, 2006, doi: 10.1109/TIP.2006.871119.

[228] L. Jiao *et al.*, “A survey of deep learning-based object detection,” *IEEE Access*, vol. 7, pp. 128837–128868, 2019, doi: 10.1109/ACCESS.2019.2939201.

[229] M. A. Hayat, G. Yang, and A. Iqbal, “Mask R-CNN Based Real Time near Drowning Person Detection System in Swimming Pools,” *Proceedings of the 2022 Mohammad Ali*

Jinnah University International Conference on Computing, MAJICC 2022, pp. 1–6, 2022, doi: 10.1109/MAJICC56935.2022.9994135.

[230] M. B. Jensen, R. Gade, and T. B. Moeslund, "Swimming pool occupancy analysis using deep learning on low quality video," in *MMSports 2018 - Proceedings of the 1st International Workshop on Multimedia Content Analysis in Sports, Co-located with MM 2018*, 2018, pp. 67–73. doi: 10.1145/3265845.3265846.

[231] H. L. Eng, K. A. Toh, W. Y. Yau, and J. Wang, "DEWS: A live visual surveillance system for early drowning detection at pool," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 18, no. 2, pp. 196–210, 2008, doi: 10.1109/TCSVT.2007.913960.

[232] G. Cosoli, L. Antognoli, V. Veroli, and L. Scalise, "Accuracy and Precision ofWearable Devices for Real-Time Monitoring of Swimming Athletes," *Sensors*, vol. 22, no. 13, 2022, doi: 10.3390/s22134726.

[233] J. Costa, C. Silva, M. Santos, T. Fernandes, and S. Faria, "Framework for intelligent swimming analytics with wearable sensors for stroke classification," *Sensors*, vol. 21, no. 15, pp. 1–17, 2021, doi: 10.3390/s21155162.

[234] E. Kałamajska, J. Misiurewicz, and J. Weremczuk, "Wearable Pulse Oximeter for Swimming Pool Safety," *Sensors*, vol. 22, no. 10, 2022, doi: 10.3390/s22103823.

[235] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, 2015, pp. 1–14.

[236] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778. doi: 10.1109/CVPR.2016.90.

[237] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the Inception Architecture for Computer Vision," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2818–2826. doi: 10.1109/CVPR.2016.308.

[238] D. Meddings, E. Altieri, J. Bierens, E. Cassell, A. Gissing, and J. Guevarra, "Preventing drowning: an implementation guide," World Health Organization. [Online]. Available: http://apps.who.int/iris/bitstream/10665/255196/1/9789241511933-eng.pdf?ua=1

[239] A. Mahony, A. Peden, C. Roberts, and P. Barnsley, "A 10 year analysis of drowning in Aquatic Facilities: Exploring Risk at Communal, Public and Commercial Swimming Pools," *Royal Life Saving*, Australia. Sydney., pp. 1–48, 2018. [Online]. Available: https://www.royallifesaving.com.au/__data/assets/pdf_file/0009/37557/RLS_PublicPools_10YearReport.pdf

[240] K. Connolly, “Child drownings in Germany linked to parents’ phone ‘fixation,’” The Guardian. Accessed: Aug. 17, 2021. [Online]. Available: https://www.theguardian.com/lifeandstyle/2018/aug/15/parents-fixated-by-phones-linked-to-child-drownings-in-germany

[241] J. Dunne, “Kids Are Drowning Because Their Parents Are Distracted By Devices,” 10 Daily. Accessed: Aug. 17, 2021. [Online]. Available: https://10daily.com.au/news/australia/a180816zwp/kids-are-drowning-because-their-parents-are-distracted-by-devices-20180816

[242] ETSI, “TS 122 280 - V15.3.0 - LTE; Mission Critical Services Common Requirements (3GPP TS 22.280 version 15.3.0 Release 15),” 2018.

[243] R. Maldonado *et al.*, “Comparing Wi-Fi 6 and 5G Downlink Performance for Industrial IoT,” *IEEE Access*, vol. 9, pp. 86928–86937, 2021, doi: 10.1109/ACCESS.2021.3085896.

[244] ETSI, “5G; System Architecture for the 5G System (3GPP TS 23.501 version 15.2.0 Release 15),” *European Telecommunications Standards Institute. Tech. Rep. V15.2.0*, vol. 15, pp. 4–220, 2018.

[245] C. Díaz *et al.*, “Metallophosphazene Precursor Routes to the Solid-State Deposition of Metallic and Dielectric Microstructures and Nanostructures on Si and SiO 2,” *Langmuir*, vol. 26, no. 12, pp. 10223–10233, Jun. 2010, doi: 10.1021/la100371w.

[246] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, “Edge Computing: Vision and Challenges,” *IEEE Internet Things J*, vol. 3, no. 5, pp. 637–646, 2016, doi: 10.1109/JIOT.2016.2579198.

[247] M. S. Elbamby *et al.*, “Wireless Edge Computing With Latency and Reliability Guarantees,” *Proceedings of the IEEE*, vol. 107, no. 8, 2019, doi: 10.1109/JPROC.2019.2917084.

