FCT Fundação para a Ciência e a Tecnologia

REPÚBLICA PORTUGUESA

PORTUGAL 2020

NORTE2020
PROGRAMA OPERACIONAL REGIONAL DO NORTE

UNIÃO EUROPEIA
Fundo Social Europeu

The Internet of People approach to Road Safety and Vulnerable Road Users in Smart Cities

Bruno Filipe Martins Fernandes

UMinho | 2021

Universidade do Minho
Escola de Engenharia

Bruno Filipe Martins Fernandes

The Internet of People approach to Road Safety and Vulnerable Road Users in Smart Cities

January, 2021

Universidade do Minho
Escola de Engenharia

Bruno Filipe Martins Fernandes

The Internet of People approach to Road Safety and Vulnerable Road Users in Smart Cities

Doctoral Thesis

Doctoral Program in Informatics

Work supervised by

Cesar Analide de Freitas e Silva da Costa Rodrigues

José Carlos Ferreira Maia Neves

January, 2021

COPYRIGHT AND TERMS OF USE OF THIS WORK BY A THIRD PARTY

Acknowledgements

Por ti e para ti, Tia.
Por ti e para ti, Avó.

> *In the 17th Chapter of St Luke it is written: "the Kingdom of God is within man". Not one man nor a group of men, but in all men! In you! You, the people have the power - the power to create machines. The power to create happiness! You, the people, have the power to make this life free and beautiful, to make this life a wonderful adventure.*
>
> — Sir Charles Spencer Chaplin

Now that the time has come for me to acknowledge all those that, somehow, supported me during this journey, I hope not to get short on words. I must start by expressing my gratitude to Professor Cesar Analide and Professor José Maia Neves, who, more than supervisors, are extraordinary friends. I will not forget Professor Paulo Novais. Thank you all for your guidance, comments, and challenges.

I am also deeply grateful to all my friends at ISLab. Thank you Ana, António, Dalila, Fábio, Filipe, Francisco, Leandro, Leonardo, Luís, Marco, Pedro, and Ricardo. Thank you all for your support and friendship. I must also thank Hector for his friendship and for helping me keep this work on the right path.

My gratitude is also expressed to the ALGORITMI Centre and FCT for providing me all the means to complete this journey, including a national PhD scholarship (with reference SFRH/BD/130125/2017).

On a personal level, I must express my uttermost gratitude to my friends and all my family. To my grandparents, my aunts, my uncles, and my cousins, thank you for always being there for me, especially after this difficult phase we all went through. I am deeply grateful to my parents, Sérgio and Lúcia. To my brother and sister, Paulo and Mariana, you know you are the source of my strength.

Finally, to the one that has been with me for many years now, Ana Bela, thank you for cherishing me, for your support and comprehension. I know that without you I would not be here. We have both made it! Since high school kids to this day - what an amazing journey.

I could not fail to thank the One who has been guiding me throughout my all life. I know that lately I have been further away but I promise I will meet You again. Thank You for making this journey with me.

Thank you all.

STATEMENT OF INTEGRITY

I hereby declare having conducted this academic work with integrity. I confirm that I have not used plagiarism or any form of undue use of information or falsification of results along the process leading to its elaboration.

I further declare that I have fully acknowledged the Code of Ethical Conduct of University of Minho.

Braga,

Assinado por : **BRUNO FILIPE MARTINS FERNANDES**
Num. de Identificação Civil: BI13945086
Data: 2021.03.05 13:42:25 Hora padrão de GMT
CHAVE MÓVEL

(Bruno Filipe Martins Fernandes)

Resumo

A Internet das Pessoas para benefício da Segurança Rodoviária e dos Utilizadores Vulneráveis da Estrada em Cidades Inteligentes

Desde o início dos tempos, tudo temos feito para melhorar a nossa qualidade de vida. A nossa caminhada tem sido incrível, mas está longe de ter terminado. Infelizmente, um problema em destaque na nossa sociedade prende-se com os acidentes rodoviários, a principal causa de morte de crianças e jovens adultos. Na estrada existem grupos de pessoas mais vulneráveis que outros, sendo estes descritos como Utilizadores Vulneráveis da Estrada (UVE). Na sua essência, este grupo caracteriza-se por pedestres, ciclistas e motociclistas. Estima-se que, actualmente, mais de metade dos mortos na estrada sejam UVE. Estes números são consequência de políticas que prioritizam veículos em vez de pessoas.

Reduzir acidentes entre UVE permanece um desafio. Nenhuma solução monolítica conseguirá resolver todos os problemas. Pelo contrário, a expectativa é que serão várias pequenas contribuições a fazer a diferença. Desta forma, a nossa contribuição é baseada na percepção de pessoas como actores activos e proactivos de uma cidade inteligente. Desta premissa surge o 3-C, o qual se foca em sensorizar a Cidade, a Comunidade e o Cidadão. O 3-C, que define as directrizes e objectivos deste trabalho, quando complementado com a metodologia de investigação adoptada, permitiu a organização de uma sequência lógica de procedimentos replicáveis, precisos e parcimoniosos para a concepção de conhecimento científico.

Esta tese de doutoramento apresenta, entre outros, um arquétipo, *SafeCity*, que agrega várias soluções que tiveram origem no processo de investigação realizado. Este arquétipo apresenta métodos e modelos disruptivos para inferência de aborrecimento e personalidade, para concepção de mapas de sentimentos e para previsão do fluxo de tráfego rodoviário, entre outros. O objectivo é o de permitir que as pessoas sejam actores participativos e que, assim, tenham acesso a um conjunto de informação sobre o estado actual e futuro do ambiente onde estão inseridas, permitindo que os UVE adaptem o seu comportamento perante cada situação, hora e local de uma cidade. Arquétipos adicionais incluem uma blockchain sobre um sistema multi-agente para gestão de grandes quantidades de dados estruturados.

Palavras-chave: Aprendizagem Automática, Cidades Inteligentes, Inteligência Ambiente, Internet das Pessoas, Utilizadores Vulneráveis da Estrada.

Abstract

The Internet of People approach to Road Safety and Vulnerable Road Users in Smart Cities

Since the dawn of times, the human being has always endeavored to improve his quality of life. Our journey has been amazing but it has not yet finished. Unfortunately, an issue that has been rising to prominence is related to road traffic accidents, the leading cause of death for children and young adults. On the road, there are groups of people more vulnerable than others to injuries and accidents. Those are known as Vulnerable Road Users (VRUs), being defined as pedestrians, cyclists, and motorcyclists. It is estimated that, currently, more than half of all road traffic deaths are amongst VRUs. These numbers are a consequence of policies that prioritize vehicles instead of people.

Reduce injuries among VRUs remains a big challenge. No monolithic solution will, singularly, settle out all problems. Instead, the expectation is that multiple small contributions may help answering important challenges raised by this group of road users. Hence, our contribution is grounded on the perception of people as active, reactive, and proactive actors of a Smart City's ecosystem. From this premise emerges the 3-C, i.e., City, Crowd, and Citizen sensing. The 3-C, which defines the guidelines and goals of this work, when complemented with the adopted research methodology, allowed the organization of a logical sequence of replicable, precise, and parsimonious procedures to build scientific knowledge.

This thesis presents, among others, an archetype platform, SafeCity, which aggregates multiple solution specimens produced as a result of the conducted research. SafeCity makes use of multiple disruptive methods and models for on-device boredom inference, to assess one's personality, to conceive emotional maps, and to provide accurate multi-step traffic flow forecasts, among many others. The goal is to allow people to be participative actors, thus having access to a set of information about the current and future status of the environment where they stand, allowing VRUs to adapt their behavior upon each situation, hour, and place of a city. Additional archetypes include a multi-agent system blockchain for the management of large amounts of data of different types and sources.

Keywords: Ambient Intelligence, Internet of People, Machine Learning, Smart Cities, Vulnerable Road Users.

Contents

List of Figures

List of Tables

Listings

List of Algorithms

Acronyms

ADAS Advanced Driver Assistance System

AI Artificial Intelligence

AmI Ambient Intelligence

ANNs Artificial Neural Networks

AP Access Point

ARIMA AutoRegressive Integrated Moving Average

ARIMAX AutoRegressive Integrated Moving Average with Explanatory Variable

ARL Association Rules Learning

ASAP Adjective Selection to Assess Personality

AUC Area Under the Curve

BLE Bluetooth Low Energy

BPTT Backpropagation Through Time

CNNs Convolutional Neural Networks

CRISP-DM Cross-Industry Standard Process for Data Mining

CRQ-4 Crossing Roads Four-Item-Questionnaire

DA Data Augmentation

DL Deep Learning

EML Extreme Machine Learning

GDPR General Data Protection Regulation

GRUs Gated Recurrent Units

GUI Graphical User Interface

HTTP Hypertext Transfer Protocol

IDE Integrated Development Environment

IEEE Institute of Electrical and Electronics Engineers

IoP Internet of People

IoT Internet of Things

ISLab Synthetic Intelligence Laboratory

ITS Intelligent Transportation System

JSON JavaScript Object Notation

KRR Knowledge Representation and Reasoning

LSTMs Long Short-Term Memory Networks

MAC Media Access Control address

MAE Mean Absolute Error

MAS Multi-Agent Systems

ML Machine Learning

MLPs Multilayer Perceptrons

MQTT Message Queuing Telemetry Transport

MSE Mean Squared Error

OUI Organizationally Unique Identifier

P2V Pedestrian-to-Vehicle

PMML Predictive Model Markup Language

PoC Proof-of-Confidence

PoW Proof-of-Work

ReLU Rectified Linear Unit

RMSE Root Mean Squared Error

RNNs Recurrent Neural Networks

RRS Road Restraint System

RSSI Received Signal Strength Indicator

SAIQ-3 Self-Assessment and Identity Three-Item-Questionnaire

tanh Hyperbolic Tangent

TMQ-5 Travel Motivation Five-Item-Questionnaire

URI Uniform Resource Identifier

UV Ultraviolet

V2I Vehicle-to-Infrastructure

V2V Vehicle-to-Vehicle

V2X Vehicle-to-X

VANETs Vehicular Ad-hoc Networks

VRUs Vulnerable Road Users

WHO World Health Organization

Chapter 1

Introduction

We all desire a better life for ourselves, for our community, and for all beings that inhabit our planet. Indeed, since the dawn of times, the human being has always endeavored to improve his quality of life. From the discovery of fire and the invention of the wheel to walking on the moon or the world's globalization. The journey has been amazing. But it has not yet finished. We must keep striving to improve our world. Our mission is to take advantage of the amazing technological revolution the world is experiencing to improve our quality of life, reduce and eliminate diseases, and ensure a prosperous future for our planet and future generations. The work here addressed aims to contribute, even if on a small scale, to this endeavor.

1.1 Research Domains

During these last decades, migration towards urban centers has significantly increased worldwide. It is estimated that in 2018 more than 55% of the world's population lived in urban settlements, being predicted that one in every three people will live in cities with at least half a million inhabitants by 2030 [1]. In fact, the number of megacities worldwide, i.e., cities with more than 10 million inhabitants, is expected to rise from 33, in 2018, to 43, in 2030 [1]. This evidence, coupled with the fascinating technological evolution we are now experiencing, makes the foundations on where this thesis rests, with the next lines describing and characterizing the main research domains at the basis of this work.

1.1.1 Road Safety and Vulnerable Road Users

Road safety has been rising to prominence in today's society. For instance, according to the World Health Organization (WHO), an agency of the United Nations responsible for global public health, the number of deaths related to road traffic accidents climbs every year, reaching 1.35 million deaths in 2016 [2]. Still according to WHO, road traffic accidents are the eighth leading cause of death for all age groups

(Figure 1.1), being the number one cause of death for children and young adults aged between 15 and 29 years. In fact, more people die as result of road accidents than from AIDS or tuberculosis [2].

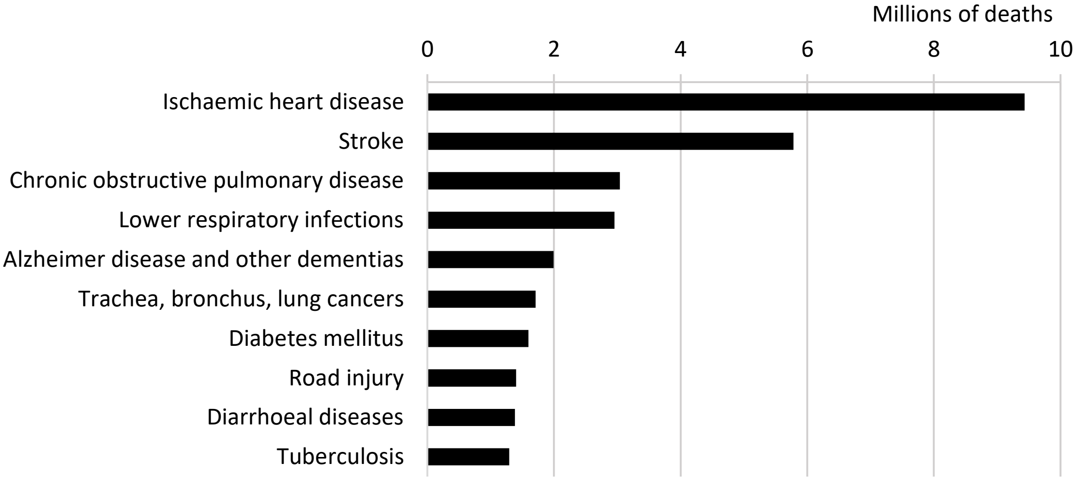


Figure 1.1: Top 10 global causes of death in 2016. Data obtained from [3].

Inequalities in the number of road traffic deaths are also visible in different regions of the globe. Figure 1.2 depicts the number of road traffic deaths per 100 000 population in different regions. Africa and South-East Asia are the regions with a higher ratio of traffic deaths. In contrast, Europe and America depict a lower number of deaths per 100 000 inhabitants. Data also show that high-income countries have a significantly lower ratio of deaths when compared to middle and low-income countries [2].

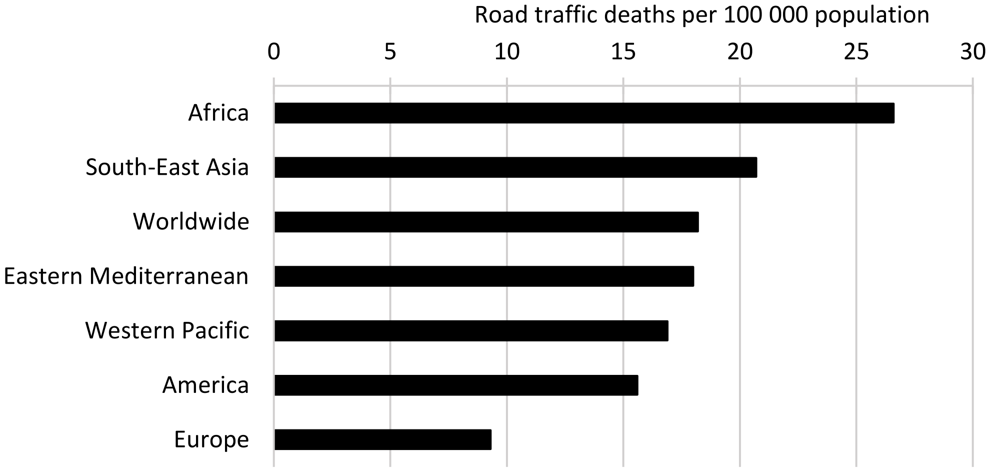


Figure 1.2: Ratio of road traffic deaths per 100 000 population in WHO regions in 2016. Data obtained from [2].

Road traffic injuries and deaths are a major health problem. In this context, road safety refers to a set of methods and measures adopted to prevent and reduce injuries in road users. Examples may include the dissemination of awareness campaigns, the improvement of physical features of roads such as road pavements [4] and road barriers [5], or the inclusion of sensorization and Artificial Intelligence (AI) to model and improve traffic dynamics [6–8], among many others.

The group of road users includes multiple distinct actors such as pedestrians, cyclists, passengers, drivers, and even public transport users. However, on the road, there are groups of people more vulnerable than others to injuries and accidents. Those are known as Vulnerable Road Users (VRUs), being defined by the European Parliament as "*non-motorized road users, such as pedestrians and cyclists as well as motorcyclists, and persons with disabilities or reduced mobility and orientation*" [9]. Their vulnerability may arise from several directions including the lack of external protection, age, physical and mobility impairments, and visual and/or hearing disabilities.

Currently, it is estimated that more than half of all road traffic deaths are amongst pedestrians, cyclists, and motorcyclists [2]. Indeed, VRUs are still often neglected by road traffic agencies of many countries. It is usual for police forces to consider accidents with such users as non-traffic crashes, which negatively influences existing records and estimations [10]. Worldwide, pedestrians and cyclists represent 26% of all road traffic deaths. Motorized two and three-wheelers represent 28% [2]. These numbers are consequence of policies that prioritize vehicles instead of people, that ignore VRUs when planning and designing roads, and that devalues the importance of crossings for pedestrians and separate lanes for cyclists. In the European Union, as depicted in Figure 1.3, in 2017, 21% of all people killed on roads were pedestrians and 26% were 2-wheelers, which includes motorcyclists (15%), cyclists (8%) and moped users (3%) [11].

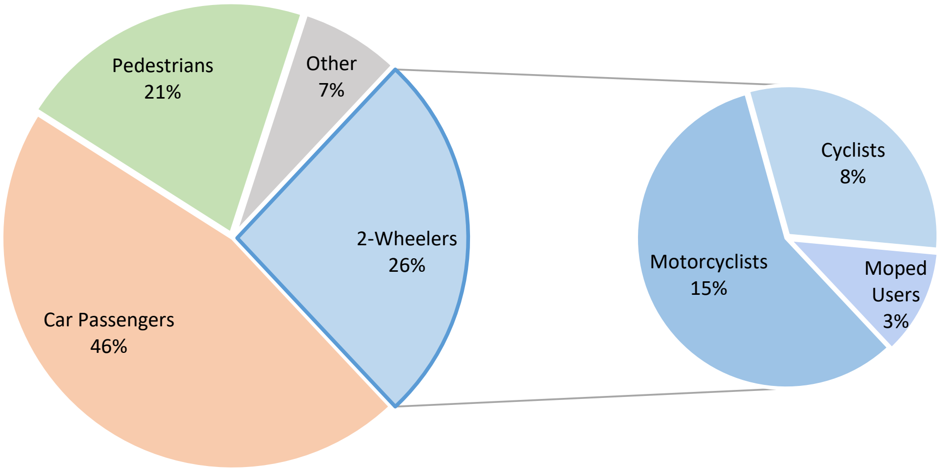


Figure 1.3: Percentage of road fatalities in the European Union by transport mode in 2017, considering all road types. Data obtained from [11].

In the European Union, most road traffic deaths in 2017 occurred on rural roads (54%), with 38% happening in urban areas and only 8% in motorways. In addition, while in 2010 only 18% of all road fatalities concerned elderly people, that number increased to 28% in 2017. These numbers are a clear indication of the increased vulnerability of those over 65 years old as pedestrians, especially in urban areas [11]. In fact, considering only urban roads, the percentage of pedestrians' fatalities nearly doubles (from 21% to 40%), with the percentage of cyclists' deaths increasing by 50%. On the other hand, car passenger fatalities reduce from 46% to 26% (Figure 1.4).

A temporal analysis of crash reports showed that pedestrian deaths are more likely to happen on holiday periods and on the last two months of the year, while most cyclists' fatalities occur during the summer or

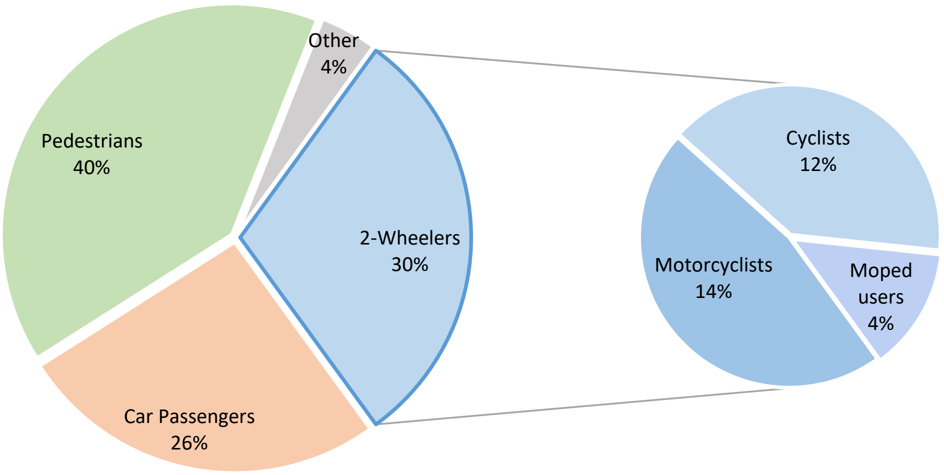


Figure 1.4: Percentage of fatalities in urban roads in the European Union by transport mode in 2017. Data obtained from [11].

early fall [12]. For cyclists, the risk of crashes also increases in urban environments [13]. The presence of retail stores, touristic places, and environmental factors, such as the weather, also increase the risk of vehicle and bicycle collisions [14]. Large urban areas, either due to its higher density of vehicles, lower number of sidewalks and pedestrian crossings, inconstant or non-modeled traffic dynamics, or higher values of pollution, pose increased risks to VRUs. In fact, on an urban context, the number of VRUs' deaths is substantially higher than the number of deaths of vehicle passengers. To reduce this number, one is required to pay attention to the needs of VRUs, who have been neglected by transport and planning policies [2]. In addition, measures that induce walking and cycling also promote physical exercise, the reduction of air pollution and greenhouse gas emissions, and lead to better road traffic dynamics.

Reduce VRUs' injuries remains a big challenge. No singular approach will solve the problem. Instead, the expectation is that multiple small contributions may help answering important challenges raised by this group of road users. Hence, we are required to have safer roads and infrastructures, to advance vehicles, to conceive passive and active devices and models, to have stronger regulations, better awareness campaigns, and we must consider VRUs as active and participative actors of the road ecosystem.

1.1.2 Internet of Things and the Internet of People

Internet of Things (IoT) is a paradigm being increasingly supported by different stakeholders. It is considered as the next big step in the evolution of the Internet [15]. The idea is to connect multiple heterogeneous devices, or ”things”, through wireless connections and unique identifiers to create a pervasive environment where a person can interact, at any time, with the digital and the physical world. IoT opens exciting opportunities but also new questions on the interaction between the citizen and the machine. Commonly known objects such as clothes, food packing, shoes, or bicycles, among many others, will be equipped with some level of internet-addressable feature, offering context awareness and communication features [16]. The large-scale implementation of IoT devices promises to transform our lives. For consumers, new products like home automation components are sending us toward smart homes, offering, for example, security and energetic efficiency. On the other hand, personal IoT devices like wearable fitness and health monitoring devices are transforming the way healthcare services are delivered to the citizen [17].

The term IoT projects the vision of a global infrastructure of networked physical objects, enabling anyplace, anytime connectivity for anything. IoT refers to a world where physical objects all interact with each other at the same space and time [18]. IoT is a recent paradigm in which "things" (objects) of our everyday life are able to communicate with one another and with the users, becoming an integral part of the internet. Thus, this concept aims at making the internet even more immersive and pervasive [19]. However, IoT is a term lacking a unique definition [16]. The term was first used in 1999, by Kevin Ashton, where the author claimed that "*computers - and, therefore, the Internet - are almost wholly dependent on human beings for information*" but now, in the twenty-first century "*because of the Internet of Things, computers can sense things for themselves*" [20]. With this in mind, a definition was proposed by Madakam, Ramaswamy and Tripathi (2015), where they define IoT as "*an open and comprehensive network of intelligent objects that have the capacity to auto-organize, share information, data and resources, reacting and acting in face of situations and changes in the environment*" (p. 165) [21].

IoT has been presenting new solutions for the benefit of the human being. However, the focus goes towards the thing, leaving aside people as an active and participative element. In its essence, adding people to the IoT leads towards the Internet of People (IoP). Being the IoP an emerging paradigm, there is a clear lack of hypothesizing meanings and properties. For instance, one study, performed in 2015, proposed a Humanized-IoT [22], where, according to the authors' understanding, the IoP is just a node of this humanized network. Such work focused on studying the extent of Fiske's psychological work, which identifies four common forms of sociability used by people in their relationships. Indeed, our vision of an IoP recognizes the existence of an ecosystem of people and things. Hence, studies like the one performed in [22] are useful to define hierarchies and rules among emerging ecosystems.

In 2011, Hernández-Muñoz et al. proposed three pillars to define the future internet: the IoT, the IoP, and the Internet of Services [23]. They took the opportunity to come up with an interesting definition for the IoP, i.e., "*envisaged as people becoming part of ubiquitous intelligent networks having the potential to seamlessly connect, interact and exchange information about themselves and their social context and environment*". In this context, wearable devices should be seen as crucial facilitators of the IoP. In fact, another study, performed by Miranda et al. (2015), describes an infrastructure that uses smartphones as a key to improve the integration of people with the IoT [24].

Our understanding of the IoP is of a dynamic network where things and people communicate and understand each other; where everyone and everything can sense the other and the world, and act on such knowledge and information, aiming to enhance people's quality of life. In order to bring people to this global network, and to make them an active, reactive, and proactive element, one is required to promote the citizen sensor, which consists of providing people with sensing capacities [25]. The goal is to have an improved efficiency and economic benefit; is to improve people's life; is to enhance quality, performance, and interactivity of urban services; to reduce costs and resource consumption; and to enhance road safety, with the longstanding goal being to prevent injuries and save lives.

1.1.3 Smart Cities

At urban centers, population growth comes with its challenges. Crowding, traffic congestion, bureaucracy, crime, sprawl, waste, and energy consumption are some of the major challenges faced by cities in the immediate future. In our days, a tremendous technological evolution is felt at all levels: devices, protocols, hardware, and software, just to name a few. Cities must act and take advantage of technology to evolve, expand, and create means to improve the quality of life of their citizens. This is what is expected from a Smart City [16].

There is no clear and definitive definition for "Smart City", but, instead, a broad range of very similar ones. Harrison et al. (2010) noted that a "*Smarter City continues the long-standing practice of improving the operational efficiency and quality of life of a city by building on advances in IT*" (p. 2) [26]. Later, Roscia, Longo, and Lazaroiu (2013) defined Smart City as the "*use of planned and wise of the human and natural resources, properly managed and integrated through the various Information and Communications Technologies already available, allows for the creation of an ecosystem that can be used of resources and to provide integrated and more intelligent systems*" (p. 371) [27]. Another definition is the one presented by Anthopoulos and Fitsilis (2014) as "*ICT-based infrastructure and services environment that enhance city's intelligence, quality of life and other attributes (i.e., environment, entrepreneurship, education, culture, transportation, etc.)*" (p. 190) [28]. More definitions may be found in the works of Allwinkle and Cruickshank (2011) [29], Hollands (2008) [30], and Nam and Pardo (2011) [31].

A candidate definition, here postulated, envisages a Smart City as having the ability to reason upon the knowledge acquired through data gathered by sensorization, with a focus on improving the quality of life at urban centers, considering sustainability and safety principles. By sustainability, we denote social, economic, and environmental factors [16].

From a computer science perspective, several authors have proposed different dimensions to classify the smartness of a city. Giffinger et al. (2007) created a Smart City ranking based on certain urban characteristics. They identified six categories in which a city's smartness could manifest itself: economy, people, governance, mobility, environment, and living [32]. Nam and Pardo (2011) introduced a model based on three dimensions: technology, people, and institutions [31]. On the other hand, Chourabi et al. (2012) have identified eight dimensions: management and organization, technology, policy, governance, people and communities, economy, built infrastructures, and the natural environment [33]. Literature also shows that it is possible to draw an artificial line between cities that opt for a highly technical infrastructure-intensive approach (Rio de Janeiro, Barcelona, or London) and cities that opt for a more citizen-centric approach (Manchester, Copenhagen, or Amsterdam) [16, 34].

Over the years, numerous attempts have been made to develop cities into smart ones. The process of making a city smart should focus on several areas and demands huge efforts. A Smart City should transverse many domains and make improvements in all of them. It should make use of sensors and information systems to gather data and, with that, create information and knowledge that is useful to its citizens. In fact, some of the main areas of intervention in Smart Cities are as follows (Figure 1.5):

- **Smart Homes**, or domotics, assume that devices and appliances are aware of the existence of others, are capable of communicating among them, and have an independent existence. This area of intervention deals with the technological enrichment of the living environment in order to offer support to inhabitants [15]. Smart homes should not only be able to react to changes in the environment but also take into account the preferences of the residents and their characteristics;

- **e-Government and e-Education** aim for more direct and convenient citizen access to government and education. While the first follows a business model based on statutes and laws, providing citizens and firms access to government data and services [35], the latter focuses on improving learning outcomes by allowing full access to education, using smart applications and analytics to support teaching [36];

- **Sustainability** is a subject of concern for the assurance of steadiness, viability, and use of a system. A common definition concerns an equilibrium on social, economic, and environmental factors. Assessing sustainability and sharing those results with the community, allows the creation of citizen awareness [37]. Indeed, it has been proved that when people are aware of the consequences of their actions they tend to adjust their behavior, thus promoting sustainability [38];

- **Water and Waste management** assume the capacity of a city to successfully manage its resources, in particular, reduce the waste of water and, at the same time, improve its quality [39]. Waste management is also a primary issue in many modern cities. New solutions may result in significant savings, and economical and ecological benefits [19];

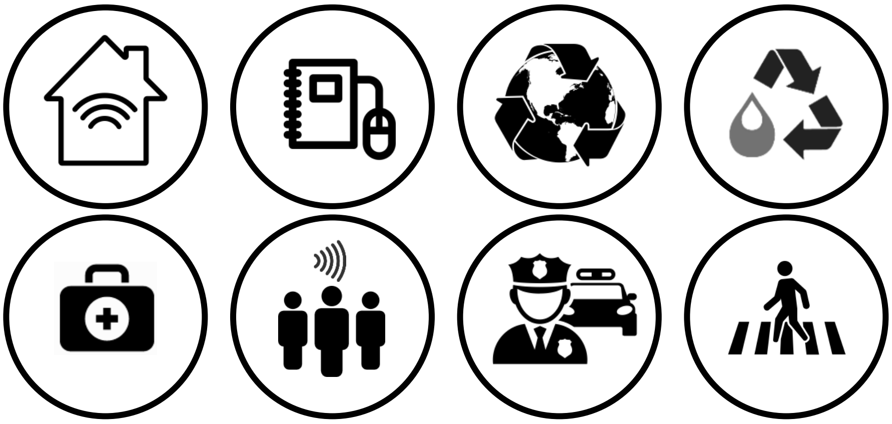


Figure 1.5: Some of the main areas of intervention in Smart Cities. From the top left to the bottom right: smart homes, e-Government and e-Education, sustainability, water and waste management, smart healthcare, citizen sensing, and public and road safety.

- **Smart Healthcare** is one of the major topics of concern in Smart Cities. Smart systems in this area comprise clinical care, remote monitoring, early intervention and diagnosis, prevention, and emergency responses, with smart devices being used by people to control diseases and their evolution [39]. It also comprises interoperability along different health services and institutions [40];

- **Citizen sensing** consists in giving people the ability to sense themselves, the others, and the environment where they stand. For that, smartphones and wearable devices are of the utmost importance since such devices, besides bringing everyone ”online”, also contain embedded sensors such as GPS, accelerometers, light sensors, or barometers, just to name a few. Enabling the citizen sensor is not only important for the IoP but also for Smart Cities, allowing people to become active, reactive, and proactive actors of the city's ecosystem [41];

- **Public safety** consists in the use of IoT devices on the benefit of law enforcement and public safety [17]. IoT devices, and the data they produce, can be used as an effective tool to fight and prevent crimes. Public safety agencies could gather and analyze data regarding the weather, traffic flow, security breaches, hazardous materials, or disasters, and provide actionable information to the citizens [36]. This topic may, however, raise legal, security, privacy, and ethical issues;

- **Road safety** has become a major issue of concern not only for car manufacturers but also for governments. The importance of this area of intervention is clear when the goal is to help save lives by preventing accidents and injuries. Indeed, road safety is a very comprehensive topic, ranging from measuring traffic congestion to increase the safety of motorcyclists or pedestrians. This work focuses on an important sub-domain of road safety, i.e, it focuses the VRUs' problem.

1.1.4 Artificial Intelligence

AI emerged during the last century as a way to emulate human intelligence in machines. For a long time, it remained as one of the most controversial research fields in the computer science domain as its main goal was to give machines the ability to reason at the same level as we, humans, do. However, the way the human being reasons and makes his decisions is far from being fully understood. In addition, the ability to reason may express itself in several different ways, including, but not limited to, self-awareness, communication, planning, or decision-making. For instance, the Cambridge dictionary defines intelligence as ”*the ability to learn and understand things*” while Merriam-Webster defines it as ”*the ability to learn or understand or to deal with new or trying situations*”. Over time, the community understood that AI, per se, encompasses several distinct domains. Hence, the following lines describe important sub-domains of AI that were considered in this PhD thesis.

Machine Learning

During these last years, Machine Learning (ML) has risen to prominence. This sub-domain of AI focuses on giving machines the ability to classify, predict, or decide, without being explicitly programmed to do so, being able to address complex non-deterministic problems. For that, it makes use of large amounts of data to create models that are able to adapt to unseen data and to new situations. Several distinct stakeholders make use of ML. Examples include the use of ML for medical imaging [42], boredom detection in smartphones [43], and quantum physics [44].

Three distinct learning paradigms can be recognized: supervised, unsupervised, and reinforcement learning. Supervised learning refers to the creation of ML models using data where the desired output is known. This desired output is called the label, or target, while the remaining features of the dataset make the input. Each row of the dataset, also known as observation, consists of a training example that is usually represented by an array or vector. The goal is to train ML models in order to learn a function that is able to map tuples of input-output. The trained model is then used, in the real world, to classify or predict the output of unseen input. Some common supervised ML methods include Decision Trees, Random Forests, Support Vector Machines, and Artificial Neural Networks (ANNs). On the other hand, unsupervised learning takes a set of data where only the input is known. The goal is typically to find structure in data and perform clustering. In this learning paradigm, dataset features are used to categorize similar observations, allowing the ability to cluster new data based on similarities to the training set. Some common unsupervised ML methods include Association Rules Learning, k-Means, and k-Nearest Neighbors. Finally, reinforcement learning uses a method based on rewards/penalties to train models. In this learning paradigm, an agent learns which actions to perform based on environmental information in order to maximize its reward or minimize its penalties. Examples include Q-learning, SARSA, and Deep Q-learning.

Within the ML field, Deep Learning (DL) has been gaining an increased importance. Due to the evolution of technology and hardware components, the community has been able to successfully apply DL models to solve complex problems and significantly improve previous solutions, including accurate traffic flow forecasting [45], autonomous driving [46], or pneumonia detection on chest X-rays [47], just to name few. In its core, DL consists of ANNs with one, or more, hidden layers. ANNs, in turn, mimic the structure of the human brain, with neurons as computational units and weighted synapses as communication routes between neurons. In essence, neural networks can be defined as a directed graph with neurons as nodes and synapses as edges. DL models can follow a supervised (Multilayer Perceptrons (MLPs), Convolutional Neural Networks (CNNs) or Long Short-Term Memory Networks (LSTMs), for example) or an unsupervised (Autoencoders and Generative Modeling) approach.

Affective Computing

Affective Computing is a sub-domain of AI that deals with emotions, mood, and the personality of subjects in computational contexts. In essence, it can refer to the capture of such characteristics through computational means or to the humanization of computational systems. The term was introduced in 1995 and coined by R. Picard [48], spanning across the fields of computer science and psychology. Interestingly, R. Picard noted that neurological studies point out that the decision-making process without any emotion may be as impaired as one with excessive emotion. The main idea is that for computers to make intelligent choices they must express emotions. Contrary to Mr. Spock, from Star Trek, who considers "*foolish* *emotions a constant irritant*", it is our emotions and personality that makes us one of a kind.

People react differently upon the same situations. This behavioral diversity may be due to one's experience, knowledge, or personality. Several studies have already established a relationship between a person's personality and aggressive reactions [49], work performance [50], or infidelity [51], just to name a few. Semantically, personality may be defined as a set of characteristics that refer to individual differences in ways of thinking, feeling, and behaving [52]. Hence, there has always been great interest in model, or quantify, a person's personality using either qualitative or quantitative metrics. However, psychological assessments are usually long and expensive. Therefore, new methods to assess personality are now emerging, i.e., methods based on audio and video [53], text [54], or even social media feeds [55]. In fact, the potential applications of affective computing technology are as vast as facial expression detection [56], voice emotion analysis [57], boredom detection [43], personality assessment [58], and emotions detection through text [59], among many others. Goals may include increasing the level of user engagement or improve the decision-making process of virtual agents.

Multi-agent Systems

Multi-Agent Systems (MAS) are computational systems that are composed of multiple intelligent agents, being especially suitable for the development of complex and dynamic systems [60]. Intelligent agent, in turn, refers to an autonomous piece of software that aims to attain its goals or the goals of the community it inhabits. In 1995, Wooldridge and Jennings proposed a weaker and a stronger notion for the term "agent". The first, more conventional, assumes that agents are (1) autonomous and operate without the direct intervention of humans, (2) are able to interact with others, (3) are reactive to changes in the environment, and (4) are able to exhibit goal-oriented behavior. On the other hand, the stronger notion assumes that agents, besides having the properties described above, are conceived using concepts usually applied to humans, i.e., agents may emulate emotions, beliefs, or knowledge [61].

Russel and Norvig (2003) further discuss the existence of five types of agents: *reflex agents*, which act solely based on their current perception of the environment, requiring the environment to be fully observable; *model-based agents*, that build an internal representation of the environment, or some parts of it, and use it to act; *goal-based agents*, that further expand the capacities of model-based ones with information describing desirable situations; *utility-based agents*, which map states to measures of utility, i.e., it further expand the notion of goal-based agents by having a strong measure of how to behave in order to achieve a particular goal; and *learning agents*, which are able to operate in unknown environments through means of a feedback system that allows agents to modify their behavior in order to perform better in the future [62].

MAS are defined by the agents that compose the system and the environment where the agents stand. In this thesis, they are seen from a computer science perspective, i.e., the software agents that compose the MAS are autonomous and work without the need of a human being, and are able to make decisions in order to achieve a common goal based on data that is shared among all peers that compose the system. Actions to be performed depend on the state of the peers and of the environment.

Ambient Intelligence

Ambient Intelligence (AmI) is a multi-disciplinary domain that deals with the development of intelligent systems, i.e., it consists of sensors and actuators incorporated into an environment, which then becomes sensitive and responsive to the presence and needs of a human being in a non-intrusive way [63]. AmI is built upon the fields of pervasive and ubiquitous computing, context awareness, and human-computer interaction [64].

The evolution of technology was a key factor in the appearance of AmI. In the last years, we have evolved from large hardware devices to smaller, connected, and more integrated processors and devices, which originated the IoT. This allowed the incorporation of computational power in familiar objects such as vehicles, fridges, bus stops, and even windows and lighting systems. In fact, sensorization is a key factor of intelligent systems, together with reasoning and acting [65]. This last factor consists in perform some action in the environment based on the output of reasoning models conceived using ML, Rule-based Systems, or Case-based Reasoning. Sensorization, in turn, is what enables AmI. Since AmI is built for real-world environments, the effective use of sensors is crucial to maximize the number and quality of the data streams that describe the environment.

In a Smart City, the goal is to invoke large scale AmI by creating several distinct sensorization levels [66]. As depicted in Figure 1.6, on the first level one can find public APIs that can be used to extract relevant information such as road traffic conditions, the weather, and pollution levels. On a second level, there are the devices that can be used by people in their daily routine, being crucial enablers of the citizen sensor. This level also includes straightforward user data such as gender, age, weight, impairments, blood pressure, consumption habits, personality, and social characteristics. The third level comprises the city in itself and methods for extracting actionable data from the environment.

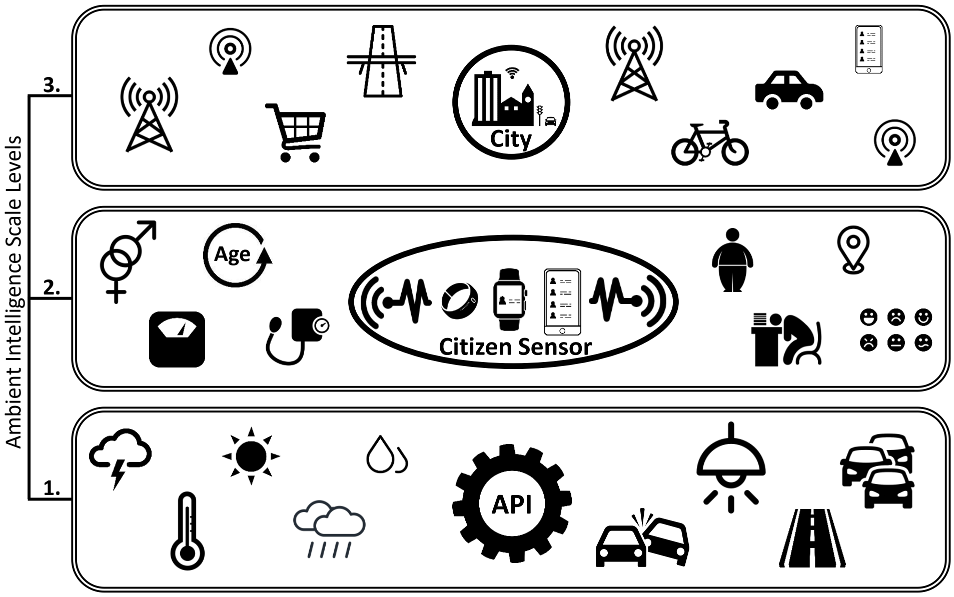


Figure 1.6: Sensorization levels in a Smart City.

Blockchain

The exponential growth of cryptocurrencies, with Bitcoin in the lead, was what put the spotlight on blockchains. Indeed, more than a paradigm, blockchain is a data structure, a technology, and a protocol, all at the same time. It is unclear who first coined the term, although it is likely to have arisen from the paper that originated Bitcoin [67], authored by Satoshi Nakamoto, where terms such as ”*block in the chain*” or ”*chain of blocks*” are used. As the name implies, a blockchain is essentially a chain of blocks that contain data, with the first block of the chain being called the genesis block. It is interesting to note that what makes a blockchain trustworthy is the fact that nobody in the network trusts anybody. Thus, the higher the number of nodes in the network the greater the security and the trustworthiness of the blockchain.

Empowered by cryptocurrencies, blockchains have been widely used to store transactions of currency, allowing nodes to store, validate, and understand from where and to where each amount of currency is being moved. However, soon researchers understood the applicability of such technology in different domains. One example is the effective development of smart contracts by Ethereum [68]. In fact, new approaches to blockchains have emerged and are, indeed, still emerging. From product traceability [69] to food safety and origin in supply chain problems [70], blockchain and smart contracts provide means to guarantee provenience and compliance rules between all the involved parties. In the field of medicine, blockchain examples can be found, essentially, in secure health records sharing [71–73]. In the field of the IoT, since such networks are data-centric and device-centric by nature, blockchain is particularly suited to deal with data correctness and to handle malicious behaviors of devices [74]. Security in data exchange is also important within IoT networks, with blockchain being used to provide decentralized authentication systems for IoT devices [75]. Indeed, it has been corroborated that blockchains have now moved beyond financial services and smart financial contracts [74, 76].

1.2 Related Work and Projects in the VRUs’ domain

Road safety has become a major issue of concern for several stakeholders. The European Union, for example, has adopted the ”Vision Zero and Safe System approach” to eliminate deaths and severe injuries on European roads by 2030. Hence, it is natural to find several distinct studies and projects being developed in the road safety domain. The next lines describe the main academic literature and industrial projects that give special importance to the VRUs’ problem.

1.2.1 Academic Literature

Several studies have already engaged in connecting the domains of road safety with the IoT. One example is the one of Mitton and Rivano (2014) who proposed a theoretical model to compute data gathered through mobile sensors in bicycles. The goal was to ”*make the bike easier and make people want to ride bicycles more often*” (p. 3) [77]. Another work, performed by Miyaji (2015), used data about traffic incidents to clarify which are the major psychosomatic states of drivers who experienced an incident [78].

Another example of a study that made use of data gathered by IoT devices is the one performed by Zheng, Rajasegarar, and Leckie (2015), where the authors employed several mechanisms to predict parking occupancy rate, thus allowing the citizen to avoid full parking lots, which may lead to better traffic dynamics and fewer accidents [79]. These examples are useful to demonstrate the importance of data made available by the IoT [16].

In the promising Smart Cities of the near future, communication between vehicles and the city will be constant [80, 81]. Hence, studies are taking advantage of Vehicular Ad-hoc Networks (VANETs) to improve not only road safety but also the driver's quality of life. VANETs are a technology that uses cars as nodes of a network to create a mobile network of vehicles [82]. This allows vehicles to easily communicate among them and with fixed infrastructures. Barba et al. (2012) developed a Smart City framework for VANETs that included intelligent traffic lights set on crossroads of a city that transmit warning messages and traffic statistics to vehicles [80]. However, a set of assumptions were made that are hardly true today. For instance, not all vehicles have global positioning systems and driver wizards with full information about the position of intelligent traffic lights, which was required by the designed framework. Another example may be seen in the work of Bui, Lee, Jung, and Camacho (2016), who, inspired by recent progress in vehicle technologies, propose a new approach to manage, in real-time, the traffic flow at road intersections by controlling the traffic light scheduling. They simulated their approach, being able to reduce the waiting time at intersections [83]. Indeed, the community has been using VANETs and Intelligent Transportation System (ITS) to improve and organize the flow of vehicles on the roads [84] and to provide real-time updated information of roads and traffic status [85]. Privacy and security of VANETs are also a hot-topic. The goal is to guarantee proper behavior of the network and the ability to resist to intrusions [86–88].

Focusing on vehicular communications, several authors have been studying possible forms of communication between vehicles, pedestrians, and infrastructures, also known as Vehicle-to-X (V2X). In this context, Vehicle-to-Vehicle (V2V) is a form of bi-directional communication between vehicles. The exchanged information may be used to calculate traffic conditions, geographical positions, to avoid inter-vehicle collisions, or to propagate rescue messages after an accident [89, 90]. Vehicle-to-Infrastructure (V2I) is a form of communication between a vehicle and an infrastructure, usually built on roads or streets. V2I may be used, for example, by a vehicle to identify its position or to obtain information about traffic and road conditions [91, 92]. In 2019, Casademont et al. proposed a cooperative-ITS for VRUs in which vehicles alert drivers of potential collisions with cyclists [93]. Even though the authors claim their framework is for VRUs, the fact is that VRUs are passive actors. The focus is only on informing the driver, i.e., the non-vulnerable user, about a possible collision. The developed framework required a large set of sensors and devices, i.e., stations, to be installed in cars and bicycles. For instance, cars required digital cockpits to show warning messages to the driver while bicycles were equipped with high precision location systems based on GPS, inertial sensors, and ultra wide-band ranging. Communication between the installed stations is provided by a network that supports V2X communications [93]. Nonetheless, other forms of vehicular communication are emerging. Forms mainly focused on pedestrians. Such forms may be defined as Pedestrian-to-Vehicle (P2V) and are shifting the focus from vehicles to pedestrians. This form of communication would allow the exchange of messages between VRUs and vehicles, in both directions [94–97]. However, P2V is still in its infancy, being, at the moment, more a concept than reality.

In 2014, Cho proposed a P2V communication system focusing on conflict zones such as unsigned single roads and intersection areas [96]. It had, as main focus, children and old people. The proposed theoretical system consisted of three parts named as on-board unit, road-side unit, and portable unit. The first one is installed in vehicles, the second one is located on the road, and the last one is used by pedestrians. Although the authors claim to use a type of P2V communication, real communication happens between vehicles and infrastructures, and only then between infrastructures and the pedestrian. There is no direct communication between people and vehicles. More examples of theoretical studies can be found in [98] and [99]. The first one proposes the combination of new wireless technologies with existing collision prevention systems. It also addresses important trade-offs for information exchange between vehicles and VRUs, namely low latency, low energy consumption, high position accuracy, and high warning reliability [98]. Besides these trade-offs, there is one that should not be forgotten, i.e., cost. A hypothetical solution for the VRUs' problem will only be implemented if it is economically viable. The second study proposes the concept of a hybrid city lighting to provide a more economical, safer, and smarter way to lighten up the way through future cities by projecting games on the pavement. With depth sensors implemented in the city's infrastructure, the authors claim they would also be able to develop car warning systems and detect clear ice areas on pavements or roads [99].