[248] R. Mijumbi, J. Serrat, J. L. Gorricho, N. Bouten, F. De Turck, and R. Boutaba, “Network function virtualization: State-of-the-art and research challenges,” *IEEE Communications Surveys and Tutorials*, vol. 18, no. 1, pp. 236–262, 2016, doi: 10.1109/COMST.2015.2477041.

[249] D. Kreutz, F. M. V. Ramos, P. Esteves Verissimo, C. Esteve Rothenberg, S. Azodolmolky, and S. Uhlig, “Software-Defined Networking: A Comprehensive Survey,” *Proceedings of the IEEE*, vol. 103, no. 1, pp. 14–76, Jan. 2015, doi: 10.1109/JPROC.2014.2371999.

[250] NGMN Alliance, “Description of Network Slicing Concept by NGMN Alliance,” *Ngmn 5G P1*, vol. 1, no. September, p. 19, 2016.

[251] F. Tango and M. Botta, “Real-time detection system of driver distraction using machine learning,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 2, pp. 894–905, 2013, doi: 10.1109/TITS.2013.2247760.

[252] K. G. Omeke, A. I. Abubakar, L. Zhang, Q. H. Abbasi, and M. A. Imran, "How Reinforcement Learning is Helping to Solve Internet-of-Underwater-Things Problems," *IEEE Internet of Things Magazine*, vol. 5, no. 4, pp. 24–29, 2023, doi: 10.1109/iotm.001.2200129.

[253] S. Cai, Y. Zhu, T. Wang, G. Xu, A. Liu, and X. Liu, "Data Collection in Underwater Sensor Networks based on Mobile Edge Computing," *IEEE Access*, vol. 7, pp. 65357–65367, 2019, doi: 10.1109/ACCESS.2019.2918213.

[254] S. Song, J. Liu, J. Guo, B. Lin, Q. Ye, and J. Cui, "Efficient Data Collection Scheme for Multi-Modal Underwater Sensor Networks Based on Deep Reinforcement Learning," *IEEE Trans Veh Technol*, vol. 72, no. 5, pp. 6558–6570, May 2023, doi: 10.1109/TVT.2022.3232391.

[255] Z. Liu, X. Meng, Y. Liu, Y. Yang, and Y. Wang, "AUV-Aided Hybrid Data Collection Scheme Based on Value of Information for Internet of Underwater Things," *IEEE Internet Things J*, vol. 9, no. 9, pp. 6944–6955, 2022, doi: 10.1109/JIOT.2021.3115800.

[256] X. Zhuo, M. Liu, Y. Wei, G. Yu, F. Qu, and R. Sun, "AUV-Aided Energy-Efficient Data Collection in Underwater Acoustic Sensor Networks," *IEEE Internet Things J*, vol. 7, no. 10, pp. 10010–10022, 2020, doi: 10.1109/JIOT.2020.2988697.

[257] Z. Fang, J. Wang, C. Jiang, Q. Zhang, and Y. Ren, "AoI-Inspired Collaborative Information Collection for AUV-Assisted Internet of Underwater Things," *IEEE Internet Things J*, vol. 8, no. 19, pp. 14559–14571, 2021, doi: 10.1109/JIOT.2021.3049239.

[258] S. Cai, Y. Zhu, T. Wang, G. Xu, A. Liu, and X. Liu, "Data Collection in Underwater Sensor Networks based on Mobile Edge Computing," *IEEE Access*, vol. 7, pp. 65357–65367, 2019, doi: 10.1109/ACCESS.2019.2918213.

[259] M. Liang, X. Su, X. Liu, and X. Zhang, "Intelligent ocean convergence platform based on IoT empowered with edge computing," *Journal of Internet Technology*, vol. 21, no. 1, pp. 235–244, 2020, doi: 10.3966/160792642020012101020.

[260] I. Carlucho, M. De Paula, S. Wang, Y. Petillot, and G. G. Acosta, "Adaptive low-level control of autonomous underwater vehicles using deep reinforcement learning," *Rob Auton Syst*, vol. 107, pp. 71–86, 2018, doi: 10.1016/j.robot.2018.05.016.

[261] L. Lin, H. Xie, and L. Shen, "Application of reinforcement learning to autonomous heading control for bionic underwater robots," *2009 IEEE International Conference on Robotics and Biomimetics, ROBIO 2009*, pp. 2486–2490, 2009, doi: 10.1109/ROBIO.2009.5420445.

[262] P. Bhopale, F. Kazi, and N. Singh, "Reinforcement Learning Based Obstacle Avoidance for Autonomous Underwater Vehicle," *Journal of Marine Science and Application*, vol. 18, no. 2, pp. 228–238, 2019, doi: 10.1007/s11804-019-00089-3.

[263] S. Sun *et al.*, “Underwater Image Enhancement With Reinforcement Learning,” *IEEE Journal of Oceanic Engineering*, vol. PP, pp. 1–13, 2022, doi: 10.1109/JOE.2022.3152519.

[264] J. Yang, M. Xi, J. Wen, Y. Li, and H. H. Song, “A digital twins enabled underwater intelligent internet vehicle path planning system via reinforcement learning and edge computing,” *Digital Communications and Networks*, 2022, doi: 10.1016/j.dcan.2022.05.005.