Another approach focuses on indirect communication between vehicles and VRUs, which can be achieved by means of interfaces installed in vehicles. In [100], the authors propose a set of interface concepts for autonomous vehicles to communicate its intention to VRUs in urban situations. The goal is to have unambiguous communication between road actors. Hence, some constraints must be considered. In particular, it should be avoided the use of language and cultural-specific symbols. In fact, in the designed interfaces, the authors considered three autonomous vehicle states: the vehicle is cruising, yielding, or it is starting to drive. Six interface concepts were proposed. One, for example, consisted of equipping the front bumper of the autonomous vehicle with a light strip that would work as a communication interface. The interface would light up with different patterns based on the car's intention. Another concept consisted of a band of light that surrounds the entire vehicle. In this concept, in urban scenarios, the band would only light a small segment on the part of the interface that is close to the detected VRUs. This would allow VRUs to acknowledge that they were, or not, detected by the autonomous vehicle and act accordingly. Colors can also be used to inform VRUs about the intention of the vehicle to yield, or not, to them [100]. Interestingly, most of the work done in this domain was performed by automobile manufacturers [101–103].

The fact is that VRUs' vulnerability may arise from many directions. The most common include the lack of external protection, road user's age, physical and mobility impairments, and visual or hearing disabilities, among others. Hence, many different methods can be followed when the goal is to increase VRUs' safety. One that has been receiving attention from the community is related to VRUs' intent and trajectory prediction. A major challenge of this method is related to the high degree of uncertainty of VRUs' actions in urban environments. For instance, in 2019, Saleh, Hossny, and Nahavandi proposed a model to predict long-term VRUs' intentions using reinforcement learning and Recurrent Neural Networks (RNNs). The authors defined long-term as 2/3 seconds ahead. Even though the models presented positive results, many decisions were made solely from an empirical point of view, without further experimentation and evaluation. Nonetheless, to train the models the authors used and pre-processed two distinct datasets [104]. One, the Stanford drone dataset [105], is available online and consists of a bird's eye view camera mounted on a drone hovering the campus of Stanford's University. This dataset is made of several videos with frame by frame bounding-boxes annotations on moving targets such as pedestrians, bicyclists, skateboarders, cars, buses, and golf carts [104, 105]. The second used dataset, which is also available online, was Daimler's path dataset [106]. This dataset consists of frame sequences recorded with a stereo camera system mounted behind the windshield of a vehicle and contains four pedestrian motion scenarios. Pedestrians are also annotated with bounding boxes.

More examples of studies that focused on RNNs to predict VRUs' intentions are [107], [108], and [109]. In [107], the authors focused on LSTMs to model human interactions in crowded spots. Data were obtained from the perspective of surveillance cameras. In [108], the authors also conceived LSTMs to predict trajectories of pedestrians, using only past positional data to forecast the next trajectory point. Finally, in [109] the authors used LSTMs with attention mechanisms to forecast pedestrians' intentions, demonstrating that forecast delays of less than 1 second may significantly affect the accuracy of models. In addition, it should be noted that cyclist velocities at intersections are typically higher than those of pedestrians, which prevents models from generalizing to both scenarios without accuracy loss [110].

A different approach focuses on the spatial and temporal analysis of crashes involving vehicles and VRUs. The goal is to identify areas where the vulnerability of such users is specially high, mainly in the framework of Smart Cities [111, 112]. In [113], the authors focused on the recognition of hazardous areas, also called blackspots, which can be defined as a particular zone that has a crash frequency significantly higher than other zones. In the performed study, blackspots are built on the density and severity of injuries, i.e., injuries are weighted based on its severity (light injuries, severe injuries, and fatalities). Several features were considered by the authors during their analysis, including VRUs' age and gender, injury severity, temporal variables, weather conditions, road location, and the surrounding environment. The authors found that September, October, and November were the months with more injuries, corroborating the results of [12], and that January and February seem to be the months with more fatalities. Thursdays and Fridays were the weekdays with more injuries. In addition, the authors found that more than 40% of pedestrian-vehicle accidents occur in the presence of crosswalks or traffic lights. No record of severely injured or dead cyclists occurred close to cycle lanes, strengthening the importance of such infrastructures to increase the safety of cyclists in an urban context. As for the used dataset, it contained VRUs crash data from three Portuguese cities between 2012 and 2015.

In 2015, Anaya et al. presented a novel Advanced Driver Assistance System (ADAS) to prevent accidents involving motorcyclists and cyclists. The authors developed a VRUs detection system where vehicles and motorcyclists have their own communication unit. On the other hand, cyclists, whose bicycles have reduced communication capabilities, are equipped with an on-board sensor known as iBeacon. An iBeacon is a low-powered bluetooth-based sensor with low-cost transmitters. It notifies bluetooth devices of one's presence. This study focused on keeping the non-vulnerable driver informed about the presence and location of VRUs by having a unidirectional information flow. In a previous study, the same authors have studied the accuracy and performance of several technologies such as GPS and Wi-Fi. Such studies are important as benchmarks since VRUs and drivers need to be informed of a possible collision as soon as possible, leaving no time for latency or packet losses [91]. Liebner, Klanner, and Stiller (2013) have also focused on the benefits that smartphones could bring to VRUs. The authors evaluated accuracy and transmission latency for smartphone-to-car communications, being able to show that the performance of the smartphone's GPS is heavily affected if the smartphone is, for example, in the breast pocket of a jacket [114].

Another study, conducted by Pyykonen, Virtanen, and Kyytinen (2015), focused on the problem of collisions between heavy goods vehicles and VRUs, in particular cyclists and motorcyclists. To address this problem the authors conceived an intelligent blind spot detection system to detect objects in blind spots and warn drivers. To achieve their goal, vehicles required new sensors such as stereo cameras, short-range radar sensors, and ultrasonic range finders. These sensors gather data that are presented to the driver through a human-machine interface based on audible and visual warnings, and a bird-eye view of the vehicle [115]. A different approach to this problem may be seen in [116]. The authors' approach is to make use of sensors to detect users and their movement, and then send alerts to vehicles nearby. The proposed system is composed of several modules, including a VRUs detection module, a road-side unit, which is part of the road's infrastructure and is responsible to transmit messages to vehicles, and an on-board unit, which is installed in the vehicle, receives messages, and notifies the driver. The model is, in its essence, based on a VANETs' approach. It is however disregarding VRUs as an active and proactive part of the system. Moreover, it showed important drawbacks such as the inability to detect pedestrians walking at higher speeds [116].

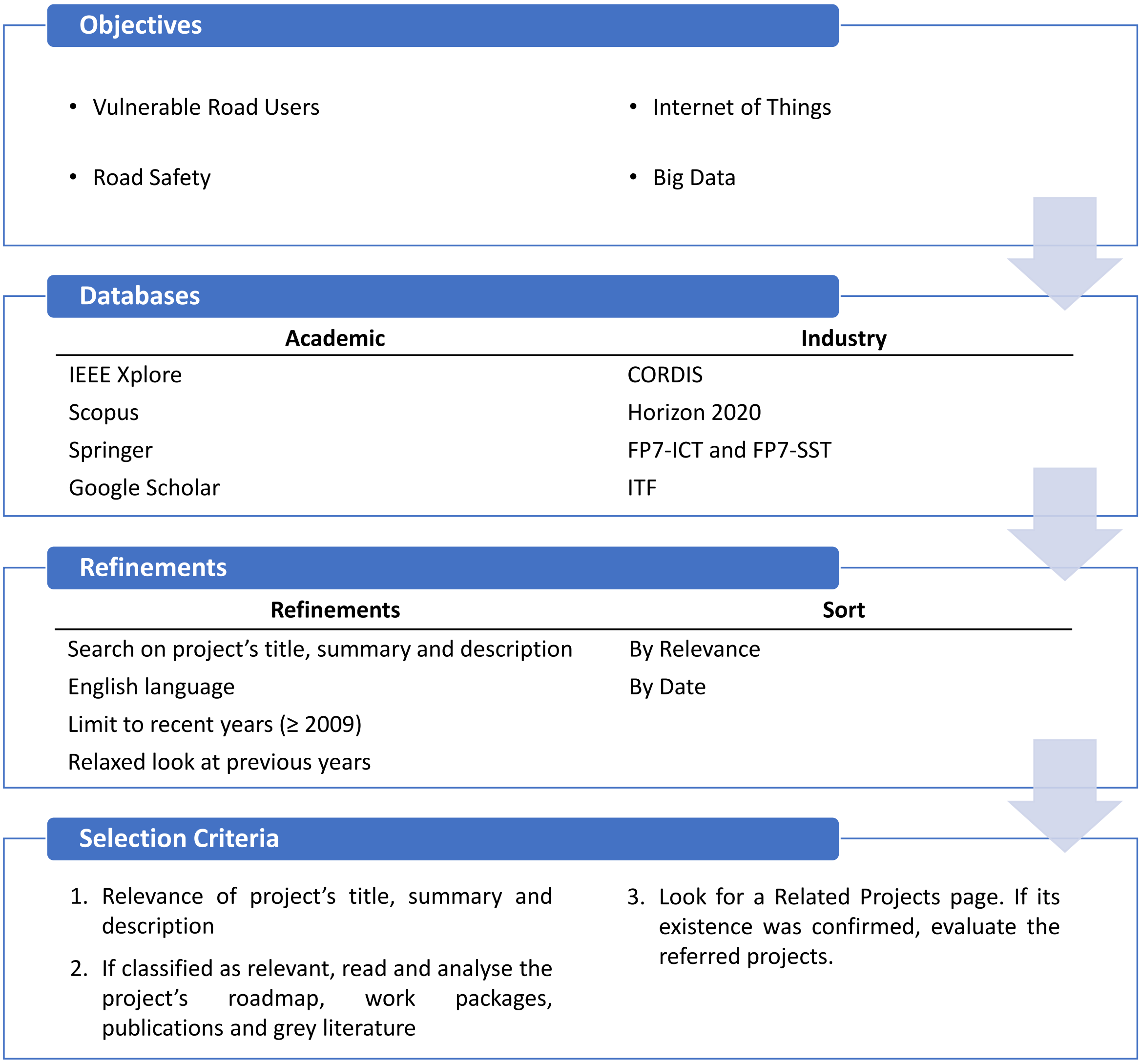
1.2.2 Related Projects

Having the VRUs' problem a significant societal impact, it is possible to find projects that are being funded and supported by associations, companies, and/or institutes, that aim to promote the evolution of industry while improving people's quality-of-life. Evidently, the difficulty of reviewing such projects is substantially higher when compared to a traditional literature review. The main documentation produced by these projects consists of gray literature, i.e., materials and research produced by companies outside of the traditional academic publishing channels. Such documentation includes work package reports, white papers, case studies, project's blog news, and government documents, among others. The lack of guidelines and standards significantly increases the difficulty of finding, analyzing, and evaluating gray literature. The concealment of results and documents is yet another adversity to overcome.

1.2.2.1 Projects' Search Strategy

A centralized platform to aggregate all projects is a non-existent situation. Therefore, to conduct the review, an extensive search was performed among different databases. In addition to typical scientific databases, a search was also performed among research and innovation programs, such as European Commission's CORDIS, The EU Framework Program for Research and Innovation (Horizon 2020), the European Commission's Seventh Framework Program (FP7-ICT and FP7-SST) and the International Transport Forum. Search engines were also used as well as the "Related Projects" page of the selected projects, which proved to be important to find projects working in similar themes and domains.

The focus went towards projects comprising road safety, VRUs, IoT, and Big Data. The more factors being pursued the best. Search terms and keywords were grouped using boolean operators, with the search being conducted in the first months of 2019. Upon inadequate or non-existent information, the project would be discarded. Figure 1.7 depicts the main steps of the followed strategy to find relevant projects.



Objectives

- Vulnerable Road Users
- Road Safety
- Internet of Things
- Big Data

Databases

Academic	Industry
IEEE Xplore	CORDIS
Scopus	Horizon 2020
Springer	FP7-ICT and FP7-SST
Google Scholar	ITF

Refinements

Refinements	Sort
Search on project's title, summary and description	By Relevance
English language	By Date
Limit to recent years (≥ 2009)	
Relaxed look at previous years	

Selection Criteria

1. Relevance of project's title, summary and description
2. If classified as relevant, read and analyse the project's roadmap, work packages, publications and grey literature
3. Look for a Related Projects page. If its existence was confirmed, evaluate the referred projects.

Figure 1.7: Projects' search strategy.

1.2.2.2 Projects' Review

Fourteen projects were considered relevant in regard to their proposal for safety enhancement and technological evolution. A critical analysis of the selected projects allowed us to summarize the main findings and advances produced by already closed projects and, at the same time, evaluate the novelty of those that are yet ongoing.

ARTRAC: Advanced Radar Tracking and Classification for Enhanced Road Safety

The ARTRAC project consisted of a consortium of seven partners including universities, automotive manufacturers, and research organizations. It is an already closed project that began on November 1st, 2011 and had a duration of three years. It focused on 24 GHz radar sensors, equipped in cars, to protect VRUs in general. ARTRAC aimed to develop a generic detection system capable of detecting pedestrians and cyclists as well as other vehicles [117, 118].

This project focused on vehicles and how to use sensor technology to increase road safety. It soon became clear that vehicles would greatly benefit from the presence of sensors to improve the driving experience and increase the safety of the road actors. This project aimed to equip cars with radar sensors to detect obstacles, such as VRUs, on the road. In fact, nowadays most vehicles use such sensors for parking assistance or distance calculations. The goal of the ARTRAC consortium was to improve vehicles, being partially successful. The non-vulnerable users were just passive elements that were only used to be marked as obstacles.

HIGHTS: High precision positioning for cooperative ITS applications

HIGHTS is an H2020-funded research project. It started in May 2015 and finished in April 2018. HIGHTS Consortium comprises fourteen partners from five countries. The consortium aimed to combine traditional satellite systems with an innovative use of on-board sensing and infrastructure-based wireless communication technologies to produce highly-accurate positioning technologies (accuracy of 25cm) for Cooperative-ITS, focusing on developing protocols and network support for data exchange between vehicles, infrastructure, and IoT devices [119]. The intention was for the developed platform to be a key enabler of cooperative adaptive cruise control and platooning. The platform intends to increase VRUs safety through bi-directional danger detection and by detecting slight deviations from driving courses [120]. Unfortunately, at the time of writing, and even though the project is now closed for a long time, no information is available regarding the completeness and implementation of the final solution.

The greatest innovation this project aimed to bring was the development of a highly accurate positioning technology for vehicles. In fact, the focus of HIGHTS was on enabling cooperative adaptive cruise control and platooning. However, it should be recognized that many applications could benefit from this new technology as it could easily allow one to apprehend the exact presence, direction, and velocity of vehicles and, thus, increase the efficiency of models to predict VRUs' intentions.

ICSI: Intelligent Cooperative Sensing for Improved traffic efficiency

ICSI was a FP7 project co-funded by the European Commission under the ICT theme, launched in November 2012 with a duration of thirty-six months. The consortium comprises partners such as INTECS and *Brisa - Auto-Estradas de Portugal*. The project's goal was to define a new architecture to enable cooperative sensing in ITS. Thus, based on real-time data, ICSI intended to develop a prototype of an European traffic monitoring system, enhancing energy efficiency in transportation systems. The consortium predicted that reductions in carbon dioxide emissions would be its most striking feature. The proposed fully-distributed architecture was able to detect and act over some of its elements, using the available local storage and computational resources. Sensor prototypes, road-side units, and communication units are required to be developed and integrated into the conceived platform [121, 122].

In this project, the targets are vehicles and traffic efficiency. ICSI focused on computer vision technologies, consisting of a set of camera-based sensors accompanied by a communication module that provided information on traffic flow and parking areas occupancy. Being difficult to unveil the true practical benefits of this project (and if/how they were achieved), it can be seen as a starting point for future projects, mainly for those aiming to advance traffic strategies based on real-time data. Within ICSI, the user is unconscious of his key role (e.g., occupying/freeing a parking lot, or increment/decrement traffic congestion) but his participation is mandatory.

Ko-TAG: Kooperative Sensorik

Ko-TAG was a collaborative research project with partners of the automotive industry and sponsored by the German Federal Ministry of Economy and Technology. It was launched in 2009 and lasted four years. The project was focused on cooperative radio-based sensing technology, mainly for V2V communication and VRUs' security. Vehicles were equipped with tracking units that query transponders carried by road users. Based on the received responses, the tracking unit determines the relative positions of nearby road users, classifies them, and assesses the risk of a collision. If there is such a risk, the system will either warn the driver or act to mitigate it, preparing the vehicle for an emergency stop. To refine the results, the tracking system uses an inertial measurement unit integrated into the transceiver to acquire data on pedestrians' speed and direction. Pedestrian behavior is predicted with a statistical-based model [123, 124].

The lack of available documentation was an adversity when analyzing this project, especially when most of the information was not in the English language. However, it is worth noting that this project aimed to bring benefits to vehicle emergency stops and autonomous emergency braking systems. Enhancing the safety of VRUs was also a priority, being Ko-TAG a project that proposed a novel concept of transponders carried by road users that would be detected by vehicles equipped with a specific antenna system. In recent years, the performance of devices, such as smartphones or sensors, has increased significantly, which reduces the need for proprietary software and hardware. In addition, considering VRUs as simple "things holders" prevents the use of algorithms for affective computing and behavioral analysis, leaving manual observation and statistics as the only possibilities for behavior modeling.

MAVEN: Managing Automated Vehicles Enhances Network

MAVEN is an H2020-funded project, coordinated by the German Aerospace Centre and Dynniq, with partners such as TomTom and MAPtm. It is a three-year project, having started in September 2016. MAVEN's main goal is to enhance intelligent urban road transport network and cooperative systems for infrastructure-assisted automated vehicles. It aims to develop algorithms to organize the flow of such vehicles and to structure the negotiation processes between vehicles and the infrastructure. The expectation is to increase traffic efficiency, improve infrastructure's capacity, and reduce emissions. In addition, it pretends to integrate new ADAS techniques to prevent dangerous situations with VRUs and contribute to V2V and V2I interactions through vehicle platooning and scheduling algorithms [125].

It is easily noticeable that most big cities are experiencing traffic problems on their roads. Future autonomous vehicles may help solve this problem and, thus, make traffic flow more homogeneous, improve driver safety and comfort, and reduce fuel consumption. MAVEN intends to study solutions to these issues through the development of new algorithms for platooning and negotiation between automated vehicles [126]. Aware that it can still take some years for fully automated vehicles to drive through our roads, MAVEN also aims to produce a roadmap for the introduction of future traffic management systems [125].

OPTIMUM: Multi-source Big Data Fusion Driven Proactivity for Intelligent Mobility

The OPTIMUM project was launched in May 2015 and had a duration of three years. The consortium had INTRASOFT International as coordinator and partners such as the Austrian Institute of Technology and *Infraestruturas de Portugal*. Due to the increasing saturation of transport networks, new solutions to improve traffic, freight transportation, and connectivity should focus on fuse data collected and processed from several sensors and service providers. Thus, OPTIMUM meant to establish a scalable and distributed architecture for the management and processing of multi-source big data, enabling the continuous monitoring of transportation systems while facilitating proactive actions in a semi-automatic manner. Among others, this project aimed to (1) capitalize on the benefits and potential of big data fusion, (2) design and develop a smart sensing system able to cope with huge amounts of heterogeneous data, and (3) enable semantic understanding of the acquired data to predict the status of transport networks [127, 128].

OPTIMUM proved to be a somehow difficult project to evaluate regarding its final result. Several reports are available to the community. However, it becomes impractical to go through several dozens of documents to look for the conceived case studies, the performed tests, and the obtained results. As prototypes, an android app and a web platform were released but it seems both have been discontinued. Nonetheless, it is worth highlighting the use of distinct approaches for data acquisition and fusion.

Prospect: PROactive Safety for PEdestrians and CyclisTs

The Prospect project was an EU research project funded by the H2020 program. The consortium included Audi, BMW, Volvo, Bosch, and Continental, among others. The project started in May 2015 and had its results published by October 2018. The overall goal of Prospect was to provide clues about relevant VRUs-related accidents and to improve the effectiveness of active VRUs safety systems [129]. Prospect focused on active safety solutions, where the vehicle senses its surroundings based on video and radar sensors. For that, new devices were developed, with deep learning techniques, in particular CNNs, being applied for object detection and classification.

Focused on vehicles, Prospect aims to enable the next generation of autonomous emergency braking systems, focusing on driver warning and vehicle control strategies (combined vehicle steering and braking) [130]. Having reached its conclusion recently, it is still difficult to measure the real impact of this project. Several prototypes and case studies were implemented but the impact in the real-world is still negligible. Another important result of this project is related to the clarification of relevant VRUs scenarios.

SAFE STRIP: Safe and green Sensor Technologies for self-explaining and forgiving Road Interactive aPplications

The SAFE STRIP consortium consists of eighteen partners from six different countries. It is an H2020-funded project with a three-year duration and a start date as May 1st, 2017. At its core, SAFE STRIP aims to introduce disruptive technology into low-cost integrated strip markers on the road and to make roads self-explanatory (with personalized in-vehicle messages) for all users and generations of vehicles with a reduced maintenance cost, as well as supporting real-time predictive road maintenance functions via dynamic road embedded sensors [131]. The premise is that, to save lives, we need inexpensive information about the road, the environment and the traffic itself, and, therefore, a novel micro/nano-sensorial system must be developed and integrated into road pavement tapes/markers at a fraction of the cost of current equipment. In addition, it intends to advance and merge the environmental parameters with vehicle tire information to dynamically estimate friction coefficient. The system is also intended to detect pedestrian crossings, work zones, railroad crossings, and other areas [132].

SAFE STRIP targets the challenging task of creating innovative micro/nano-sensorial systems with network capabilities to integrate into road pavements. Indeed, the world is now moving towards the creation of Smart Cities whose existence will always be linked to the city's ability to gather data and extract knowledge from diversified data sources. The development and introduction of new sensors, along with the use of AmI techniques, will be decisive for the successful creation of truly Smart Cities.

SAVE-U: Sensors and system architecture for vulnerable road users protection

SAVE-U is the oldest project in this review. However, its importance relies on the fact that it was one of the first to use multiple sensors to address the VRUs' problem. The project's consortium consisted of partners such as Faurecia, Siemens Automotive, and Daimler-Chrysler. It began in March 2002 and ended after three years. SAVE-U aimed to develop a high-performance sensing platform for cars, consisting of different sensor technologies, i.e., a radar network composed of several 24 GHz sensors, and an imaging system composed of passive infrared and color video cameras for the detection of unprotected road users. The main idea is that sensors recognize these users and calculate the probability of a collision with the vehicle, thus warning the driver or applying automatic braking if there is a risk of collision [133].

SAVE-U value is undeniable. In fact, the work realized under SAVE-U provided significant insights into contemporary autonomous emergency braking systems. Systems like the one referred to above are now widely spread across most vehicles, having been a crucial step in the enhancement of road safety. The biggest innovation of this project, and one of the reasons why it is worth mentioning it, relates to the use of different technologies of sensors for data fusion and knowledge extraction, and to the creation of models for the prediction of possible dangerous situations. Now that many years have passed, sensors have become smaller, faster, more efficient, and effective in the task of data collection.

SMART RRS: SMART Road Restraint Systems

SMART RSS was a project co-funded by the European Commission under the FP7-SST program between November 2008 and May 2012. The main partners of the project included, among others, Idiada Automotive Technology, University of Zaragoza, and the Federation of European Motorcyclist Associations. The main goal of this project was to develop a new Road Restraint System (RRS) by integrating primary and tertiary sensor systems into a smart RRS. With that, the consortium aimed to (1) provide greater protection to road users, (2) alert drivers and emergency services about the current state of the road and traffic flow, and (3) alert authorities about accidents to minimize the response time. The conceived smart RRS is free from cutting fixing posts and uses new materials and fixations to absorb crash energy [134].

This project can be analyzed from two perspectives. On the one hand, this project brought important insights to standardize roadside barriers [135]. On the other hand, this project raised the idea of using RRS to sense the road and propagate messages between road users and authorities. This last perspective seems to have not yet been deployed in real-world scenarios.

TEAM: Tomorrow's Elastic Adaptive Mobility

TEAM was a FP7-ICT funded project. It finished in October 2016. The consortium consisted of twenty-eight partners such as BMW and Fraunhofer. TEAM's vision was to bring together drivers, travelers, and infrastructure operators in a collaborative network, considering the needs and constraints of the participants and the network itself. The intention was to use mobile devices, such as smartphones or on-board units, to improve transport safety and efficiency. The project aimed to (1) improve traffic coordination to reduce fuel consumption and emission levels, (2) build interfaces for traffic visualization and interaction, (3) inform drivers of potential hazards through other vehicles, and (4) implement games to support better driving. Project partners state that the developed technologies are expected to enter the market within five to ten years after project completion [136, 137].

Several different challenges were addressed in this project. However, the idea of a collaborative network takes precedence, aiming to unite drivers, travelers, and infrastructure operators together in a collaborative network. TEAM's approach was based on social networks for vehicles, making use of gamification to increase user engagement. The difficulty with such scenarios is on building a strong community of users in order to have an acceptable dissemination of information and effective gamification techniques. Being difficult to measure the success of the project, the consortium itself realizes the struggle to enter the market as it is. Consequently, rather than requiring users to be the ones who make the wheel spin, a system must be able to deal with users' passivity and lack of engagement.

TIMON: Enhanced real time services for an optimized multimodal mobility relying on cooperative networks and open data

TIMON was an EU research project funded by the H2020 program. It was launched in June 2015 and lasted until November 2018. The TIMON consortium consisted of eleven organizations from eight different countries. The consortium believes that persistent problems related to traffic congestion, safety, and environmental challenges could be addressed if all users of the transport ecosystem, i.e., VRUs, vehicles, infrastructures, and enterprises, were connected on a cooperative platform. This project resulted in a mobile application that consists of a map that provides real-time planning services and alerts. The application feeds and is fed by a cloud-based platform that consumes data from app users, open and closed data, and V2X hybrid communications. The services provided by the application include collision alert, emergency vehicle approaching, road hazard warnings, vehicle density awareness, dynamic route re-planning, and dynamic route solutions for drivers. TIMON's approach can sense users that use their proprietary application and have GPS enabled, which limits the number of detected users [138]. Unfortunately, only a very limited amount of documentation from the project was made available.

TIMON's vision of a cooperative platform to deal with traffic-related issues shares similar premises with the vision of an IoP for Smart Cities. In fact, the IoP itself consists of an ecosystem, where things and people communicate, sense, and understand each other and the world, aiming to improve the quality of life of the latter. Therefore, more than a cooperative platform, we are required to focus on a cooperative environment, composed of conscious things and citizen sensors, with the latter as an active, reactive, and proactive actor.

Transforming Transport (TT): Validate the technical and economic viability of Big Data to reshape transport processes and services

TT is an EU-funded project representing a large consortium of forty-seven stakeholders in transport, logistics, and information technology in Europe. It began in 2017 and run for two and a half years. While business and tourism are expected to grow significantly in the upcoming decades, it is estimated that freight transport will increase by forty percent by 2030, requiring new alternatives for the way transport is carried out [139]. In addition, the reduction of pollution is leading the search for more efficient and sustainable transport paradigms. TT project aims to study and demonstrate how big data can transform the transport industry, making it more efficient while providing additional value in both mobility and logistics domains [140]. It addresses seven pilot case studies, such as smart highways, sustainable vehicle fleets, or multi-modal urban mobility.

TT's approach is to study the impact of big data to reshape the transportation industry. Several case studies will be conducted to show concrete, measurable, and verifiable evidence of data value for mobility and logistics. Indeed, big data can aid in the evolution of the sector, but it should not be forgotten that the methods to deal with these data and the extraction of knowledge are equally important.

XCYCLE: Advanced measures to reduce cyclists' fatalities and increase comfort in the interaction with motorised vehicles

XCYCLE was an EU-funded project under the H2020 program. The consortium, composed by the University of Bologna and Volvo, among others, started the project in June 2015, with all activities being completed by November 2018. On the road, cyclists, who are not treated equally by traffic systems, suffer a disproportionate share of serious injuries and fatalities [141]. Thus, XCYCLE aimed to develop user-friendly systems to equalize the treatment of cyclists in traffic and to improve active/passive detection of cyclists. The overall architecture included an in-vehicle detection system, a threat assessment system, and a communication gateway to efficiently interact with in-vehicle and roadside systems.

Having reached its conclusion recently, its current impact is still negligible and hard to calculate. Indeed, this project focused on a subset of the entire VRUs spectrum. However, in the market, there are already available several different on-bike devices aimed at car detection. Unfortunately, these devices did not meet great success mainly due to their dimensions, weight, and significant cost [142]. Nonetheless, an interesting result of XCYCLE is related to the in-depth analysis of cyclists' behavior in terms of trust, reactions to false alarms, and risk perception to address issues such as the temptation of cyclists to violate red traffic lights.

1.2.3 Summary

Smart Cities are a multi-disciplinary domain, with smart homes, sustainability, and public safety as some of the main areas of intervention. One that still requires extensive studies is road safety since road traffic injuries and deaths are still a major health problem in today's society. VRUs are a particular group of affected users that have an increased vulnerability on the road. Several distinct approaches are being followed to enhance VRUs' safety. While some focus on drivers, others consider vehicles or road-side infrastructures. In fact, this may be seen as the main conclusion of the conducted literature review. Clearly, the main emphasis is on vehicles and drivers, setting the vulnerable user aside. However, if VRUs are part of the problem they should also be part of the solution. One must seize the citizen as a sensor and as an active and participative actor of the road ecosystem, and, with that, allow the IoP to be a reality. As soon as the citizen sensor becomes effective, data will be available for data mining and knowledge extraction. It will allow Smart Cities to understand and act on problematic situations and solve problems where the human is the weaker element.

Even though the academic community has been promoting significant advances in several domains, the vast contribution of the industry to the technological progress of our world is undeniable. Hence, considering the significant amount of attention road safety has been arousing, it became important to study, analyze, and review projects that are being carried out by associations, companies, and/or institutes that aim to promote the evolution of industry while improving people's quality-of-life and empowering the social good. From the selected projects, the great majority is, or already has, concluded its work. This may be related to the due dates of innovation programs. It is also interesting to note the strong engagement and commitment of the automotive industry, telecom operators, and public institutions. Table 1.1 presents a summary analysis of the reviewed projects.

The reviewed projects followed different approaches in regard to the problems being studied. It is interesting to note the benefits of using anonymized network data and massive mobile data to create more robust and accurate solutions, seizing the opportunities that big data brings to maximize social benefits. Interestingly, older projects such as Ko-TAG, SAVE-U, ARTRAC, and SMART RRS were important to advance state of the art radar, camera, and video-based sensors for road safety, in particular, for vehicles. On the other hand, recent projects such as XCYCLE, HIGHTS, MAVEN, and TT use smartphones, high-precision location technologies, and infrastructure-based wireless communication technologies to increase road safety. Others, like TEAM and TIMON, are focused on developing frameworks for collaborative networks and on future automated vehicles and their challenges. Indeed, the goal of all these projects is to foster road safety, by advancing existing knowledge on sensors and collaborative networks.

Like the academic community, excessive focus is being given to the non-vulnerable user. This means that VRUs in themselves are being set aside, with works foregoing vital information about VRUs. The goal is usually to increase VRUs safety by keeping the non-vulnerable user, i.e., the driver, informed about the presence and location of VRUs, maintaining an unidirectional information flow.

Table 1.1: Summary of the projects considered for review.

Project	**Domains**				**Date**	**Program**
	Big Data	**IoT**	**Road Safety**	**VRUs**		
ARTRAC			✓	✓	2011-2014	FP7-Transport
HIGHTS		✓	✓	✓	2015-2018	H2020
ICSI		✓	✓		2012-2015	FP7-ICT
Ko-TAG		✓	✓	✓	2009-2013	Private
MAVEN		✓	✓	✓	2016-2019	H2020
OPTIMUM	✓	✓	✓		2015-2018	H2020
Prospect		✓	✓	✓	2015-2018	H2020
SAFE STRIP		✓	✓		2017-2020	H2020
SAVE-U			✓	✓	2002-2005	FP5-IST
SMART RRS		✓	✓		2008-2012	FP7-SST
TEAM		✓	✓		2012-2016	FP7-ICT
TIMON	✓	✓	✓	✓	2015-2018	H2020
TT	✓		✓		2017-2019	H2020
XCYCLE		✓	✓	✓	2015-2018	H2020

1.3 Conceptualization

The next lines are dedicated to the conceptualization and formalization of the research plan. It starts by detailing the problem being addressed, the motivation, and the main goals. Then, based on the analysis of state of the art, research questions are defined and the hypothesis is set. The research methodology being followed in this thesis is detailed afterwards, along with the research methods and the research plan. A brief presentation of the host institution and the research team is also provided. Finally, the structure of this document is compiled.

1.3.1 The Problem, Motivation, and Goals

It is important to start by detailing the nature of the problem here addressed, in order to explain the underlying motivation to achieve the desired goals. In a broader sense, this work aims to increase road safety. In particular, it aims to reduce the vulnerability of those more fragile on the road, while promoting an environment where people are active and participative actors. Indeed, the main problem that has been identified in sections 1.1 and 1.2 is related to the lack of importance that is given to VRUs. In fact, these are seen as mere obstacles that should be avoided. VRUs are considered "flat things" without personality neither emotions. Undoubtedly, until now, the main emphasis has been set on advancing vehicles and on keeping the non-vulnerable user informed about the presence of someone, or something, on the road. However, road fatalities, the eighth leading cause of death, significantly affect pedestrians, cyclists, and motorcyclists. These, together, accounted for 70% of all road traffic deaths, in urban roads, in the European Union in 2017 [11].

The technological revolution we are experiencing demands for new approaches to road safety. In particular, one must seize the citizen as a sensor and as an active and participative actor of the road ecosystem. One must take advantage of the dissemination of devices such as smartphones, smartwatches, and other sensors, to enable the IoP, an environment where people are not set aside. With that, one will also be promoting the technological evolution of cities, creating new means to sense the city's environment, and allowing Smart Cities to understand and act on problematic situations where the human being is the weaker part. In addition, the consistent population growth of large urban centers demands for new ways to tackle a set of problems that include crowding, traffic congestion, bureaucracy, crime, sprawl, and the security of those more vulnerable on the road.

This work focuses on three large concepts, i.e, VRUs, the IoP, and Smart Cities. These seemingly detached concepts will be handled as one in this thesis, i.e., one expects to contribute to the enhancement of VRUs safety by considering these users as participative actors of the road ecosystem with an increased awareness of their surroundings, contributing, at the same time, to the development of Smart Cities. This is what motivates and drives this work. Take advantage of the amazing technological innovation we are experiencing to improve our quality of life and ensure a better future for us all. Another catalyst is the possibility of extending the current state of the art of several distinct domains, including AI.

VRUs vulnerability may arise from several directions. In fact, many distinct approaches, with different goals, can be followed and implemented. No solution will settle out all problems nor will it eliminate all fatalities. Instead, the difference will be made through the sum of several small contributions. Hence, the approach followed in this work aims, first and foremost, to recognize VRUs as active, reactive, and participative actors of the city ecosystem, guaranteeing that VRUs are aware of their surroundings. From this premise emerges what we define as the 3-C. The 3-C correspond to *City*, *Crowd*, and *Citizen* sensing, i.e., it consists of three sensorization levels and the corresponding models for knowledge extraction. Based on that, the following goals were defined:

- City Sensing - one major goal of this work focuses on sensing the city in itself and on conceiving methods to extract knowledge from the city's ecosystem. On the one hand, such knowledge can be shared with VRUs with the goal of making such users aware of their surroundings. On the other hand, this goal is directed towards the promotion of smarter cities, i.e., cities able to sense their environment, act on problematic situations, and provide actionable information to their citizens;

- Crowd Sensing - also known as mobile crowd sensing, consists in sensing large groups of individuals through means of mobile devices with sensing and computing capabilities. This level lies between sensing the city and the citizen. Two approaches may be followed. A passive approach to crowd sensing allows gathering knowledge about the density of people at certain points of interest for VRUs, such as pedestrian crossings or busier roads. An active approach, on the other hand, incites subjects to participate and opine about their city;

- Citizen Sensing - another goal of this work consists of developing means to sense the citizen in order to establish them as more than simple "things" holders. To achieve such goal, special focus should be given to the use of mobile and wearable devices as well as the design and conception of ML models for affective computing.

To strengthen the foundations of this work, two additional goals were elicited:

- Smart Cities introduce novel problems related to the management of inordinate amounts of data of different types and sources. Hence, research must be conducted to study new ways to store data in such a context, guaranteeing that data may be committed by any actor, being, at the same time, freely available to anyone that aims to derive knowledge from it;

- The development of a collaborative platform that allows VRUs to have access to a set of data and information regarding the environment where he/she stands.

The goals here described are the ones to be tackled during this research work. The knowledge and specialization of the research team, as described in a subsequent section, was crucial for the definition of this line of attack.

1.3.2 Hypothesis and Research Questions

Having settled the problem and goals that underlie this work, it is now time to define the main research hypothesis that will guide the experiments to be carried out:

RESEARCH HYPOTHESIS

The combination of AmI techniques and Affective Computing with the IoP is an effective approach, in terms of efficiency and efficacy, to enhance VRUs' safety by making them participative actors of the city's ecosystem with an increased awareness of their surroundings.

The main research hypothesis goes through several fields to achieve the main goal of enhancing VRUs' safety. In fact, this research work is expected to conduct research in the fields of (1) AmI, to deal with the development of intelligent environments and crowd sensing in Smart Cities; (2) Affective Computing, to give dimension and depth to VRUs; (3) IoP and MAS, to create a collaborative platform to be used by those more vulnerable on the road. The conceived platform is expected to live by itself, using ML models and other techniques to learn and provide actionable information to VRUs. The prediction is that if one uses such techniques and paradigms in the context of a Smart City, it will be easier for VRUs to participate and have access to a set of information about the current and future status of their city, which will make them feel safer and assume better behaviors on the road.

Given this array of features, the research hypothesis is broken down into specific solutions' specimens in the form of exploratory, evaluation, and action research questions:

Research Question 1: *Is a MAS able to autonomously capture city's data and produce actionable knowledge from such sources?*

A MAS is expected to emerge from the conducted research, being comprised of a set of agents that are able to autonomously capture data, pre-processing it, and then make it available for ML models, which are also to be embedded by intelligent agents. The goal is to have a fully-autonomous ML pipeline, going from data collection to model conception. The validity of this research question is to be assessed by means of the produced and deployed prototypes, and their performance.

Research Question 2: *With the use of Affective Computing techniques and mobile devices, is it possible to accurately measure boredom and the personality of subjects in real-time with a non-invasive, non-intrusive approach?*

Affective computing is here seen as a way to capture emotions, mood, and personality through computational means. This use case aims to conceive means to measure boredom and personality in real-time, using ML and mobile devices in a non-invasive, non-intrusive manner. This research question is of the utmost importance as it aims to give systems the ability to obtain a deeper understanding of people based on their personal characteristics. This question will be validated through the conception, development, and evaluation of models for boredom and personality assessment, being further validated by the scientific community.

Research Question 3: *Is it possible to implement passive methods for crowd sensing that require no participation from people?*

This research question focuses on the conception of passive methods for crowd sensing, i.e., methods to sense people without the need to install or carry proprietary software. The goal is to be able to enhance crowd control and people's safety without requiring citizens to have, in their possession, any mobile application. The validity of this research question is to be assessed through means of prototypes and experimentation as well as by peers in scientific gatherings.

Research Question 4: *Is blockchain a viable solution (in terms of efficiency and performance) to store structured data? Can intelligent agents work, and maintain, a blockchain?*

Data collected about a Smart City's environment should be made available to all. Hence, solutions are required where data may be committed by any actor, being, at the same time, freely available to anyone that aims to derive knowledge from it. Combined with the need for an immutable, distributed, and decentralized approach and the blockchain technology emerges as an evident choice. This research question also deals with the ability of intelligent agents to maintain a blockchain in a MAS context. Given the large spectrum of this use case, it is to be considered validated by means of a case study and the attained results.

Research Question 5: *Can a collaborative platform allow VRUs to have a better perception of their surroundings and become active actors of the city's ecosystem?*

This research question deals with the ability of a platform to make use of models to sense the city's environment and provide actionable information to its users so that they can use it to have a better perception of their surroundings. Having a larger perception of the environment will allow VRUs to make informed, fact-based decisions. The collaborative aspect is related to the ability of users to contribute to the platform. This research question will be considered validated if such a platform is produced, deployed, and legitimized by the scientific community.

The previously elicited research questions aim to further specify the research hypothesis by taking into consideration the addressed domains, thus providing a stronger validation to the research hypothesis.

1.3.3 Research Methodology and Plan

Adopting a well-defined research methodology assumes an increased importance when conducting scientific research. The goal is to have an organized and logical sequence of standardized procedures to build scientific knowledge. The process must be replicable, precise, falsifiable, and parsimonious.

The research conducted in this PhD thesis follows a scientific research methodology that combines both the Hypothetical-Deductive scientific method with the Engineering Process, as depicted in Appendix A. Following the engineering background of this doctoral course, one should bear in mind that it intends to research and develop a solution for a problem. Thus, this work focused on creating knowledge for a purpose produced by scientific applied research, having its validity in a matter of efficacy, efficiency, and utility. It did not aim at creating knowledge that conveys understanding human, group, or social phenomena, which is often referred to as theories or laws.

The followed methodology starts with the identification of the problem based on the reviewed literature. A research hypothesis is then devised as a solution to the identified problem. This initial phase is followed by the Design one, which focuses on designing and proposing solutions' specimens based on the devised hypothesis. In this work, such specimens are to be understood as the research questions defined in Section 1.3.2. The Validation phase is where one focuses on planning and executing experiments, recording observations, interpret results, and draw conclusions. The goal is to validate the conceived specimens in order to prove the research hypothesis.

This doctoral plan aims to produce empirical research, i.e., aims to test the feasibility and validity of an hypothesis and the corresponding solution with a set of experiments. In fact, in this work the research methods may be defined as experiments, in the laboratory, both through computer and simulated experiments, as well as proof-of-concepts and field experiments, when deploying the prototypes for people to use it for a period of time.

Five main tasks were defined to lead this project to a fruitful conclusion and achieve the proposed objectives. Such tasks were developed during the last three years of the doctoral program, the research period, being as follows:

Task 1: *Definition, design, and conception of an IoP architecture and a MAS on the context of the VRUs problem.*
Duration: October 2017 to December 2017 (3 months).

IoT has been assuming an increased importance in our lives. However, it leaves aside people as an active, reactive, and participative actor. The IoP came to address this issue. However, being the IoP an emerging paradigm, there is a clear lack of hypothesizing meanings and properties. This task aims to postulate a set of essential properties in the development of the IoP and, based on such features, proceed towards the design of a descriptive IoP-based architecture. This task aids defining the foundations on where the research work rests and intends to help providing an answer to research question n° 1.

Task 2: *Affective computing and ML models for the citizen sensor.*
Duration: October 2017 to September 2018 (12 months).

The final system is intended to consider the emotional states of its users in order to establish them as more than simple "things" holders. Therefore, models to extract VRUs-relevant information are defined using ML and AI approaches, with particular focus on Affective Computing. Two main features are considered, i.e., boredom prediction and personality assessment using mobile and/or wearable devices and non-intrusive approaches. This task provides the answer to research question n° 2.

Task 3: *Conception, definition, and implementation of methods and models for city and crowd sensing.*
Duration: July 2018 to June 2019 (12 months).

This task aims to give answer to two goals. On the one hand, it consists in the design and conception of AmI methods to create a sensed city environment. It focuses on two major features: traffic flow forecasting and environmental sustainability, in particular, air pollutants, using DL models. Such information is of the utmost importance for VRUs to have a better perception of their city status. It also focuses on the design and conception of an agent-based system to autonomously go through the entire ML pipeline. On the other hand, this task aims to sense large groups of individuals without requiring the crowd to install any software on their mobile and wearable devices, allowing the possibility of gathering insights at certain points of interest for VRUs such as pedestrian crossings or busier roads. This task provides an answer to research questions n° 1 and 3.

Task 4: *Experimentation, exploration, and definition of standards and ontologies for data storage.*
Duration: January 2019 to March 2020 (15 months).

Smart cities, to emerge as such, are required to have the ability to create knowledge and reason upon data describing its environment. Such an ability introduces novel problems related to the management of inordinate amounts of data of different types and sources. Hence, solutions are required where data may be committed by any actor, being, at the same time, freely available to anyone that aims to derive knowledge from it. Combined with the need for an immutable, distributed, decentralized approach and the blockchain technology emerges as evident. This task aims to design and conceive a MAS blockchain and evaluate its ability to store structure data committed by any actor. This task provides the answer to research question n° 4.

Task 5: *Platform implementation.*
Duration: April 2019 to June 2020 (15 months).

The last task focuses on the development of the project's platform archetype, making use of the conceived solution specimens. The goal is to have a collaborative platform where VRUs are felt, have opinion, and have access to a set of information that can increase their safety on the road. The platform is intended to have front-ends where information is visible. This task provides the answer to research question n° 5, the last research question.

Two more tasks go throughout the entire research period. One is related to the analysis of the state of the art. This happens due to the novelty of the addressed domains and to guarantee that one is fully aware of how the art advances during the three years of the research period. The second one is related to the scientific dissemination, being focused on the production and publishing of scientific papers in peer-reviewed conference proceedings, book chapters, and international journals, sharing the attained results and the main research conclusions.

It is important to highlight that the followed methodology is entirely non-invasive, and preserves data and people's privacy. This research work has no intention of harm. Instead, it aims to commit benefits to our society by contributing with solutions to improve our quality of life.

1.3.4 Research Team and Host Institution

The research team is made of three people. The candidate, Bruno Fernandes, holds a Master's degree in Informatics Engineering from the University of Minho, in Braga, Portugal. He is a scientific researcher at the ALGORITMI Centre, a research unit of the School of Engineering at the University of Minho. He is also an Invited Assistant Professor at the same university, lecturing ML and Intelligent Systems. His current research interests include Smart Cities, IoP, ML, MAS, blockchain, and road safety.

Cesar Analide, the supervisor, is a scientific researcher in the AI field. He is also a Professor at the Department of Informatics of the University of Minho, being a founder member of the Synthetic Intelligence Laboratory (ISLab). His main research interests are in the domains of AmI, knowledge representation, MAS, and ML. In the road safety field, he has been producing knowledge regarding the assessment of road traffic expression, driving behavior, well-being, and ambient sensorization.

José Neves, the co-supervisor, is an *Emeritus* Professor in Computer Science at the Department of Informatics of the School of Engineering at the University of Minho, being also a researcher at the ALGORITMI Centre. He was the founder of the AI area at the University of Minho. His research interests include, among others, AI, ML, knowledge representation and reasoning, and evolutionary computing.

The research team conducts scientific research at the ALGORITMI Centre, a research unit of the School of Engineering at the University of Minho, focusing its activity on projects that explore a strong link with the community. It is a center of excellence, receiving a Very Good in FCT's last R&D Units evaluation. Bruno Fernandes, José Neves, and Cesar Analide belong to the ISLab, which has, over the years, made important contributions to the domains of knowledge representation, sensorization, MAS, and ML, just to name a few. The group also has strong partnerships with the School of Medicine, the School of Law, and the Institute of Social Sciences of the University of Minho as well as with other national and international institutions, having large experience in joint projects.

1.3.5 Document Structure

This document is organized into six chapters as follows:

Chapter 1 established the foundations on where this work rests. It initiated with an introduction to the main research domains. It then provided a thorough review of the state of the art as well as an analysis of related projects. This chapter ends with the conceptualization and formalization of the research plan, research questions, and working hypothesis.

Chapters 2 to 5 detail the experiments that were conducted throughout the execution of the research plan to validate the postulated research hypothesis. Each chapter has a similar structure, i.e., starts with the theoretical foundations and the rationale behind the performed experiments, it then details each experiment, ending with an analysis and discussion of the case study and the obtained results. These chapters are presented using the 3-C concept, i.e., *City*, *Crowd*, and *Citizen* sensing.

Chapter 2, City Sensing, presents and details all experiments that aim to sense the city in itself. It starts with the MAS that was conceived to collect data for a city and automatically create and fit ML models. It then details two major experiments in regard to a city's environment, i.e., traffic flow forecasting and Ultraviolet (UV) index forecasting.

Chapter 3, Crowd Sensing, details the experiments that were conducted for crowd sensing. Particularly, it describes and discusses a passive sensing approach, where users are passive actors and sensed without the need to carry anything, and an active sensing approach, using geofences and an emotional map that quantifies people's feelings towards certain zones of a city.

Chapter 4, Citizen Sensing, describes and discusses experiments that aim to sense the citizen. In particular, this chapter presents an approach to boredom detection in smartphones as well as a novel method for personality assessment, and knowledge representation and reasoning.

Chapter 5 presents, details, and discusses *KnowLedger*, a MAS blockchain solution to store structured data, where a new protocol, entitled as Proof-of-Confidence (PoC), is used to award those who commit accurate and reliable data. To reduce the computational load of the entire solution *KnowLedger* includes a novel multi-chain architecture.

Chapter 6, the last chapter of this thesis, provides an in-depth analysis of the obtained results, presenting the conceived archetype, *SafeCity*, which aggregates multiple solution specimens. It also presents fundamental considerations regarding the research hypothesis and outlines further research directions. Quantitative results, in the form of scientific publications, participation and organization of scientific events, lectures, invited talks, and supervision of students, are also described.

Chapter 2

City Sensing

Cities, to become smart, must evolve technologically, must implement systems and techniques to enhance data collection and knowledge creation, and, last but not the least, must create means for their citizens to become active actors of the city's ecosystem. With road safety as background, cities must promote new measures to guarantee the safety of their citizens in order to decrease accidents and injuries. The way to address such points is not monolithic. Instead, it should be grounded on multiple contributions that, when summed, allow people to have available a set of information that may help increase their quality-of-life.

2.1 Rationale

An important goal of this research work consists of sensing the city in itself and on conceiving methods to extract knowledge from the city's environment. Such knowledge may then be shared with VRUs so that they become aware of their surroundings and may adapt their behavior accordingly. Obviously, many distinct parameters can be sensed and modeled, depending on the underlying domain. This research work focused on two main features: traffic flow forecasting and environmental sustainability. The first aims to create models that are able to accurately forecast how the traffic flow will stand in future hours. The second focuses on forecasting the UV index. Forecasts are then used to provide actionable information to VRUs, allowing them to avoid streets and zones with high traffic as well as to adapt their behavior upon dangerous UV index values.

To achieve the proposed goals, state of the art RNNs based-models were designed, conceived, tuned, and evaluated. In particular, several candidate LSTMs models were experimented. It should not be forgotten that data are mandatory. Hence, relevant attention was given to the development of an agent-based system responsible for collecting data from a city's environment, as detailed in subsequent lines.

Considering the strong ML background of this research work, all ML-based tasks followed the Cross-Industry Standard Process for Data Mining (CRISP-DM) methodology. This methodology provides a structured approach for planning and executing a ML project, being a robust, well-proven, and widely used methodology. Figure 2.1 details the methodology's pipeline, starting from business and data understanding, followed by data preparation, modeling, evaluation, and deployment of the best candidate model. This iterative process allows tasks to be performed in a different order, allowing one to backtrack to previous tasks and repeat actions when required.

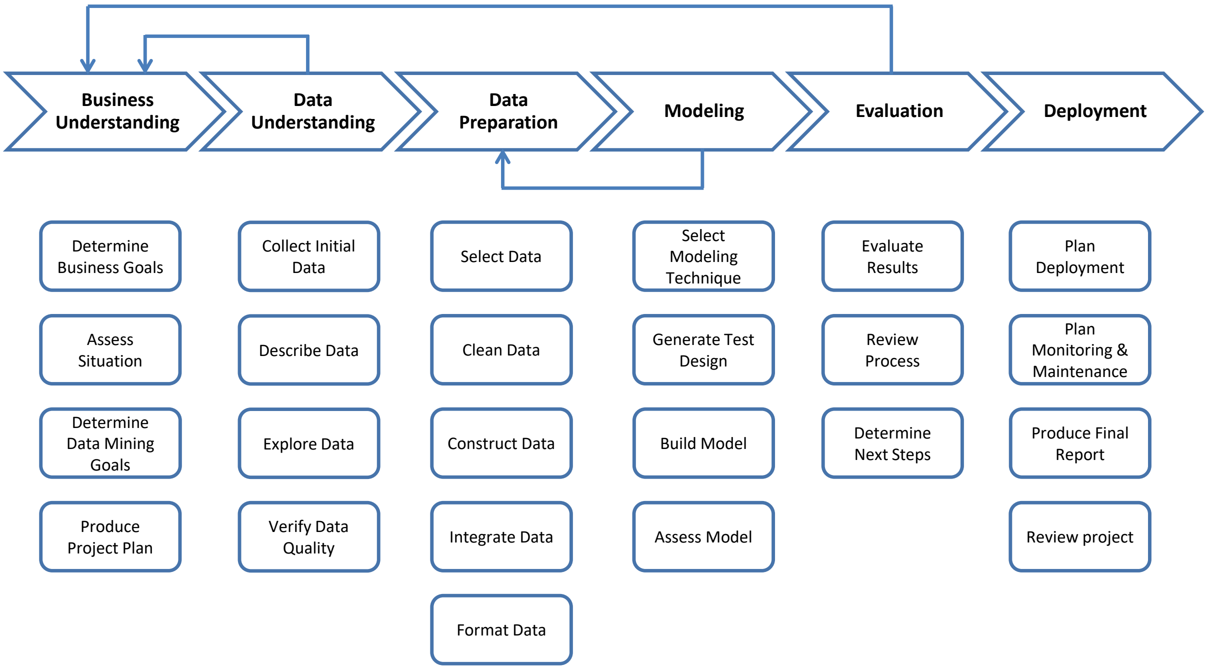


Figure 2.1: CRISP-DM methodology for ML projects. Adapted from *crisp-dm.eu*.

2.2 A Conceptual Architecture for the Internet of People

The IoP aims to complement the IoT by considering people as active, reactive, and proactive actors of the city's ecosystem [25]. However, being in its infancy, the foundations on where the IoP rests are not yet well established. To help strengthen the IoP, this research work proposes a set of properties that should be seen as relevant when conceiving IoP-based environments:

- *Interoperability*: ensuring proper communication between all elements of the IoP is an essential feature. A possible definition of interoperability corresponds to the ability of heterogeneous systems to exchange information transparently [40]. It assumes the use of open and well-defined standards and protocols, rather than having each manufacturers using proprietary technologies;

- *Citizen Sensor*: the exponential use of smartphones and wearable devices, which have exceptional sensing capabilities, empowers the citizen sensor. This enables citizens to sense themselves, others, and the environment where they stand, allowing the creation of knowledge from extended and deeper sets of data [41];

- *Ambient Intelligence*: with devices becoming smaller, integrated, and autonomous, they tend to go unnoticed. One should take advantage of such devices to create intelligent environments, allowing the conception of truly Smart Cities [63].

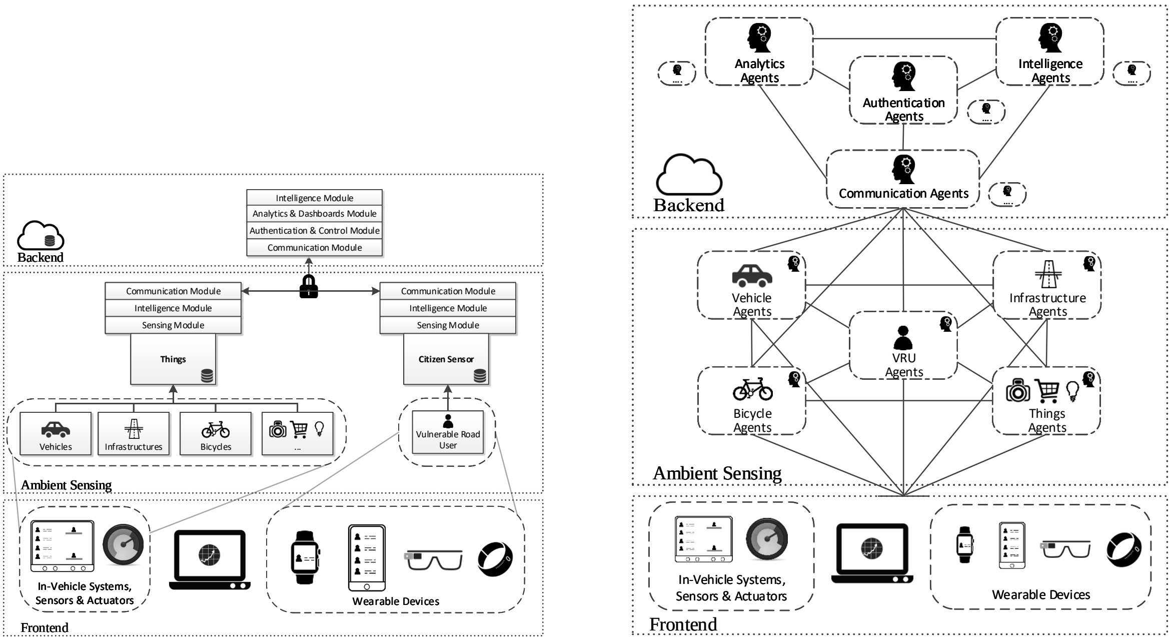
More properties to have a secure, powerful, and open IoP may be set as follows:

- *Intelligence*, *Analytics*, and *Dashboards*: IoP systems are expected to be able to gather important insights and depict it using front-ends, allowing the creation of user awareness;
- *Cloud Platforms*: due to the lower computational power of wearable devices, it should be possible to communicate with centralized platforms in case they require on-demand access to services, storage, or reporting, among others;
- *Device Authentication*, *Privacy*, and *Security*: privacy and security of things and people is a major issue, ranging from the collection of information to its exposure. Sensors, actuators, and other devices typically tend to delegate their authentication to reduce their exposure [143];
- *Transparency*, *Monitoring*, and *Auditing*: communication among things and people should follow open and robust standards and protocols for higher transparency. Users should be able to define their privacy policies and to monitor how their data are being used [143];
- *Scalability*, *High-Availability*, and *Efficiency*: platforms should be scalable and as flawless as possible. Possibilities such as the use of asynchronous communication or MAS are interesting solutions to support scalability, availability, and fault tolerance [144].

Examples of IoT architectures have already been proposed in the literature [23, 81, 145]. However, when the subject matter is the IoP, there is a clear shortage of proposals. Hence, this research work conceptualized a high-level architecture for the IoP that is compliant with the properties defined before. Such architecture, as depicted in Figure 2.2a, consists of three layers. One, the *Backend*, should ideally be built on a cloud platform for scalability and availability factors. It should also contain a set of modules, including (1) an *Intelligence module* responsible for providing the platform with reasoning capabilities; (2) an *Analytics and Dashboarding* module focused on the analysis and production of accessible information for people and entities; (3) an *Authentication and Control* module responsible for protecting and securing the platform; and (4) a *Communication* module, the only entry point to the backend [25].

On a second layer, entitled as *Ambient Sensing*, one may find the actors of the platform, with VRUs in the form of citizen sensors. It is in this middle layer that the environment is felt, with the elements ideally equipped with three distinct modules, i.e., (1) a *Sensing* module, responsible for the use of sensors and techniques to sense and understand the environment and the others; (2) an *Intelligence* module, which intends to work on the data gathered by the first module and employ models for Affective Computing or traffic forecasting, among others; and (3) a *Communication* module, responsible for sharing relevant data with others and for communicating with the backend [25].

Finally, the last layer consists of a *Frontend*, where information is visible to people and the community, in the form of dashboards or relevant information [25].



(a) High-level IoP architecture in the context of the VRUs' problem.

(b) A MAS over the proposed IoP architecture.

Figure 2.2: An high-level architecture and MAS for the IoP. Adapted from [25].

Figure 2.2b depicts a MAS proposal over the defined IoP architecture. In such a scenario, agents communicate with each other and with the environment with a focus on understanding the latter and reason upon the conceived models, coordinating their efforts to achieve their goals and the one of the ecosystem where they live [25].

2.3 Software Agents for City Sensing

Based on the proposed understanding of the IoP, a MAS was conceived to gather data from a city's environment and to go through the entire ML pipeline to make use of the best candidate models. The next lines describe such agents, entitled as *The Collector* and the *ML Architect*.

2.3.1 The Collector

There are a multitude of alternatives when the goal is to sense a city's environment. In fact, one may consider three distinct sensorization levels for AmI in a Smart City (as depicted previously, in Figure 1.6). Throughout this research work, there was a strong need to sense and get real data from a city's environment and, based on such data, design and conceive models for knowledge creation.

A software agent, entitled as *The Collector*, was developed from scratch to collect data from a set of public APIs. *The Collector* monitors a set of cities and their environmental parameters. It is a modular and configurable software agent that can, at any time, start collecting data for any city worldwide. It was developed using Java as programming language, being remotely deployed in an Ubuntu 18.04 machine hosted at the University of Minho. Its data collection service is available as open-source on GitHub[1] under an Apache License. In this research work, *The Collector* is responsible for the full non-mobile data collection process.

The Collector went live on 24th July 2018, and has been collecting data uninterruptedly. It has been collecting data for five different Portuguese cities and places, in particular, Cabeceiras de Basto, Braga, Guimarães, Porto, and Lisboa. Nonetheless, more cities worldwide can be easily added to the fetch list. As soon as *The Collector* starts, it will autonomously collect data and dump it to a database shared with other software agents. The Collector works by making API calls, using Hypertext Transfer Protocol (HTTP) Get requests, every 120 minutes for pollution APIs, every 60 minutes for weather APIs, and every 20 minutes for the traffic ones. These timings are the default ones and all are runtime configurable. *The Collector* then parses the received JavaScript Object Notation (JSON) object and stores it in a MySQL database (version 14.14 and distribution 5.7.29) using an InnoDB engine (version 5.7.29).

At the time of writing, four distinct API clients are being queried. One, the *Pollution Client*, uses Open Weather Maps[2] and Open Air Quality[3] APIs. From the first, the agent obtains data about air pollution, including ozone indexes as well as UV, carbon monoxide, sulfur, and nitrogen dioxide ones. From the second, it extracts data such as black carbon, and particulate matters 2.5 and 10, among others. Table 2.1 describes the features available in the pollution dataset.

Table 2.1: Features of the pollution dataset.

#	Feature	Description
1	*id*	Record identifier
2	*pollution_type*	Possible values include BC, CO, NO_2, O_3, SO_2, PM10, PM2.5 and UV
3	*city_name*	Name of the city
4	*city_type*	Type of API used for the city
5	*seq_num*	Road identification (not used for pollution purposes)
6	*value*	Value of the *pollution_type*
7	*data_precision*	Precision of data
8	*source_name*	Source of data gathered by the API
9	*last_updated*	Date when the values were gathered by the API
10	*creation_date*	Timestamp (YYYY-MM-DD HH24:MI:SS)

On the other hand, the *Weather Client* uses Open Weather Maps API to obtain data about the weather including, among others, the weather description, temperature, humidity, atmospheric pressure, wind speed, and cloudiness. It is also obtains three days of weather forecasts (Table 2.2).

[1] github.com/brunofmf/SmartCity-API

[2] openweathermap.org/api

[3] docs.openaq.org

Table 2.2: Features of the weather dataset.

#	Feature	Description
1	*id*	Record identifier
2	*city_name*	Name of the city
3	*city_type*	Type of API used for the city
4	*seq_num*	Road identification (not used for weather purposes)
5	*weather_description*	Textual description of the weather
6	*temperature*	In degree Celsius
7	*atmospheric_pressure*	Atmospheric pressure at sea level (hPa)
8	*humidity*	In percentage
9	*wind_speed*	In meter/sec
10	*cloudiness*	In percentage
11	*rain*	Precipitation volume for the last hour (mm)
12	*calculation_date*	Date when the values were gathered by the API
13	*current_luminosity*	Current luminosity (categorical)
14	*sunrise*	Sunrise time (unix, UTC)
15	*sunset*	Sunset time (unix, UTC)
16	*forecast_date*	Forecast date (YYYY-MM-DD HH24:MI:SS)
17	*creation_date*	Timestamp (YYYY-MM-DD HH24:MI:SS)

The *Traffic Flow Client* uses TomTom Traffic Flow API[4] to obtain data about the flow of vehicles in several roads of the sensed cities. As features it includes the road identification, the functional road class describing the road type, the current speed in km/h at the observed road, the free-flow speed in km/h expected under ideal free-flow conditions, the current travel time in seconds, and the travel time in seconds that would be expected under ideal conditions, among others (Table 2.3).

Table 2.3: Main features of the traffic flow dataset.

#	Feature	Description
1	*id*	Record identifier
2	*city_name*	Name of the city
3	*city_type*	Type of API used for the city
4	*seq_num*	Road identification
5	*functional_road_class*	Road category description
6	*latitude*	Road latitude
7	*longitude*	Road longitude
8	*current_speed*	Current speed observed on the road (km/h)
9	*free_flow_speed*	Speed expected under ideal conditions (km/h)
10	*current_travel_time*	Current travel time observed on the road (s)
11	*free_flow_travel_time*	Travel time expected under ideal conditions (s)
12	*confidence*	API's confidence on the values
13	*realtime_ratio*	Ratio between live and historical data
14	*creation_date*	Timestamp (YYYY-MM-DD HH24:MI:SS)

[4] developer.tomtom.com/traffic-api/traffic-api-documentation/traffic-flow

The *Traffic Incidents Client* uses TomTom Traffic Incidents API[5] to obtain data about road incidents in cities. As incidents, we may consider jams, accidents, or closed roads, among others. As features, one may find the incident description, its category, the magnitude of the delay it is causing, the cause of the incident, the delay in seconds, and the length in meters of the traffic jam, among others (Table 2.4).

Table 2.4: Main features of the traffic incidents dataset.

#	Feature	Description
1	*id*	Record identifier
2	*city_name*	Name of the city
3	*city_type*	Type of API used for the city
4	*description*	Incident description
5	*cause_of_incident*	The cause of the incident
6	*from_road*	Road where the incident starts
7	*to_road*	Road where the incident ends
8	*incident_category*	Category of the incident
9	*magnitude_of_delay*	Magnitude of the delay
10	*length*	Length, in meters, of the traffic jam caused by the incident
11	*delay_in_seconds*	Delay, in seconds, caused by the incident
12	*affected_roads*	All roads affected by the incident
13	*latitude*	Latitude of the incident
14	*longitude*	Longitude of the incident
15	*incident_date*	Date of the incident (YYYY-MM-DD HH24:MI:SS)
16	*creation_date*	Timestamp (YYYY-MM-DD HH24:MI:SS)

In addition, as explained in future sections, *The Collector* is also responsible for collecting data in regard to Crowd Sensing. Appendix B depicts and describes *The Collector*'s data model.

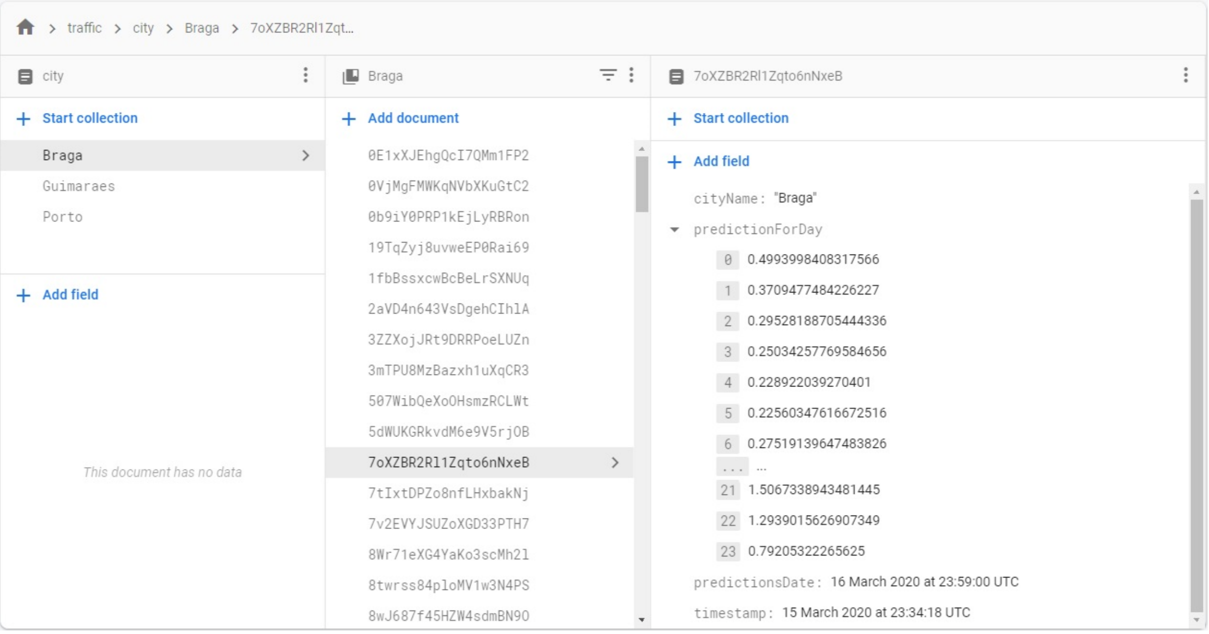
2.3.2 ML Architect

The *ML Architect* is the software agent responsible for using ML models to provide accurate forecasts and predictions, implementing Automated ML. It was developed using Python, version 3.7, and is remotely deployed in an Ubuntu 18.04 machine hosted at the University of Minho. The *ML Architect* holds all non-mobile ML models and uses real time data gathered by *The Collector* to obtain predictions and forecasts. As soon as the *ML Architect* intends to use a ML model, its first behavior is to obtain the required data. Then, it autonomously prepares the data as required by the model it aims to use. It then uses a pre-trained model to obtain forecasts, dumping the obtained values to a cloud platform.

The *ML Architect* uses a publisher-subscriber pattern, i.e., after executing the ML models and obtaining the respective forecasts, the model's output is then committed to a feed that is subscribed by other agents. In particular, the *ML Architect* uses Firebase's Cloud Firestore, a flexible and scalable database for mobile, web, and server development from Google's Cloud Platform. Cloud Firestore is a cloud-hosted,

[5] developer.tomtom.com/traffic-api/traffic-api-documentation/traffic-incidents

NoSQL database, with a data model that supports flexible and hierarchical data structures, with data being stored in documents that contain fields mapping values. Documents are, in turn, stored in collections, which are documents' containers used to organize data and build queries. Documents support several data types, including strings, numbers, and complex nested objects. Figure 2.3 depicts a view of the Graphical User Interface (GUI) of Firebase's Cloud Firestore containing a set of forecasts made by the *ML Architect* in regard to the status of the traffic flow for the city of Braga.



```
Braga
Guimaraes
Porto
```

```
0E1xXJEhgQcI7QMm1FP2
0VjMgFMWKqNVbXKuGtC2
0b9iY0PRP1kEjLyRBRon
19TqZyj8uvweEP0Rai69
1fbBssxcwBcBeLrSXNUq
2aVD4n643VsDgehCIhlA
3ZZXojJRt9DRRPoeLUZn
3mTPU8MzBazxh1uXqCR3
507WibQeXoOHsmzRCLWt
5dWUKGRkvdM6e9V5rjOB
7oXZBR2Rl1Zqto6nNxeB
7tIxtDPZo8nfLHxbakNj
7v2EVYJSUZoXGD33PTH7
8Wr71eXG4YaKo3scMh2l
8twrss84ploMV1w3N4PS
8wJ687f45HZW4sdmBN90
```

```
cityName: "Braga"
predictionForDay
    0   0.4993998408317566
    1   0.3709477484226227
    2   0.29528188705444336
    3   0.25034257769584656
    4   0.228922039270401
    5   0.22560347616672516
    6   0.27519139647483826
    ...  ...
    21  1.5067338943481445
    22  1.2939015626907349
    23  0.79205322265625
predictionsDate: 16 March 2020 at 23:59:00 UTC
timestamp: 15 March 2020 at 23:34:18 UTC
```

Figure 2.3: Firebase's Cloud Firestore GUI depicting a set of forecasts made by the *ML Architect*.

2.4 Traffic Flow Forecasting

Time series problems are among those who have significantly benefited from the progress of DL. In its essence, a time series problem consists in forecasting future values based on previous observations. Verily, there are an endless number of time series problems, with the major constraint being the availability of past observations. Until recently, the focus was on using statistical-based models such as the AutoRegressive Integrated Moving Average (ARIMA) or AutoRegressive Integrated Moving Average with Explanatory Variable (ARIMAX) to forecast future points [146, 147]. However, recent times came with very good results in regard to the use of DL models for time series forecasting, in particular RNNs [7, 8, 148]. The promise of RNNs is that the temporal dependence and contextual information in the input data can be learned and generalized to produce reliable predictions. LSTMs, a specific type of RNNs, are among those that have been showed to produce accurate results on time series data [6, 7].

Traffic flow forecasting is an acknowledged time series problem. Indeed, the possibility of knowing, beforehand, how the traffic flow will stand in the next minutes or hours would enable a driver to opt for a different road, a cyclist to opt for another hour to go cycling, or a pedestrian to choose a less polluted zone. However, due to the difficulty of having real contemporary data, the great majority of studies in this domain uses open datasets. This poses a problem due to the traffic dynamics of different countries, cities, and even roads. Secondly, the lack of real-time data prevents the models from being, indeed, deployed, and used to predict the traffic flow in a real-life scenario.

To overcome such problems, this research work uses data provided by *The Collector*, having the features depicted in Table 2.3. The goal is to conceive the best candidate model for traffic flow forecasting, which is then given to the *ML Architect* for real-time traffic flow forecasting. The focus is on state of the art LSTMs models, which are benchmarked against statistical-based ones, in particular, ARIMA and ARIMAX. Due to the novelty of the topics here addressed, the conducted experiments also aim to clarify several misconceptions shown in the literature that lead to important pitfalls. In particular, to clarify the importance of an adequate forecasting architecture and the tuning of LSTMs models in regard to the used time frames. Moreover, existing models do not clearly describe the method used for multi-step forecast, which may be indicative of forecasting errors [45]. Therefore, this study aims to give an answer to the following research questions:

RQ2.1. Do LSTMs have better accuracy than statistical-based models, such as ARIMA and ARIMAX, for traffic flow forecasting?

RQ2.2. Are LSTMs capable of accurately forecast the traffic flow of a road for multiple future timesteps (multi-step)?

RQ2.3. Do recursive multi-step LSTMs have better accuracy than multi-step vector output ones?

RQ2.4. Do multi-variate LSTMs have better accuracy than uni-variate ones?

2.4.1 State of the Art

2.4.1.1 ARIMA Models and LSTM Networks

ARIMA is a forecasting model originally developed by Box and Jenkins [149]. It belongs to a class of uni-variate auto-regressive algorithms used in forecasting applications based on time series data. ARIMA models are generally defined by the tuple (q, d, p), where q is the order of the auto-regressive components; d is the number of differencing operators; and p is the highest order of the moving average term. These parameters control the complexity of the model and, consequently, the auto-regression, integration and moving average components of the algorithm [150], i.e.:

$$\hat{y}_t = \Phi_1 \gamma_{t-1} + \Phi_2 \gamma_{t-2} + \cdots + \Phi_p \gamma_{t-p} + a_t - \Theta_1 a_{t-1} - \cdots - \Theta_p a_{t-q} \tag{2.1}$$

Where:

- y denotes a general time series;
- $\hat{y}_t$ is the forecast of the time series y for time t;

- y_{t-1}, ..., y_{t-p} are the previous p values of the time series y (and form the auto-regression terms);
- Φ_1, ..., Φ_p are coefficients to be determined by fitting the model;
- a_t, ..., a_{t-q} is a zero mean white noise process (and forms the moving average terms);
- Θ_t, ..., Θ_{t-q} are coefficients to be determined by fitting the model.

ARIMAX is considered an ARIMA extension to a multi-variable problem, i.e:

$$\phi(L)(1-L)^d Y_t = \theta(L)\varepsilon_t \tag{2.2}$$

The expression $\phi(L)$ represents the auto-regressive polynomial $1 + \phi_1 L + \cdots + \phi_q L^q$, $\theta(L)$ the moving average polynomial $1 - \theta_1 L - \theta_2 L^2 - \cdots - \theta_p L^p$ and L the lag operator. ARIMAX(p, d, q) extends ARIMA(p, d, q) equation, resulting in the following formulation:

$$\Phi(L)(1-L)^d Y_t = \theta(L)X_t + \theta(L)\varepsilon_t \tag{2.3}$$

X_t is the exogenous variable at time t. This model is adequate to forecast a stationary phenomenon with additional multi-variate data with context. ARIMA has been implemented on many domains such as temperature and pollution prediction [146] as well as short-term traffic flow forecasting [151]. On the same note, ARIMAX has been applied in scenarios where there is the need to use a multi-variate forecasting model. Regarding traffic flow forecasting, a comparison between ARIMA and ARIMAX is depicted in [152]. The obtained results demonstrate the better performance of ARIMAX and the use of additional contextual variables to achieve better forecasting accuracy [45].

Recent times came with very promising results in regard to the use of RNNs [6, 8, 45]. As opposed to classical ANNs, RNNs allow information to persist due to its recurrence and chain-like nature. The promise of RNNs, and LSTMs in particular, is that the temporal dependence and contextual information in the input data can be learned. LSTMs were introduced in 1997, by Hochreiter and Schmidhuber, but many more contributed to current LSTMs [153, 154]. LSTMs are a type of RNNs capable of learning long and short-term temporal dependencies. LSTMs computational units are called memory cells and have cell state. In addition, unlike typical RNNs, each memory cell of LSTMs contains additional neural network layers interacting with each other. These layers, or gates, give LSTMs the ability to further govern the information flow. Gates are composed of a sigmoid neural network layer, which specifies how much information the cell wants to forget (f_t), input (i_t) or output (o_t) at a timestep t, and a point-wise multiplication operation over the cell state:

- The forget gate, f_t, is the first gate inside the memory cell and decides which information the cell should discard from the cell state;
- The input gate, i_t, decides which values to add to the cell state;

- The output gate, o_t, decides what to output based on the input and the cell state.

Due to its characteristics, LSTMs have achieved important results in sequence problems such as text classification [155, 156], music generation [157, 158], handwriting recognition [159, 160], or speech recognition [161, 162], just to name a few. Traffic forecasting is yet another domain where LSTMs have been applied with some success [148, 163, 164].

2.4.1.2 The Literature

Recent studies have emerged where LSTMs are used for traffic flow forecasting. However, it is possible to verify that many exhibit flaws and leave untreated several important aspects of LSTMs. For instance, in [148], the authors conceived two distinct RNNs models for short-term traffic flow forecasting over the PeMS dataset, an open dataset describing the occupancy rate of several car lanes at the San Francisco bay area. Their experiments were based on an input sequence of six timesteps of five minutes each (thirty minutes) to predict the traffic flow for the next five minutes (single-step). No experimentation was performed in order to find the optimized values for any of the time parameters. No reference is made about the update pattern of the internal state of a memory cell.

In [163], the PeMS dataset was again used to develop LSTMs for short-term traffic flow forecasting. The focus is again to develop a single-step model. However, several figures depict predictions for a time window far superior (one day) without any explanation on the method that was used to evaluate this multi-step model. On the other hand, the authors clarify the features provided as input to the model. The authors opted to tune the number of memory cells of each layer and the size of the input layer. In regard to this last parameter, it is expected that the authors are indeed referring to the input shape of the first layer. Hence, the authors are tuning the number of timesteps that make an input sequence. All values from one to twelve were tried, meaning, in the first case, a sequence with just one timestep (five minutes of data) and, in the last case, a sequence with twelve timesteps (one hour of data).

In [164], the authors propose a deep stacked bidirectional LSTMs in order to consider both forward and backward dependencies in time series data. Their goal is to predict traffic speed using two distinct datasets, even though later the authors claim that only one was used to evaluate the model. To fully capture the spatial dimension of the problem, and since the dataset contains observations of multiple roads, the authors opted to develop a multi-variate, multi-road model receiving, as input, multiple observations from different roads at the same timestep. Each input sequence is composed of ten timesteps of five minutes, with a prediction being provided for the subsequent timestep (single-step). The number of timesteps in the input sequence was set based on some experiments, using six, eight, ten, and twelve timesteps. The obtained results were very similar, which can be explained by the fact that all attempted values are also very close. Indeed, it would be interesting to know how would the model behave with higher input timesteps that could represent, for example, an entire day instead of a few tens of minutes. No reference is made about the update pattern of the cell state.

In [165], the authors left aside traffic and focused on electric load forecasting using a uni-variate dataset collected by themselves. Interestingly, the authors of this work clearly describe the multi-step forecasting concept, having used a forecasting horizon of ninety-six timesteps. Such a long forecasting horizon directly affects the accuracy of the model. Nonetheless, the authors found that LSTMs still outperformed traditional forecasting methods, such as Seasonal ARIMA. On the other hand, input sequences were composed of ten days of timesteps. No details were provided regarding how this value was found.

Other studies focused on different deep learning models for time series forecasting. Indeed, new trends are emerging regarding the use of CNNs [166–168] and attention mechanisms [169, 170]. In [166], the authors focused on deep learning-based techniques, in particular RNNs and CNNs, for day-ahead multi-step load forecasting in commercial buildings, comparing the obtained results with ARIMAX. The gated 24-h CNNs based model was the one achieving the best results, improving ARIMAX's accuracy by 22.6%. On the other hand, in [168], the authors integrated, in a platform called DeepSense, both CNNs and RNNs. CNNs were responsible for learning intra-interval interactions, with the intra-interval representations along time being inputted to the RNNs. In [167] the authors proposed a deep learning fault detection approach based on CNNs, with the goal being to diagnose anomalies in the output of wind turbine systems. Time series data were converted as 2d images and fed to the conceived CNNs, with deeper CNNs outperforming shallow ones. On the other hand, attention mechanisms, initially applied to machine translation [171], have been growing in popularity for time series forecasting. Indeed, in [170] the authors propose a $Transformer$ that is solely based on attention mechanisms, discarding recurrence and convolutions. Such mechanisms allow models to selectively focus on some parts of the input data, which may be important when handling long input sequences. For instance, in [169], convolutional attention mechanisms based on the temporal output of CNNs were used to conceive a network that is able to convert variable-length time to a fixed-length low-dimensional time series representation [45].

2.4.2 Exploring spatial-temporal dependencies in time series

Traffic flow forecasting is a stochastic, non-linear time series problem, where space and time are of the utmost importance. For instance, no two countries are the same in regard to traffic dynamics. Even inside a country, cities and roads may have significantly different dynamics. Hence, it becomes extremely difficult, not to say unfeasible, to deploy a model to predict the traffic of several roads of a specific city when the model was trained on data from California roads, for example. To deploy accurate solutions that make real-time predictions, models require information on the space they are working. Hence, possible solutions would include the conception of road-specific models or the categorization of roads that share similar dynamics. Another possibility, even though computationally heavier, would be to create a multi-variate, multi-road model receiving, as input, multiple observations from different roads at the same timestep.

In typical MLPs, when tuning the network, the goal is to find the best set of hyperparameters that provide the best accuracy. It is usual to find models that have been tuned in regard to their depth, the number of neurons, the learning rate used by the optimizer, or the activation function. However, in time series problems, special attention should be given to all hyperparameters that are related to time. Tuning such hyperparameters assumes, as demonstrated by our experiments, an increased importance in a model's accuracy and performance.

There are, essentially, two main time hyperparameters to consider. The first is the number of timesteps that compose an input sequence. This assumes critical importance when performing Backpropagation Through Time (BPTT), a gradient-based technique that unfolds a RNNs in time to find the gradient of the cost, allowing LSTMs to learn from input sequences of timesteps. Let us consider, for example, an hourly dataset: if it is defined that each input sequence is composed of twelve timesteps, it means that each input sequence will correspond to twelve hours. If we set this value to twenty-four, then it will correspond to an entire day being used when computing. Longer input sequences were problematic for classical RNNs mainly due to the vanishing and the exploding gradient problems. LSTMs, on the other hand, are able to handle longer sequences with success. Alternatively, it is possible to use a truncated version of BPTT, which limits the number of used timesteps when finding the gradient of the cost. A second hyperparameter that may influence the model's accuracy is related to the reset frequency of the cell state. Indeed, memory cells are stateful. However, different libraries handle these states differently. For instance, by default, Keras assumes that internal states are reset after each batch. If one aims to maintain state between batches, one must explicitly define such behavior. Usually, such behavior is useful when it is assumed that information from past sequences may be useful for future sequences. Hence, some logic should be applied so that the model may understand patterns between input sequences and how they relate to each other. There is no obvious rule of thumb but some experimentation may be performed to find a tuned value for the problem, and data, in hands [45].

Finally, it should be noted that the goal of any model is to provide accurate and reliable forecasts. In time series problems, the goal is to forecast what will happen in a future time. Hence, models could be conceived to be single-step, i.e., provide a single forecast for the next immediate timestep, or multi-step, i.e., provide a set of forecasts for several future timesteps. Using the hourly dataset example, a single-step model that receives twelve input timesteps will give a forecast for the thirteenth timestep. A multi-step model would give forecasts for the thirteenth, fourteenth, and fifteenth timesteps, for example. If single-step models are fairly easy to conceive and evaluate, multi-step models are, on the other hand, harder. In essence, there are two main options to consider when conceiving multi-step models. The first is to have as many neurons in the output layer of the model as timesteps to forecast (Multi-Step Vector-Output). Hence, if one aims to forecast three future timesteps the model would have three output neurons. The second option would be to conceive a single-step model and recursively call it for as many future timesteps as one aims to forecast (Recursive Multi-Step). Again, as an example, let us consider the hourly dataset and an input sequence of twelve timesteps. First, we would conceive the model as single-step, i.e., the model would receive twelve hours as input and would output the thirteenth hour. Then, we would evaluate its performance, i.e., how accurate the model is in forecasting the thirteenth timestep. We would then push the forecast value of the thirteenth timestep to the input sequence, remove the oldest timestep (to ensure that the input sequence consists of just twelve timesteps) and give the updated input sequence to the model to get the forecast of the fourteenth timestep. We would keep repeating this process until the forecasting horizon is reached. This is what we call a *blind prediction* using a Recursive Multi-Step model. Obviously, this strategy suffers from the accumulation of errors with higher forecasting horizons [165]. However, there is no other valid way to evaluate such a model since the only way to know multiple future values is by using forecasts of the future itself. Literature is evasive in regard to the used methodology for forecasting. Nonetheless, as we show in the next sections, even though Multi-Step Vector-Output models are easier to conceive and evaluate, Recursive Multi-Step models perform better, both in terms of forecast accuracy and computational performance [45].

2.4.3 Data Preparation and Pre-processing

Data used in this study was made available by *The Collector*, having the features depicted in Table 2.2 and Table 2.3, i.e., weather and traffic flow features, respectively. Both datasets included observations from 24th July 2018 to 30th April 2019, in twenty minutes intervals. In the traffic flow dataset, as feature engineering, two more features were created:

- *speed_diff* feature, which corresponds to the difference between the speed, in km/h, that is expected under ideal conditions on a road and the speed, in km/h, that is being currently observed on that same road;
- *time_diff* feature, which corresponds to the difference between the travel time that is expected under ideal conditions on the road and the travel time that is being currently observed on that same road, in seconds.

The *speed_diff* feature will be the one used to quantify traffic. High *speed_diff* values mean that one is going much slower than what would be expected at that road, while low values mean that one is going almost at the ideal speed. A *speed_diff* of zero means no traffic for that road while a value of, for example, 40 km/h means that one is going 40 km/h slower than expected.

Then, to prepare the datasets, a specific road of the city of Braga was selected, with all other cities and roads being filtered out. The next step was to remove all static and non-informative features, such as the *id*, *city_type*, *seq_num*, *latitude*, *longitude*, and the *functional_road_class*, among others. The *time_diff*, *free_flow*, and *current_speed* features, which are highly correlated with the *speed_diff*, were also filtered out. Regarding the weather dataset, the only considered features were, besides the *creation_date* and *city_name*, the *temperature* and *precipitation*.

For both datasets, six new temporal and contextual features were created based on the *creation_date* of each observation. The *year*, *month*, *day*, *week_day*, *hour*, and *minutes* were the created features. The *minutes* feature was further normalized to have one of three possible values: 0, 20 or 40. Hence, all observations with *minutes* between $[0, 20[$ were normalized to belong to the 0 split, all that were between $[20, 40[$ belong to the 20 split and all those between $[40, 60[$ belong to the 40 split. Having now consistent dates, a date-time index was created. It became now possible to join both datasets by *city_name* and *index*. By the end of this step, we had a dataset, in ascending order by index, with five features, i.e., the *speed_diff*, *temperature*, *precipitation*, *week_day*, and *hour* (Table 2.5) [45].

Table 2.5: Features present in the dataset used for traffic flow forecasting.

#	Feature	Observation Example
0	*timestep index*	2019-04-23 17:40
1	*speed_diff*	32
2	*temperature*	10
3	*precipitation*	0
4	*week_day*	1
5	*hour*	17

No missing values were present. However, due to API limitations or to the fact that *The Collector* was down, there are missing timesteps. In a time series, missing timesteps may lead to the creation of incorrect patterns in input sequences. Consider, for example, the situation where a sequence contains observations for the fourth, fifth, and sixth hours of a day and then, due to the nonexistence of observations, it is followed by the fourteenth and fifteenth hours of that same day. This sequence would not describe the traffic pattern of a road. Different approaches may be followed to solve this situation. The approach followed in this work was to first include all missing timesteps with *NaN*, which means ”Not a Number”. After performing this operation, all timesteps, from July until April, were present in the dataset in twenty minutes intervals. Missing values, previously filled with *NaN*, were interpolated when the amount of consecutive missing observations corresponded to less than six hours for weather features and less than ten hours for traffic features. Weather features (*temperature* and *precipitation*) were linearly interpolated. On the other hand, *speed_diff* was computed based on the mean value of that same timestep at the three previous weeks. The dataset was iterated, observation per observation, from the oldest to the newest. In the case there was no information about the three previous weeks, the subsequent three weeks were the ones used. In the case of timestep gaps longer than ten hours, the entire day would be removed. Algorithm 2.1 describes, using pseudocode, the method that was developed for the computation of missing timesteps.

Considering that LSTMs work internally with the Hyperbolic Tangent (tanh), all features were normalized to fit into the interval $[-1, 1]$, i.e.:

$$\frac{x_i - \min(x)}{\max(x) - \min(x)} \tag{2.4}$$

Finally, the dataset was grouped by hour. The final multi-variate dataset contains 5050 timesteps, or observations, with a shape of (5050, 5). However, it was still not ready to be used by LSTMs networks. In order to create such networks one is required to set the problem as a supervised one, with inputs and corresponding labels. Hence, data were divided into smaller sequences, with its length depending on the number of input timesteps to be used. The label of each new sequence was the next immediate timestep, or a sequence of timesteps, depending if the model was to be Single-Step and Recursive Multi-Step, or Multi-Step Vector-Output, respectively (Algorithm 2.2).

Algorithm 2.1 Computation of missing timesteps.

```
  dataset.resample('20min')
  dataset['temperature', 'precipitation'].interpolate(method='linear', limit_direction='forward', limit='6h')
  *initialize* consequent_missing_obs
  for i, row ∈ enumerate(dataset, 0) do
      if row['speed_diff'] == NaN and consequent_missing_obs < 10h then
          *increment* consequent_missing_obs
          if i − (3 × one_week) > 0 then
              value_1w_before ⇐ dataset[i − one_week]
              value_2w_before ⇐ dataset[i − one_week × 2]
              value_3w_before ⇐ dataset[i − one_week × 3]
              row['speed_diff'] ⇐ mean(value_1w_before, value_2w_before, value_3w_before)
          else
              value_1w_after ⇐ dataset[i + one_week]
              value_2w_after ⇐ dataset[i + one_week × 2]
              value_3w_after ⇐ dataset[i + one_week × 3]
              row['speed_diff'] ⇐ mean(value_1w_after, value_2w_after, value_3w_after)
          end if
      else if row['speed_diff'] ≠ NaN then
          *reset* consequent_missing_obs
      end if
  end for
```

To evaluate the effectiveness of the candidate models, two error metrics were used. The first one corresponds the Root Mean Squared Error (RMSE), allowing one to penalize outliers and easily interpret the obtained results since they are in the same unit of the feature that is being predicted by the model. Its formula is as follows:

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^{n}(y_i - \widehat{y}_i)^2}{n}} \tag{2.5}$$

The second error metric corresponds to the Mean Absolute Error (MAE). It was mainly used to complement and strengthen the confidence on the obtained values. Its formulas is as follows:

Algorithm 2.2 From an unsupervised to a supervised problem.

```
Require: dataset, timesteps, multi_steps
Ensure: X, y
  X, y = list(), list()
  while i ∈ range(len(dataset) − (timesteps + multi_steps)) do
      input_index ⇐ i + timesteps
      label_index ⇐ input_index + multi_steps
      X.append(dataset[i:input_index, :])
      y.append(dataset[input_index:label_index, speed_diff])
  end while
```

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n} |y_i - \widehat{y}_i| \tag{2.6}$$

All the developed models were evaluated in regard to the mentioned metrics. A time series cross-validator was also used to provide train/test indices. In this case, the k^{th} split returns the first k folds as train set and the $(k+1)^{th}$ fold as test set. Unlike standard cross-validation methods, successive training sets are supersets of those that came before. Each training set was further split into training and validation sets in a ratio of 9:1. A forecast function was also created to predict three days of the last test set (72 timesteps). The predicted value is compared with the real value in order to compute both error metrics. In the case of Recursive Multi-Step models, blind forecasting pushes prediction values to the input sequence to forecast subsequent timesteps [45].

2.4.4 Experiments

The main goal of this research study was to develop and tune the best possible LSTMs for traffic flow forecasting while addressing important pitfalls. All candidate models were conceived as multi-step, i.e., all models aim to forecast twelve consecutive hours.

Python, version 3.6, was the used programming language for data preparation and pre-processing as well as for model development and evaluation. Pandas, NumPy, scikit-learn, matplotlib, and statsmodels were the used libraries. Tensorflow v1.12.0 was the machine learning library used to conceive the deep learning models. tf.keras, TensorFlow's implementation of the Keras API specification, was also used. For increased performance, and to fit the models in reasonable times, Tesla T4 GPUs were used as well as an optimized version of LSTMs using CuDNN (CudnnLSTM), NVIDIA's deep neural network library of primitives for deep neural networks. It is worth mentioning that all this hardware was made available by Google's Colaboratory.

Due to the random initialization of LSTMs' weights, multiple repetitions of each combination of hyperparameters were performed on incremental sets of data, taking the mean RMSE and MAE to verify the model's quality. Several experiments were performed to find the best set of hyperparameters for the final model. An experiment was also set where ARIMA models are deployed, evaluated, and later compared with LSTMs.

Algorithm 2.3 details the generic function that was written for the conception and compilation of the LSTMs. This function is used in all experiments that use LSTMs, with the inputs of the function defining the model's structure and time frames [45].