[265] J. Yan, Y. Gong, C. Chen, X. Luo, and X. Guan, “AUV-Aided Localization for Internet of Underwater Things: A Reinforcement-Learning-Based Method,” *IEEE Internet Things J*, vol. 7, no. 10, pp. 9728–9746, 2020, doi: 10.1109/JIOT.2020.2993012.

[266] F. Xu, F. Yang, C. Zhao, and S. Wu, “Deep Reinforcement Learning Based Joint Edge Resource Management in Maritime Network,” *China Communications*, vol. 17, no. 5, pp. 211–222, 2020, doi: 10.23919/JCC.2020.05.016.

[267] X. Hou, J. Wang, T. Bai, Y. Deng, Y. Ren, and L. Hanzo, “Environment-Aware AUV Trajectory Design and Resource Management for Multi-Tier Underwater Computing,” *IEEE Journal on Selected Areas in Communications*, vol. 41, no. 2, pp. 474–490, 2023, doi: 10.1109/JSAC.2022.3227103.

[268] B. J. A. Krose, “Learning from delayed rewards,” *Rob Auton Syst*, vol. 15, no. 4, pp. 233–235, 1995.

[269] C. J. C. H. Watkins and P. Dayan, “Q-learning,” *Mach Learn*, vol. 8, no. 3, pp. 279–292, 1992, doi: 10.1007/BF00992698.

[270] V. Mnih *et al.*, “Human-level control through deep reinforcement learning,” *Nature*, vol. 518, no. 7540, pp. 529–533, 2015, doi: 10.1038/nature14236.

[271] T. P. Lillicrap *et al.*, “Continuous control with deep reinforcement learning,” *4th International Conference on Learning Representations, ICLR 2016 - Conference Track Proceedings*, 2016.

[272] D. Silver, G. Lever, N. Heess, T. Degris, D. Wierstra, and M. Riedmiller, “Deterministic policy gradient algorithms,” *31st International Conference on Machine Learning, ICML 2014*, vol. 1, pp. 605–619, 2014.

[273] C. M. G. Gussen, P. S. R. Diniz, M. L. R. Campos, W. A. Martins, F. M. Costa, and J. N. Gois, “A Survey of Underwater Wireless Communication Technologies,” *Journal of Communication and Information Systems*, vol. 31, no. 1, pp. 242–255, 2016, doi: 10.14209/jcis.2016.22.

[274] Q. Wang, H. N. Dai, Q. Wang, M. K. Shukla, W. Zhang, and C. G. Soares, “On connectivity of UAV-assisted data acquisition for underwater internet of things,” *IEEE Internet Things J*, vol. 7, no. 6, pp. 5371–5385, 2020, doi: 10.1109/JIOT.2020.2979691.

[275] X. Hou *et al.*, "Machine-Learning-Aided Mission-Critical Internet of Underwater Things," *IEEE Netw*, vol. 35, no. 4, pp. 160–166, 2021, doi: 10.1109/MNET.011.2000684.

[276] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "A Survey on Mobile Edge Computing: The Communication Perspective," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 4, pp. 2322–2358, 2017, doi: 10.1109/COMST.2017.2745201.

[277] W. You, C. Dong, Q. Wu, Y. Qu, Y. Wu, and R. He, "Joint task scheduling, resource allocation, and UAV trajectory under clustering for FANETs," *China Communications*, vol. 19, no. 1, pp. 104–118, 2022, doi: 10.23919/JCC.2022.01.009.

[278] X. Hou *et al.*, "Machine-Learning-Aided Mission-Critical Internet of Underwater Things," *IEEE Netw*, vol. 35, no. 4, pp. 160–166, 2021, doi: 10.1109/MNET.011.2000684.

[279] R. J. Urick, *Principles of Underwater Sound*. McGraw-Hill, 1983.

[280] A. F. Harris and M. Zorzi, "Modeling the Underwater Acoustic Channel in ns2," *ACM International Conference Proceeding Series*, 2007, doi: 10.4108/nstools.2007.2024.

[281] L. Berkhovskikh and Y. Lysanov, "Fundamentals of Ocean Acoustics." New York: Springer, 1982.

[282] M. Stojanovic, "On the Relationship between Capacity and Distance in an Underwater Acoustic Communication Channel," *SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 11, no. 4, pp. 34–43, Oct. 2007, doi: 10.1145/1347364.1347373.

[283] H. Huang, D. Zhu, and F. Ding, "Dynamic task assignment and path planning for multi-auv system in variable ocean current environment," *Journal of Intelligent and Robotic Systems: Theory and Applications*, vol. 74, no. 3–4, pp. 999–1012, 2014, doi: 10.1007/s10846-013-9870-2.

[284] T. Q. Dinh, J. Tang, Q. D. La, and T. Q. S. Quek, "Offloading in Mobile Edge Computing: Task Allocation and Computational Frequency Scaling," *IEEE Transactions on Communications*, vol. 65, no. 8, pp. 3571–3584, 2017, doi: 10.1109/TCOMM.2017.2699660.