2.4.4.1 ARIMA and ARIMAX

When compared to LSTMs, ARIMA and ARIMAX are considered more traditional approaches to time series forecasting. The literature shows several examples of their usage. Therefore, ARIMA and ARIMAX models shall be used as benchmarks for the developed LSTMs. Hence, using the collected data, models were

Algorithm 2.3 LSTM model's conception and compilation.

```
Require: timesteps, multisteps, features, h_layers, h_neurons, activation, dropout_rate, deep_dense
Ensure: Sequential LSTM Model
  model $\Leftarrow$ Sequential()
  while i $\in$ range(h_layers) do
      if i == 0 then
          if i+1 == h_layers then
              model.add(LSTM(h_neurons, return_seq=False, input_shape=(timesteps, features)))
          else
              model.add(LSTM(h_neurons/2, return_seq=True, input_shape=(timesteps, features)))
              model.add(Dropout(dropout_rate))
          end if
      else if i+1 == h_layers then
          model.add(LSTM(h_neurons$\times$2, return_seq=False))
      else
          model.add(LSTM(h_neurons, return_seq=True))
          model.add(Dropout(dropout_rate))
      end if
  end while
  model.add(Dense(h_neurons, activation=activation))
  model.add(Dropout(dropout_rate))
  if deep_dense then
      model.add(Dense(h_neurons/2, activation=activation))
      model.add(Dropout(dropout_rate))
  end if
  model.add(Dense(multisteps))
  model.compile(loss=rmse, optimizer=Adam(), metrics=[mae, rmse])
```

trained in order to mimic the conditions in which LSTMs were trained, i.e., using ARIMA for uni-variate and ARIMAX for multi-variate problems.

The experiments made with the ARIMA model used a grid search approach to fine tune the (p, d, q) parameters. For a better comparison between ARIMA and LSTMs, the p parameter was based on large windows ($[12, 8]$), with LSTMs models achieving even higher windows. The remaining parameters were computed from the range $[0, 3]$. An obvious limitation of this approach is the unfeasibility of training models with some specific parameters due to internal errors of the ARIMA algorithm. These errors are essentially related with stationary data. ARIMAX followed the same approach, however contextual data were added through the use of exogenous variables. The fine tune of the (p, d, q) parameters was the same as with ARIMA. The features introduced as contextual data included the precipitation and weekday, for each timestep.

In order to tackle multi-step predictions, a forecasting method was used, which allows to directly forecast multiple instances from the last point in the train dataset. This is consistent with the blind multi-step forecasting approach. For a multi-step approach incorporating real observations, the model needs to be re-trained after each dataset change. The time used to train an ARIMA or ARIMAX model is usually faster than LSTMs, but in the case of forecasting one step ahead with real observations, the need for retrain at each timestep, and the number of experiments being performed, made the ARIMA and ARIMAX slower than the worst performing LSTMs. For simplicity, only blind multi-step forecasting models were considered [45].

2.4.4.2 Recursive Multi-Step vs Multi-Step Vector Output LSTMs

The goal of this experiment was to compare the performance, both computationally and in terms of accuracy, of two distinct multi-step approaches. It soon became obvious that the Multi-Step Vector Output model would require a more complex architecture. That came, however, with a significantly higher computational and training time cost. Therefore, for this experiment, only uni-variate models were considered. Both models receive input sequences of one feature, i.e., $speed\_diff$, and provide $speed\_diff$ forecasts for the next 12 hours. Random search was used to reduce the hyperparameters' configuration space.

Recursive Multi-Step LSTMs were conceived and tuned in regard to a set of hyperparameters. As explained before, such models predict the next immediate timestep, being then called recursively twelve times. RMSE and MAE values are gathered both for blind and non-blind predictions. Non-blind predictions use real values when iterating recursively. Obviously, non-blind predictions produce a better result than blind ones since this last approach suffers from the accumulation of errors with higher forecasting horizons. This serves the purpose of showing that misconceptions may lead researchers to present models that have been inappropriately evaluated. Nonetheless, blind predictions are the ones to be considered. Table 2.6 describes the hyperparameters searching space considered for the candidate models.

Table 2.6: Recursive Multi-Step uni-variate hyperparameters' searching space.

Parameter	**Searched Values**	**Rationale**
Epochs	[200, 300, 500]	-
Timesteps	[12, 24, 48, 96]	Input of 0.5 to 4 days
Batch size	[24, 168, 252, 336, 672]	1 day to 4 weeks
LSTM layers	[3, 4, 5]	Number of LSTM layers
Dense Layers	1	Number of dense layers
Dense Activation	[ReLU, tanh]	Activation function
Neurons	[32, 64, 128]	For dense and LSTM layers
Dropout rate	[0.0, 0.2, 0.5]	For dense and LSTM layers
Learning rate	Tuned via callback	Keras callback
Multisteps	12	12 hours
Features	speed_diff	Uni-variate
CV Splits	3	Time series cross-validator

As soon as the first Multi-Step Vector Output candidate model started its training, it became obvious that, computationally, it would be much expensive and it would take a considerable amount of time to go through all hyperparameters' possibilities. Therefore, a smaller hyperparameters searching space was considered (Table 2.7). Intuition and random search helped reduce the number of fits performed and the amount of time it took to gather all results.

Table 2.7: Multi-Step Vector Output uni-variate hyperparameters' searching space.

Parameter	**Searched Values**	**Rationale**
Epochs	[500, 700]	-
Timesteps	[48, 96]	Input of 2 and 4 days
Batch size	[168, 336, 672]	1, 2 and 4 weeks
LSTM layers	[4, 5]	Number of LSTM layers
Dense Layers	[1, 2]	Number of dense layers
Dense Activation	[ReLU, tanh]	Activation function
Neurons	[64, 128, 256, 512]	For dense and LSTM layers
Dropout rate	[0.2, 0.5]	For dense and LSTM layers
Learning rate	Tuned via callback	Keras callback
Multisteps	12	12 hours
Features	speed_diff	Uni-variate
CV Splits	3	Time series cross-validator

2.4.4.3 Uni-Variate vs Multi-Variate LSTMs

This experiment aimed to compare the performance, both computationally and in terms of accuracy, of uni-variate and multi-variate LSTMs. The first type, uni-variate, refers to models that use a single feature, while multi-variate ones use multiple features, with the expectation of accurately generalize the problem in hands. Both uni-variate and multi-variate models were conceived as Recursive Multi-Step. This decision was based on the fact that Recursive Multi-Step forecasting was shown to be less expensive and more accurate than Multi-Step Vector Output. Intuition and random search were, again, used.

For the multi-variate experiments, from the features considered in Table 2.5, *temperature* was removed and experiments were initially made with the *speed_diff*, *precipitation*, and *week_day* features. Later, the *hour* was also added. Again, it soon became obvious that the model was becoming more expensive to train and that it was requiring a more complex architecture. However, making the model deeper came with a considerable time cost. Table 2.8 describes the hyperparameters searching space used for the conception of the Recursive Multi-Step multi-variate LSTMs candidate models [45].

2.4.5 Results and Discussion

The conceived models where evaluated in regard to their RMSE and MAE. A time series cross-validator was used to provide train and test indices to split time series data. Each training set was further split into training and validation sets in a ratio of 9:1. The results presented in the following lines are the output of a forecast function that was developed to plot and predict three entire days of the test set (72 timesteps).

2.4.5.1 ARIMA Models vs LSTM Networks

ARIMA based-models were used as a benchmark. Through this analysis, it was possible to observe that ARIMA and ARIMAX models tend to be unreliable when addressing a dynamic problem such as traffic flow in an open data stream format. Issues when training these models also emerged. For instance, if the

Table 2.8: Recursive Multi-Step multi-variate hyperparameters' searching space.

Parameter	**Searched Values**	**Rationale**
Epochs	[500, 700]	-
Timesteps	[12, 24, 48, 96]	Input of 0.5 to 4 days
Batch size	[84, 168, 252, 672]	0.5 to 4 weeks
LSTM layers	[2, 3, 4, 5, 6]	Number of LSTM layers
Dense Layers	[1, 2]	Number of dense layers
Dense Activation	[ReLU, tanh]	Activation function
Neurons	[32, 64, 128]	For dense and LSTM layers
Dropout rate	[0.2, 0.5]	For dense and LSTM layers
Learning rate	Tuned via callback	Keras callback
Multisteps	12	12 hours
Features	speed_diff, precipitation, week_day and hour	
CV Splits	3	Time series cross-validator

dataset of past observations is constantly updated then there is a need to re-train models every time such events happen.

In terms of accuracy, both ARIMA and ARIMAX can yield good results for smaller forecasts. However, the mean average of metrics such as MAE and RMSE is frequently higher. These models behave specially badly when the series incurs, for some time, in stagnant data, i.e., when there is no speed variation for several timesteps. Figure 2.4 depicts the best multi-step predictions using these statistical-based models. In Table 2.9, a summary of the best model specification for both models is depicted. The higher the order we can achieve with the arguments (p, d, q) the higher the performance. On the other hand, as we increase these parameters values so does increase the number of times the models cannot be computed and the time needed to train the model. The presence of contextual data makes the ARIMAX model slightly more accurate. On the other hand, ARIMAX models are costly to train as the computational overhead by the contextual data are directly translated to an increase in training times. Contrary to what may be expected, ARIMAX breaks less often than the standard uni-variate ARIMA.

Table 2.9: Top-four ARIMA and ARIMAX uni-variate model parameters' configuration.

Moving Average (p)	**Dif. Operators (d)**	**Auto-regressive (q)**	**RMSE**	**MAE**
		$ARIMA$		
12	1	2	**6.336**	**5.088**
12	1	1	6.452	5.200
8	1	2	7.967	6.275
8	1	1	8.869	7.474
		$ARIMAX$		
12	1	2	**6.110**	**4.918**
12	1	1	6.171	4.972
8	1	2	8.301	6.822
8	1	1	8.325	6.736

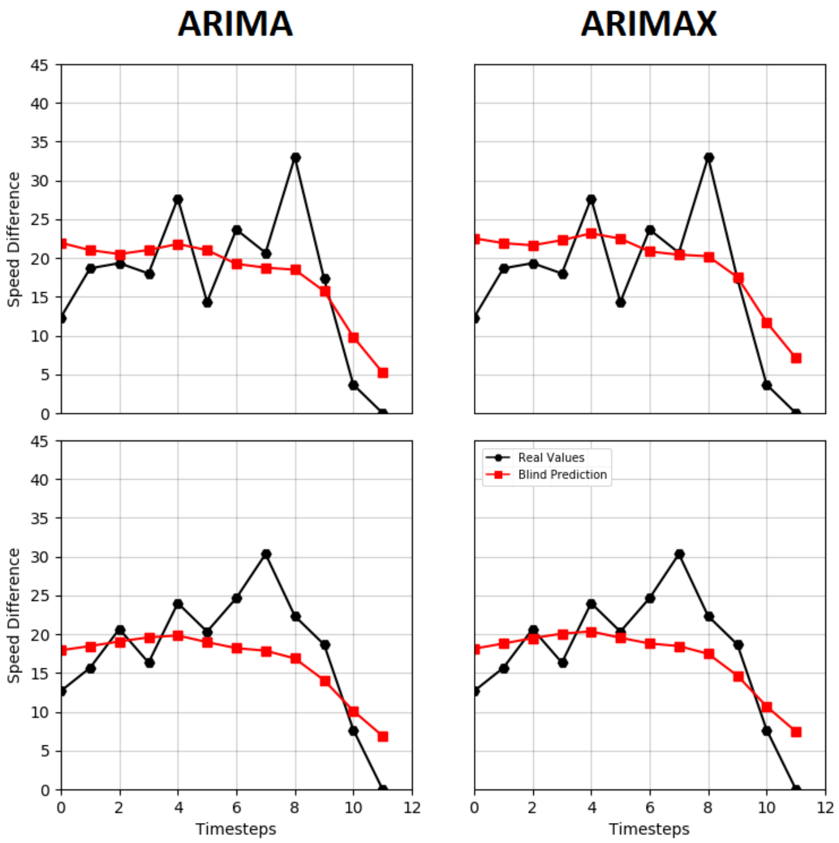


Figure 2.4: Two random blind multi-step predictions for the best ARIMA and for the best ARIMAX model.

When comparing ARIMA-based models with LSTMs ones, it is worth mentioning that LSTMs can achieve higher forecasting performances and produce stable results despite data properties. On the other hand, ARIMA-based models need to enforce data to be stationary, which might not be possible in continuous streams of data. Regarding training times, ARIMA and ARIMAX have short training times but require re-training when new observations are added to the dataset and the forecast should occur after the latest observation. LSTMs do not share this problem as they can forecast the next sequence of data without the need to retrain the network. This detail makes LSTMs better suited for real-time applications, where pre-trained models can deliver high accuracy results without constant need for re-training.

2.4.5.2 Recursive Multi-Step vs Multi-Step Vector Output LSTMs

Recursive Multi-Step LSTMs had significantly lower RMSE and MAE values when compared to Multi-Step Vector Output ones. As depicted in Table 2.10, the best Recursive Multi-Step LSTMs had a RMSE of 3.496 and a MAE of 1.567 while the best Multi-Step Vector Output LSTMs had a RMSE of 5.040 and a MAE of 1.846. Indeed, after more than 200 experiments, only one Recursive Multi-Step LSTMs had worst accuracy than the best Multi-Step Vector Output one.

It is worth highlighting that vector output models have as many neurons in the output layer as timesteps to forecast. Intuitively, this would lead the model to demand a more complex architecture when compared to those that have a single neuron as output. This can be indeed verified in the performed experiments.

Table 2.10: Recursive Multi-Step vs Multi-Step Vector Output LSTMs top-five results.

#	Timesteps	Batch	Layers	Neurons	Dropout	Act.	RMSE	MAE
				Recursive Multi-Step Uni-Variate				
209	96	672	5	64	0.2	relu	**3.496**	**1.567**
188	96	672	4	64	0.2	tanh	3.518	1.567
95	96	252	5	32	0.2	relu	3.555	1.592
195	96	672	4	128	0.5	tanh	3.583	1.598
12	48	252	3	64	0.5	relu	3.649	1.629
				Multi-Step Vector Output Uni-Variate				
24	96	672	4	128	0.5	relu	**5.040**	**1.846**
31	96	672	5	128	0.2	relu	5.134	1.839
29	96	672	5	128	0.2	tanh	5.272	1.852
20	48	168	5	512	0.5	relu	5.279	1.880
30	96	672	5	128	0.5	tanh	5.325	1.885

The first round of experiments with vector output models limited the maximum number of neurons to 128 and the number of dense layers to 1. This maximum number was the one used by all the best models. Then, the second round of experiments with vector output models considered a higher number of neurons, layers, and training epochs. However, very few experiments were performed with combinations of 256 and 512 neurons, 4 and 5 hidden LSTMs layers, and 2 dense layers. This is related to the fact that each candidate model was taking more several hours to train.

It can be concluded that deeper and more complex architectures could improve the performance of Multi-Step Vector Output models. This comes, however, with a significant increase in training times. On the other hand, Recursive Multi-Step LSTMs were between eight and sixteen times faster to train, required a shallower architecture, and produced results that are more than 40% better. It is also interesting to note that the best models were the ones using a bigger batch size combined with input sequences of four entire days (96 timesteps). There was no clear distinction between using Rectified Linear Unit (ReLU) or tanh. The same argument is applied for dropout values of 20% and 50%.

Table 2.10 describes the top-five results achieved with each multi-step approach. The best Recursive Multi-Step LSTMs had a RMSE of 3.496, which means that such a model is able to forecast, by a margin of around 3 km/h, the expected speed difference at a road for each one of the next twelve hours. These results prove the feasibility of using LSTMs networks for accurate multi-step prediction. Figure 2.5 presents six random multi-step predictions (12 timesteps) of the best Recursive Multi-Step LSTMs. On the other hand, Figure 2.6 presents six random multi-step predictions for the best Multi-Step Vector Output LSTMs.

2.4.5.3 Uni-Variate vs Multi-Variate LSTMs

An increase of the multitude of input features led to a decrease of the RMSE and MAE values. As depicted in Table 2.11, the best uni-variate model had a RMSE of 3.496 and a MAE of 1.567 while the best multi-variate LSTMs had a RMSE of 2.907 and a MAE of 1.346, which corresponds to a decrease of more than

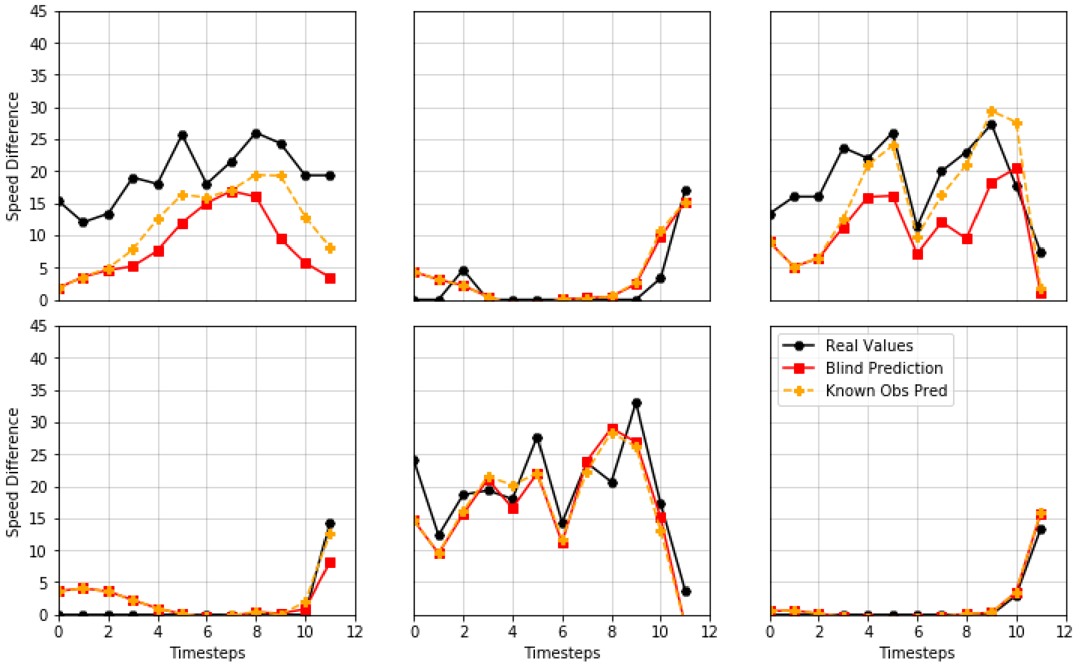


Figure 2.5: Six random multi-step predictions of the best Recursive Multi-Step Uni-Variate LSTM model (#209). Comparison of real values vs predicted ones using blind forecasting vs predicted ones using known observations.

20% of the RMSE metric. It is worth noting that including the hour of the day as input feature allowed the model to make more accurate forecasts. On the other hand, the presence of the precipitation feature led to worst results. Moreover, it is interesting to note that the presence of more input features led to a decrease in the number of input timesteps. Indeed, the best uni-variate models required 96 input timesteps, while

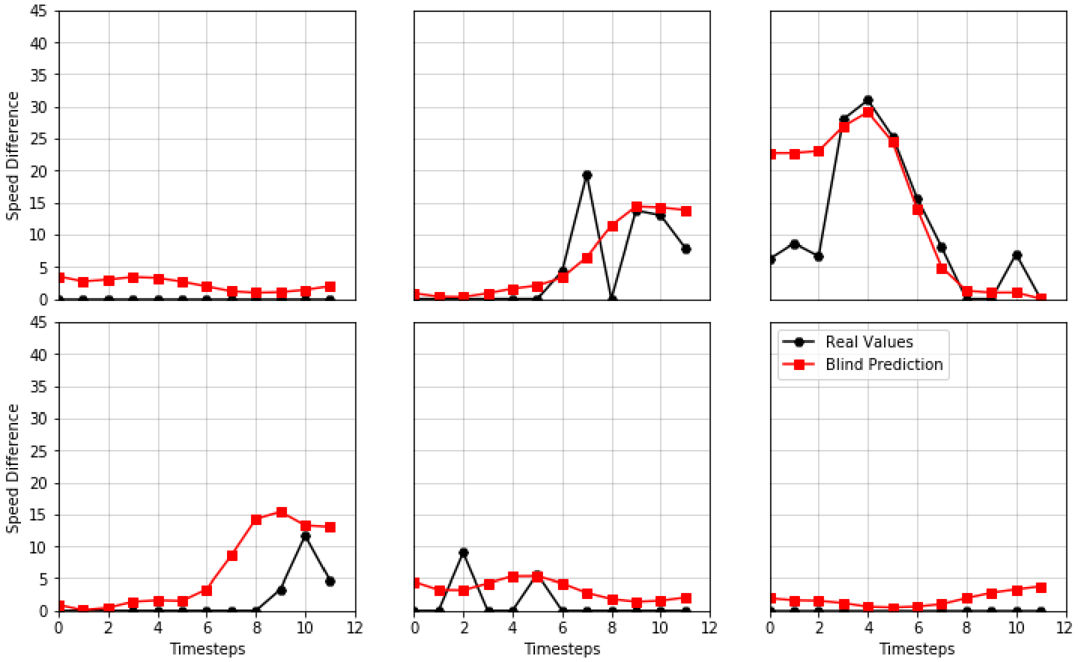


Figure 2.6: Six random multi-step vector output predictions of the best Multi-Step Vector Output Uni-Variate LSTM model (#24). Comparison of real values vs predicted ones using vector output forecasting.

Table 2.11: Uni-variate vs multi-variate LSTMs top-five results.

#	Timesteps	Batch	Layers	Neurons	Dropout	Act.	RMSE	MAE
				Recursive Multi-Step Uni-Variate				
209	96	672	5	64	0.2	relu	**3.496**	**1.567**
188	96	672	4	64	0.2	tanh	3.518	1.567
95	96	252	5	32	0.2	relu	3.555	1.592
195	96	672	4	128	0.5	tanh	3.583	1.598
12	48	252	3	64	0.5	relu	3.649	1.629
				Recursive Multi-Step Multi-Variate				
53*	24	168	4	64	0.5	tanh	**2.907**	**1.346**
24*	24	96	4	32	0.5	tanh	3.006	1.412
16*	24	48	4	32	0.5	tanh	3.031	1.419
17*	24	168	5	64	0.5	tanh	3.037	1.402
37*	48	84	2	64	0.5	tanh	3.038	1.425
*	Used features: *speed_diff*, *week_day* and *hour*.							

the best multi-variate models require just 24 input timesteps. In simple terms, while the uni-variate models require four days of input to forecast the next twelve hours, the multi-variate ones require just a single day as input to forecast the same amount of hours. The batch size required by multi-variate models is also substantially lower when compared to uni-variate ones. Regarding the activation function, while there is no clear distinction between ReLU and tanh in uni-variate models, it is clear that tanh performs better in multi-variate ones with a dropout of 50%.

Even though multi-variate models took longer to train than uni-variate ones, such variation is negligible. Moreover, the addition of more features to the model allowed it to perform better than an uni-variate one. Indeed, the addition of the hour of the day was the factor that allowed the candidate models to achieve lower error values. Table 2.11 describes the top-five results achieved with each approach. The best Recursive Multi-Step multi-variate LSTMs is able to predict, by a margin inferior to 3 km/h, the expected speed difference at a road for each one of the next twelve hours. Figure 2.7 presents six random multi-step predictions (12 timesteps) of the best Recursive Multi-Step multi-variate LSTMs.

2.4.5.4 Conclusions

From all candidate models, and after several weeks of combined training times, the model that was able to more accurately forecast the traffic flow of a road for multiple future timesteps was the fifty-third Recursive Multi-Step multi-variate candidate model, with a RMSE and MAE of 2.907 and 1.346, respectively. Uni-variate models also presented interesting results, with its best model being 20% worst than the best multi-variate one. On the other hand, the best Vector Output model had a RMSE that was more than 73% worst than the best multi-variate model and 40% worst than the best uni-variate. In addition, Vector Output models took significantly more time to train when compared to the other models. The training time difference between the uni-variate and the multi-variate models can be considered negligible. On the other hand,

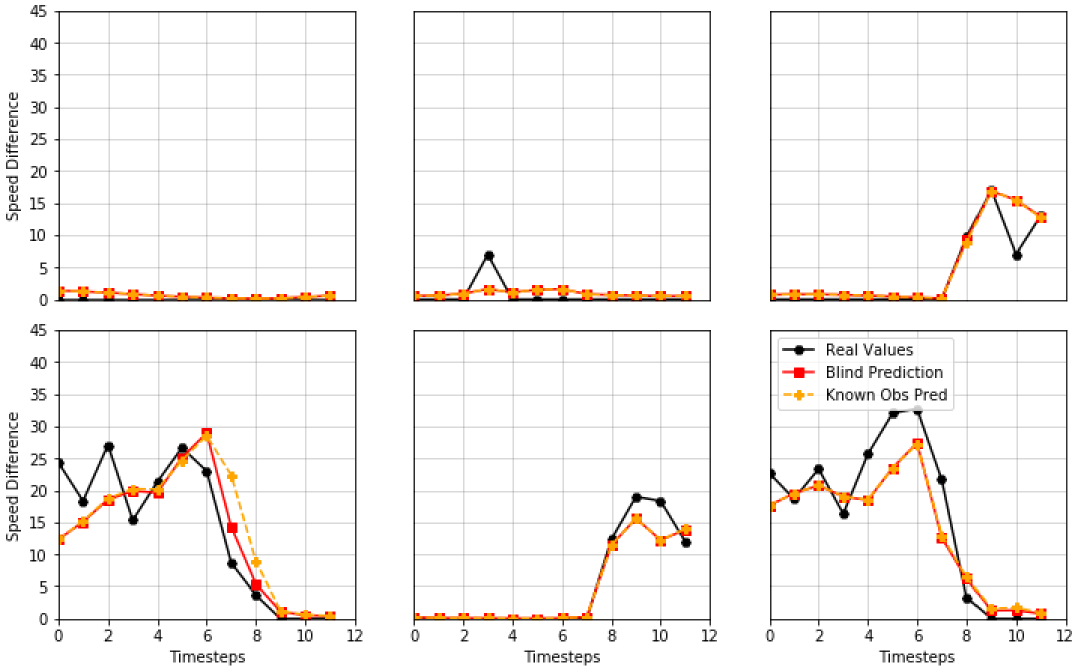


Figure 2.7: Six random multi-step predictions of the best Recursive Multi-Step Multi-Variate LSTM model (#53). Comparison of real values vs predicted ones using blind forecasting vs predicted ones using known observations.

ARIMA and ARIMAX presented results that are two times worse than the best LSTMs. In addition, the fit frequency of ARIMA-based models is significantly higher when compared to LSTMs, which may be problematic in open data streams scenarios. All this is in agreement with what the literature has been presenting, confirming that these statistical-based models have now been superseded by deep learning ones. Table 2.12 summarizes the achieved results.

Table 2.12: Summary results for the best model for each approach.

Model	**RMSE**	**MAE**
Recursive Multi-Step Multi-Variate	**2.907**	**1.346**
Recursive Multi-Step Uni-Variate	3.496	1.567
Multi-Step Vector Output Uni-Variate	5.040	1.846
ARIMAX (Multi-Variate)	6.110	4.918
ARIMA (Uni-Variate)	6.336	5.088

As expected, the time frames impact the model's accuracy. The obtained results show that the number of input timesteps, as well as the batch size, may affect accuracy significantly. It was interesting to note that the presence of more input features led to a decrease in the number of input timesteps required by the model. While the best uni-variate models required four days of input, the multi-variate ones require just a single day. The batch size required by multi-variate models is also substantially lower when compared to uni-variate ones. As direct answers to the elicited research questions, it can be said that (RQ2.1) LSTMs have significantly better forecasting accuracy than ARIMA-based models; (RQ2.2) LSTMs are able to accurately forecast several future timesteps; (RQ2.3) it has been demonstrated that recursive multi-step LSTMs have better accuracy than multi-step vector output ones, with these last ones requiring a more complex and deeper architecture, which, in turn, increases training times; and (RQ2.4) the addition of more input features, namely the day of the week and the hour of the day, allowed LSTMs to behave 20% better when compared to the uni-variate ones.

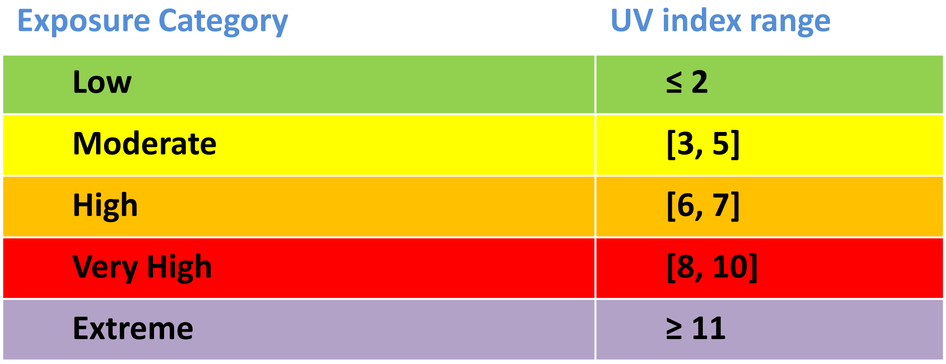
2.5 UV Index Forecasting

Recent times came with an increased importance being given to sustainability, in several domains. In fact, sustainability refers to the assurance of steadiness, viability, and efficiency in any domain, i.e., the ability to meet present needs without compromising future generations [37]. The three main pillars of sustainability are the social, economic, and environmental ones [172], with many stakeholders being focused in implementing sustainable transportation systems [173, 174], smart agriculture [175, 176], sustainable houses [177], and sustainable energy recovery from municipal solid wastes [178], just to name a few.

The work here presented focuses on environmental sustainability. In particular, it focuses on forecasting the UV index to provide actionable information to people, i.e., having access to such forecasts allow VRUs to adapt their behavior on the road, avoiding solar exposure in dangerous days or by taking other protective measures [179]. In fact, a significant increase in the incidence of skin cancers has been observed since the early 1970s, being strongly correlated to sun exposure and the thought that a tan is desirable [180]. The most effective strategy to control skin cancer is to avoid sun exposure during peak UV levels [179]. Other primary prevention strategies include the use of protective clothing, sunscreen protection, and avoiding artificial tanning devices [181]. In fact, it is believed that four out of five skin cancer cases could be prevented since UV damage is highly avoidable [182].

The sun emits several types of radiation that are characterized by their wavelength. Such emissions include UV, visible light, and heat, among others. UV covers the wavelength range between 100–400 nm, being split into three bands: UVA in the range 315-400 nm, UVB in the range 280-315 nm, and UVC in the range 100-280 nm. UV radiation reaching the Earth's surface is almost entirely composed of UVA radiation, since UVC is fully absorbed when passing through the atmosphere and approximately 90% of all UVB radiation suffers the same fate. UVA can penetrate into tissue, being less biologically damaging than UVB [180]. UV radiation levels are influenced by sun elevation, latitude, cloudiness, altitude, ozone levels, and ground reflection. Despite harmful at high concentrations, UV radiation is still essential to the humanity. For instance, small amounts of UV radiation are essential for the production of vitamin D, a key regulator in calcium and phosphate homeostasis [183]. UV radiation, under medical supervision, is also used to treat rickets, psoriasis, and eczema [180].

The UV index is a standard metric describing the magnitude of solar UV radiation reaching Earth's surface at a particular time, at a given region [180]. It has been adopted and standardized by the WHO and the World Meteorological Organization. In its essence, the higher the index value, the greater the risk for damage. The UV index should be presented as a value rounded to the nearest integer, with values between $[0, 2]$ describing low risk; between $[3, 5]$ describing moderate risk; between $[6, 7]$ start carrying some risk; between $[8, 11]$ being very dangerous to the human being; and values higher than 11 being of extreme danger (Figure 2.8). When cloudiness is variable, the UV index should be presented as a range of values. Forecasts not considering cloud effects should refer to themselves as "clear sky" or "cloud free" UV index forecasts [180].



Exposure Category	UV index range
Low	≤ 2
Moderate	[3, 5]
High	[6, 7]
Very High	[8, 10]
Extreme	≥ 11

Figure 2.8: UV index exposure categories. Colors are associated to each category. Adapted from [180].

Radiation levels differ throughout the day. However, most reports are focused on depicting the maximum UV radiation level on a given day, which occurs during the four-hour period around the solar noon [180]. Hence, this study makes use of data gathered by *The Collector* to conceive the best possible candidate model to forecast the maximum daily "clear sky" UV index, for the next three days. State of the art LSTMs models are conceived, tuned, and evaluated, based on the results obtained from the traffic flow forecasting experiment, which allowed us to perceive the better performance of Recursive Multi-Step Multi-Variate networks. This study aims to give an answer to the following research question:

RQ2.5. Are LSTMs networks capable of accurately forecast the maximum daily "clear sky" UV index for multiple future timesteps?

2.5.1 State of the Art

The literature depicts two main UV index forecasting methods: one based on solar total radiance and other environmental parameters, and another based on previous UV index values. The second method has been shown to produce better results, with the first being more suitable for epidemiological studies [184]. In this study, we are particularly interested in framing UV index forecasting as a time series problem, where one considers a series of past UV points ordered in time.

UV index forecasting is essentially performed through means of static mathematical models. It is interesting to note that online platforms, such as *Open Weather Maps* or *OpenUV*, provide several days of UV index forecasts. However, many of them do not disclose how forecasts are performed and which features are used. On the other hand, scientific literature using ML or DL-based models is practically non-existent. One example is the work of Deo et al. (2017), where the authors focused on short-term UV index forecasting (10 minutes) [185]. The authors employed an Extreme Machine Learning (EML) model based on a shallow ANNs, using RMSE and MAE as evaluation metrics. The conceived model outperformed mathematical based ones [185].

A different study proposed a DL model to forecast UV index with the goal being to predict erythema formation in the human skin, which corresponds to skin redness due to the dilation of superficial capillaries [186]. The conceived model receives, as input, the temperature, cloudiness, insolation, and the UV index of the day before the one it aims to forecast. The used dataset contained records extracted from the National Aeronautics and Space Administration. Models achieved an overall accuracy of 66.8% of correct classifications [186].

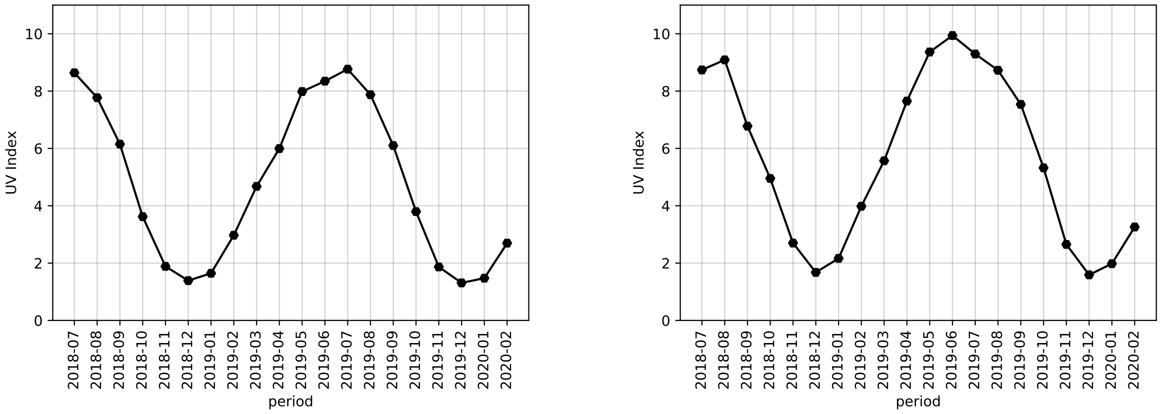
In 2018, Purananunak et al. proposed an UV index monitoring, alert, and prediction system. To forecast future timesteps the authors employed a linear regression model using the Weka software. The conducted experiments were poorly explained, not describing how were the models evaluated, which metrics were used, and how the tuning was performed [187]. In 2017, Mei et al. (2017) proposed the use of smartphone cameras to measure UV radiation, developing a mobile application for such end [188]. The rationale is that sensors inside smartphone cameras are very sensitive to UV radiation. The authors conducted several experiments, under different conditions, evaluating the feasibility of using smartphones as UV sensors. In fact, results showed that the proposed procedure achieved an average 95% accuracy when compared to professional digital UV meter, being significantly better than UV sensors equipped on smartwatches. No forecasts were performed.

2.5.2 Data Preparation and Pre-processing

Data used in this study were made available by *The Collector*, having the features depicted in Table 2.1, i.e., pollution features. Since only pollution data are used, from this point forward, forecasts are to be understood as "clear sky" UV index forecasts [180]. The used dataset includes 16375 observations collected by three hard sensors from 24th July 2018 to 24th February 2020. Only observations with *pollution_type* as "Ultraviolet" were considered. The case study focused on one Portuguese city. Observations from all other cities were discarded. Since at the same timestep several UV measures are made available by distinct sensors, data were grouped by *creation_date* to contain the mean UV value at each timestep. Data were further grouped, to contain the maximum UV value per day for the selected city.

Features without any value assigned or always filled with the same value, such as the *last_updated* or *data_precision* ones, were removed. In fact, since the goal is to forecast the UV index using previous UV values as input, all features were removed except for the *value* and *creation_date* ones. Based on the latter, a feature engineering process allowed us to create a set of features including the *year*, *month*, and *day* for each observation.

Being this a time series problem, one must be aware about the existence of missing timesteps. In the collected dataset, 102 timesteps were missing. Some of which corresponded to periods of roughly 1 month, between December 13th, 2018 to January 14th, 2019, and between March 7th, 2019 to April 9th, 2019. As discussed previously, missing timesteps are problematic. Hence, to fill the missing gaps, Open



(a) Mean UV index per month.

(b) Max UV index per month.

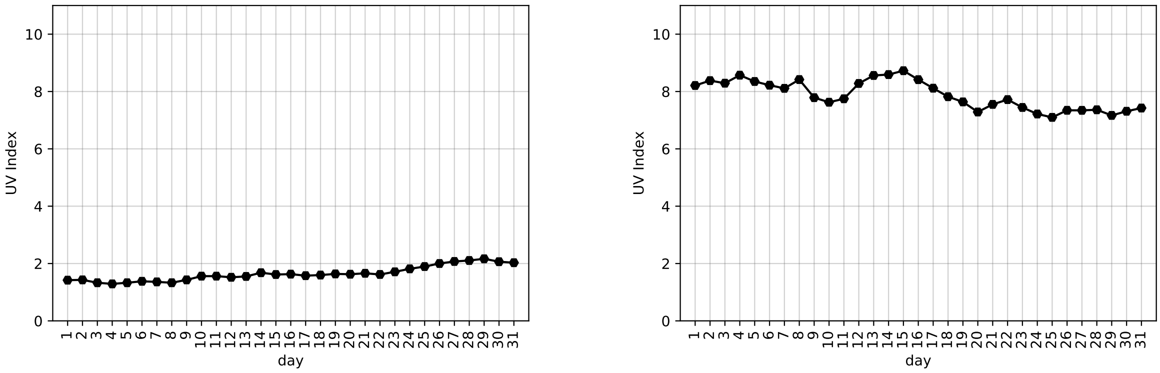
Figure 2.9: UV index values per month in the collected dataset.

Weather Map's *Historical Ultraviolet API* was used. By the end of this step, the dataset consisted of 581 daily timesteps. No missing values are present in the dataset. As explained in Section 2.4.3, since LSTMs work internally with the tanh activation function, all features were further normalized to fit into the interval $[-1, 1]$.

Figure 2.9 depicts the mean (Figure 2.9a) and max (Figure 2.9b) UV index values, per month, in the dataset. It is easily noticeable the existence of seasonality and cycles. Lower UV index values emerge during autumn and winter, with higher values appearing during spring and summer (data for the the northern hemisphere). Overall, the mean UV index is higher in July, but the peak value is usually found in June.

Figure 2.10 depicts the maximum UV value, per day, of two months in 2019. Figure 2.10a shows a steady line of low values during January. On the other hand, during August the values are considerably higher and with greater amplitude, offering an increased risk to the population (Figure 2.10b).

Two distinct datasets were created to achieve the goal of accurately forecast the UV index using previous UV values as input. Both contain 581 timesteps, varying only in the number of features. One dataset, the *uni-variate*, only contains the value of the UV index for each timestep (shape of (581, 1)). The second dataset, the *multi-variate*, also contains temporal information, i.e., the day and the month of the year



(a) UV index values for January, 2019.

(b) UV index values for August, 2019.

Figure 2.10: UV index values per day during two months of 2019 in the collected dataset.

(shape of (581, 3)) [189]. Algorithm 2.2 was used to set the problem as a supervised one, with inputs and corresponding labels. Recursive Multi-Step LSTMs were the ones considered since the goal is to forecast three days ahead and also because such multi-step architecture has already been shown to produce better results than others [45].

2.5.3 Experiments

To achieve the goal of forecasting the UV index for the next three days, we were required to develop and tune several candidate LSTMs. The effectiveness of the candidate models was evaluated using the RMSE and MAE error metrics. The predicted value is compared with the real value in order to compute both values. A time series cross-validator was also used to provide the train/test indices. Algorithm 2.3 details the generic function that was used to conceive and compile the DL models.

Several experiments were conducted to find the best combination of hyperparameters for both uni-variate and multi-variate approaches. The first uses previous UV index values to recursively forecast the future value. The latter, besides the UV index value also uses the *day* and *month*, giving a stronger temporal context to the models. Table 2.13 describes the searched hyperparameter configuration, for both approaches.

The Knime platform was used for data exploration. On the other hand, Python, version 3.7, was the used programming language for data exploration and preparation, and for model development and evaluation. Pandas, NumPy, scikit-learn, and matplotlib were the used libraries. Tensorflow v2.0.0 was used to implement the deep learning models. Tesla T4 GPUs were used as well as CUDNNLSTM layers for optimized performance in a GPU environment. Hardware was made available by Google's Colaboratory.

Table 2.13: Uni-variate vs multi-variate hyperparameters' searching space for UV forecasting.

Parameter	**Uni-Variate**	**Multi-Variate**	**Rationale**
Epochs	[150, 300]	[300, 500]	-
Timesteps	[7, 14]	[7, 14]	Input of 1 to 2 weeks
Batch size	[16, 23]	[16, 23]	Batch of 2 to 3 weeks
LSTM layers	[3, 4]	[3, 4]	Number of LSTM layers
Dense Layers	1	1	Number of dense layers
Dense Activation	[ReLU, tanh]	[ReLU, tanh]	Activation function
Neurons	[32, 64, 128]	[32, 64, 128]	For dense and LSTM layers
Dropout rate	[0.0, 0.5]	[0.0, 0.5]	For dense and LSTM layers
Learning rate	callback	callback	Keras callback
Multisteps	3	3	3 days forecasts
Features	1*	3**	Used features
CV Splits	3	3	Time series cross-validator

* Used features: *UV index*

**Used features: *UV index*, *day*, and *month*

2.5.4 Results and Discussion

The conducted experiments outputted similar results for both approaches. The best overall candidate model was a multi-variate one, with a RMSE of 0.306. This indicates that a stronger temporal context may lead to an overall decrease in error metrics. On the other hand, the best uni-variate models present a lower MAE than the best multi-variate ones, which implies that the multi-variate models output more stable forecasts (as RMSE penalizes outliers). Table 2.14 depicts the top-three results for each approach when forecasting three days.

Table 2.14: Uni-variate vs multi-variate LSTMs top-three results for UV forecasting.

#	**Timesteps**	**Batch**	**Layers**	**Neurons**	**Dropout**	**Act.**	**RMSE**	**MAE**
				Recursive Multi-Step Uni-Variate				
116	7	16	4	64	0.5	tanh	**0.325**	**0.236**
108	7	16	3	128	0.5	tanh	0.349	0.271
8	7	16	3	64	0.5	tanh	0.354	0.260
				Recursive Multi-Step Multi-Variate				
31	14	16	3	64	0.0	tanh	**0.306**	**0.249**
125	14	16	3	64	0.0	relu	0.339	0.284
73	14	23	3	32	0.0	relu	0.340	0.275

Overall, the best uni-variate model had a RMSE of 0.325 and a MAE of 0.236. On the other hand, the best multi-variate one had a RMSE and MAE of 0.306 and 0.249, respectively. Since both metrics are in the same unit of measurement as the UV index, such error values testify the ability of these models to accurately predict the UV value for the next three days. Interestingly, multi-variate models required a higher number of input timesteps (14 days) when compared to uni-variate ones (7 days). Both approaches required batches of two weeks. All the best uni-variate models required the tanh as activation function.

The best uni-variate model required a more complex architecture, of 4 hidden LSTMs layers, to achieve similar performances to the multi-variate ones, who required a shallow architecture. In addition, this shallow architecture ruled out the use of dropout, which was required by the best uni-variate models. On the other hand, the deeper architecture of the uni-variate LSTMs made them costly, in terms of time, to train. Figure 2.11 illustrates the architecture of the best multi-variate model.

The obtained results show promising prospects for UV index forecasting, with the best candidate models being able to accurately forecast the UV index for the next three days using only previous UV index values and some temporal data (an error of 0.3 units of measure). As a direct answer to the elicited research question (RQ2.5), it can be said that deep LSTMs models are able to accurately forecast the maximum daily ”clear sky” UV index value for multiple future timesteps using a Recursive Multi-Step approach [189].

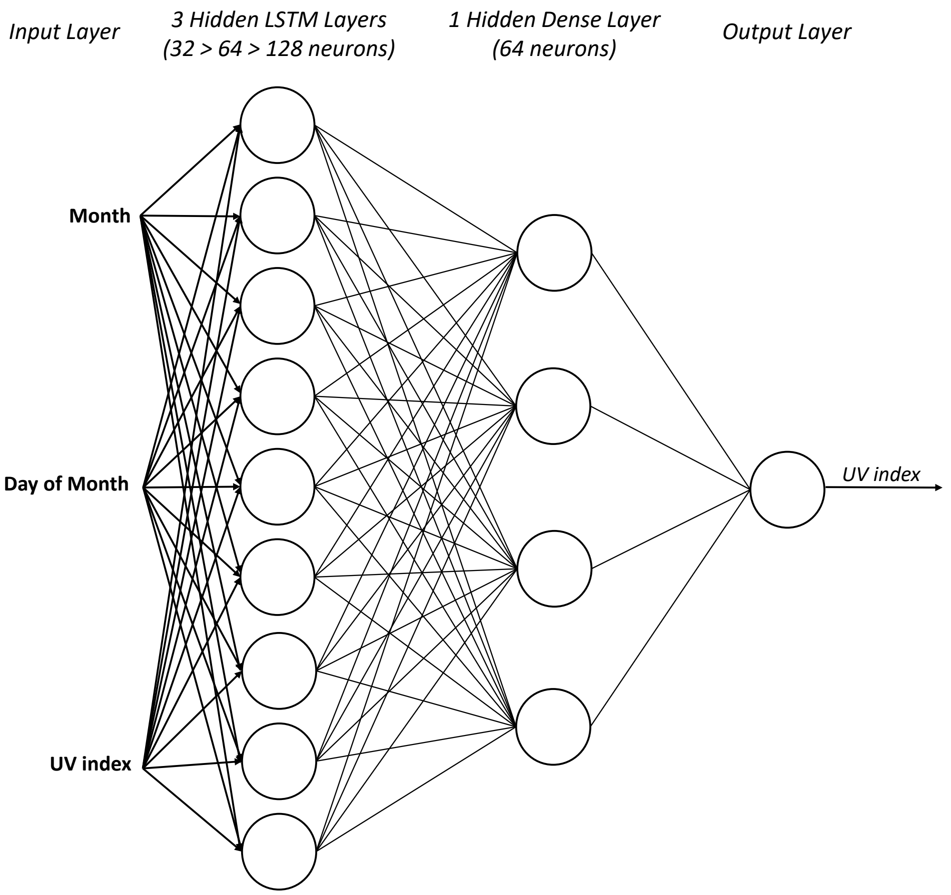


Figure 2.11: Representative architecture of the best multi-variate model for clear-sky UV index forecasting.

2.6 Summary

One important goal of this doctoral work (Task 3) focuses on sensing the city and conceiving methods to extract knowledge from the city's ecosystem, which may then be shared with VRUs in order to make them more aware of their surroundings and adapt their behavior accordingly. It also aims to promote Smart Cities, i.e., give cities the ability to sense their environment, act on problematic situations, and provide actionable information to their citizens.

The first step to enable *City Sensing* was to develop a MAS responsible for the entire data collection process and ML pipeline (Automated ML), based on the envision of an architecture for the IoP (Task 1). The MAS is remotely deployed in an Ubuntu 18.04 machine hosted at the University of Minho, being up and running since 24th July 2018. One software agent, *The Collector*, is a modular and configurable software agent that can, at any time, start collecting data for any city worldwide. It currently gathers data about the weather, pollution levels, traffic flow, and traffic incidents. Another software agent, the *ML Architect*, is responsible for using ML and DL models to provide accurate forecasts and predictions, following a publisher-subscriber pattern. It holds all non-mobile ML models and uses real-time data gathered by *The Collector*.

Establishing the data collection process was a crucial step towards the conception of models to be used by the *ML Architect*. To conceive such models, state of the art LSTMs were designed, conceived, tuned, and evaluated. Considering the strong ML background of this research work, CRISP-DM proved to be an effective methodology. To provide actionable information to VRUs, two main case studies were implemented. The first focused on forecasting the traffic flow status for the next twelve hours. With such information, VRUs can opt to avoid highly congested hours, which not only pose an increased risk of road injuries but also correspond to a scenario of higher atmospheric pollution given the number of vehicles on the road. This study also allowed us to understand and clarify the better performance of LSTMs when compared to statistical-based models, as well as their ability to accurately forecast future timesteps using a Recursive Multi-Step Multi-Variate approach. The second case study focused on forecasting the "clear sky" UV index for the next three days. The goal is again to allow VRUs to have a better perception of the city's environment and adapt their behavior in situations of increased risk. It must be noted that the most effective strategy to control skin cancer is to avoid sun exposure during peak UV radiation levels. Multiple experiments were conducted, using a wide combination of hyperparameters for all the candidate models. The best candidate model was able to forecast the UV index for the next three days with an error of just 0.3 units of measure.

The research methodology being followed in this PhD work considered an initial *Design* phase where solutions' specimens, i.e., the main Research Questions in Section 1.3.2, were defined based on the devised hypothesis. The work presented in this chapter is part of the *Validation* phase, which aims to validate the defined solutions' specimens to prove the research hypothesis. In particular, the work here presented aims to give an answer to ***Research Question 1****: Is a MAS able to autonomously capture city's data and produce actionable knowledge from such sources*? It can be said that it is viable to conceive a MAS to collect data about a city's environment by taking advantage of several public APIs. In addition, ML and DL models can be trained and then used by software agents to provide forecasts and predictions to the community. The work performed in this chapter closes **Task 1** and fulfills, by 50%, **Task 3**, which also encompasses *Crowd Sensing*, as described in the next chapter. The first *C*, from the *3-C*, i.e., ***City***, *Crowd*, and *Citizen* sensing, has now been addressed.

The work presented in this chapter gave origin to multiple scientific papers ([8, 16, 25, 45, 189]) as well as to the participation in several scientific events. The work performed on the UV domain was also distinguished with the *Best Application Paper* award by the program committee of the 17th International Conference on Distributed Computing and Artificial Intelligence (DCAI'20), held in L'Aquila, Italy. To conclude, the work presented in this chapter was performed in collaboration with researchers from the ALGORITMI Centre, Braga, Portugal; the Polytechnic Institute of Porto, Felgueiras, Portugal; and the Department of Electrical and Systems Engineering, Universidad de Leon, Spain.

Chapter 3

Crowd Sensing

Smart Cities, to emerge as such, are required to sense and understand their environment. Such an environment comprises vehicles, infrastructures, the weather, pollution levels, and many others, including people. In fact, people are not only the city's main stakeholders but also an active element of the city's ecosystem. Hence, the ability to sense groups of people, here entitled as crowd sensing, assumes an increased importance for the development of truly Smart Cities, the rise of the IoP, and the emergence of new proposals to enhance road safety.

3.1 Rationale

The research work presented in the next lines aims to sense people in order to gather insights at certain points of interest for VRUs. Such places can include pedestrians' crossings, sidewalks, or busier roads, among others. More than tracking individuals, the goal is to sense large groups of people in an anonymized fashion.

Crowd sensing approaches can be divided into two distinct groups: the first, passive sensing, aims to sense people without requiring them to do or carry anything, i.e., they are completely passive actors; the second, active sensing, requires people to install software or carry specific hardware in order to be sensed. Both have their advantages and disadvantages. Achieve passive sensing is more challenging but allows one to build artifacts that are able to sense everyone without requiring users to actively participate in the conceived solution. Active sensing, on the other hand, is easier to conceive and develop, but requires users to carry specific software/hardware artifacts that make them visible and recognizable.

The next lines describe and analyze the conception of passive methods for crowd sensing, i.e., the ability to sense people without requiring them to do anything. On the other hand, a set of active sensing solutions are also described. While the former aims to sense people at specific points of interest for VRUs, the latter focuses on conceiving geofences and emotional maps for the benefit of VRUs.

3.2 Passive Sensing

The prospect of sensing people without requiring them to be part of a system opens important perspectives for enhanced crowd sensing and control. However, the implementation of such a system requires extensive studies in order to understand its feasibility, both technologically and monetarily. Hence, a first step consists in studying and understanding the feasibility of developing such a system through means of new sensors and algorithms. Then, if viable, the goal should be put in terms of conceiving prototypes to analyze the system's performance. The next lines aim to give an answer to the following research questions:

RQ3.1. Do we already have the technological means to implement passive crowd sensing?

RQ3.2. Is the development of a passive crowd sensing system feasible, both technologically and monetarily?

RQ3.3. If viable, how much range does a passive crowd sensor has?

RQ3.4. If viable, how accurate is such a system (in terms of sensed people)?

3.2.1 State of the Art

The literature shows that several studies have already engaged on employing different means to track and count people at specific places. Some have focused on methods that force people to carry specific devices while others employ invasive technologies like cameras. However, the acceptance of such methods is low due to development and deployment costs, inefficiency, and privacy issues. On the other hand, non-invasive methods are easier to implement and to be accepted by the community, even if less accurate [66].

In [190], the authors aim to track users' smartphones using Wi-Fi monitors, being interested in finding users' trajectory. The authors' system uses common Access Point (AP) hardware to both collect and deliver detections, requiring a central server. The proposed tracking system consists of one, or more, Wi-Fi monitors placed at specific locations, as well as a central server where Wi-Fi observations are analyzed. Wi-Fi monitors are built from standard Wi-Fi AP hardware, namely the Ubiquity PicoStation 2, which retails for around 70$. For internet access, the monitors have to be equipped with a cellular modem, incurring in significant monthly costs, which can be prohibitive for long-term or large-scale deployments. The analyzed work provides no estimation of crowd density, being, in turn, focused on tracking people [66].

In [191], the authors estimate crowd density using a smartphone to scan the environment for bluetooth devices. Their approach requires not only users to be moving through the environment with their smartphones scanning for devices but also to have discoverable bluetooth enabled. Moreover, the authors derive the basis for their technique on a problematic assumption, i.e., that many people have the bluetooth transceivers of their smartphone in the discoverable mode as default setting. Even though we are now witnessing to the exponential use, by people, of smartwatches, smartbands, and even smart clothing, all of which hold bluetooth connectivity, the use of standalone bluetooth scanning devices can provide inaccurate results for crowd sensing. When possible, the focus should be set on Wi-Fi detection, complemented with bluetooth [66].

Differently, in [192], both Wi-Fi APs and bluetooth beacons are used to determine the reliability of the estimation of people in a certain area. In particular, attention is devoted to Wi-Fi technology with active and passive scanning of Media Access Control address (MAC) addresses and the Received Signal Strength Indicator (RSSI). The study was conducted at a major boarding airport in Germany, with boarding passes being used as validators. Considerations about users' devices, bluetooth, and Wi-Fi policies are also shown to influence the number of captured devices. In general, bluetooth is considered to underestimate the number of people while Wi-Fi tends to overestimate. The authors suggest a hybrid approach for more accurate results [66].

One study that leverages Wi-Fi technology to count the total number of people walking inside a building is the one performed in [193]. The authors' approach is based on Wi-Fi transceivers, receivers, and Wi-Fi RSSI measurements. The transceivers transmit wireless signals that interact with walking people and static objects, being then received by receivers. The fluctuation in RSSI values between transceiver and receiver may be indicative of the presence of people, animals, or objects, in the area between the two devices. Both wireless cards need to be interfaced with a computer, with the authors using a Raspberry Pi. This interesting approach to the problem presents some drawbacks. On the one hand, the authors employed tests with very few people inside the area of interest, leaving doubts about the accuracy of the system with a real crowd. On the other hand, the cost of the solution may be significant if one aims to cover a meaningful area, requiring several transmitters, receivers, and computers. In addition, new studies would have to be conducted if one changes any variable, such as the distance between devices or their relative position to each other, as models would have to learn to identify new RSSI fluctuations [66].

In [194], the authors provide an application scenario for the use of Wi-Fi probe-based detection of user occupancy. In this case, the estimation is demonstrated as an input for modeling user occupancy and optimize heating, ventilation, and air conditioning control. Even though an interesting approach, the conducted research aims to create occupancy models using Dynamic Markov Time-Window inference from occupancy estimation. The passive scan used in the approach is not detailed, but it is said to be based on a simple detection of devices. Enhanced Wi-Fi and bluetooth strategies can be used to improve occupancy estimation, but they are not in the scope of the referred paper [66].

Another approach may be seen in [195], where the authors, with a non-invasive and non-participatory mechanism, aim to measure bus passenger loads by detecting the periodic network probing activity from smartphones. The authors experimented two methods for data collection. The first relied on bluetooth device detection. However, poor results enforced a second method, which relies on Wi-Fi probing detection. The authors' approach was to use the wireless network interface controller of a Linux computer (Atheros AR9285 wireless network interface of a Hewlett-Packard Pro-Book 4320s laptop running Ubuntu) placed in monitor model and airodump-ng for packet capture during sixty minutes at a bus stop, on a busy afternoon, and during a ten-minute ride aboard a bus. Indeed, the use of a computer and the need for someone to operate the device within a bus and at a bus stop significantly reduces the practicability and feasibility of such a system [66].

Another study that aims to make the bridge between Smart Cities, and monitoring and tracking systems is the one presented in [196]. Here, the authors present a system to track the movement of people and vehicles, monitoring the radio-electric space by capturing Wi-Fi and bluetooth signals. The system collects people's and vehicles' mobility data and is able to track them with a grid of devices that are connected to a central server, allowing the construction of mobility graphs. The devices that compose the system are based on a single-board Raspberry Pi 1 and have a cost of 100$ each. A positive aspect of this work relates to the anonymization of data for privacy purposes, even though MAC randomization when probing AP is already employed by many vendors. Finally, there is no reference to the software used for device detection neither if it is using proprietary technology nor if new software was developed [66].

3.2.2 Materials and Methods

The reviewed literature shows that researchers have already engaged in tracking Wi-Fi and bluetooth signals passively, through means of the Raspberry Pi, APs, or even computers. These are rather large and expensive devices, specially for large-scale implementations. Hence, the first major goal of the conducted research was to understand if we already have the technological means to implement passive methods for crowd sensing using a smaller and cheaper piece of hardware. The next lines describe what we call the ”Smart Scanner”, i.e., a hardware device that has been programmed by us to perform crowd sensing.

3.2.2.1 Smart Scanner - The Hardware

During the conducted research, we decided to take advantage of the huge potential of small Wi-Fi integrated boards, in particular, second-generation ESP8266 ESP-12E NodeMCU Amica boards (Figure 3.1). The ESP8266 NodeMCU board is a low-power Arduino-type board, with a Wi-Fi interface that allows the board to work as an AP, station, or both. In fact, the board integrates a 802.11b/g/n HT40 Wi-Fi transceiver, which means that the board is able to connect itself to a network while working as a wireless AP, allowing other Wi-Fi devices to connect to a network through the board.

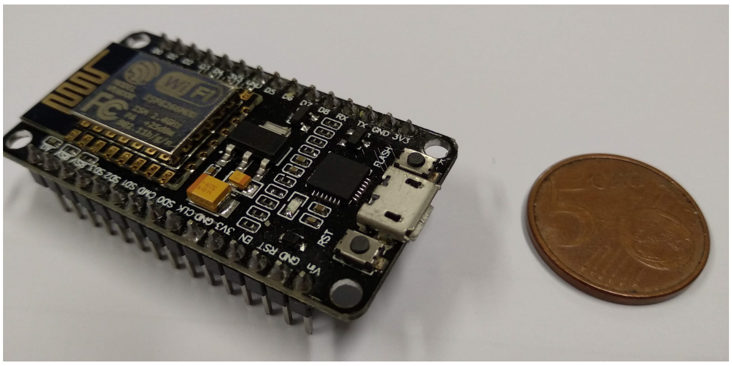


Figure 3.1: A second-generation ESP8266 ESP-12E NodeMCU Amica board.

It is suitable for the IoT and can facilitate the bridge towards Smart Cities, removing the need for wired communication and processing. Besides the reduced cost of 2.4€ per board, it also has an impressive set of features that must be highlighted. It has Wi-Fi capability, eliminating the need for wired communication, 128 KB of RAM and 4 MB of Flash memory (for program and data storage), which proved to be more than enough, a micro-USB interface, a built-in antenna, open-source support, small dimensions (4.8 x 2.4 x 0.5 cm), and low weight (109 g), making it a desirable hardware for future Smart Cities.

The board also has a Tensilica Xtensa 32-bit LX106 RISC microprocessor, which operates at 80 to 160 MHz clock frequency. It has a low-dropout voltage regulator to keep the voltage steady. The board is powered through an on-board micro-USB connector. It also has two buttons. One to reset the board and another, the flash button, used while upgrading the board's firmware. In addition, it also features a LED indicator, which can be used by the developer to obtain visual feedback from the board. Finally, the board includes a CP2102 USB-to-UART bridge controller, which converts USB signals to serial, allowing computers to program and communicate with the board [197]. ESP8266 was released a few years ago. This means that ESP8266-based boards are now stable, have several libraries and extensive documentation, and have full support from the community.

Recently, an upgraded board, the ESP32, was released (Figure 3.2). It is ESP8266 successor, i.e., it not only has Wi-Fi support but it also features Bluetooth 4.0 connectivity (dual-mode). It has a processor similar to ESP8266, but it has two individually controlled CPU cores, which operate at 80 to 240 MHz clock frequency (dual-core). It has 448 KB of ROM, 520 KB of SRAM, and 4MB of flash memory. The board integrates the same Wi-Fi transceiver as ESP8266. However, the ESP32 chip also holds dual-mode bluetooth connectivity, supporting bluetooth 4.0 and classic bluetooth. The board is also powered through an on-board micro-USB connector, it also offers visual feedback, and it also has a bridge controller to convert USB signals to serial [198]. Being this a recent board, the availability of libraries and documentation for the ESP32 is significantly lower when compared to the ESP8266. In fact, it proved to be more difficult to program and use than ESP8266, particularly when used in dual-mode, i.e., simultaneously capturing bluetooth and Wi-Fi probes. For this reason, ESP8266 was the board used in the conducted experiments.

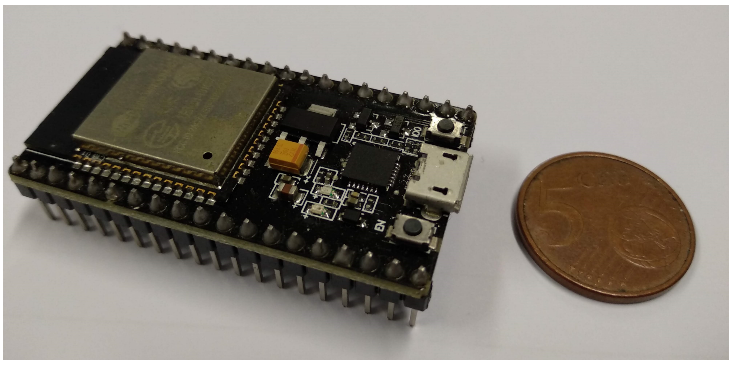


Figure 3.2: An ESP32 board.

3.2.2.2 Smart Scanner - The Software

Both boards integrate Wi-Fi transceivers, which means that both are able to work as a station while working as an AP. This feature caught our attention. One could take advantage of the fact that these boards work as an AP to sense a point of interest for Wi-Fi probe requests. In its essence, a probe request is a frame sent by a client station (e.g., smartphone, smartwatch, or laptop, among others) requesting information from a specific AP or all APs in a given area. In other words, this means that devices carried out by people in their daily lives are constantly scanning the environment to gather information about existing APs and networks that are available for connection. This process is performed whenever the WiFi interface is enabled, regardless of being, or not, connected to a wireless network [199]. The goal is to provide the user the best possible connection quality, looking for networks with higher signal strength. Hence, if properly programmed, both boards can be set to work as an AP and, with that, passively detect and capture probe requests sent by devices that are carried out by people. Then, since the boards have limited memory, one could take advantage of the fact that these boards are able to simultaneously work as a station to send the captured data to a cloud server.

Recent years came with significant improvements to Wi-Fi probing in order to increase users' privacy. Since transmissions are made prior to connecting to a network, data travels on the medium without any kind of encryption. Hence, a major improvement consisted in hide the real MAC address when probing. The MAC randomization procedure varies according to the used device, the operating system, and the manufacturer. This does not pose a problem to this study since it does not aim to track users through means of MAC addresses. Instead, it aims to simply detect devices, which may be indicative of the presence of people in the vicinity of the ESP boards. It will, however, require us to implement methods that take into consideration the fact that the same device may send several probe requests with different MAC addresses in a short period of time, i.e., we cannot have a direct mapping between a MAC address and a device.

The Arduino Integrated Development Environment (IDE), a cross-platform application written in Java, makes it very easy to develop, compile, and upload code to arduino-based boards, such as the ESP ones. Sketch is the name given to a program developed with this IDE, which is written in C or C++. The open-source nature of the Arduino project has facilitated the release of many open-source software libraries. Programming in the Arduino IDE requires the developer to define, at least, two functions: *setup()* and *loop()*. The former one is called once when a sketch starts after powering up the board, being used to initialize variables, input and output pin modes, and other libraries needed in the developed sketch. The latter is executed repeatedly in the main program until the board is powered off or reset, allowing the program to change and adapt.

Considering all constraints associated with ESP32, the focus went towards the conception of a probe request counter for crowd sensing using an ESP8266 board. The sketch was released in GitHub[1] under a MIT license, being ready to use (plug and play). The sketch, besides the two mandatory functions, also implements a few auxiliary ones:

[1] https://github.com/brunofmf/Crowd-Sensing

1. *onProbeRequestCaptureData(constWiFiEventSoftAPModeProbeRequestReceived&evt)*, a callback function that triggers each time the ESP board captures a probe request from any device. If the board has enough memory available, and if the captured request is a new sighting (it was not seen previously), the board will save it while waiting for data to be pushed to a cloud server (Listing 3.1);

Listing 3.1: Crowd Sensing *onProbeRequestCaptureData* function.

```
void onProbeRequestCaptureData(const WiFiEventSoftAPModeProbeRequestReceived& evt) {
 //if enough memory and a new sighting
 if(currIndex < ARRAY_SIZE){
   if(newSighting(evt)){
     probeArray[currIndex].mac = macToString(evt.mac);
     probeArray[currIndex].rssi = evt.rssi;
     probeArray[currIndex++].previousMillisDetected = millis();
   }
 } else{
   //not enough memory, set log message
   Serial.println(F(”*** Array Limit Achieved!! Push probe requests! ***”));
 }
}
```

2. *onProbeRequestPrint(const WiFiEventSoftAPModeProbeRequestReceived& evt)*, a callback function that triggers each time the ESP board captures a probe request and prints it to a monitor (useful for debugging purposes);

3. *onStationConnected(constWiFiEventSoftAPModeStationConnected&evt)* and *onStationDisconnected(const WiFiEventSoftAPModeStationDisconnected& evt)*, two callback functions that execute each time a client station connects to, or disconnects from, the ESP board, respectively;

4. *clearData()*, required to clear the internal memory of the board to avoid out-of-memory errors. Executed after pushing the data to the cloud server;

5. *setupMqtt()* and *setupFirebase()*, functions responsible for setting the connection with a Message Queuing Telemetry Transport (MQTT) broker and Firebase, respectively;

6. *buildAndPublish(bool clearD)*, builds a JSON object with all the captured probe requests and then pushes the created object to a cloud server (defaults to a MQTT broker);

7. *newSighting(const WiFiEventSoftAPModeProbeRequestReceived& evt)*, a function that verifies whether the captured request, through its MAC address, is a new sighting or if it was already captured in the last 60 seconds (default value) (Listing 3.2);

Listing 3.2: Crowd Sensing *newSighting* function.

```
bool newSighting(const WiFiEventSoftAPModeProbeRequestReceived& evt){
  String mac = macToString(evt.mac);
  long currTime = millis();
  //start by the end as array is ordered - new sightings are at the end
  //break as soon as it finds the mac in the list
  for(int i = currIndex-1; i>=0; i--){
    //if this mac has already been captured
    if(mac.equals(probeArray[i].mac)){
      //check if enough time has passed
      //(sightingsInterval milliseconds since last sighting)
      if(currTime-probeArray[i].previousMillisDetected < sightingsInterval){
        return false;
      }
      break;
    }
  }
  return true;
}
```

8. *stopHandlers()* and *restartHandlers()*, responsible for stopping/starting the capture handlers and timers;

9. *startTimer()* and *stopTimer()*, which starts/stops a timer that decides when to push data to a cloud server (default value as 45 seconds);

10. *printProbeArray()*, to print the internal array with the captured probe requests (useful for debugging purposes);

11. *macToString(const unsigned char* mac)*, that sets a MAC address to a string (Listing 3.3).

Listing 3.3: Crowd Sensing *macToString* function.

```
String macToString(const unsigned char* mac){
  char buf[20];
  snprintf(buf, sizeof(buf), "%02x:%02x:%02x:%02x:%02x:%02x",mac[0], mac[1], mac[2], mac
      ↪ [3], mac[4], mac[5]);
  return String(buf);
}
```

The two mandatory functions, *setup()* and *loop()*, make use of the auxiliary ones to achieve the goal of capture probe requests of client stations in the vicinity of the ESP board. Listing 3.4 depicts the *setup()* function responsible for setting the data rate in bits per second (baud) for serial data transmission; the Wi-Fi mode of the board as WIFI_AP_STA, meaning the board will work, simultaneously, as an AP and as a station; initialize the Firebase or MQTT connection; setup the timer for handling data collection and the transmission of information to the cloud server; and register the handlers that are called every time a connection/disconnection is made and every time a probe request is captured by the board [66].

Listing 3.4: Crowd Sensing sketch's *setup* function.

```
void setup() {
  Serial.begin(9600);
  //don't save WiFi configuration in flash - optional
  WiFi.persistent(false);
  //set up an access point
  WiFi.mode(WIFI_AP_STA);
  WiFi.softAP(AP_SSID, AP_PASSWORD);
  //connect to the network
  WiFi.begin(STATION_NETWORK, STATION_PASSWORD);
  if(WiFi.status() == WL_CONNECTED) {
    Serial.print(”Connected to ”); Serial.print(STATION_NETWORK);
    Serial.print(”; IP address: ”); Serial.println(WiFi.localIP());
    isConnected = true;
  } else{
    Serial.print(” Connection to ”); Serial.print(STATION_NETWORK);
    Serial.println(” failed! Board will only work as an AP......”);
  }
  //setup the MQTT/Firebase Connection
  if(useMqtt){
    setupMqtt();
  } else {
    setupFirebase();
  }
  //setup timer
  os_timer_setfn(&theTimer, timerCallback, NULL);
  os_timer_arm(&theTimer, sendTimer, true);
  //callbacks
  stationConnectedHandler = WiFi.onSoftAPModeStationConnected(&onStationConnected);
  stationDisconnectedHandler = WiFi.onSoftAPModeStationDisconnected(&onStationDisconnected);
  probeRequestPrintHandler = WiFi.onSoftAPModeProbeRequestReceived(&onProbeRequestPrint);
  probeRequestCaptureDataHandler = WiFi.onSoftAPModeProbeRequestReceived(&
       ↪ onProbeRequestCaptureData);
  Serial.print(”*** All setup has been made! ”);
  if(isConnected){
    startTimer();
  } else {
    Serial.println(”Timer is DISABLED! ***”);
  }
}
```

On the other hand, listing 3.5 depicts the *loop()* function that, if the board is connected to a computer, is capable of receiving additional user commands such as *Stop/Restart*, *Count*, *Start/Stop_Timer* and

Clear, among others. This allows one to enforce additional behavior from the board, which, by default, starts with all handlers and services enabled [66].

Listing 3.5: Crowd Sensing sketch’s *loop* function.

```
void loop() {
  if(Serial.available() > 0) {
    command = Serial.readString();
    if(command.equalsIgnoreCase(CMD_RESTART)){
      restartHandlers();
    } else if(command.equalsIgnoreCase(CMD_STOP)){
      stopHandlers();
    } else if(command.equalsIgnoreCase(CMD_COUNT)){
      Serial.println(currIndex);
      delay(1000);
    } else if(command.equalsIgnoreCase(CMD_PRINT)){
      printProbeArray();
      delay(1000);
    } else if(command.equalsIgnoreCase(CMD_CLEAR)){
      clearData();
      delay(1000);
    } else if(command.equalsIgnoreCase(CMD_SEND)){
      sendDataCmd();
      delay(1000);
    } else if(command.equalsIgnoreCase(CMD_START_TIMER)){
      startTimer();
    } else if(command.equalsIgnoreCase(CMD_STOP_TIMER)){
      stopTimer();
    } else{
      Serial.println(F(”Unknown command!”));
    }
  }
  if(sendNow){ //if timer told us it is time to send data
    buildAndPublish(true);
    sendNow = false;
    if(useMqtt){
      client.loop();
    }
  }
}
```

When using of the conceived sketch, one may change variables’ default values for specific behavior, in particular:

- *DEVICE_ID*, which sets the device identifier;
- *useMqtt*, which defines the use of a MQTT broker or Firebase;

- *FIREBASE_HOST*, *FIREBASE_AUTH* and *FIREBASE_PUSH*, to set Firebase authentication;
- *MQTTSERVER*, *MQTTPORT*, *MQTTUSER* and *MQTTPASSWORD*, to set MQTT authentication;
- *STATION_NETWORK* and *STATION_PASSWORD*, to define the network the board is going to connect to;
- *AP_SSID* and *AP_PASSWORD*, to define the network's SSID and password that is provided by the board to client stations.

The sketch should then be uploaded to the board using Arduino IDE. An upload speed (baudrate) of, at least, 57600 should be used otherwise timeouts will occur when uploading the sketch. A set of libraries are also required, including *FirebaseArduino* and *PubSubClient*. While the first allows the board to push data to Firebase, the second provides the ability to publish and subscribe to a server that supports MQTT. It uses a quality-of-service set to 0, which means that there are no guarantees data will reach the broker. In addition, the *PubSubClient* library limits the payload to 128 bytes on the *MQTT_MAX_PACKET_SIZE* variable. As we may sense several devices at once, this value will limit the payload from being sent to a MQTT broker. Hence, *MQTT_MAX_PACKET_SIZE* value in *PubSubClient.h* was updated to 4096 bytes.

3.2.3 Range Measurement Experiment

The ESP boards come with a built-in antenna. However, there is no default value for its range. In fact, its range significantly depends on the environment where the board stands. Indoor scenarios, with walls, chairs, and tables, will limit the antenna's range to a few dozen meters. On the other hand, line-of-sight propagation in outdoor scenarios can reach several hundred meters. This value will, however, decrease if a shell is used to protect the board from the weather. All experiments were performed using only the board's built-in antenna. No external antenna was used.

3.2.3.1 Outdoor Experiment

For this study, it was important to have a clear sense of the range of the ESP8266 ESP-12E NodeMCU Amica board in an outdoor scenario with a shell all around the board (to make it weatherproof). This allows one to know the exact area covered by a single board in such conditions. The adopted procedure focused on field experiments and was as follows:

1. Upload the sketch to the board and place the board inside its shell;
2. Select the point-of-interest, which consisted of a pedestrians' crossing;
3. Power up the board;

4. Connect a smartphone to the board's network;

5. Periodically connect/disconnect the smartphone from the board's network, while moving away from the board in a straight line. While performing this action, the board collects the RSSI, which decreases to the point where one is unable to connect to the board's network and no RSSI is received by the board;

6. Repeat the previous step multiple times, in different directions.

Then, with the collected data, a straight line was drawn and it was calculated the distance between the board and the place where the signal was lost. The mean value indicated a range of, approximately, 27 meters on a straight line, covering an area of 2290 meters^2 (Figure 3.3) [66].

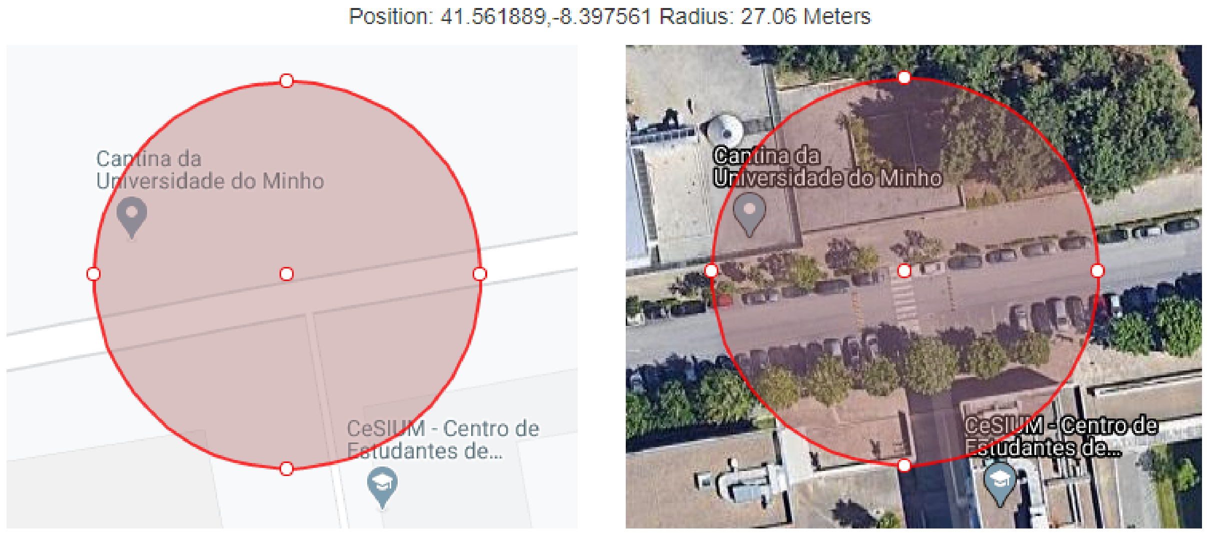

Position: 41.561889,-8.397561 Radius: 27.06 Meters
Cantina da Universidade do Minho
CeSIUM - Centro de Estudantes de...
Cantina da Universidade do Minho
CeSIUM - Centro de Estudantes de...

Figure 3.3: Range of the second-generation ESP8266 ESP-12E NodeMCU Amica board within a point-of-interest for VRUs: a pedestrian crossing.

3.2.3.2 Indoor Experiment

To have an estimate of the indoor range of the ESP8266 ESP-12E NodeMCU Amica board, a similar procedure to the one described in the outdoor experiment was conducted. The sketch was initially uploaded to the board, which was then placed on a table, against a wall, at ISLab (Figure 3.4). An important difference is related to the lack of a shell surrounding the board, which is unnecessary in indoor environments. The board was then powered up, with a smartphone connecting to the board's network. The next step was to periodically connect/disconnect the smartphone from the board's network while moving away from the board in several directions, both inside and outside the laboratory. While performing this action, the board collects the RSSI, which decreases to the point where one is unable to connect to the board's network and no RSSI is received by the board.

Interestingly, the board's range in an indoor environment is similar to its range in an outdoor scenario with a shell all around the board. Inside ISLab, which has an area of approximately 68 meters^2, the board's

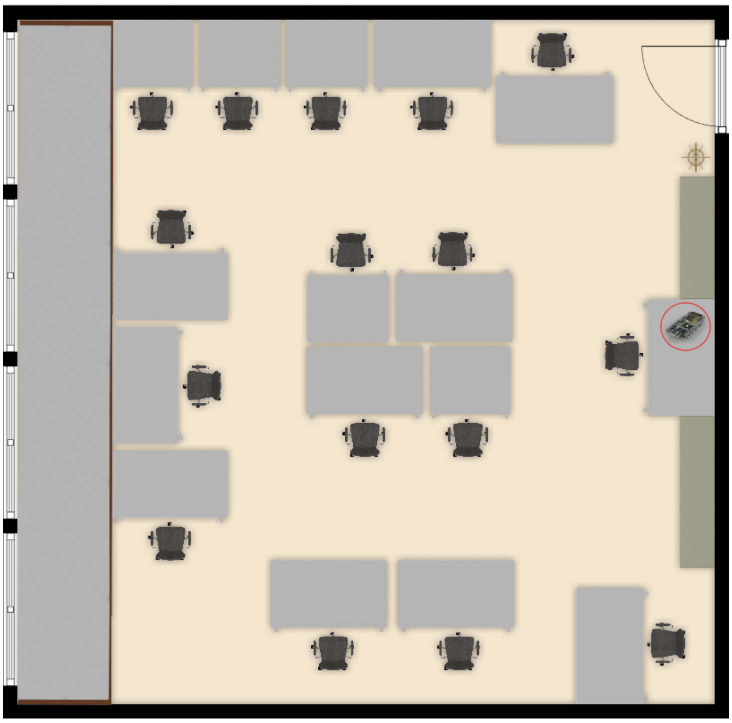


Figure 3.4: ISLab blueprint illustrating the location of the second-generation ESP8266 ESP-12E NodeMCU Amica board (circled in red) for the indoor range measurement experiment.

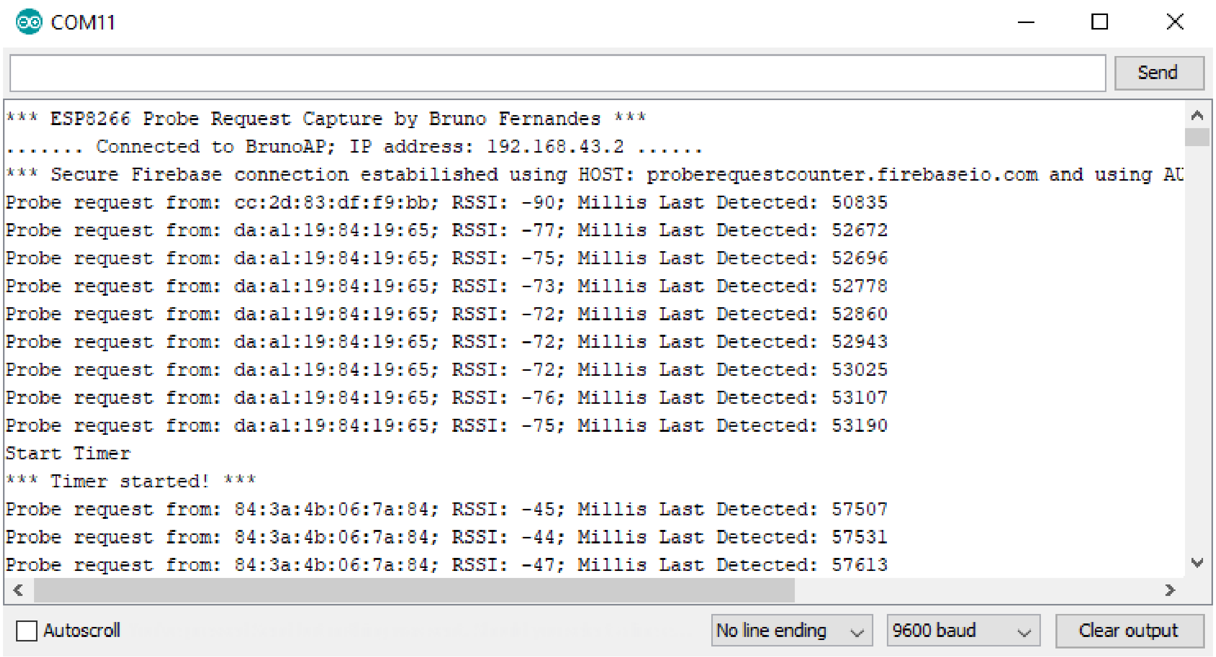
signal was perfect. Outside ISLab, the board's signal was able to cross several horizontal and vertical walls. The signal was received with low to medium strength on the floor below ISLab, in a vertical line. Two floors below, no signal was caught. Within the same floor as ISLab, the signal's range varied according to the number of obstacles it faced. The signal was still caught 20 meters way from the sensor with four to five walls as obstacles. With fewer walls as obstacles, the signal had a range higher than 30 meters. Such signals were caught inside other laboratories, offices, and corridors. Based on the obtained results, it is estimated that one board is able to cover half a floor of the Department of Informatics of the University of Minho.

3.2.4 Passive Crowd Sensing Experiment

The ESP board, i.e., the Smart Scanner when deployed with the developed software, was used for crowd sensing in both outdoor and indoor scenarios. Greater importance was given to the sensorization of the outdoor environment since it allows one to sense and gather insights at important points-of-interest for VRUs. The indoor experiment is to be understood as a further validation of the Smart Scanner's sensing capabilities.

3.2.4.1 Outdoor Experiment

Having defined the Smart Scanner's range in an outdoor environment, the crowd sensor was again placed in the location depicted in Figure 3.3 (geographic coordinates = 41.561889, -8.397561). On March 23rd, 2018, four measurements were made with a duration of fifteen minutes each. While the Smart Scanner was sensing the environment and communicating its findings, an observer would be counting each person and each vehicle that would enter within the Smart Scanner's range (observational method). This allowed one to have a validation measure. Figure 3.5 depicts the serial monitor output of the ESP8266 board, with several probe requests being captured, logged, and sent to Firebase's cloud server.



```
*** ESP8266 Probe Request Capture by Bruno Fernandes ***
....... Connected to BrunoAP; IP address: 192.168.43.2 ......
*** Secure Firebase connection estabilished using HOST: proberequestcounter.firebaseio.com and using AU
Probe request from: cc:2d:83:df:f9:bb; RSSI: -90; Millis Last Detected: 50835
Probe request from: da:al:19:84:19:65; RSSI: -77; Millis Last Detected: 52672
Probe request from: da:al:19:84:19:65; RSSI: -75; Millis Last Detected: 52696
Probe request from: da:al:19:84:19:65; RSSI: -73; Millis Last Detected: 52778
Probe request from: da:al:19:84:19:65; RSSI: -72; Millis Last Detected: 52860
Probe request from: da:al:19:84:19:65; RSSI: -72; Millis Last Detected: 52943
Probe request from: da:al:19:84:19:65; RSSI: -72; Millis Last Detected: 53025
Probe request from: da:al:19:84:19:65; RSSI: -76; Millis Last Detected: 53107
Probe request from: da:al:19:84:19:65; RSSI: -75; Millis Last Detected: 53190
Start Timer
*** Timer started! ***
Probe request from: 84:3a:4b:06:7a:84; RSSI: -45; Millis Last Detected: 57507
Probe request from: 84:3a:4b:06:7a:84; RSSI: -44; Millis Last Detected: 57531
Probe request from: 84:3a:4b:06:7a:84; RSSI: -47; Millis Last Detected: 57613
```

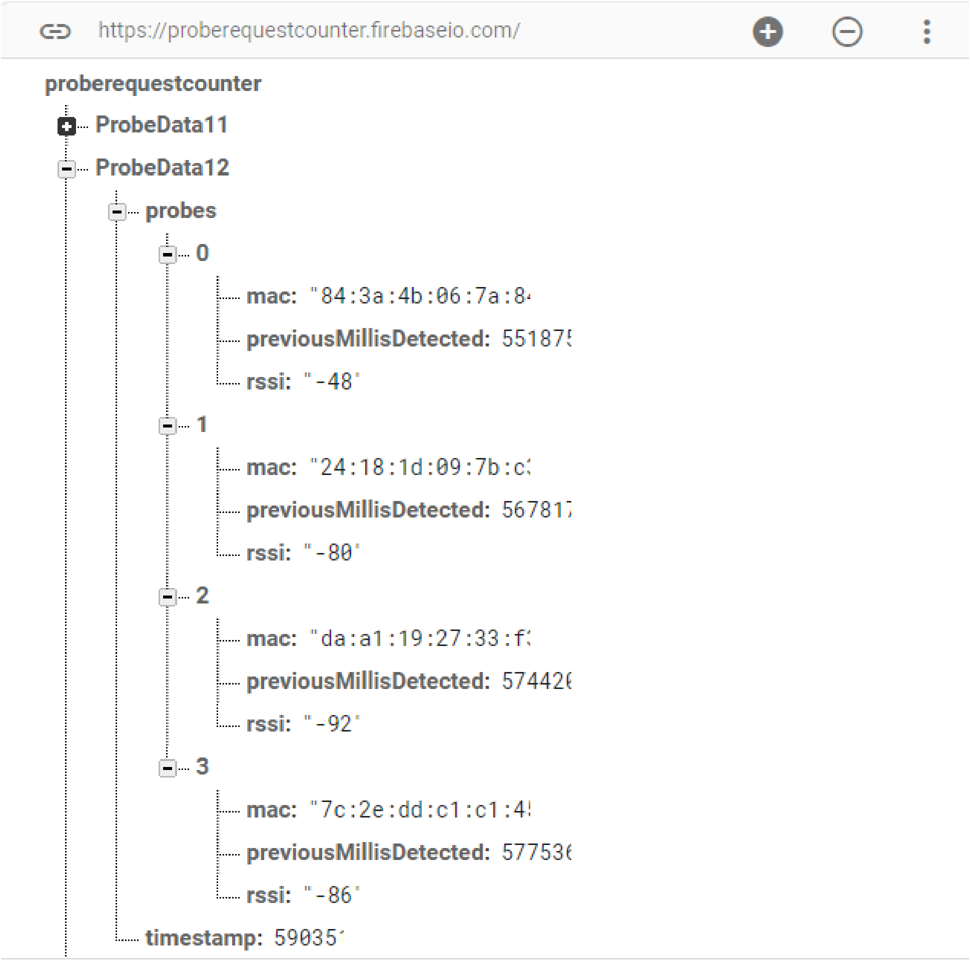
Figure 3.5: Serial monitor output of the ESP8266 board when working as a Smart Scanner.

Figure 3.6 depicts the cloud-hosted database that was set to store probe requests captured and communicated by the Smart Scanner. It uses Firebase Realtime Database, a NoSQL database where data are stored as one large JSON tree. *The Collector* uses this feed to collect its crowd sensing data.

Table 3.1 shows the outcome of the observational method vs the Smart Scanner's data, for each one of the four periods. During these periods, the observer reported a total of 640 people and 90 vehicles within the Smart Scanner's range, for a total of 730 people (assuming one occupant per vehicle). On the other hand, the Smart Scanner handled a grand total of 1062 requests from 722 distinct devices. The number of distinct devices is unscrambled from the received MAC address, the corresponding RSSI, and the timestamp associated with each request.

Table 3.1: Outcome of the observational method vs the Smart Scanner's data.

Measurement	**Observation**			**Smart Scanner**	
	People	Vehicles	**Total**	Distinct	**Total**
1st Period (11h46 - 12h01)	104	10	**114**	91	**172**
2nd Period (12h01 - 12h16)	164	20	**184**	224	**322**
3rd Period (12h16 - 12h31)	180	21	**201**	170	**266**
4th Period (12h31 - 12h46)	192	39	**231**	237	**302**
Total	640	90	**730**	722	**1062**



```
https://proberequestcounter.firebaseio.com/

proberequestcounter
  ProbeData11
  ProbeData12
    probes
      0
        mac: "84:3a:4b:06:7a:8
        previousMillisDetected: 551875
        rssi: "-48"
      1
        mac: "24:18:1d:09:7b:c
        previousMillisDetected: 567817
        rssi: "-80"
      2
        mac: "da:a1:19:27:33:f
        previousMillisDetected: 574426
        rssi: "-92"
      3
        mac: "7c:2e:dd:c1:c1:4
        previousMillisDetected: 577536
        rssi: "-86"
    timestamp: 59035
```

Figure 3.6: Probe requests' data model in Firebase Realtime Database.

3.2.4.2 Indoor Experiment

For the indoor experiment, the crowd sensor was again placed in the location depicted in Figure 3.4 (geographic coordinates = 41.561391, -8.397263). The board was powered up on July 25th, 2018, and sensed the environment until September 19th of the same year, for a total of 57 days. However, between August 21st and 26th, inclusive, the board was down and, so, no data were collected during the referred days. During the 51 days the crowd sensor was on, it collected a total of 232421 WiFi probe requests. From the grand total, 43590 correspond to distinct MAC addresses. However, if the analysis is performed per month, this number increases to 44762, meaning that several MAC addresses were sensed in multiple months (Table 3.2).

Table 3.2: Data captured by the Smart Scanner for indoor crowd sensing.

Measurement	Active Days	Probe Requests	Probe Requests/Day	Distinct MACs
July 25th-31st	7	47205	6744	10188
August 1st-31st	25	45057	1802	9094
September 1st-19th	19	140159	7377	25480
Total	**51**	**232421**	**15923**	**44762**

3.2.5 Results and Discussion

First and foremost, it must be highlighted that the followed method is entirely non-invasive, requiring no user interaction with the developed system. Not in any way did we try to decrypt any data or perform active actions to stimulate or alter normal network behavior. This study presents no expectation of harm. Instead, it aims to commit a benefit to our society.

3.2.5.1 Outdoor Experiment

Regarding the results of the outdoor experiment, as depicted in Table 3.1, it is possible to identify some fluctuations between the values reported by the observer and the distinct devices detected by the Smart Scanner. The overall value displays, however, promising results, with the Smart Scanner showing only a two percent deviation from the real count of people in the sensed area (722 vs 730 people). It is also noticeable that the number of people tends to increase over time (Figure 3.7). This is explained by the fact that the sensed point-of-interest, a pedestrian crossing, is near a restaurant and the experiment took place close to lunch time.

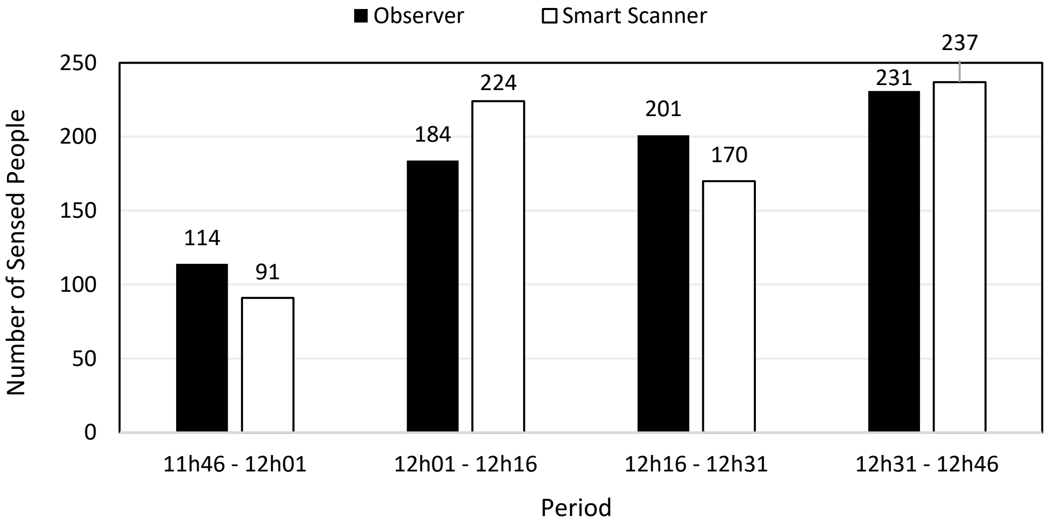


Figure 3.7: Observer's count vs the number of distinct devices captured by the Smart Scanner, per period.

Altogether, the Smart Scanner pushed a total of 1062 Wi-Fi probe requests to the cloud-hosted database. It should be noted that this value corresponds only to new sightings captured by the Smart Scanner as it filters repeated sightings that are captured during a certain period of time. Figure 3.8 presents a comparison between the number of pushed probe requests by the Smart Scanner versus the number of distinct ones, per period. For instance, during the second period (12h01 - 12h16) the scanner dealt with a total of 322 Wi-Fi probe requests but only 224 where distinct, meaning that 98 requests were discarded as repeated. This happens, for example, if a person enters the point-of-interest but remains in the region for several minutes. In fact, that was a situation reported by the observer, where people were noticed to stay within the point-of-interest for a few minutes, meaning that several probe requests would be received by the crowd sensor.

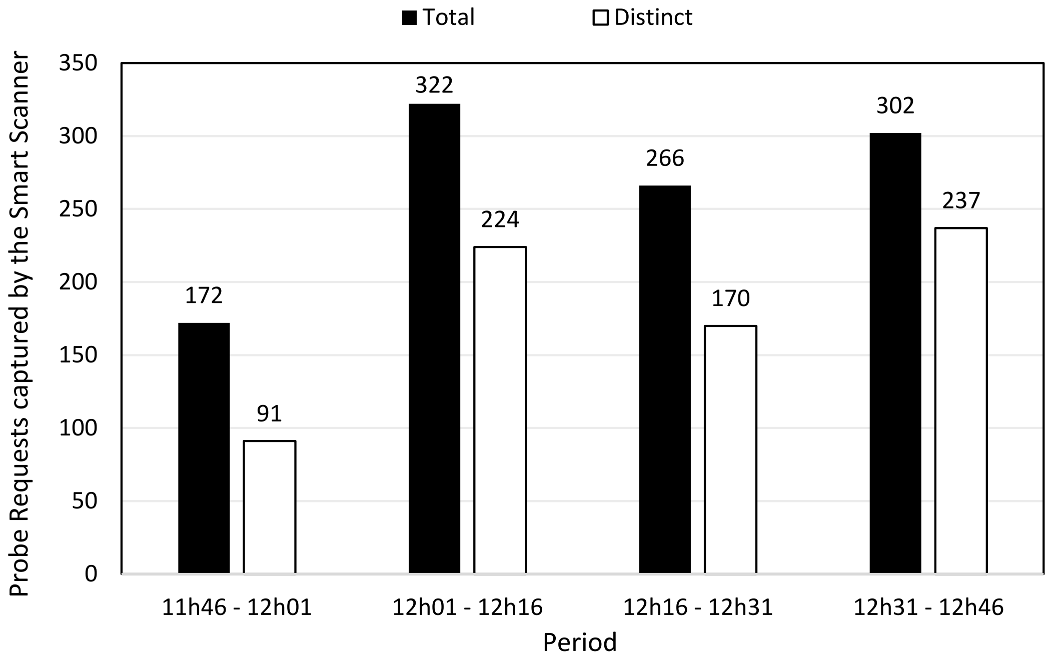


Figure 3.8: Probe requests pushed by the Smart Scanner to the cloud-hosted database, per period.

Interestingly, when aggregating data for the four periods, the number of distinct devices sensed by the Smart Scanner reduces from 722 to 678. The most obvious explanation is that the same person was sensed in two, or more, periods, meaning that there were 44 people within the designated area in different periods. This is indeed useful because the point-of-interest is a pedestrian crossing near a restaurant, with the measurement periods happening just before lunchtime. Therefore, it is quite plausible that a person went through the point-of-interest on a period (e.g., the first one) to have lunch and then passed again in another period (e.g., the last one) after the meal (Figure 3.9).

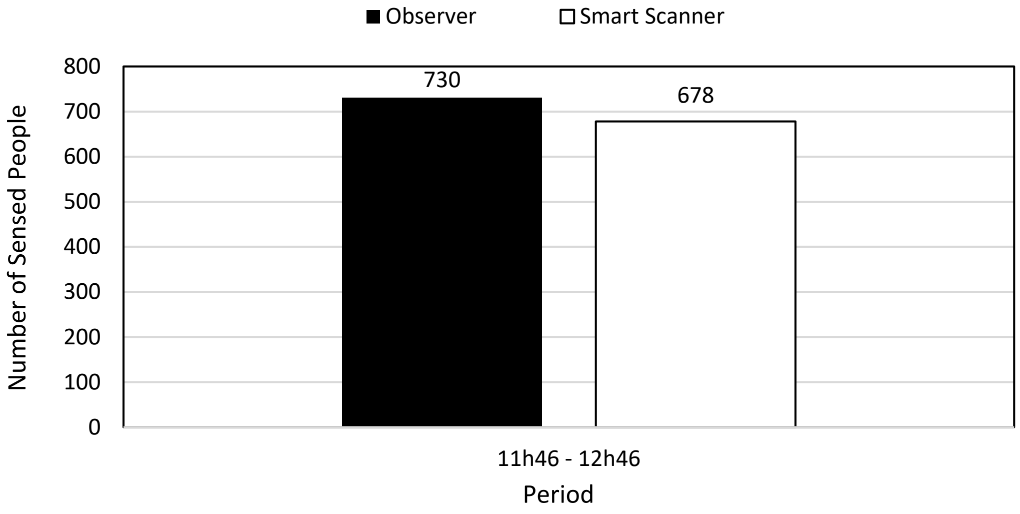


Figure 3.9: Observer's count vs the number of distinct devices captured by the Smart Scanner for all periods together.

Interesting insights can be gathered from the analysis of the captured MAC addresses. In fact, the first three octets of such an address correspond to the Organizationally Unique Identifier (OUI), which identifies the vendor of the device. Although Institute of Electrical and Electronics Engineers (IEEE) offers the ability to purchase a private OUI that does not contain the company's name, such privacy measure seems not to be used by any major manufacturer. The remainder of the address is assigned by the organization, subject to the constraint of uniqueness. In this study, from the 1062 sensed Wi-Fi probe requests, 302 were emitted from Android Devices (da:a1:19) and did not discriminate the vendor (Google, Inc.); 127 had a locally administered address; 66 were from Samsung devices (88:83:22 and 04:d6:aa, among others); 20 from Apple; and, among many others, 2 from Continental Automotive Czech Republic (9c:28:bf) and 1 from PARROT SA (a0:14:3d), a drone's company (Figure 3.10). When these data are decrypted, it is interesting to note a significant number of devices transmitting Google's id or having a locally administered address. This is due to recent measures for MAC randomization, in order to increase one's privacy and hinder the tracking of a person's trajectory using the MAC address.

Overall, the achieved results show promising prospects. The Smart Scanner was able to sense 98% of the crowd and provide relevant insights on important points-of-interest for VRUs. In addition, respecting budgetary limits imposed on any public and private organization, these results were obtained using a 2.4€ arduino-based board, which covered more than 2200 meters^2. In addition, all the produced software was released as open-source, advocating for transparency and reliability. Out of the road safety domain, the Smart Scanner can be used to improve crowd control at big events, improve urban planning, or assemble relevant insights on how visited a specific place is, among many others.

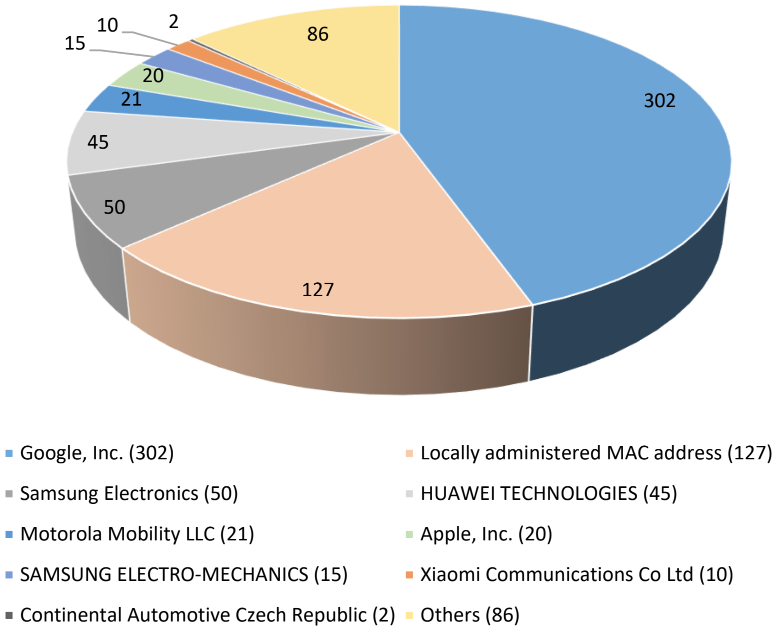


Figure 3.10: Distribution of the sensed devices, by vendor, for the outdoor experiment.

3.2.5.2 Indoor Experiment

Regarding the indoor experiment, as depicted in Table 3.2, a total of 232421 WiFi probe requests were sensed by the Smart Scanner. Interestingly, even though during August the Smart Scanner worked more days than in any other month, it was the month with the smaller amount of sensed probe requests. In fact, in August, the Smart Scanner sensed a mean value of 1802 probe requests per day. On the other hand, in July and September, this value increases to 6744 and 7377 probe requests/day, respectively. This is in line with expectations since in August the university is closed and many people are on vacation. Figure 3.11 shows the variation on the number of sensed probe requests per day as well as the variation of distinct MAC addresses probing each day. In both cases, it is easily noticeable the lower number of sensed probes during August when compared to July and September. It is also noticeable a drastic reduction in the number of sensed probes during weekends when compared to weekdays. In addition, the number of distinct MAC addresses probing is significantly lower when compared to the total count of probes captured each day, meaning that the same MAC is part of multiple probe requests.

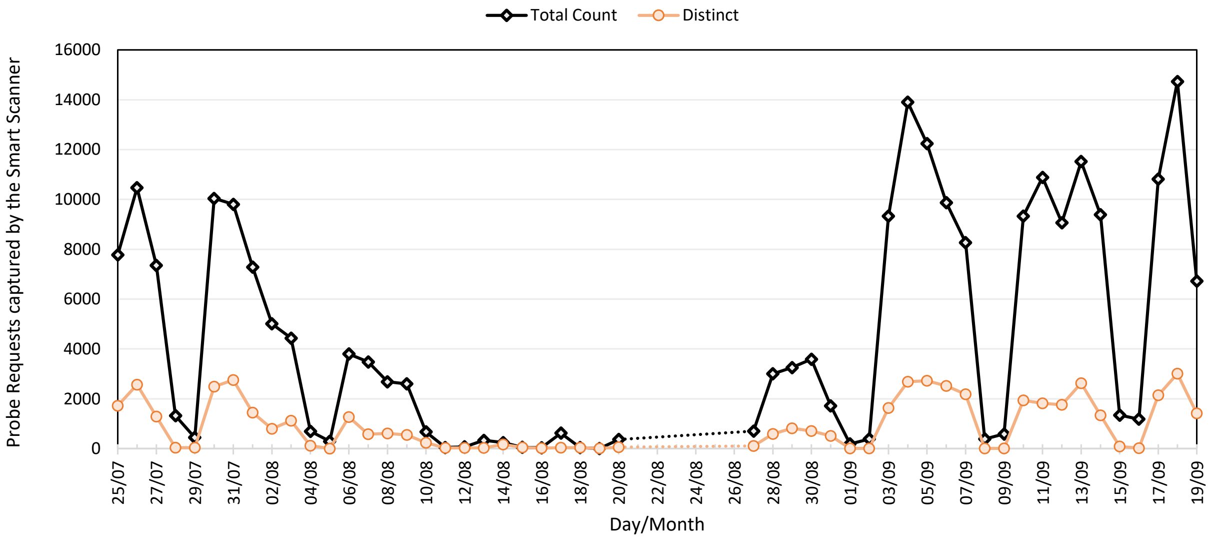


Figure 3.11: Captured probe requests vs distinct MAC addresses probing each day.

The OUI of the 232421 MAC addresses sensed by the Smart Scanner shows the presence of 61 distinct vendors. Besides well-known companies such as Intel (b4:b6:76, among others), Apple (80:e6:50, among others), or Xiaomi (ec:d0:9f, among others), other companies emerge such as the Raspberry Pi Foundation (b8:27:eb and dc:a6:32), Garmin International (10:4e:89) and NVIDIA (00:04:4b). The OUI of 14741 probes were non-identifiable, meaning that they were likely to be locally administered MAC addresses. Overall, since the indoor experiment was carried out inside the ISLab at the Department of Informatics of the University of Minho, it comes as no surprise that Intel Corporate was the vendor with the higher count of sensed probe requests (total of 56993 probes), closely followed by AzureWave Technology (45204). Locally administered MAC addresses are the third most sensed type of OUI (total of 38440 probes, without counting the non-identifiable ones). Figure 3.12 depicts the distribution of the sensed devices, by vendor, for the indoor experiment.

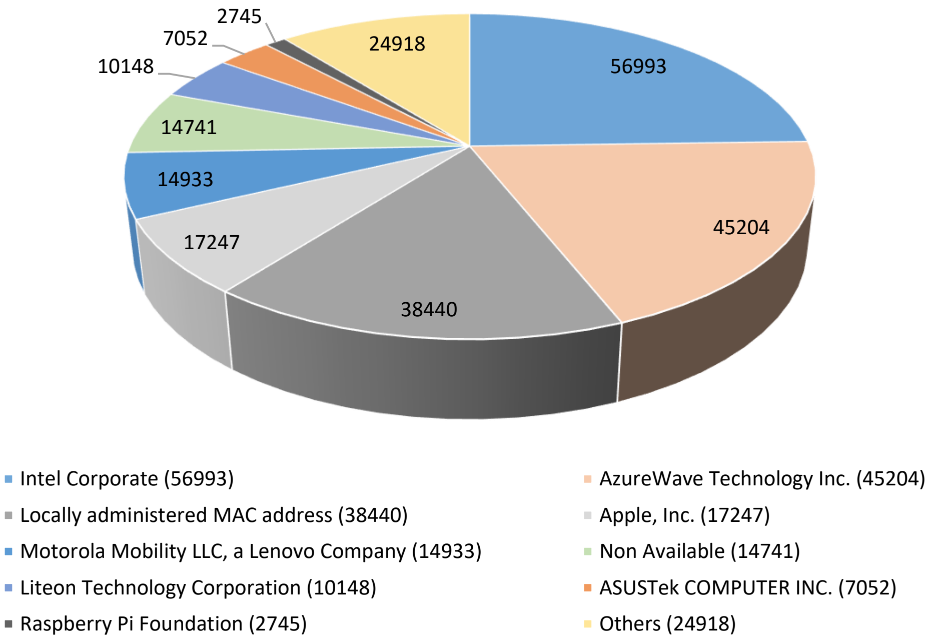


Figure 3.12: Distribution of the sensed devices, by vendor, for the indoor experiment.

3.2.5.3 Summary

As answers to the elicited research questions, it can be said that (RQ3.1 and RQ3.2) arduino-based boards, such as the ESP ones, when properly programmed, provide the ability to implement passive crowd sensing methods in an efficient and economic manner (a cost of 2.4€ per board). Such boards (RQ3.3) were shown to have an outdoor range of, approximately, 27 meters on a straight line, covering an area of 2290 meters^2 when inside a shell to make the board weatherproof. In indoor scenarios, the board's signal was able to cross several horizontal and vertical walls, being caught inside other laboratories, offices, and corridors. In terms of accuracy (RQ3.4), the outdoor experiment showed the ability of the Smart Scanner to provide accurate estimates on the number of people within a certain zone based on the sensed WiFi probe requests (in particular, in specific points-of-interest for VRUs such as pedestrians' crossings). The indoor experiment, on the other hand, showed the ability of the Smart Scanner to work within walls, with practical examples of deployment being the use of the Smart Scanner to control the access to buildings or to know the density of people within libraries, stores, or public services, among many others.

3.3 Active Sensing

A second approach to crowd sensing can be defined as active sensing, i.e., an approach that requires people to install specific software or carry specific hardware that makes them visible to a physical or virtual sensor. Such an approach requires people to become users of a system, being the lack of engagement a relevant challenge. However, active sensing approaches allow one to propose original and contemporary solutions to sense groups of people.

Here, the research conducted in the aforementioned domain aims to sense groups of people in two distinct ways. On the one hand, it aims to define a set of geofences (Figure 3.13), i.e., a location-aware service that establishes virtual perimeters for real-world geographic areas, to alert people, in particular, VRUs, as soon as they enter problematic zones of a city. The goal is to provide useful information to users when they are near an area of interest. On the other hand, it aims to create emotional maps based on the feelings of people, i.e., a city's map where positive feelings of people are quantified and illustrated. For this, users are required to provide their feedback, using a mobile application, about their feelings in regard to certain zones of a city. This study aims to follow an engineering process to give answers to the following research questions:

RQ3.5. Is it possible to conceive and establish virtual perimeters for real-world geographic areas?

RQ3.6. Is it possible to warn people as soon as they enter the vicinity of such areas?

RQ3.7. Is it possible to represent and quantify people's feelings in an emotional map?

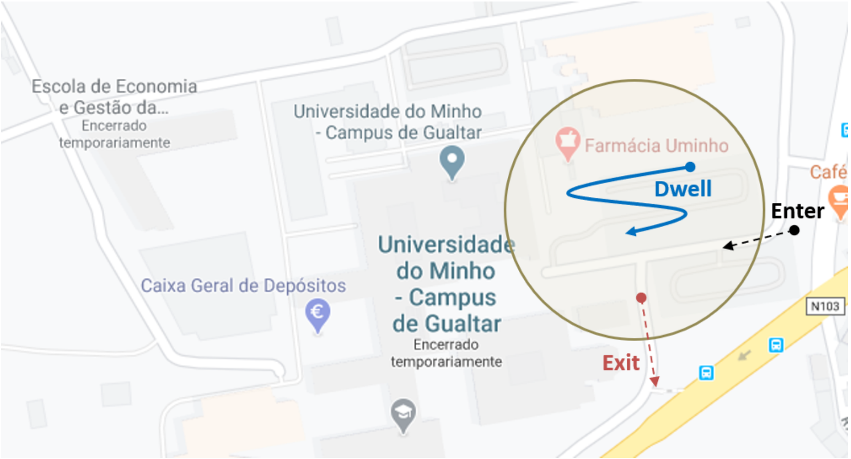

Escola de Economia e Gestão da...
Encerrado temporariamente
Universidade do Minho - Campus de Gualtar
Farmácia Uminho
Café
Dwell
Enter
Universidade do Minho - Campus de Gualtar
Encerrado temporariamente
Caixa Geral de Depósitos
Exit
N103

Figure 3.13: A geofence example.

3.3.1 State of the Art

During the last years, companies have been taking advantage of geofences for activities associated with advertising and sales. For instance, in 2018, Burger King incorporated in their new mobile application a set of geofences that were set within the vicinity of their competitors. As soon as an user entered such geofences, which could mean he was about to visit a Burger King's competitor, the mobile application would give the user a great discount in Burger King's most famous burger [200]. This goes in line with another study that focused on understanding how the use of spatial data can improve advertising performance for customers using geofences [201]. Another example is the one used by real estate agents to sell houses. In such example, geofences are set in places where houses are for sale and, as soon as a potential client walks or drives within range of the property, a notification message is triggered [202]. Both examples focus on the use of APIs rather than physical devices to set the boundaries of the geofences. A different example, however, is presented in [203], where the authors propose an application to track the location of young child's smartphone in real-time. Parents are able to define locations and receive text alerts as soon as the used handset reaches a pre-defined location.

Besides advertising and sales, geofences can also be used to promote and advance other use cases. An example is related with new ways to evolve confinement approaches, tracking people that enter an area of interest or to make people stay within an area during confinement periods. Other use cases focus on defining no-go zones, polluted zones, or shopping areas, among many others. Another example where the use of geofences is common is related to dock-less shared bicycles and scooters, where users can find, ride, and return such means of transport anywhere via GPS-based smartphone apps. In such use case, geofences are used to designate optimal parking spots for renting and returning bicycles and scooters [204]. Finally, a very interesting example is presented in [205], with the authors developing a bio-telemetry system that provides real-time alerts if a telemetered bird flies within the proximity of geofenced wind farms, allowing farmers and authorities to take the appropriate measure to minimize the risk of collision and injury. This study arises from the premise that wind farms have a severe impact on resident and migratory avifauna populations. Another example of a bio-telemetry system using geofences is the one presented by Manley (2017), with the author using such a system to monitor telemetered grizzly bears [206].

In regard to the quantification of people's feelings in an emotional map, as argued in [207], even though emotions are what make us human, their presence on maps and in spatial data are uncommon. In addition, it should be highlighted that emotional maps, more than emotions, can map people's perceptions and experiences in particular places. This goes in line with the assumptions of Mody et al. (2009) that emotions and places are interconnected [208]. In their work, the authors propose a location-based social networking tool that allows its users to share and store their emotional feelings about distinct locations. Their application uses a geo-emotional tagging system, running on a smartphone, to credit user emotions to such locations. On the other hand, in [207], the author developed a web application for the collection of user emotions as well as the visualization of the collected data using a colored hexagonal grid to create a heat-map. To collect data, a set of emotional mapping workshops were carried out in distinct municipalities, with each one having its own set of questions that were performed to the users to quantify their feelings. Our approach, on the other hand, uses a mobile application that allows users to quantify and visualize his/her feelings, and the feelings of the community, using a heat-map over a world's map. Feelings are positive, binary, and organized into distinct categories, which indicate areas suitable to do sports, shopping, that are crime safe, and historically important.

3.3.2 Materials and Methods

Soft and virtual sensors were used to implement both the geofences and the emotional map. In fact, a first attempt was made using beacons, a bluetooth based sensor with low cost, low power transmitters. The beacons transmit a Bluetooth Low Energy (BLE) signal that is captured by smartphones when close to a beacon. The beacons periodically wake up, transmit a BLE signal, and then return to a low-power state, having no user interface or GPS capabilities. This requires people to install software in their smartphones to capture and process the signal. The software is typically programmed to perform a specific action when such signals are captured. An example includes using beacons to signalize stores and then give discounts to the user as soon as he/she is close to the store. Our experiments used beacons to set the boundaries of geofences and, thus, warn users as soon as they entered, or exited, a geofence. The conducted experiments focused on using Gimbal Series 10 beacon, a battery-powered device with a retail price of 4€ each. These beacons use standard CR2032 replaceable batteries that last several months, have small dimensions (40 x 28 x 5.5 mm) and weight (6.52 g), and hold an omni-directional antenna working on the 2.4GHz channels. As advertised by the manufacturer, it has a range of 50 meters, although it can often be detected at further distances in clear line-of-sight conditions. Our tests confirmed these values.

These beacons, however, showed some major drawbacks. Firstly, each beacon needs to be manually activated and configured at Gimbal's Manager page. Secondly, the use of Gimbal's SDK for beacon detection and feature extraction is mandatory, requiring internet access at all times. Thirdly, Gimbal's SDK is not open-source, which created severe problems when trying to debug and understand its behavior. And, finally, since 2018, Gimbal is blocking all purchased beacons and requesting developers to sign a contract and going through a confusing and peculiar process to comply with General Data Protection Regulation (GDPR) even though no personal information is being gathered about the user. The price of each beacon was also a major drawback that prevented scalability. The alternative was to use soft sensors that are made available by existing APIs. This is, in fact, a benefit of APIs, as they do not rely on physical sensors that must be set up in each location. They require, however, users to have the smartphone's GPS on. The next lines describe the used APIs for the creation of geofences and heat-maps.

3.3.2.1 Geofencing API

Several APIs provide the ability to create virtual geofences for real-world geographic areas. Such APIs are typically supported by software companies that offer map-based services, including Google, Here, or TomTom. After analyzing and comparing the available options, the choice was to use Google's Geofencing API. This API, besides being less restrictive than others, offers a set of features suitable for the development of geofences. It not only allows the creation/elimination of geofences, but it also allows the implementation of notifications upon enter, dwell, or leave an area of interest. To define the geofence, one must specify its latitude, longitude, radius, and expiration duration. Multiple geofences can be defined, with a limit of 100 geofences per app, per device.

To use Google's Geofencing API, at least two permissions are required (*ACCESS_FINE_LOCATION* and *ACCESS_BACKGROUND_LOCATION*), with a broadcast receiver being required to listen for geofence transitions. A set of methods must be implemented, including:

1. The creation and addition of geofences using *Geofence* objects. The *GEOFENCE_TRANSITION_ENTER* transition triggers when a device enters a geofence, and the *GEOFENCE_TRANSITION_EXIT* when a device leaves it (Listing 3.6);

Listing 3.6: Creating geofences in Android.

```
geofenceList.add(new Geofence.Builder()
  .setRequestId(geo_name) //requesting the geofence ID, i.e., identify the geofence
  .setCircularRegion(latitude, longitude, radius)
  .setExpirationDuration(Geofence.NEVER_EXPIRE)
  .setTransitionTypes(Geofence.GEOFENCE_TRANSITION_ENTER | Geofence.
       ↪ GEOFENCE_TRANSITION_EXIT)
  .build()
);
```

2. The *GeofencingRequest* class and its nested *GeofencingRequestBuilder* to specify the geofences to monitor and to set how related geofence events are triggered. Specifying *INITIAL_TRIGGER_ENTER* tells that *GEOFENCE_TRANSITION_ENTER* should be triggered if the device is already inside the geofence (Listing 3.7);

Listing 3.7: Monitoring geofences.

```
private GeofencingRequest getGeofencingRequest(){
  GeofencingRequest.Builder builder = new GeofencingRequest.Builder();
  builder.setInitialTrigger(GeofencingRequest.INITIAL_TRIGGER_ENTER);
  builder.addGeofences(geofenceList);
  return builder.build();
}
```

3. Adding, to the *geofencingClient* object, the created geofences (Listing 3.8).

Listing 3.8: Adding the created geofences.

```
GeofencingClient geofencingClient = LocationServices.getGeofencingClient();
geofencingClient.addGeofences(getGeofencingRequest(), getGeofencePendingIntent())
  .addOnSuccessListener(this, new OnSuccessListener<Void>() {
    @Override
    public void onSuccess(Void aVoid){
      //geofences added
    }
  })
  .addOnFailureListener(this, new OnFailureListener() {
    @Override
    public void onFailure(@NonNull Exception e) {
      //failed to add geofences
    }
  });
```

3.3.2.2 Heat-maps API

While Google's Geofencing API is to be used in the Geofencing experiment, for the emotional map we are required to illustrate feelings in a map. For that, the option was to use Google Maps' Android Heat-map Utility API, which gives the possibility of representing the distribution and relative intensity of data points on a map. With heat-maps, rather than placing markers at each location, one can use colors to represent the density of data points. To use this API a set of methods must be implemented, including:

1. Creation and addition of a collection of geographically tagged data points to create an *HeatmapTileProvider* with a customized color gradient. Then calling *GoogleMap.addTileOverlay()* to add the created overlay to the map (Listing 3.9);

Listing 3.9: Creating the heat-map in Android.

```
private fun addHeatMapForCategory(mMap: GoogleMap, categoryData: MutableList<
    ↪ FirebaseFeelings>, userCategoryData: MutableList<FirebaseFeelings>){
  var latLngFeelings = getLatLngPointsFromFirebaseFeelings(categoryData)
  latLngFeelings.addAll(getLatLngPointsFromFirebaseFeelings(userCategoryData))
  if(latLngFeelings.isNotEmpty()){
    val gradient = Gradient(heatMapColors, heatMapStartPoints)
    mProvider = HeatmapTileProvider.Builder()
      .data(latLngFeelings)
      .radius(HEATMAP_RADIUS)
      .gradient(gradient)
      .opacity(HEATMAP_OPACITY)
      .build()
    mOverlay = mMap.addTileOverlay(TileOverlayOptions().tileProvider(mProvider))
  }
}
```

2. Ability to change the dataset upon which a heat-map is built (Listing 3.10);

Listing 3.10: Changing the dataset of geographically tagged data points that make the heat-map.

```
private fun changeDataset(categoryData: MutableList<FirebaseFeelings>, userCategoryData:
    ↪  MutableList<FirebaseFeelings>){
    if(this::mOverlay.isInitialized) {
        var latLngFeelings = getLatLngPointsFromFirebaseFeelings(categoryData)
        latLngFeelings.addAll(getLatLngPointsFromFirebaseFeelings(userCategoryData))
        //set new data
        mProvider.setData(latLngFeelings)
        //clear cache
        mOverlay.clearTileCache()
    }
}
```

3. Ability to remove a heat-map (Listing 3.11).

Listing 3.11: Removing the heat-map.

```
fun removeHeatmap(){
   if(this::mOverlay.isInitialized)
      mOverlay.remove()
}
```

3.3.2.3 Technologies

Both the geofences as well as the city's map from people's feelings experiments use APIs that are made available by Google. Both experiments target the Android operating system, using Java and Kotlin as programming languages. Android Marshmallow is the lowest supported version (API version 23), with Android Pie as target version (API version 28). Android Studio 3.5.2 is the used IDE. To store and synchronize data, Firebase's Cloud Firestore was used.

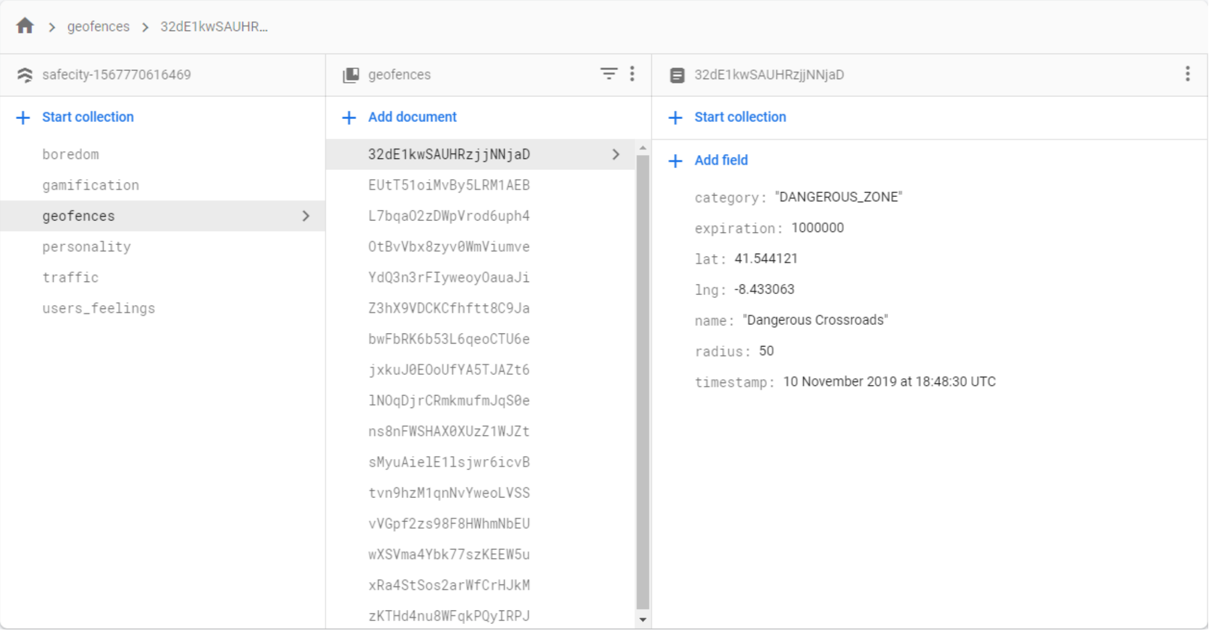
3.3.3 Geofencing

With the goal of providing useful information to people when they are near an area of interest, geofences were conceived to establish virtual perimeters over real-world geographic areas. The focus was set in VRUs and on alerting them as soon as they enter zones of the city where their vulnerability increases. In fact, four distinct categories were elicited and defined, namely high-traffic, polluted, crowded, and dangerous zones. Each geofence is associated with one and only one category, being further characterized by its geographic coordinates, a radius, a name, an expiration duration, and the timestamp that specifies when the geofence was created (Listing 3.12).

Listing 3.12: *FirebaseGeofence* object.

```
data class FirebaseGeofence (
   val name: String? = null,
   val category: GeofenceCategory? = null,
   val lat: Double? = null,
   val lng: Double? = null,
   val radius: Float? = null,
   val expiration: Long? = null,
   @ServerTimestamp
   var timestamp: Date? = null
)
```

Geofences are stored in Firebase's Cloud Firestore, a cloud-hosted NoSQL database that supports hierarchical data structures (as detailed in Section 2.3.2). In this database, the *geofences* collection contains all existing geofences in the form of documents. Each document has a unique identifier and a set of fields that characterize the geofence. Figure 3.14 depicts the hierarchical data structure to store geofences as well as an example of a geofence categorized as being a dangerous zone due to the existence of unsafe crossroads within the covered area.



geofences > 32dE1kwSAUHR...

safecity-1567770616469

Start collection

boredom
gamification
geofences
personality
traffic
users_feelings

geofences

Add document

32dE1kwSAUHRzjjNNjaD
EUtT51oiMvBy5LRM1AEB
L7bqa02zDWpVrod6uph4
OtBvVbx8zyv0WmViumve
YdQ3n3rFIyweoyOauaJi
Z3hX9VDCKCfhftt8C9Ja
bwFbRK6b53L6qeoCTU6e
jxkuJ0EOoUfYA5TJAZt6
lNOqDjrCRmkmufmJqS0e
ns8nFWSHAX0XUzZ1WJZt
sMyuAielE1lsjwr6icvB
tvn9hzM1qnNvYweoLVSS
vVGpf2zs98F8HWhmNbEU
wXSVma4Ybk77szKEEW5u
xRa4StSos2arWfCrHJkM
zKTHd4nu8WFqkPQyIRPJ

32dE1kwSAUHRzjjNNjaD

Start collection

Add field

category: "DANGEROUS_ZONE"
expiration: 1000000
lat: 41.544121
lng: -8.433063
name: "Dangerous Crossroads"
radius: 50
timestamp: 10 November 2019 at 18:48:30 UTC

Figure 3.14: Firebase's Cloud Firestore cloud-hosted NoSQL database holding geofence objects.

Geofences are created by users with an administrator permission (read and write permissions). Only authenticated users are able to access the existing geofences (read permission). Access rules are set in Firebase's Cloud Firestore database, as depicted in Listing 3.13.

Listing 3.13: Access rules to the *geofences* collection.

```
rules_version = '2';
service cloud.firestore {
  match /databases/{database}/documents {
    //default rule
    match /{document=**} {
      allow read, write: if false;
    }
    //geofences Collection
    match /geofences/{geofences} {
      allow read: if request.auth != null;
    }
  }
}
```

With the data model and the support infrastructure defined, a mobile application, targeting the Android operating system, was conceived to provide alerts to its users as soon as they enter, or leave, a geofence. It presents a map-based layout with geofences being drawn in the map with different colors depending on its category. As permissions, it requires the *ACCESS_FINE_LOCATION* one, which allows the application to obtain the user's location as precise as possible from the available location providers, including the GPS, WiFi, and mobile cell data, and the *ACCESS_BACKGROUND_LOCATION* permission, which allows the application to access the user's location when running in the background. The structure of the developed application consists of several packages, viz.

1. *util* package, where the *GeofenceCategory* enum class represents all possible geofence's categories;

2. *model* package, containing all classes that represent data objects, including the *FirebaseGeofence* object (Listing 3.12);

3. *handlers* package, containing all handlers responsible for the domain's logic:

   a) *GeofenceHandler*, which is responsible for implementing Google's Geofencing API. Includes sub-classes and methods to load, monitor, and draw geofences as well as to monitor geofence transitions and send notifications as soon as a transition is detected;

   b) *FirebaseHandler*, which includes all logic to access, write, and read from Firebase's Cloud Firestore database;

   c) *PermissionHandler*, implementing the required methods to obtain users' permission. Includes methods to check if a permission was granted as well as to ask the user to grant a permission;

   d) *GpsHandler*, which contains the logic to enable and disable the device's GPS;

   e) *LocationHandler*, being responsible for obtaining the user's current location if the user granted such permission. Includes methods to obtain the current location as well as to start and stop periodic location updates.

4. *activity* package, where the *GeofenceActivity* is defined. An activity provides the window in which the application draws its interface, with the Android system initiating the code in an activity by invoking methods that correspond to specific stages of the activity's life cycle. *GeofenceActivity* implements methods to inflate the layout, set all button behaviors, and use the referred handlers.

Figure 3.15 presents the mobile application's logic in a simplified sequence diagram. The first steps consist of inflating the view and initializing the buttons' behavior. Then, if permission is granted by the user to access his position, the application starts the smartphone's GPS, and the *LocationHandler* starts updating, periodically, the user's position in the map in parallel to the main processing flow. This keeps happening until the GPS is turned off or permissions are revoked. Geofences are then loaded and monitored, with the *GeofenceHandler* handling all notification callbacks as soon as the user enters, or leaves, a geofence. Notifications are still sent even with the application dismissed or running in the background. If the GPS is off or if no permission is granted, the user is still able to see the geofences in the application's GUI but notifications will not be pushed.

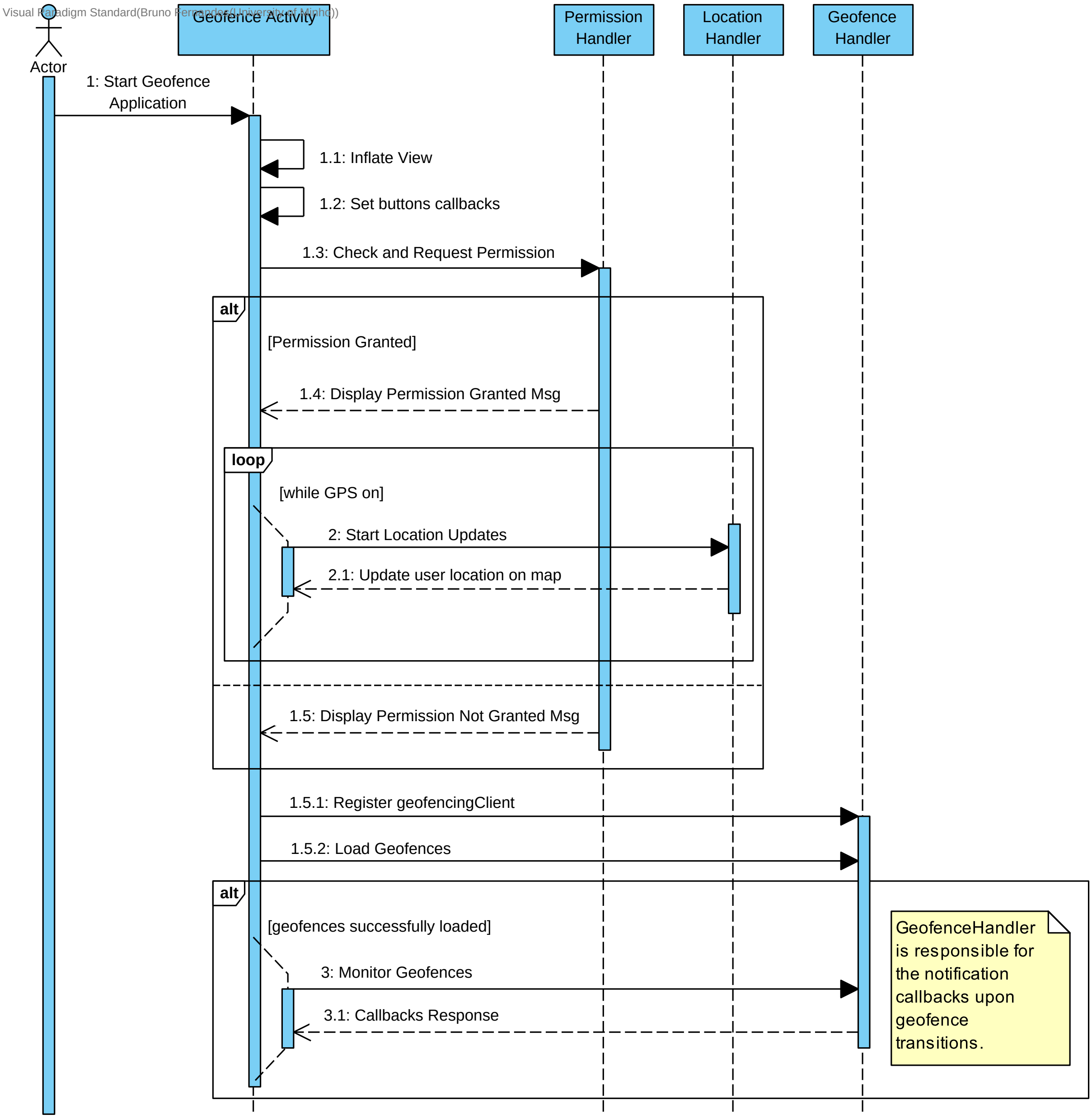
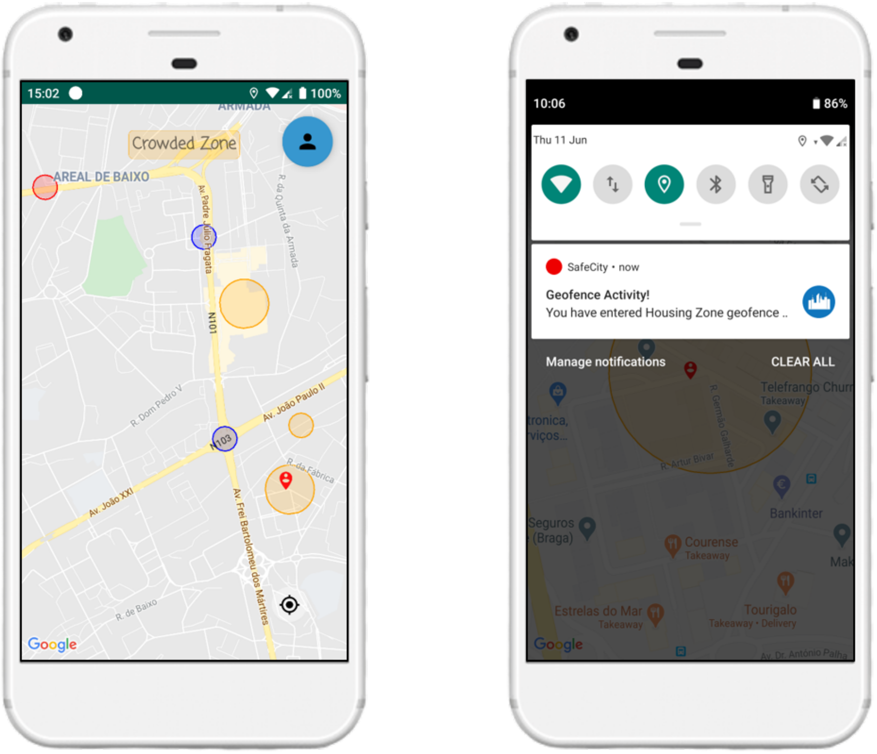

Visual Paradigm Standard(Bruno Fernandes(University of Minho))
Actor
Geofence Activity
Permission Handler
Location Handler
Geofence Handler
1: Start Geofence Application
1.1: Inflate View
1.2: Set buttons callbacks
1.3: Check and Request Permission
alt
[Permission Granted]
1.4: Display Permission Granted Msg
loop
[while GPS on]
2: Start Location Updates
2.1: Update user location on map
1.5: Display Permission Not Granted Msg
1.5.1: Register geofencingClient
1.5.2: Load Geofences
alt
[geofences successfully loaded]
3: Monitor Geofences
3.1: Callbacks Response
GeofenceHandler is responsible for the notification callbacks upon geofence transitions.

Figure 3.15: Simplified sequence diagram of the geofencing activity.

In regard to the application's layout, it consists of a *ConstraintLayout* with an embedded google maps fragment where geofences are illustrated. The user can navigate anywhere on the map and zoom in/out. A button in the bottom right corner allows the user to center the view on his current position. If a geofence is clicked, it displays its category at the top of the screen. As soon as the user enters or leaves a geofence, a notification emerges in the smartphone, warning the user of such a fact. Figure 3.16 depicts the main view of the conceived mobile application. In Figure 3.16a it is shown the map view of the application with several geofences of distinct categories and radius, painted over the map, with the user inside a crowed geofence. On the other hand, Figure 3.16b presents the notification received by the user as soon as he enters a geofence.

A case study was implemented to evaluate the conceived application. The first step focused on defining



SafeCity • now

Geofence Activity!

You have entered Housing Zone geofence ..

(a) User entered a crowded geofence. Red marker sets the user location.

(b) Notification from entering a crowded geofence named "Housing Zone".

Figure 3.16: Main views of the mobile geofencing application.

a set of geofences in the city of Braga, in Portugal. Sixteen were defined and categorized. Then, with the GPS on and the application installed in a Xiaomi Mi A1 smartphone running Android 9 (API level 28), the subject moved around the city on a rainy day using two distinct transportation means: on foot and in a car. The goal was to analyze the application's performance in terms of latency between entering/leaving a geofence and the reception of the corresponding notification. As expected, when walking the application's performance was better than when inside a car, i.e., the application was quicker to perceive that the subject entered/left a geofence when on foot. In fact, it took around 2 seconds for notifications to be pushed when entering a geofence. To push exit notifications, the application took approximately the same time.

When using a car as mean of transport, the performance clearly deteriorates as speed increases. Indeed, in cases where the geofence had a small radius (less than 50 meters), since vehicles go through the geofence almost instantaneously, no entry neither exit notification is pushed to the subject's smartphone. This happens because the *LocationHandler* is unable to query the user's location within an acceptable time frame. On the other hand, geofences with a higher radius would be detected depending on the speed of the vehicle or the traffic status, for example. At low speed, the obtained results on a vehicle show similarities of those when walking, with notifications being pushed successfully and in an acceptable time (a delay of three to four seconds). No experiments were conducted when riding a bike. However, since in an urban context the speed of a cyclist is closer to that of a pedestrian, response times are expected to be similar but with a possible deterioration. On the other hand, when driving a motorcycle, response times are expected to be closer to that of a car.

Some aspects of the conducted experiments must be highlighted. Firstly, all experiments were conducted on a rainy day, which can significantly influence GPS precision and negatively affect response times. Secondly, smartphone's GPS quality is yet another factor that may positively, or negatively, affect response times. Finally, with regard to the background operation of the conceived application, notifications were successfully pushed even with the application dismissed or in the background, if the smartphone's screen is not locked. However, if the subject locks the smartphone's screen, the notification is only pushed when the screen is unlocked, which can happen several minutes after going through a geofence.

As direct answers to the elicited research questions, it can be said that (RQ3.5) Google's Geofencing API facilitated the conception of geofences and the monitoring of geofence transitions, which were drawn over a map for user convenience. The conceived application (RQ3.6) was able to successfully warn pedestrians and runners as soon as they entered, or left, a geofence within an acceptable time frame. In regard to car occupants, response times depend on the size of the geofence as well as the vehicle's speed.

3.3.4 Emotional Map from People's Feelings

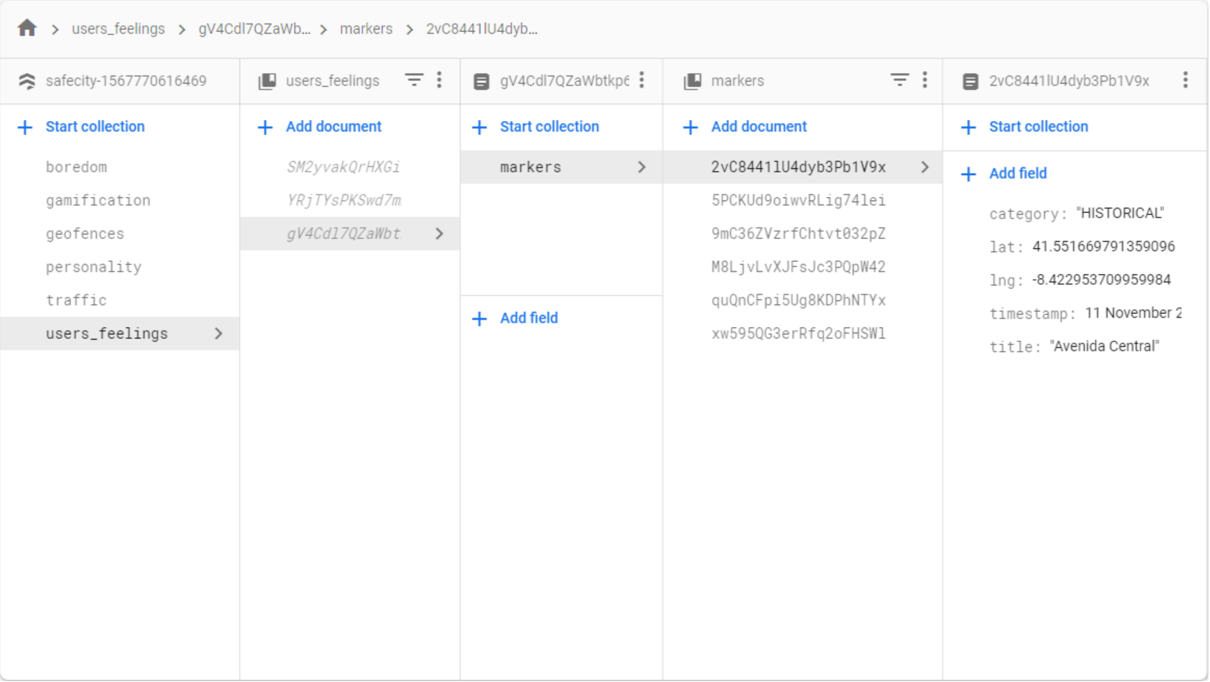
With the goal of gathering insights from a city's crowd in regard to their feelings towards certain zones and categories of a city, a mobile application containing an emotional map was conceived to quantify and illustrate such feelings. The expectation is to listen to several individuals and, thus, take advantage of the wisdom of crowds to quantify the overall feeling of people towards their city.

Three distinct positive categories were defined, in particular, attractive zones in terms of shopping, sports, and history. One negative category is used to refer to crime hotspots. Feelings are binary and categorically tagged, being further characterized by its geographic coordinates, a title, and the timestamp that specifies its creation date (Listing 3.14).

Listing 3.14: *FirebaseFeelings* object.

```
data class FirebaseFeelings(
   val title: String? = null,
   val category: FeelingsCategory? = null,
   val lat: Double? = null,
   val lng: Double? = null,
   @ServerTimestamp
   var timestamp: Date? = null
)
```

Similar to geofences, feelings are also stored in an instance of Firebase's Cloud Firestore database. The *users_feelings* collection holds a document for each user of the application. Each document contains a nested collection of *markers*. This collection, in turn, contains all feelings of the corresponding user in the form of documents (one document per feeling). Each document is an instance of the *FirebaseFeelings* object. Figure 3.17 depicts the hierarchical data structure to store feelings as well as an example of a feeling for a historical zone (geographic coordinates = 41.551669, -8.422953) named "Avenida Central".



users_feelings > gV4Cdl7QZaWb... > markers > 2vC8441lU4dyb...

safecity-1567770616469
boredom
gamification
geofences
personality
traffic
users_feelings

users_feelings
SM2yvakQrHXGi
YRjTYsPKSwd7m
gV4Cdl7QZaWbt

gV4Cdl7QZaWbtkp€
markers

markers
2vC8441lU4dyb3Pb1V9x
5PCKUd9oiwvRLig74lei
9mC36ZVzrfChtvt032pZ
M8LjvLvXJFsJc3PQpW42
quQnCFpi5Ug8KDPhNTYx
xw595QG3erRfq2oFHSWl

2vC8441lU4dyb3Pb1V9x
category: "HISTORICAL"
lat: 41.551669791359096
lng: -8.422953709959984
timestamp: 11 November 2
title: "Avenida Central"

Figure 3.17: Firebase's Cloud Firestore cloud-hosted NoSQL database holding feelings objects.

Listing 3.15 details the access rules to the *users_feelings* collection. Users are able to write feelings only to their documents (read and write permissions). Authenticated users can read the feelings of the entire community (read permission). This is necessary since the mobile application needs to access all community feelings to draw the emotional map.

Listing 3.15: Access rules to the *users_feelings* collection.

```
service cloud.firestore {
 match /databases/{database}/documents {
  match /{document=**} {
   allow read, write: if false;
  }
  match /users_feelings/{userId} {
   allow create, update: if belongsTo(userId);
  }
  match /users_feelings/{userId}/{document=**} {
   allow write: if belongsTo(userId);
  }
  match /{path=**}/markers/{document} {
   allow read: if request.auth != null;
  }
  function belongsTo(userId) {
   return request.auth.uid == userId
  }
 }
}
```

With the data model and the support infrastructure defined, a mobile application, targeting the Android operating system, was conceived to allow users to add categorized feelings to certain zones of a city as well as view the feelings of the entire community using a heat-map perspective. As permissions, the application only requires the *ACCESS_FINE_LOCATION* one to obtain the user's location and display it on the emotional map. An internet connection is required to commit new feelings and update the emotional map with the community's feelings. In terms of structure, the conceived application consists of the following packages, viz.

1. *util* package, where the *FeelingsCategory* enum class represents all possible feelings's categories;

2. *ui* package, containing the fragment view for each category;

3. *model* package, containing all classes that represent data objects, including the *FirebaseFeelings* object (Listing 3.14);

4. *handlers* package, containing all handlers responsible for the domain's logic:

   a) *FeelingsMapHandler*, responsible for loading the user's and the community's feelings. It is also responsible for implementing Google Maps' Android Heat-map Utility API. Includes sub-classes and methods to compute and draw the heat-map, per category; to draw the user's feelings in the map; to add/remove user's feelings; and to validate that the minimum distance between feelings is respected;

   b) *FirebaseHandler*, which includes all logic to access, write, and read from Firebase's Cloud Firestore database;

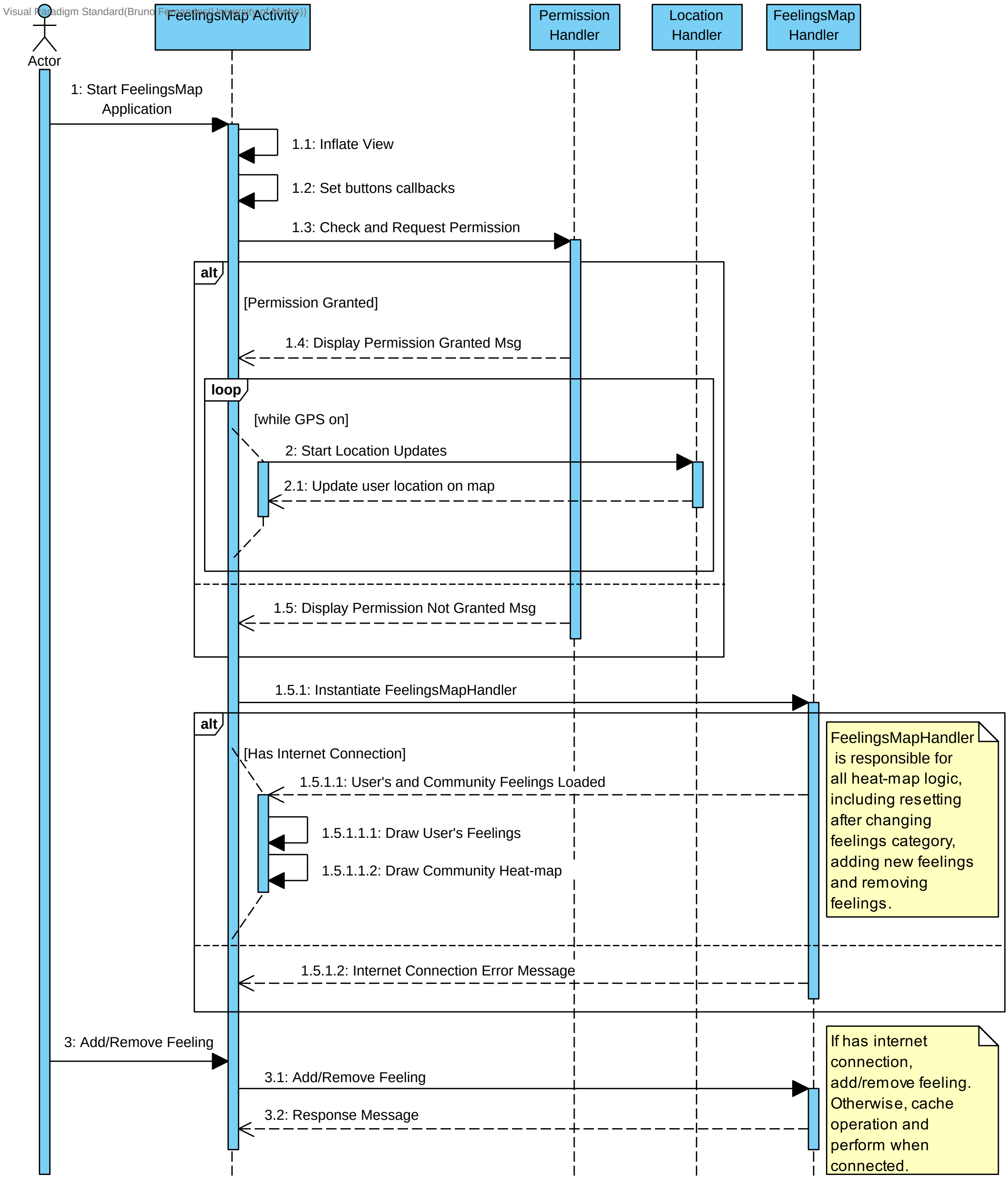
   c) *PermissionHandler*, implementing the methods to obtain the required permission;

   d) *GpsHandler*, which contains the logic to enable and disable the device's GPS;

   e) *LocationHandler*, being responsible for obtaining user's current location if the user granted such permission. Includes methods to obtain the current location as well as to start and stop periodic location updates.

5. *activity* package, where the *FeelingsMapActivity* is defined. It implements methods to inflate the layout, set all button behaviors, and use the referred handlers.

Figure 3.18 depicts the application's logic in a simplified sequence diagram. After inflating and initializing the activity's view, if the user granted permission, the application starts the smartphone's GPS and the *LocationHandler* starts updating, periodically, the user's position in the map in parallel to the main processing flow. The activity then instantiates the *FeelingsMapHandler*, which immediately checks if an active internet connection is available. If true, all feelings are loaded, the heat-map is built, and the user's feelings are drawn on the map. The sequence diagram also shows the addition/removal of a feeling

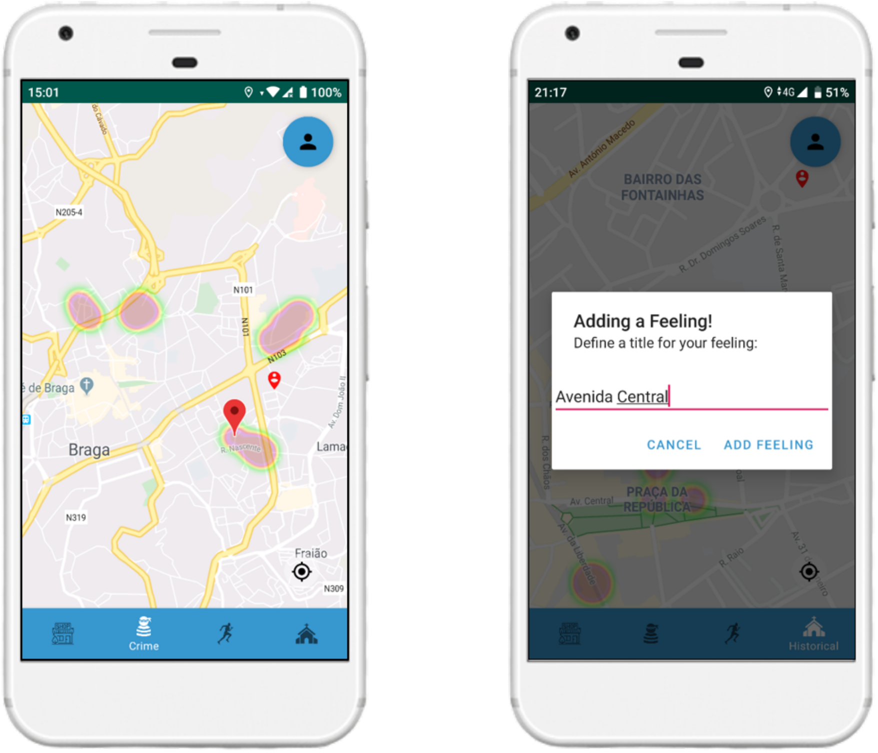


Visual Paradigm Standard(Bruno Fernandes(University of Minho))
Actor
FeelingsMap Activity
Permission Handler
Location Handler
FeelingsMap Handler
1: Start FeelingsMap Application
1.1: Inflate View
1.2: Set buttons callbacks
1.3: Check and Request Permission
alt
[Permission Granted]
1.4: Display Permission Granted Msg
loop
[while GPS on]
2: Start Location Updates
2.1: Update user location on map
1.5: Display Permission Not Granted Msg
1.5.1: Instantiate FeelingsMapHandler
alt
[Has Internet Connection]
1.5.1.1: User's and Community Feelings Loaded
1.5.1.1.1: Draw User's Feelings
1.5.1.1.2: Draw Community Heat-map
1.5.1.2: Internet Connection Error Message
FeelingsMapHandler is responsible for all heat-map logic, including resetting after changing feelings category, adding new feelings and removing feelings.
3: Add/Remove Feeling
3.1: Add/Remove Feeling
3.2: Response Message
If has internet connection, add/remove feeling. Otherwise, cache operation and perform when connected.

Figure 3.18: Simplified sequence diagram of the feelings map activity.

by the user, which is managed by the *FeelingsMapHandler*. If no internet connection is available, the operation is cached and realized, if still valid, as soon as an active connection becomes available.

Figure 3.19 illustrates the main views of the feelings map application. In Figure 3.19a it is depicted the emotional map using a heat-map view, representing the distribution and relative intensity of data points on the map. This view presents the feelings of the community towards crime hotspots in the city of Braga. To change to a different category, the user is just required to use the navigation controller at the bottom of the view. This figure also shows a red marker that corresponds to a feeling added by the logged user. All other



Adding a Feeling!
Define a title for your feeling:
Avenida Central
CANCEL ADD FEELING

(a) Heat-map view of the emotional map.

(b) Adding a new feeling to the map.

Figure 3.19: Main views of the emotional map application.

feelings that make the heat-map belong to other users. To add a feeling to the emotional map, the user is required to make a long click in the zone where he intends to add the feeling. Figure 3.19b shows the dialog that emerges as soon as a long click is performed. It asks the user to give a name to the feeling. Internally, it stores the geographic coordinates where the click was made as well as the corresponding category. After adding the feeling, it is inserted in the heat-map, it is committed to the user's collection in Firebase, and becomes immediately available to the community. To prevent users from spamming a zone with feelings, the feelings of a same user for the same category must be, at least, 200 meters apart in a straight line. An error message emerges if the user tries to add a feeling that does not respect this distance. To remove a feeling, the user must click on the desired feeling (for example, the red marker in Figure 3.19a) and then confirm the removal. Switching categories reloads the feelings of the corresponding category and draws them on the map. In itself, the application's layout consists of several $ConstraintLayout$s with a $BottomNavigationView$ and an embedded google maps fragment. The user can navigate anywhere on the map and zoom in/out. A button in the bottom right corner allows the user to center the view on his current position.

The heat-map view (RQ3.7) provided a simple yet intuitive way to represent binary feelings (or there is a feeling or not) from a community of people over a map. It also allowed an easy management of one's feelings. The application was tested and tuned using distinct smartphones. It was then made available online. The goal becomes now to advertise and distribute the application to a wider audience in order to obtain more data from where a recommender system can be developed. Future improvements also include setting a scale for feelings instead of a binary approach.

3.4 Summary

The study and experiments detailed in this chapter aim to provide an answer to Task 3. For that, this chapter focuses on the second C of what we call the 3-C, i.e., *City*, ***Crowd***, and *Citizen* sensing. The goal is to sense large groups of people. To achieve such a goal, two distinct approaches were followed. On the one hand, a passive crowd sensing approach allowed us to sense people, at specific points-of-interest for VRUs without requiring them to carry any hardware neither install any software. After studying and analyzing the available hardware, arduino-based boards, in particular ESP ones, showed to provide the ability to capture WiFi and bluetooth probe requests, when properly programmed. In the conducted experiments, ESP8266 ESP-12E NodeMCU Amica boards were used to demonstrate the feasibility of conceiving Smart Scanners for passive crowd sensing. These boards have a reduced cost and a set of impressive features.

Range measurement experiments allowed us to further clarify the board's range in an outdoor environment (with a shell all around the board to make it weatherproof) and in an indoor environment. The Smart Scanner was able to properly sense crowds both in outdoor as well as in indoor scenarios. Outdoor scenarios allow the sensorization of important points-of-interest for VRUs such as pedestrians' crossings (the outdoor case study), roads, and sidewalks, among many others. In indoor environments, passive crowd sensing allows the conception of systems to control the access to buildings or to know the density of people within libraries, stores, or public services, among many others. It becomes also possible to further unscramble the captured MAC addresses, which, to some extent, can describe the sensed population in regard to their devices (computers, smartwatches, and smartphones, among other types of devices).

To complement the conducted experiments, an active sensing approach to crowd sensing focused on designing and conceiving a mobile application for geofencing as well as an emotional map to quantify people's feelings. The first focused on providing useful information to VRUs when they are near zones with high traffic, high pollution, crowed, or dangerous, allowing such users to be more aware of their surroundings or simply avoid such areas. The conceived application showed an acceptable latency between entering/leaving a geofence and the reception of the corresponding notification when on foot. On the other hand, when inside a car, performance deteriorates with higher speeds and lower geofence radius. The second approach, on the other hand, focused on gathering insights from a city's crowd in regard to their feelings towards certain zones and categories of the city, in particular, if an area is suitable to do sports, shopping, if it is crime safe, and historically important. A mobile application was conceived to depict the emotional map using a heat-map approach based on the community's feelings. An user may, at any time, add or remove new feelings to quantify a certain area. Both mobile applications were tested and tuned with distinct smartphones and made available online. The expectation is that the emotional map grows with the community engagement, allowing recommender systems to emerge, providing useful information to VRUs if, for example, they intend to do sports in an unfamiliar city.

To aid proving the research hypothesis and validate the solutions' specimens, the conducted experiments focused on providing an answer to ***Research Question 3****: Is it possible to implement passive methods for crowd sensing that require no participation from people?*. The obtained results show promising results in regard to the feasibility of implementing passive methods for crowd sensing. In fact, arduino-based boards, such as the used ESP8266, when properly programmed, provide the ability to capture probe requests emitted by devices carried out by people in their daily lives efficiently and economically, working in indoor and outdoor environments. In addition, two active crowd sensing approaches were implemented to fulfill the performed experiments from a different angle, i.e., establishing people as an active user that is required to install pieces of software in their devices.

The work performed in this chapter closes **Task 3**. Two Cs, from the 3-C, have now been addressed. The next chapter details the last C. In addition, the work presented in this chapter gave origin to a book chapter ([209]) and a journal paper ([66]), being performed in collaboration with researchers from the ALGORITMI Centre, Braga, Portugal, and the Polytechnic Institute of Porto, Felgueiras, Portugal.

Chapter 4

Citizen Sensing

Each individual has its own characteristics. This diversity is what makes us special and unique. Hence, in this chapter we present and discuss new ways to capture such features in a manner that, when complemented with methods to sense a city and groups of people, allows the proposal of contemporary solutions to improve the safety of those more vulnerable on the road.

4.1 Rationale

The ability to sense a person is what drives the work presented in this chapter. In fact, to aspire for an IoP we are required to include each person as an active, proactive, and reactive element of the city's ecosystem. To do that, we must promote the citizen sensor, enabling people to sense themselves, others, and the environment where they stand, allowing the creation of knowledge from extended sets of data.

The work here presented focuses on three distinct aspects of citizen sensing. The first approach consists in conceiving and evaluating ML models for personality assessment, i.e., the goal is to further reduce Saucier's Mini-Marker test to a "game of words" where the subject, instead of rating forty adjectives, only has to select those he relates the most, removing the need to rate adjectives. The proposed Adjective Selection to Assess Personality (ASAP) method replaces the entire process of rating adjectives by an adjective selection process. To collect a significant dataset we were required to design and conceive a platform for data collection, requesting subjects to rate each adjective and select those describing him the most. The goal is to reduce the complexity of tests, the time it takes to perform a test, and to make the test more attractive and easier to implement in current and future technological platforms.

The second approach consists in establish the ability to detect and predict boredom when using a smartphone, i.e., using only data that is made available by smartphones' sensors, without considering any biometric neither intrusive data. To achieve this goal, it is of the utmost importance the existence of a dataset from where one could conceive and evaluate candidate models. Therefore, we were again required to design and conceive a mobile application for data collection. This opens new ways to entertain people, to provide actionable information, and to create more robust recommender systems, which may vary from recommending tasks from to-do lists to targeted advertising, among many others.

Finally, the third approach consists of a meta-analytic study focused on proposing a method to Knowledge Representation and Reasoning (KRR) centered on logic programming, which establishes a formal logical inference engine that is complemented with an ANNs line to computation. The data used in the case study are based on the questionnaires prepared by Papadimitriou et al. (2017), which were developed to capture key components affecting pedestrian walking and crossing behaviors, attitudes, motivations, and habits, i.e., the human factors that characterize VRUs [210]. Understanding the behavior of VRUs on the road, on the one hand, may enhance their safety and, on the other, may help those that design and plan pedestrian crossings and all other road features.

4.2 An Adjective Selection Personality Assessment Method

People react differently when experiencing the same situations. This behavioral diversity may be due to one's experience, knowledge, or even personality. Indeed, several studies have already established a relationship between a person's personality and aggressive reactions [49], work performance [50], or infidelity [51], just to name a few. Semantically, personality may be defined as a set of characteristics that refer to individual differences in ways of thinking, feeling, and behaving [52]. Personality has a great impact on the way we live our lives, either by the way we behave, feel, or interact with others. Hence, there has always been great interest in model, or quantify, a person's personality using either qualitative or quantitative metrics [58].

Several personality tests are available such as HEXACO-60 [211], Myers-Briggs Type Indicator [212], the Enneagram of Personality [213], and NEO-personality-inventory [214]. However, Goldberg's 100 Unipolar Markers remains one of the most popular ways to measure personality traits, in particular, the Big Five [215]. It consists of a total of 100 adjectives, or markers, that the subject must rate on how they relate to each adjective, with 1 being *Extremely Inaccurate* and 9 as *Extremely Accurate*. Among the full set of markers one may find adjectives such as *kind*, *careless*, *efficient*, *philosophical* or *energetic*. Goldberg's test allows one to measure five domains, in particular, *Surgency*, *Agreeableness*, *Conscientiousness*, *Emotional Stability*, and *Intellect*. The OCEAN model, on the other hand, consists of the following five factors [216, 217]:

- *Openness*: related to one's curiosity, imagination, and openness to new experiences. Higher values usually emerge on people that enjoy new adventures and ideas. Lower values tend to emerge on more conservative people;
- *Conscientiousness*: related to self-discipline, being careful and diligent, organized and consistent, pursuing long-term goals. Less conscientious people tend to be more spontaneous and imaginative;

- *Extraversion*: related to a state where a person seeks stimulation from being with others. Extroverted people tend to be energetic and talkative, while introverted ones are reserved;

- *Agreeableness*: related to behavioral characteristics such as being kind and sympathetic. Agreeable people tend to be friendly, cooperative, and empathetic. Non-agreeable people are less cooperative and more competitive;

- *Neuroticism* (opposite of *Stability*): related to being moody and showing signs of emotional instability. Neurotic people tend to be stressed and nervous. Non-neurotic people are more emotionally stable [217].

Goldberg's test consist of 100 unipolar markers that must be quantified by a subject. An important reduction to the set of markers was performed by Gerard Saucier with The Mini-Marker test, using a sub-set of 40 markers to assess the Big Five with an acceptable performance, leading to the use of less difficult markers [218]. Saucier's test uses the same rating scale, being made of five disjoint-sets of eight unipolar markers each:

- *Intellect or Openness* trait is made of six positively weighted adjectives (*intellectual*, *creative*, *complex*, *imaginative*, *philosophical*, and *deep*) and two negative ones (*average* and *ordinary*);

- *Conscientiousness* trait is made of four positively weighted adjectives (*systematic*, *practical*, *efficient*, and *orderly*) and four negative ones (*disorganized*, *careless*, *inefficient*, and *sloppy*);

- *Extraversion* trait is made of four positively weighted adjectives (*extroverted*, *talkative*, *energetic*, and *bold*) and four negative ones (*shy*, *quiet*, *withdrawn*, and *bashful*);

- *Agreeableness* trait is made of four positively weighted adjectives (*kind*, *cooperative*, *sympathetic*, and *warm*) and four negative ones (*cold*, *harsh*, *rude*, and *distant*);

- *Emotional Stability* trait is made of two positively weighted adjectives (*relaxed* and *mellow*) and six negative ones (*moody*, *temperamental*, *envious*, *fretful*, *jealous*, and *touchy*).

During the last years, multiple studies have already engaged in using ML models for personality assessment using images, videos, audio, or text as input data. However, no proposals have been made to reduce the complexity of tests using ML. Hence, the conducted research aims to give an answer to the following research question:

RQ4.1. Is it possible to use ML-based models to further reduce Saucier's Mini-Marker to a "game of words" where the subject, instead of rating forty adjectives, only has to select those he relates the most?

The next lines describe the process that was carried out to answer this question and propose the ASAP method, which replaces the entire process of rating adjectives by an adjective selection process.

4.2.1 State of the Art

The use of ML to predict personality traits has been gaining popularity within the field of Affective Computing, with several studies engaging in conceiving ML models for personality assessment [54, 55, 217, 219]. In 2017, Majumder et al. conceived and evaluated DL models to assess personality from text. They conceived and fit a total of five ANNss, one for each personality trait. All networks had the same architecture, with each ANNs behaving as a binary classifier to predict whether the trait was positive or negative [54]. As dataset, the authors used James Pennebaker and Laura King's stream-of-consciousness essay dataset, which contains 2468 anonymous essays tagged with the binary value for each of the Big Five [220]. This dataset seems to be currently unavailable.

In 2017, Yu and Markov conceived and evaluated multiple DL models to learn suitable data representation for personality assessment, using Facebook status update data. This dataset consisted of raw text, user's information, and standard Big Five labels, which were obtained using self-assessment questionnaires [55]. In fact, it is possible to find several studies focused on inferring personality based on social media feeds. For instance, Kosinski et al. (2014) focused on examining how an individual's personality manifests in his/her online behavior [217]. The obtained results showed psychologically meaningful links between individuals' personalities, website preferences, and social media data. The potential applications of these works are essentially related to targeted advertising and personalized recommender systems, which take into consideration one's personality to deliver useful content.

In 2012, Sumner et al., based on Twitter use, focused on identifying signals of the Dark Triad, i.e., the anti-social traits of *Narcissism*, *Machiavellianism*, and *Psychopathy*. Almost three thousand Twitter users, from 89 countries, participated in the study, with an in-built Twitter application being developed to collect self-reported ratings on the Short Dark Triad questionnaire, which measures the anti-social traits, and the Ten Item Personality Inventory test. The authors conclude that even though possible to examine large groups of people, the conceived ML models behave poorly when applied to individuals, being imprecise when predicting Dark Triad traits just from Twitter activity [219].

Another study, performed by Cerasa et al. (2018), focused on conceiving and evaluating ML models to identify individuals with gambling disorder. To build the dataset, a set of healthy and sick individuals were asked to perform the NEO-personality-inventory test. The authors employed Classification and Regression Trees achieving interesting performances evaluated using the Area Under the Curve (AUC). In fact, the best candidate model was able to identify individuals with gambling disorder with an AUC of approximately 77% [221].

On the other hand, studies have been performed where audio and video data are used by DL-based models to predict personality [53]. One study, performed by Levitan et al. (2016), focused on the automatic identification of traits such as gender, deception, and personality using acoustic-prosodic and lexical features [222]. In particular, the authors focused on automatic detection of deception. The authors used Columbia deception corpus, which consists of deceptive and non-deceptive speech from standard American-English and Mandarin-Chinese native speakers, including more than one hundred hours of speech with self-identified truth/lie labels [223]. The authors then collected demographic data from each subject and administered a NEO-FFI personality test to access the Big Five. Each trait was binned as a three-class classification problem (*low*, *medium*, and *high*), which created an highly unbalanced dataset since the majority of subjects fell into the *medium* class. Several ML models and feature sets have been experimented, with AdaBoost and Random Forests being the best performing classifiers for personality assessment [222].

Another study, performed by Gurpinar et al. (2016), focused on using DL to predict the Big Five of faces appearing in videos [224]. The authors employed transfer-learning and CNNs to extract facial expressions, as well as ambient information. The conceived models were evaluated on the *ChaLearn Challenge Dataset on First Impression Recognition*, which consists of ten thousand clips collected from more than five thousand YouTube videos. The label of each clip corresponds to the Big Five personality traits of the person appearing in that clip. Their best candidate model achieved an accuracy of over 90% on the test set [224].

It is also usual to find the use of different data sources combined through means of data fusion for personality assessment. Indeed, personality assessment from multi-modal data has been assuming a greater importance in the computer vision field [225]. For instance, Gucluturk et al. (2017), aimed to analyze what features are used by personality trait assessment models when making predictions, conducting several experiments that characterized audio and visual information that drive such predictions [225]. On the other hand, Zhang et al. (2016) proposed a Deep Bi-modal Regression framework to capture rich information from both the visual and audio aspects of videos, winning the *ChaLearn Looking at People* challenge. CNNs were conceived to exploit visual cues, while linear regressors were used for audio [58, 226].

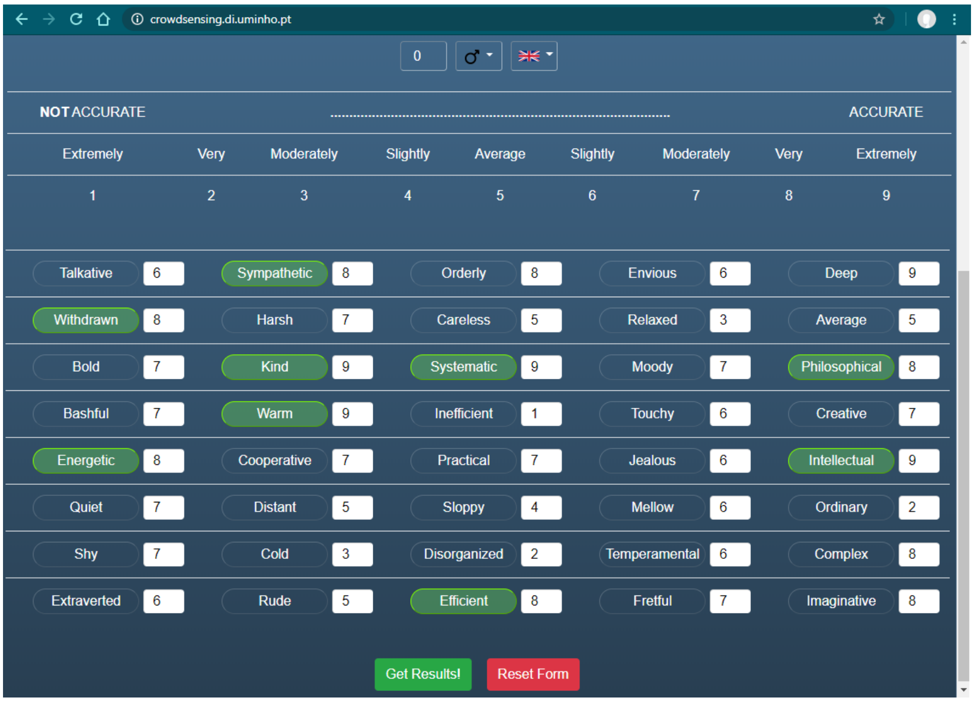
4.2.2 Data Collection and Exploration

Due to the non-availability of data and the particularities of the proposed method, we were required to develop a web platform for data collection, requesting subjects to rate adjectives and select those describing them the most. This allowed us to build a dataset containing self-reported ratings on Saucier's Mini-Marker test, the corresponding values of the Big Five as well as the adjectives selected by the subjects. The next lines describe in detail the developed platform, exploring and explaining the collected dataset and the implemented data augmentation techniques. It also details the conceived ML architectures and the experimental setup.

4.2.2.1 Web Platform for Data Collection

To bring this study to a fruitful conclusion, we were required to collect a dataset from where we could derive knowledge. Hence, a platform was conceived and made available online[1]. The platform displays all 40 adjectives used by Saucier's Mini-Marker test, asking the subject to rate each one. It also allows the subject to select a set of adjectives that describe him the most. Figure 4.1 depicts the main page of the

[1] http://crowdsensing.di.uminho.pt/



NOT ACCURATE				....				ACCURATE
Extremely	Very	Moderately	Slightly	Average	Slightly	Moderately	Very	Extremely
1	2	3	4	5	6	7	8	9

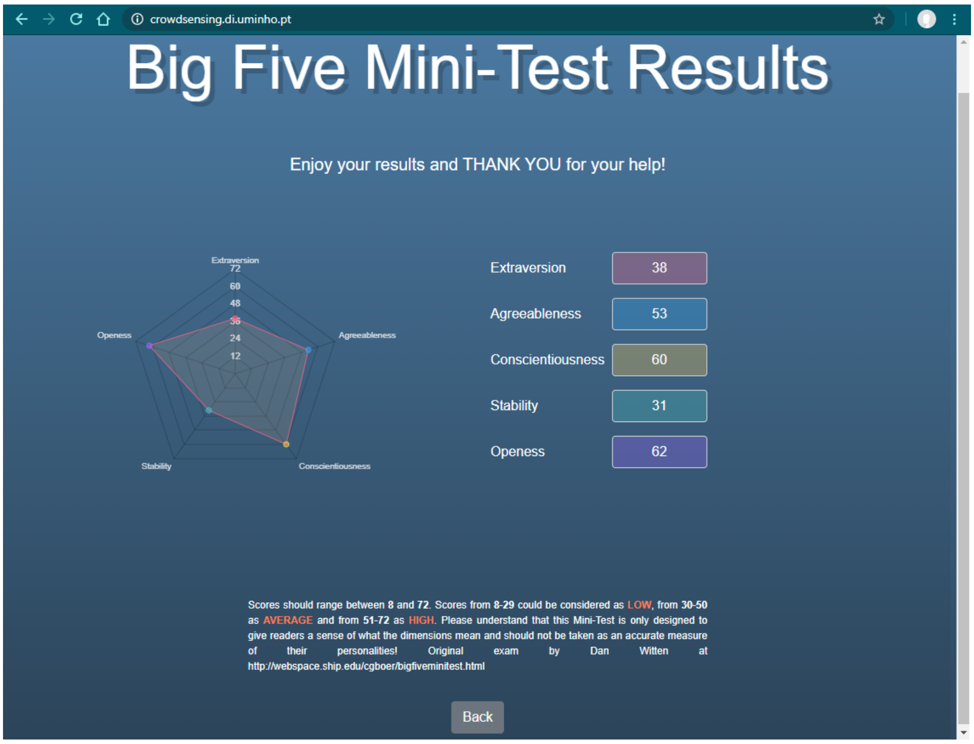
Talkative	6	[✓ Sympathetic]	8	Orderly	8	Envious	6	Deep	9
[✓ Withdrawn]	8	Harsh	7	Careless	5	Relaxed	3	Average	5
Bold	7	[✓ Kind]	9	[✓ Systematic]	9	Moody	7	[✓ Philosophical]	8
Bashful	7	[✓ Warm]	9	Inefficient	1	Touchy	6	Creative	7
[✓ Energetic]	8	Cooperative	7	Practical	7	Jealous	6	[✓ Intellectual]	9
Quiet	7	Distant	5	Sloppy	4	Mellow	6	Ordinary	2
Shy	7	Cold	3	Disorganized	2	Temperamental	6	Complex	8
Extraverted	6	Rude	5	[✓ Efficient]	8	Fretful	7	Imaginative	8

Figure 4.1: Web platform for data collection allowing the subject to perform Saucier's Mini-Marker test and, at the same time, select a set of adjectives that describe him the most.

conceived platform. The subject can then get the test results and obtain the value of each personality trait (Figure 4.2). The platform provides a rationale to explain the subject how he/she is contributing to the study. No personal data are stored neither it is possible to link subjects to their answers - only information about age, genre, and language are stored, and only if the user explicitly provides it. The platform is available online and any person can access and use it. It was published online on 21st September 2018, and was deployed in an Ubuntu 18.04 machine hosted at the University of Minho. The platform was shared among a diversified population, using social media and the university's mailing lists. Data were also collected in person, which allowed us to increment the dataset size with records containing both the ratings and the selected list of adjectives [58].

To facilitate the data collection process, the developed platform allows subjects to perform Saucier's Mini-Markers in three distinct languages. All translations were performed by three Portuguese and Spanish native speakers fluent in English. It should also be highlighted that this study does not aim to examine the psychometric properties of the Portuguese or Spanish versions neither to provide sound validity evidence for the performed translations (even though *Tau-Equivalent* estimates of score reliability are later examined). The assumption is that ML models are able to quantify, or qualify, the personality traits without requiring any contextual information about the region, genre, language, or age of the subjects.

The web platform was conceived using *Node.js*, an asynchronous and open-source event-driven JavaScript runtime environment that executes JavaScript outside the browser, allowing one to write server-side scripts. A web application framework, *Express.js*, was also used. Several external packages were used, including *i18n*, for language translation using dynamic JSON, *cookie-parser*, to parse cookies attached to the client request object, and *mysql*, for loading the appropriate *Node.js* database driver.



Big Five Mini-Test Results

Enjoy your results and THANK YOU for your help!

Extraversion	38
Agreeableness	53
Conscientiousness	60
Stability	31
Openess	62

Scores should range between **8** and **72**. Scores from **8-29** could be considered as LOW, from **30-50** as AVERAGE and from **51-72** as HIGH. Please understand that this Mini-Test is only designed to give readers a sense of what the dimensions mean and should not be taken as an accurate measure of their personalities! Original exam by Dan Witten at http://webspace.ship.edu/cgboer/bigfiveminitest.html

Figure 4.2: Results of Saucier's Mini-Marker test provided by the web platform.

A set of routes were created to regulate how the platform answers to client requests for particular endpoints, which includes an Uniform Resource Identifier (URI) and a specific HTTP request method. The routes specify a handler function that is called when the application receives a request to the specified route. Listing 4.1 depicts how the conceived platform routes the received requests.

Listing 4.1: Routing of the web platform for personality assessment and data collection.

```
var express = require('express');

var mainRouter = require('./routes/main');
var resultsRouter = require('./routes/results');
var translationRouter = require('./routes/translation')

var app = express();

//for any HTTP request method
app.use('/', mainRouter);
app.use('/main', mainRouter);
app.use('/results', resultsRouter);
app.use('/translation', translationRouter);

//catch 404 and forward to error handler
app.use(function(req, res, next) {
  next(createError(404));
});
```

Listing 4.2 presents the *resultsRouter*, the one responsible for rendering the subject's personality traits in the web browser. For the sake of simplicity, only this router is here presented. The remaining routers are similar in terms of structure.

Listing 4.2: The *resultsRouter* is the one responsible for rendering the subject's personality traits.

```
var express = require('express');
var router = express.Router();

router.get('/', function(req, res, next) {
  if(!req.session.results {
   return res.redirect('/main');
  }
  //get data from the session
  var results = req.session.results;
  var extraVal = results.ExtraVal;
  var agreeVal = results.AgreeVal;
  var consVal = results.ConsVal;
  var stableVal = results.StableVal;
  var openVal = results.OpenVal;
  //render view
  res.render('results', { title: 'Big Five Mini-Test',
                   extraversion: extraVal,
                   agreeableness: agreeVal,
                   conscientiousness: consVal,
                   stability: stableVal,
                   openess: openVal }
            );
});
module.exports = router;
```

The platform consists of three view files developed using EJS, a templating language to generate HTML markup with JavaScript. The *main.ejs* file is used to dynamically build the main page of the platform (Figure 4.1), the *results.ejs* is used to build the results page (Figure 4.2) and, finally, the *error.ejs* is used to display errors. A set of scripts were conceived and are used to compute the traits' value based on the answers of the subject and to optimize the applications' GUI. The subject's answer to the personality test is stored in a MySQL database (version 14.14 and distribution 5.7.29) using an InnoDB engine (version 5.7.29), remotely deployed in the same machine as the web platform.

4.2.2.2 Data Exploration

The collected dataset contains 255 observations. Each observation is made of 50 features, viz, *age*, *genre*, *language*, 40 *adjectives*, 5 *personality traits*, the *selected adjectives*, and the *creation date*. The *age*, the 40 *adjectives*, and the 5 *personality traits* are integers. The *genre* is a binary attribute and *language* is either *es*, *en*, or *pt*. On the other hand, the *selected adjectives* feature (also

Table 4.1: Features available in the personality dataset.

#	Feature	#	Feature	#	Feature
1	*age*	18	*philosophical*	35	*cold*
2	*genre*	19	*bashful*	36	*disorganized*
3	*language*	20	*warm*	37	*temperamental*
4	*talkative*	21	*inefficient*	38	*complex*
5	*sympathetic*	22	*touchy*	39	*extraverted*
6	*orderly*	23	*creative*	40	*rude*
7	*envious*	24	*energetic*	41	*efficient*
8	*deep*	25	*cooperative*	42	*fretful*
9	*withdrawn*	26	*practical*	43	*imaginative*
10	*harsh*	27	*jealous*	44	*extraversion_trait*
11	*careless*	28	*intellectual*	45	*agreeableness_trait*
12	*relaxed*	29	*quiet*	46	*conscientiousness_trait*
13	*average*	30	*distant*	47	*stability_trait*
14	*bold*	31	*sloppy*	48	*openess_trait*
15	*kind*	32	*mellow*	49	*selected_attr*
16	*systematic*	33	*ordinary*	50	*creation_date*
17	*moody*	34	*shy*		

described as *selected_attr*) consists of a string where the selected adjectives are comma separated. Table 4.1 presents all available features in the personality dataset.

In the final dataset, 159 observations have the *selected_attr* feature filled with the selected adjectives. On the other hand, 96 observations only have the adjectives' ratings. A few observations have adjectives rated with the value 0. 200 observations belong to male subjects, while 55 belong to female ones. Only two languages were used: 220 observations were done in Portuguese while 35 were done in English. More than 90% of the observations were collected in 2019. The mean age value of subjects is of 30.1 years.

Adjectives with lower mean value are essentially related to negative ones such as *rude*, with 3.13, *inefficient*, with 3.26, and *ordinary*, with 3.28. The adjectives that have higher mean value are *kind*, with 6.004, *imaginative*, with 6, and *cooperative*, with 5.73. Mean standard deviation of the 40 adjectives is 2.5, with the lower value being 0 and the maximum 9. Mean skewness is of 0.03, representing a symmetrical distribution. Mean kurtosis is of −0.98, representing a somewhat "light-tailed" dataset in regard to the 40 adjectives. In regard to the Big Five (Table 4.2), the one having lower mean value is *Extraversion*, with *Agreeableness* being the one with higher mean value. Mean standard deviation of all traits is of approximately 10 units of measure. The coefficient alpha for the forty items is of 0.82 [227]. For each individual trait, the *Tau-Equivalent* estimates of score reliability are lower, specially for the *Stability* factor. Except for the *selected_attr* feature, no missing values are present in the dataset.

With all features assuming a non-Gaussian distribution (under the Kolmogorov-Smirnov test with $p < 0.05$), the non-parametric Spearman's rank correlation coefficient was used. A few pairs of correlated features, in the form (*trait, adjective*), appear in the dataset. This is in line with expectations since the Big Five are mathematically based on the adjectives. Higher correlations appear for the pairs (*Agreeableness,*

Table 4.2: Descriptive statistics for the Big Five.

	Openness	**Conscientiousness**	**Extraversion**	**Agreeableness**	**Stability**
N° of Items	8	8	8	8	8
Mean	44.976	46.476	39.428	47.148	46.140
Median	46	47	40	48	46
Std. Deviation	10.315	10.547	10.017	10.056	8.860
Skewness	−0.212	−0.207	−0.135	−0.216	−0.047
Kurtosis	−0.260	−0.492	−0.344	−0.328	−0.484
Coef. Alpha	0.62	0.61	0.56	0.58	0.42

Warm), (*Conscientiousness, Efficient*), (*Openess, Complex*) and (*Extraversion, Extraverted*).

The *selected_attr* feature consists of a string where adjectives are separated by commas. An example of a valid value would be "*Talkative, Sympathetic, Kind, Energetic, Jealous, Intellectual, Extraverted, Efficient, Fretful*". From all 159 observations that have the *selected_attr* feature filled, 157 are unique values meaning that only three subjects chose the same adjectives. Interestingly, all adjectives were selected at least once. In fact, the least selected adjectives were *ordinary*, which was selected 14 times, *touchy*, 18 times, *rude*, 19 times, and *cold* and *fretful*, 23 times. These are, essentially, adjectives with negative connotation. On the opposite spectrum, *kind* was selected 67 times, *imaginative*, 59 times, *sympathetic*, 58 times, *creative*, 57 times, and *withdrawn*, 56 times (Figure 4.3). Excluding those who opt not to select adjectives, 10 subjects only chose one adjective to describe themselves, while 14 subjects selected fifteen, or more, adjectives. The mean value is of approximately ten selected adjectives per subject.

Approximately 38% of the total number of observations do not have adjectives selected. To overcome this issue, it becomes important to understand the relation between selecting an adjective and its respective rating. For instance, considering all subjects that selected the adjective *efficient*, the mean rating of that same adjective is of 5.794. On the other hand, the mean rating of the *sloopy* adjective considering all

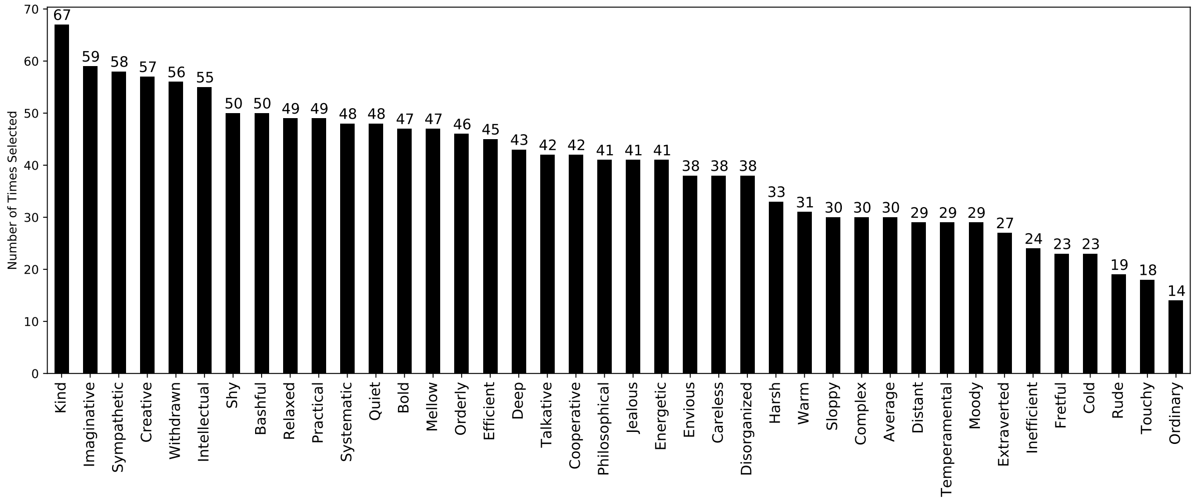


Figure 4.3: Number of times each adjective was selected.

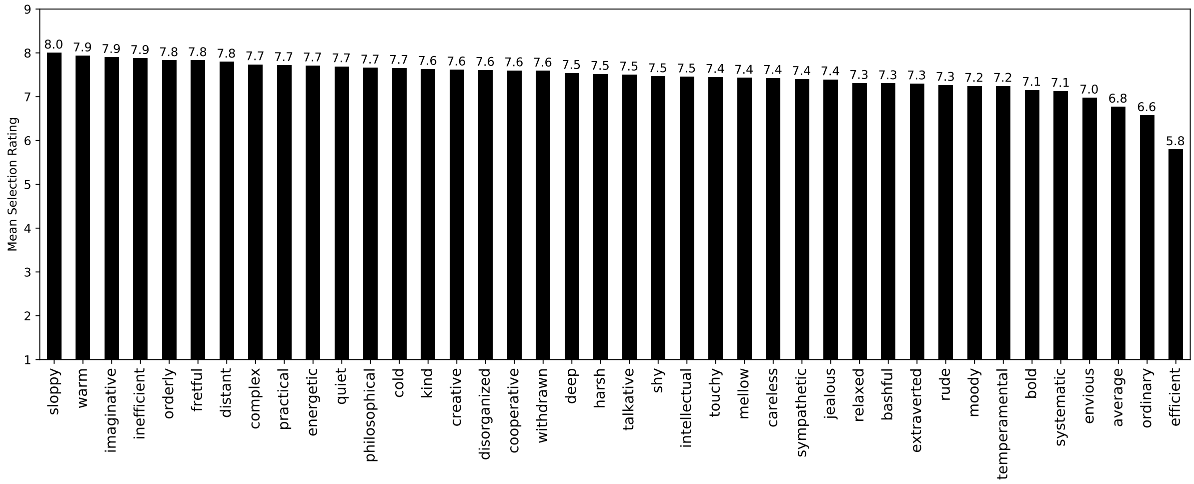


Figure 4.4: Mean rating values to set an adjective as selected.

subjects that selected that adjective is of 8. This tells us that *sloppy* tends to be selected when receiving higher ratings. On the other hand, *efficient* is selected even with average ratings. The overall mean, 7.448, tells us that, as expected, adjectives tend to be selected when receiving high values. Figure 4.4 depicts the mean rating values to set an adjective as selected.

To discover relations between the selected adjectives, a ML and a pattern mining method, entitled as Association Rules Learning (ARL), was applied. ARL does not consider the order of the items, neither it extracts an individual's preference. Instead, it looks for frequent itemsets. The goal is to find associations and correlations between adjectives that were selected to describe subjects. In particular, the APRIORI algorithm was used to analyze the list of selected adjectives, and provide rules in the form *Antecedent* -> *Consequent*, where -> may be read as "implies". To find these rules, three distinct metrics were used: *Support*, which gives an idea of how frequent an itemset is in all existing transactions, helping identifying rules worth considering; *Confidence*, an indication of how often a rule is true; and *Lift*, which measures how much better the rule is at predicting the presence of an adjective compared to just relying on the raw probability of the adjective in the dataset. The returned rules go both ways, i.e., if *A* implies *B* then the reverse is also true. Table 4.3 presents all rules with a support value higher than 0.15. In fact, the support value was tuned to find a representative set of rules. Such a lower support value tells us that rules tend to be less frequent than expected. On the other hand, the obtained confidence values strengthen the possibility of both the antecedent and the consequent being found together for a subject. Lift values higher than 1 tells us that the adjectives are positively correlated [58].

4.2.3 Data Preparation and Pre-processing

A random seed, as 91190530, was defined for replicability purposes. Five observations that had abnormal values were removed. In particular, one observation had abnormally high values, while the other four were filled with the same exact dummy value for all adjectives. None of these observations had the *selected_attr* feature filled. The final dataset is made of 250 observations.

Table 4.3: Rules with support higher than 0.15 using Association Rules Learning and the APRIORI algorithm.

Support	Confidence	Lift	Antecedent		Consequent
0.195	0.525	1.466	Creative	–>	Imaginative
0.189	0.508	1.207	Kind	–>	Imaginative
0.176	0.418	1.146	Sympathetic	–>	Kind
0.170	0.551	1.511	Sympathetic	–>	Relaxed
0.170	0.491	1.394	Withdrawn	–>	Intellectual
0.170	0.491	1.165	Kind	–>	Intellectual
0.170	0.540	1.281	Kind	–>	Shy
0.164	0.464	1.273	Sympathetic	–>	Withdrawn
0.164	0.634	1.505	Kind	–>	Jealous
0.157	0.521	1.428	Sympathetic	–>	Systematic
0.157	0.500	1.371	Sympathetic	–>	Bashful
0.157	0.500	1.187	Kind	–>	Bashful
0.151	0.510	1.400	Sympathetic	–>	Bold
0.151	0.358	1.017	Withdrawn	–>	Kind
0.151	0.585	1.389	Kind	–>	Energetic

4.2.3.1 Handling zero-ratings

The lowest accepted value by Saucier's test is one, however, zeros are present in the dataset. To correct this situation and to make all observations mathematically valid, such values were updated to the nearest valid value, with traits' values being re-calculated based on such changes.

4.2.3.2 One-hot encoding the selected_attr feature

The $selected\_attr$ feature consists of a string with comma-separated adjectives. Such data were one-hot encoded using a Multi-Label Binarizer, allowing these data to become easier to handle by ML models. Forty new features were created, being entitled as $\{adjective\}\_selected$, with $\{adjective\}$ being a placeholder for the corresponding adjective name. A value of 0 means that the adjective was not selected, with a value of 1 meaning selection.

4.2.3.3 Filling the selected_attr feature when empty

Approximately 38% of all observations do not have adjectives selected, i.e., the $selected\_attr$ feature is empty because the subject did not choose any adjective. However, to be able to propose the ASAP method, we are required to have as many observations as possible with the $selected\_attr$ feature filled. Hence, a method was conceived to synthetically mark adjectives as selected based on adjectives' ratings and frequent patterns of selected adjectives.

The first step consists in iterating through the observations without selected adjectives. Then, for each observation, iterate through each adjective. If the adjective's rating is higher than the mean selection rating of that same adjective (as depicted in Figure 4.4), then the adjective is a candidate to be $selected$.

Being a candidate means that the adjective may, or may not, be selected. To reduce bias, this decision is randomized, with the adjective having a three-quarters chance of being selected. If the adjective is to be selected, then the corresponding *{adjective}_selected* one-hot feature is selected (marked with 1). The next step is to see if the selected adjective is part of any rule (as depicted in Table 4.3). If it is, then the consequent will have half a chance of being selected as well. The upper limit is of fourteen selected adjectives per observation, with the lower limit being one selected adjective. To respect this last condition, for each observation, it is stored a list of all adjectives that are above the selection threshold. If no adjective was previously selected, then a random adjective from the referred list is selected. Algorithm 4.1 describes, using pseudo-code, the implemented method.

Algorithm 4.1 Filling the *selected_attr* feature.

```
Input: dataset, limit = 14
adj_thresholds = getAdjectivesThresholds(dataset)
for row ∈ dataset do
    if row.selected_attr == 'na' then
        initialize enabled_adjectives = 0
        initialize obs_without_selection = {}
        for adjective ∈ row.adjective do
            if adjective.value >= adj_thresholds[adjective] then
                if enabled_adjectives < limit then
                    if random.choice(4) < 3 then
                        enabled_adjectives += 1
                        dataset[row][{adjective}_selected] = 1
                        list_of_consequents = getConsequents(adjective)
                        for consequent_adjective ∈ list_of_consequents do
                            if random.choice(2) < 1 then
                                enabled_adjectives += 1
                                dataset[row][{consequent_adjective}_selected] = 1
                                if enabled_adjectives == limit then
                                    break
                                end if
                            end if
                        end for
                    else
                        obs_without_selection[adjective] = adjective.value
                    end if
                end if
            end if
        end for
        if enabled_adjectives == 0 then
            random_adjective = random.choice(obs_without_selection.keys())
            dataset[row][{random_adjective}_selected] = 1
        end if
    end if
end for
```

The method described in Algorithm 4.1 enabled all observations to have adjectives selected. Considering only the affected observations, the mean value is of 7.8 selected adjectives per observation, with a minimum of 1 and a maximum of 14 selected adjectives. Several randomized decisions are made based on a probabilistic approach in order to reduce any possible bias [58].

4.2.3.4 Data Augmentation

Since the small size of the dataset may pose a problem to ML models, we aimed to investigate how models would behave in non-data scarce environments. Hence, Data Augmentation (DA) techniques were conceived to increase the dataset's size. It is worth highlighting that there is no standardized DA process that can be applied to every domain. Instead, DA refers to a process that is highly dependent on the domain where it is to be implemented. The goal is to increase the dataset size while maintaining relations and data specificities, using randomness to reduce bias.

With the use of DA techniques, a second dataset was conceived. Hence, two distinct input datasets will be fed to the candidate ML models. On the one hand, models are to be trained and evaluated with the original dataset, without any DA (*No DA*). On the other, candidate models will also be trained and evaluated using an augmented dataset (*With DA*). In the augmented dataset, new observations were generated from every single observation. The number of new observations that can be generated from one observation varies according to a random variable that outputs, with the same probability, a number between 15 and 25. For every new observation, another random variable will decide how many and which adjectives to vary from the original observation. A minimum of 5 and a maximum of 20 adjectives must vary. Each of these adjectives can stay the same or go up/down one or two units, always respecting the test limits of 1 and 9. Then, the Big Five are calculated for the new observations. Finally, the last step consists in selecting and deselecting adjectives. In particular, in finding out if the adjective that varied is a candidate to be selected or deselected, similarly to what was done to fill the $selected\_attr$ feature when empty. If the adjective that had its value updated is a candidate to be selected and if it was indeed chosen to be selected (three-quarters chance), then if it is an antecedent of any rule, the consequent would also have half a chance of being selected as well. Finally, a final random variable, varying from 5 to 14, defines how many selected adjectives the new observation can hold. If such a limit is exceeded, then, randomly, selected adjectives are deselected until the upper limit is respected.

Data augmentation processes may add an intrinsic bias to ML models. Hence, to reduce bias to its minimum, several randomized decisions were made based on a probabilistic approach to create a more generalized version of the dataset.

4.2.3.5 Binning

In Saucier's original study [218], trait scores were divided into three bins. Trait scores between $[8, 29]$ were considered to be low, between $[30, 50]$ were considered to be $average$, and between $[51, 72]$ were considered to be *high*. This assumes an increased importance since one of the conceived ML architectures, as explained later, uses classification models, where labels (the personality traits) are required to be binned. Hence, considering the original split and the need to create trait bins, labels were binned using the three bins defined originally. As depicted in Figure 4.5, after binning the dataset using the original intervals, bins get imbalanced, with all five traits having a higher number of observations falling within the range $[30, 50]$. In fact, for all traits, around 60% of observations fall within the *average* bin. Regarding the other two bins, *high* contains significantly more observations than *low* for all traits except for the *Extroversion* trait, which contains approximately the same amount of observations in the *high* and *low* bins. This distribution of observations must be taken into consideration when conceiving and training the ML models. In fact, this distribution will lead to the use of error metrics that take into account the presence of imbalanced sets of data.

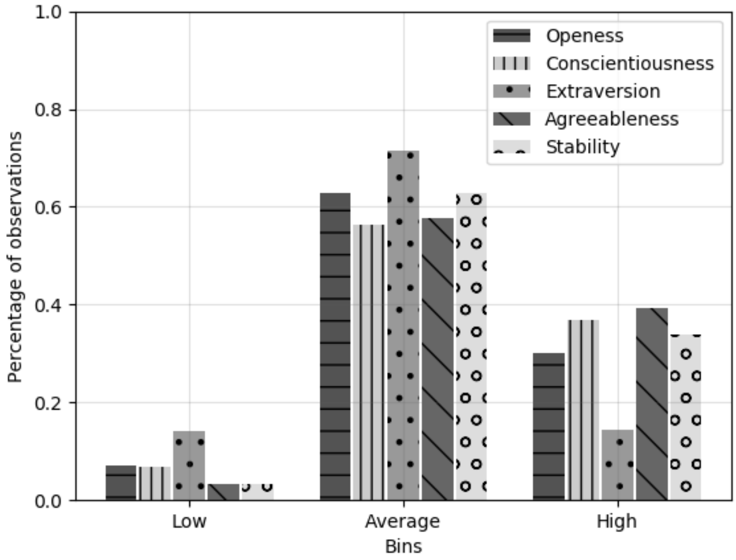


Figure 4.5: Distribution of observations per bin and personality trait.

4.2.3.6 Final Considerations

Two datasets were created. The *age*, *language*, and *genre* features were removed from both datasets as well as the rating of the 40 adjectives since those will not be used by the models (models will only use the binary selection of adjectives). Dataset with *No DA* consists of 250 observations, while the dataset *With DA* consists of 5230 observations. Both datasets contain 50 features that correspond to the 40 one-hot encoded adjectives, the 5 personality traits' scores, and the 5 binned personality traits.

4.2.4 Experiments

Based on the collected dataset, its characteristics, and the essence of the ASAP method, two different ML architectures were conceived and evaluated. The first architecture consists of five supervised trait regressors while the second one consists of five supervised trait classifiers. The goal is to obtain the Big Five scores based on the selection of adjectives.

Both architectures use gradient boosting, in particular, Gradient Boosted Trees to tackle this supervised learning problem. The "gradient boosting" term was first used by J. Friedman [228], being used as a ML technique to convert weak learners, typically Decision Trees, into strong ones, allowing the optimization of a differentiable loss function, with the gradient representing the slope of the tangent to the loss function. Gradient boosting trains weak learners in a gradual, additive, and sequential manner. A gradient descent procedure is performed so that trees are added to the gradient boosting model in order to reduce the model's loss. Being this a greedy algorithm, it can overfit. Hence, to control overfitting, it is common to use regularization parameters, limit the number of trees of the model, and tree's depth and size. Another benefit of using Gradient Boosted Trees is the ability to compute estimates of feature importance [58].

4.2.4.1 Architecture I - Big Five Regressors

The first proposed architecture uses a total of five different Gradient Boosted Trees regression models to obtain the score of the Big Five, with each model mapping a specific trait (Figure 4.6).

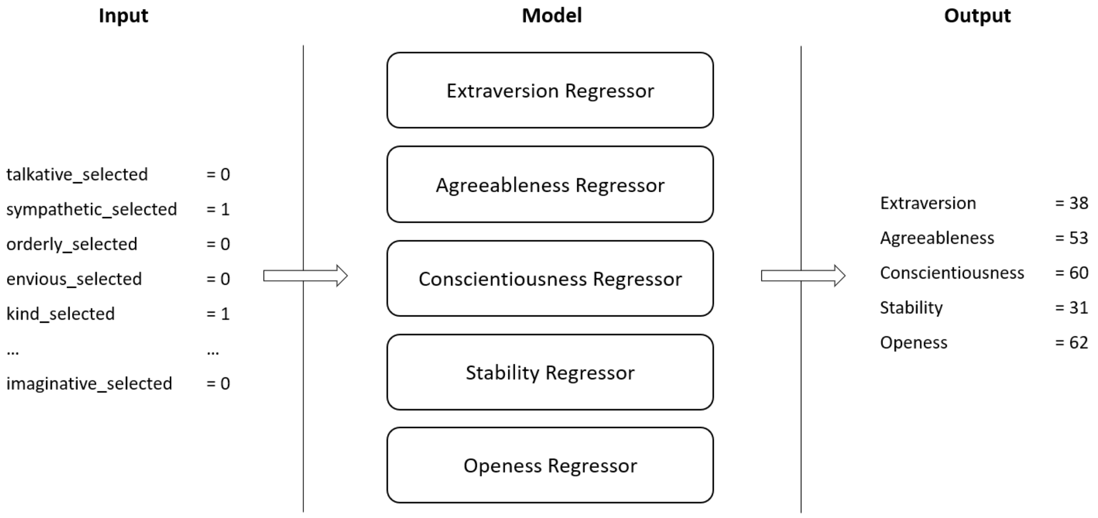


Figure 4.6: Architecture I - Big Five regressors.

As input, each model receives the one-hot encoded adjectives' selection (whether the adjective was selected or not). The main characteristics of this architecture may be summarized as follows:

- *Input*: the one-hot encoded adjectives' selection;
- *Output*: the score of each personality trait;
- *Evaluation*: two independent trials using nested cross-validation with Mean Squared Error (MSE) as objective function, and RMSE and MAE as evaluation metrics;
- *Models*: personality traits are computed independently of others, i.e., five independent regression models are trained, one for each trait.

4.2.4.2 Architecture II - Big Five Bin Classifiers

The second proposed architecture uses a total of five different Gradient Boosted Trees classification models to obtain the binned score of the Big Five, with each model mapping a specific trait (Figure 4.7). As input, each model receives the one-hot encoded adjectives' selection (whether the adjective was selected or not). The main characteristics of this architecture may be summarized as follows:

- *Input*: the one-hot encoded adjectives' selection;
- *Output*: the bin (low/average/high) of each personality trait;
- *Evaluation*: two independent trials using nested cross-validation for multi-output multi-class classification with softmax as objective function and accuracy, f1-score, and mean error as metrics;
- *Models*: personality traits are computed independently of others, i.e., five independent classification models are trained, one for each trait.

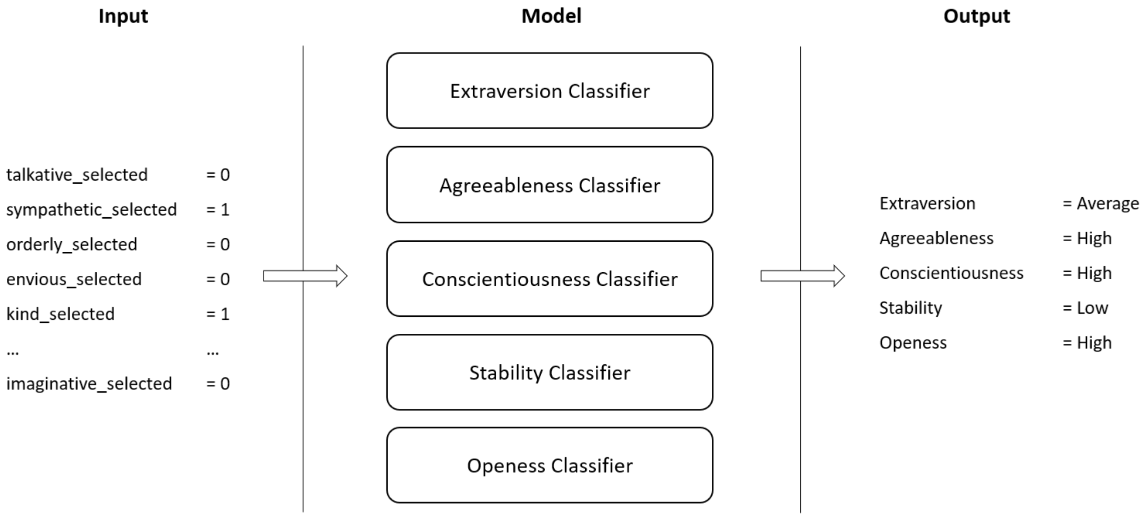


Figure 4.7: Architecture II - Big Five bin classifiers.

4.2.4.3 Models' Evaluation

All the conceived models follow a supervised learning approach, i.e., models are trained on a sub-set of data and are then evaluated on a distinct sub-set. In fact, we went further and implemented nested cross-validation to estimate the skill of the candidate models on unseen data as well as for hyperparameter tuning. Hyperparameter selection is performed in the inner loop, while the outer one computes an unbiased estimate of the candidate's accuracy. Nested cross-validation assumes an increased importance since, otherwise, the same data would be used to tune the hyperparameters and to evaluate the model's accuracy [229]. Inner cross-validation was performed with $k = 4$ and outer cross-validation used $k = 3$. Two independent trials were performed.

To evaluate the effectiveness of Architecture I, two error metrics were used. Both take as input the model's predicted value and the actual value from Saucier's test, computing a metric of how far the model is from the real known value. The first one, RMSE, allows us to penalize outliers and easily interpret the obtained results since they are in the same unit of the feature that is being predicted by the model. The second error metric, MAE, was used to complement and strengthen the confidence in the obtained values.

Since Architecture II consists of several classification models, confusing matrix-based metrics were used to evaluate the classifier's output quality, in particular the f1-score, where the relative contribution of precision and recall are equal, and the mean error, which penalizes wrongly classified observations. Being this a multi-class problem and considering that bins are imbalanced, both micro and macro-averaged f1-scores are used. Macro-average computes the error metric independently for each class and averages the errors, treating all classes equally. On the other hand, micro-average aggregates all classes' contributions to compute the final error metric.

4.2.4.4 Experimental Setup

Python, version 3.7, was the used programming language for data exploration and pre-processing as well as for model development and evaluation. *Pandas*, *NumPy*, *scikit-learn*, *XGBoost*, *matplotlib*, and *seaborn* were the used libraries. The Knime platform was also used for data exploration. All hardware was made available by Google's Colaboratory.

XGBoost was the library used to conceive the Gradient Boosted Trees. It is a distributed gradient boosting library that is efficient and flexible. Contrary to other boosted trees based libraries, XGBoost implements regularization and parallel processing, having already been used in several studies [230–232]. Algorithm 4.2 describes, using pseudo-code, the method used to conceive the boosted regressors and classifiers, depending on the inputted architecture.

Algorithm 4.2 Building the Gradient Boosted models.

```
Input: architecture
if architecture == 1 then
    estimator = XGBRegressor(booster = 'gbtree', objective = 'reg:squarederror')
    multi_estimator = MultiOutputRegressor(estimator)
else
    estimator = XGBClassifier(booster = 'gbtree', objective = 'multi:softmax', num_class = 3)
    multi_estimator = MultiOutputClassifier(estimator)
end if
Return: multi_estimator
```

4.2.4.5 Hyperparameter Search Space

Models were tuned in regard to a set of hyperparameters using Random Search limited to 175 combinations (out of 486) for efficiency purposes. Architecture I uses MSE as objective function while Architecture II uses softmax. Table 4.4 describes the searching space for each hyperparameter.

Table 4.4: Hyperparameters' searching space.

Parameter	**Searched Values**	**Rationale**
a. Number of Estimators	[300, 400, 500]	Number of trees in a model
b. Eta	[0.01, 0.05, 0.1]	Learning rate
c. Gamma	[0.02, 0.04, 0.08]	Minimum loss reduction required to make a further partition on a leaf node
d. Trees' Max Depth	[4, 12, 18]	Maximum depth of a tree
e. Minimum Child Weight	[4, 6, 8]	Minimum sum of instance weight needed in a child (higher values for conservative models)
f. Colsample by tree	[0.2, 0.3]	Fraction of columns to be sub-sampled (controlling correlation between trees)

4.2.5 Results and Discussion

Two distinct ML architectures were experimented. One uses Gradient Boosted Trees regressors to obtain the exact value of each personality trait (Architecture I) while the other uses Gradient Boosted Trees classifiers to obtain the bin of each personality trait (Architecture II). Different experiments were conducted with two distinct datasets. One with 250 observations (*No DA*) and another with 5230 observations (*With DA*). Both architectures receive, as input, the one-hot encoded selection of adjectives.

Nested cross-validation was performed to tune the hyperparameters and to have a stronger validation of the obtained results. Inner cross-validation was performed using $k = 4$, with random search being used to find the best set of hyperparameters. In the inner loop, 700 fits were performed (4 folds × 175 combinations). The outer cross-validation loop used $k = 3$, totaling 2100 fits (3 folds × 700 fits). Two independent training trials were performed, with a grand total of 4200 fits (2 trials × 2100 fits) per architecture per dataset [58].

4.2.5.1 Architecture I - Big Five Regressors

All candidate models were evaluated in regard to RMSE and MAE error metrics. Table 4.5 depicts the best hyperparameter configuration for Architecture I, for both datasets. What immediately stands out is the better performance of the candidate models when using the larger dataset. In fact, RMSE decreases about 30% when using the dataset *With DA*. This was already expected since the dataset with *No DA* was made of only 250 observations.

Overall, for Architecture I with *No DA* the error is of approximately 8 units of measure. Since RMSE outputs an error in the same unit of the features that are being predicted by the model, it means that this Architecture is able to obtain the value of each personality trait with an error of 8 units. On the other hand, for Architecture I *With DA*, RMSE is of approximately 5.6 units of measure. It is also possible to discern that RMSE tends to be more stable when using the *With DA* dataset when compared to the *No DA*

Table 4.5: Architecture I results with and without data augmentation, for each independent trial, with RMSE as metric. Hyperparameters described by letters as follows: *a.* number of estimators, *b.* eta, *c.* gamma, *d.* trees' max depth, *e.* minimum child weight, and *f.* colsample by tree.

Trial	**CV Split**	**Best Score**	**Evaluation**	**Time (min)**	***a.***	***b.***	***c.***	***d.***	***e.***	***f.***
				No Data Augmentation						
1	1	7.813	8.078	3.8	300	0.05	0.04	4	6	0.2
1	2	8.015	7.560	3.8	300	0.05	0.02	4	4	0.2
1	3	8.203	7.512	3.7	300	0.01	0.04	4	4	0.2
2	1	8.024	7.594	3.7	300	0.10	0.08	4	8	0.2
2	2	8.184	7.161	3.7	300	0.05	0.02	4	8	0.2
2	3	7.847	7.961	3.7	300	0.05	0.04	4	8	0.2
				With Data Augmentation						
1	1	5.692	5.464	64.8	300	0.10	0.02	12	4	0.3
1	2	5.604	5.602	65.9	300	0.01	0.02	18	4	0.3
1	3	5.646	5.520	69.6	300	0.01	0.02	18	4	0.3
2	1	5.637	5.537	68.4	300	0.01	0.02	12	4	0.3
2	2	5.632	5.482	65.7	300	0.01	0.04	18	6	0.3
2	3	5.673	5.467	67.4	300	0.01	0.08	12	4	0.3

dataset, which shows higher error variance. In Table 4.5, the *Evaluation* column presents the error value of the best candidate model in the outer test fold. These values provide a second and stronger validation of the ability to classify of the best model.

The hyperparameter tuning process is significantly faster for Architecture I with *No DA*, taking around 3.7 minutes to perform 700 fits and around 22 minutes to perform the full run. On the other hand, Architecture I *With DA* takes more than 1 hour to perform the same amount of fits, requiring more than 6.5 hours to complete. Overall, the models that behaved the best used 300 gradient boosted trees. Interestingly, when using the dataset with *No DA*, all models required 20% of the entire feature set when constructing each tree (colsample by tree) and used a maximum depth of 4 levels, building shallower trees, which helps controlling overfitting in the smaller dataset. On the other hand, when using the dataset *With DA*, the best models not only required 30% of the feature set but also required deeper trees, which indicate the need for more complex trees to find relations in the larger dataset. To strengthen this assertion, the learning rate is also smaller in Architecture I *With DA* allowing models to move slower through the gradient.

Focusing the results obtained from testing in the test fold of the outer-split, Architecture I *With DA* presents a global RMSE of 5.512 and MAE of 3.979. On the other hand, Architecture I with *No DA* presents higher error values, with a global RMSE and MAE of 7.644 and 6.082, respectively. The fact that RMSE and MAE have relatively close values implies that not many outliers, or distant classifications, were provided by the models. It is also interesting to note that, independently of the dataset, *Openness* is the most difficult trait to classify. All these data are given by Table 4.6, where the MSE is also displayed, being used to compute the RMSE.

Table 4.6: Evaluation results of Architecture I, with and without data augmentation, obtained from the test folds of the outer-split.

Metric	Global	Extraversion	Agreeableness	Conscient.	Stability	Openness
No Data Augmentation						
MAE	**6.082**	5.495	5.942	5.616	6.205	7.153
MSE	**58.527**	45.973	55.984	49.230	58.933	82.514
RMSE	**7.644**	6.778	7.468	7.000	7.668	9.061
With Data Augmentation						
MAE	**3.979**	3.937	4.071	3.832	3.798	4.259
MSE	**30.385**	29.529	31.630	28.813	27.069	34.884
RMSE	**5.512**	5.433	5.623	5.367	5.201	5.906

Figure 4.8 provides a graphical view of RMSE and MAE for Architecture I for both datasets, being possible to discern that both metrics present a lower error value when conceiving models over the augmented dataset.

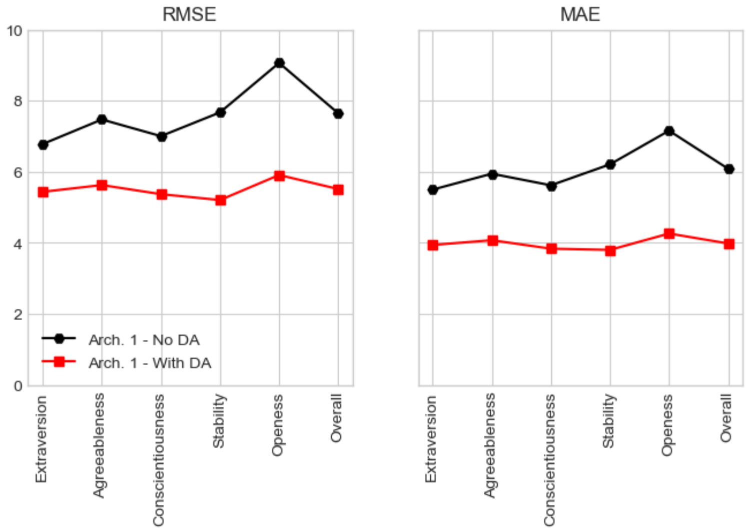


Figure 4.8: Graphical view of Architecture I RMSE and MAE for both datasets.

4.2.5.2 Architecture II - Big Five Bin Classifiers

Architecture II candidate models, which classify personality traits in three bins (low, average, and high), were evaluated using several classification metrics. Table 4.7 depicts the best hyperparameter configuration for Architecture II, for the two datasets, using accuracy as metric. Again, models conceived over the dataset *With DA* outperform those conceived over the dataset with *No DA*, more than doubling the accuracy value. In addition, their evaluation values also tend to be more stable and less prone to variations. However, one may argue that the accuracy values attained by the candidate models and presented in Table 4.7 are low. Hence, it is of the utmost importance to assert that such accuracy values correspond to samples that had all five traits correctly classified. I.e., if one trait of a sample was wrongly classified then that sample would be considered as badly-classified even if the remaining four traits were correctly classified. To provide a stronger validation metric, Table 4.8 provides metrics based on traits' accuracy instead of samples' accuracy, presenting significantly higher values.

Table 4.7: Architecture II results with and without data augmentation, for each independent trial, with sample accuracy as metric. Hyperparameters described by letters as follows: *a.* number of estimators, *b.* eta, *c.* gamma, *d.* trees' max depth, *e.* minimum child weight, and *f.* colsample by tree.

Trial	CV Split	Best Score	Evaluation	Time (min)	*a.*	*b.*	*c.*	*d.*	*e.*	*f.*
			No Data Augmentation							
1	1	0.144	0.131	8.4	500	0.05	0.02	4	6	0.3
1	2	0.180	0.181	8.6	400	0.01	0.08	4	4	0.2
1	3	0.138	0.108	8.4	300	0.01	0.04	12	8	0.3
2	1	0.175	0.143	8.5	300	0.05	0.04	4	4	0.2
2	2	0.138	0.181	8.4	500	0.10	0.02	18	4	0.3
2	3	0.155	0.133	8.4	400	0.05	0.08	18	8	0.3
			With Data Augmentation							
1	1	0.458	0.486	128.6	300	0.10	0.02	12	4	0.3
1	2	0.464	0.466	130.1	300	0.01	0.08	12	4	0.3
1	3	0.453	0.468	133.6	400	0.10	0.08	12	4	0.2
2	1	0.465	0.490	129.4	300	0.10	0.02	18	4	0.3
2	2	0.466	0.466	123.6	300	0.01	0.04	12	4	0.3
2	3	0.448	0.480	125.3	500	0.10	0.08	18	4	0.2

It becomes clear that the tuning process is significantly faster for Architecture II with *No DA*, taking around 50 minutes to complete the process. On the other hand, when using the larger dataset, the process takes more than 12 hours to complete. Overall, models tend to use 300 gradient boosted trees and require 30% of the entire feature set per tree. The best classifiers also require deeper trees, with 12 or 18 levels. It is also worth mentioning that all the best models conceived over the dataset *With DA* required a minimum child weight of 4. This hyperparameter defines the minimum sum of weights of all observations required in a child node, being used to control overfitting and prevent under-fitting, which may happen if high values are used when setting this hyperparameter.

As stated previously, all metrics provided in Table 4.8 are based on traits' accuracy. Using class accuracy instead of sample accuracy, the mean error of Architecture II candidate models using the dataset *With DA* is of 0.165, which corresponds to an accuracy higher than 83%. On the other hand, the mean error with *No DA* increases to 0.338 (an accuracy of 66%).

In this study, both micro and macro-averaged metrics were evaluated. However, since we are interested in maximizing the number of correct predictions each classifier makes, special importance is given to micro-averaging. In fact, micro f1-score of the classifiers conceived over the dataset *With DA* displays an interesting overall value of 0.835, with the *Openness* trait being, again, the one showing the lower value. It is worth mentioning that micro-averaging in a multi-class setting with all labels included, produces the same exact value for the f1-score, precision and recall metrics, being this the reason why Table 4.8 only

Table 4.8: Evaluation results of Architecture II, with and without data augmentation, based on trait's accuracy obtained from the test folds of the outer-split.

Metric	**Global**	**Extraversion**	**Agree.**	**Conscient.**	**Stability**	**Openness**
			No Data Augmentation			
Mean Error	**0.338**	-	-	-	-	-
Micro F1-Score	**0.663**	0.728	0.660	0.620	0.648	0.656
Macro F1-Score	**0.459**	0.532	0.462	0.419	0.443	0.438
Macro Precision	**0.477**	0.590	0.468	0.413	0.445	0.468
Macro Recall	**0.464**	0.525	0.469	0.434	0.447	0.447
			With Data Augmentation			
Mean Error	**0.165**	-	-	-	-	-
Micro F1-Score	**0.835**	0.846	0.834	0.831	0.843	0.822
Macro F1-Score	**0.776**	0.770	0.795	0.801	0.731	0.782
Macro Precision	**0.830**	0.826	0.854	0.840	0.809	0.819
Macro Recall	**0.742**	0.731	0.758	0.774	0.691	0.755

displays micro f1-score. On the other hand, macro-averaging computes each error metric independently for each class and then averages the metrics, treating all classes equally. Hence, since models depict a lower macro f1-score when compared to the micro one, this could mean that there may be some classes that are less used when classifying, such as *low* or *high*. Nonetheless, macro f1-score still presents a very interesting global value of 0.776. Macro-averaged precision also depicts a high value, strengthening the ability of models to correctly classify true positives and avoid false positives. Finally, models' global macro-averaged recall is of 0.742, a promising value that tells us that the best candidate models are able, to some extent, to avoid false negatives. Figure 4.9 provides a graphical view of micro and macro-averaged f1-score and precision for Architecture II for both datasets, being again possible to recognize a better performance when using the dataset *With DA*.

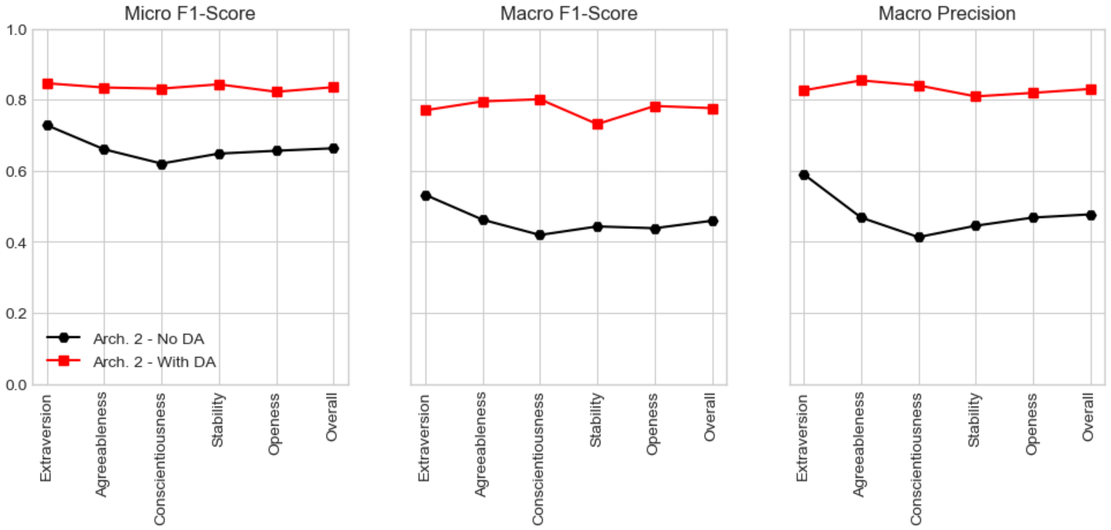


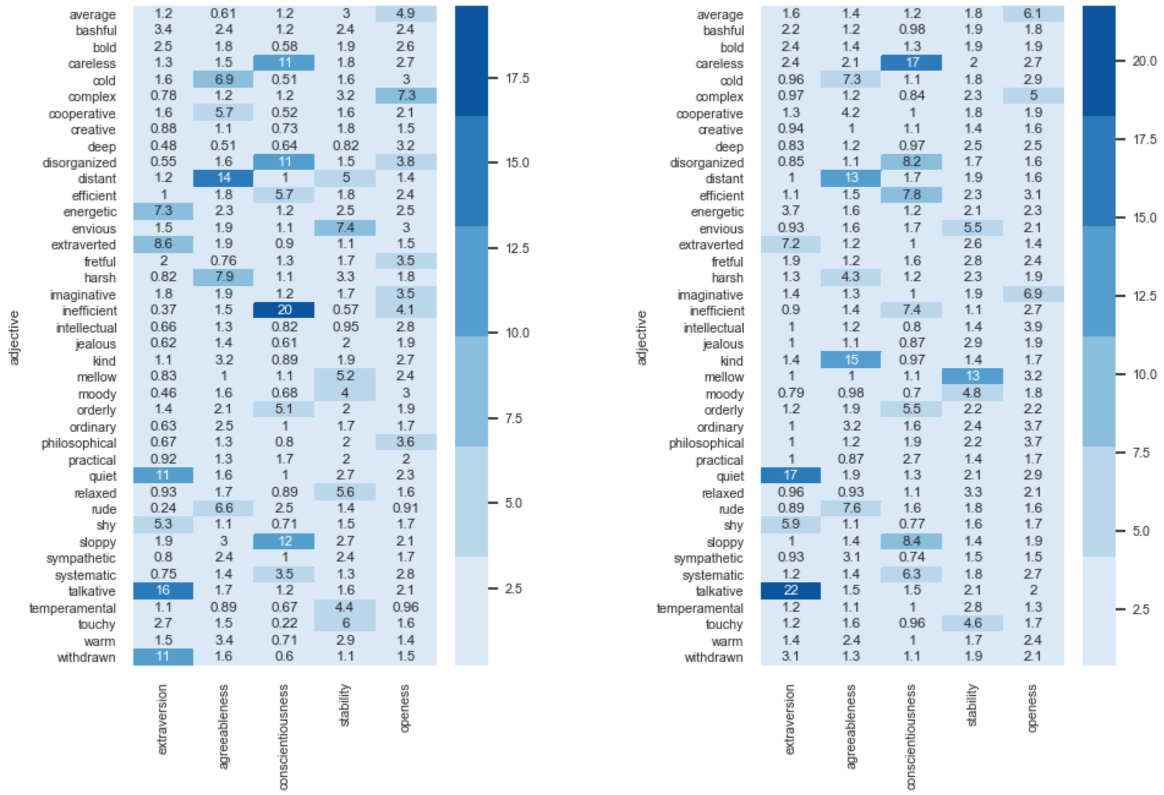
Figure 4.9: Graphical view of Architecture II micro and macro-averaged f1-score, and precision for both datasets.

4.2.5.3 Feature Importance

Gradient Boosted Trees allow the possibility of estimating feature importance, i.e., a score that measures how useful each feature was when building the boosted trees. This importance was estimated using *gain* as importance type, which corresponds to the improvement in accuracy brought by a feature to the branches it is on. A higher value for a feature when compared to another implies it is more important for classifying the label.

Figure 4.10 presents the estimated feature importance of Architecture I using an heat-map view. Interestingly, models conceived using the dataset with *No DA* (Figure 4.10a) give a higher importance to the selection of the adjective *inefficient* when classifying the *Conscientiousness* trait. *Sloppy*, *disorganized*, and *careless* are other adjectives that assume special relevance when classifying the same personality trait. Regarding the *Extraversion* trait, *talkative*, *quiet*, and *withdrawn* are the most important adjectives, being only then followed by the *extroverted* and *energetic* ones. The *Agreeableness* trait gives higher importance to *distant*, *harsh*, *cold*, and *rude*. On the other hand, feature importance is more uniform in the *Stability* and *Openness* personality traits, with the most important adjectives assuming a relative importance of about 7%. Another interesting fact that arises from these results, is that some adjectives have lower importance for all five traits. Examples include *bashful*, *bold*, *intellectual*, and *jealous*.

As for the models conceived using the dataset *With DA* (Figure 4.10b), results are similar to the smaller



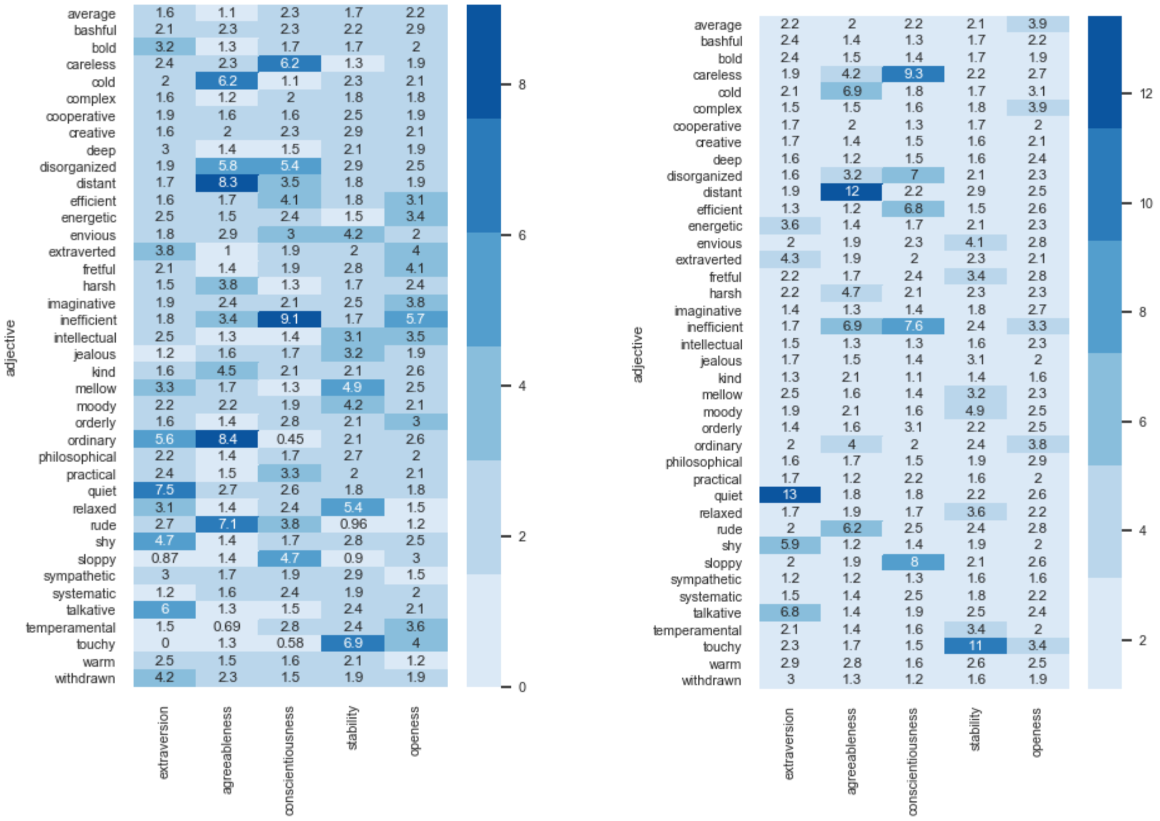
(a) Using dataset with *No DA*.

(b) Using dataset *With DA*.

Figure 4.10: Feature importance heat-map of Architecture I.

dataset. In these models, there is a lower number of important features, but the ones considered as important have stronger importance. An example is the case of the adjective *talkative* for the *Extraversion* trait, which increases its importance from 16% to 22%, and *quiet*, which increases from 11% to 17%. *Withdrawn* and *energetic* have a reduced importance. Interestingly, for the *Agreeableness* trait, the adjective *kind* becomes the most important one, increasing from 3.2% to 15%. The *Openness* trait still assumes a more uniform importance for all features, being this one of the reasons why it was the trait showing worst performance using Architecture I models.

Regarding Architecture II, Figure 4.11 presents the estimated feature importance for both datasets. What immediately draws one attention is the fact that importance values are much more balanced when compared to Architecture I. Indeed, the highest importance value is of 9.1% with *No DA* and 13% *With DA* when compared to 20% and 22% of Architecture I, respectively. Nonetheless, except for a few exceptions, adjectives assuming higher importance in Architecture I also assume higher importance in Architecture II. The main difference is that values are closer together, having a lower amplitude.



(a) Using dataset with *No DA*.

(b) Using dataset *With DA*.

Figure 4.11: Feature importance heat-map of Architecture II.

4.2.5.4 Summary

The proposed ASAP method aims to use ML-based models to reinstate the process of rating adjectives or answering questions by an adjective selection process. To achieve this goal, two different ML architectures were proposed, experimented, and evaluated. The first architecture uses Gradient Boosted Trees regressors to quantify the Big Five personality traits. Overall, this architecture is able to quantify such traits with an error of approximately 5.5 units of measure, providing an accurate output given the limited amount of available records. On the other hand, Architecture II uses Gradient Boosted Trees classifiers to qualify the bin in which the subject stands, for each trait. This architecture was able to quantify the personality traits with a micro-averaged f1-score of more than 83%. A better performance of both architectures in the augmented dataset was also expected since the original dataset had a limited amount of records. The implemented data augmentation techniques aimed to increase the dataset size following well-defined rationales but also included several randomized decisions based on a probabilistic approach in order to reduce bias and create a more generalized version of the dataset. For this, data exploration and pattern mining, in the form of ARL, assumed an increased importance, allowing us to understand relations between selected adjectives. Results for records with very few adjectives selected may be biased to the dataset used to train the models since the ability to quantify traits based on the selection of just one or two adjectives is of an extreme difficulty. Hence, for the ASAP method to behave properly, subjects should be encouraged to select four, or more, adjectives.

A further validation was carried out by means of a significance analysis between the correlation differences of predicted and actual scores. The best overall candidate model of Architecture I was trained using, as input data, 90% of the original dataset, with the remaining being used to obtain predictions. Predictions were compared with the actual scores of the five traits. As expected, the p-value returned a high value (0.968), with a z-score of 0.039. Such values tell, with a high degree of confidence, that the null hypothesis should be retained and that both correlation coefficients are not significantly different from each other. This is in line with expectations since the conceived models are optimizing a differentiable loss function, using a gradient descent procedure that reduces the model's loss to increase the correlation between predictions and actual scores.

Architecture II took significantly more time to fit than Architecture I. However, it provides more accurate results, which are less prone to error. It should be noted that Architecture II only provides an approximation to the Big Five of the subject, i.e., it does not numerically quantify each trait. Instead, it tells in which bin the subject finds himself. This can be useful in cases where the general qualification of each trait is more important than the specific score of the trait. On the other hand, Architecture I will provide an exact score for each personality trait based on a selection of adjectives. Indeed, the working hypothesis has been confirmed, i.e., it is possible to achieve promising performances using ML-based models where the subject, instead of rating forty adjectives or answering long questions, selects the adjectives he relates the most with (RQ4.1). This allows one to obtain the Big Five using a method with reduced complexity and that takes a small amount of time to complete. The obtained results are just estimates, with an underlying error. The conducted experiments showed the ability of ML-based models to compute estimates of personality traits, and should not be seen as a definitive psychological assessment of one's personality traits. For a full personality assessment, tests such as the one proposed by Saucier, Goldberg, or the NEO-personality-inventory should be used.

The use of augmented sets of data may bring an intrinsic bias to the candidate models. In all cases, preference should always be given to the collection and use of real data. However, in scenarios where data are extremely costly, an approximation may allow ML models to be analyzed with augmented data. In such scenarios, data augmentation processes should make use of several randomized decisions based on probabilistic approaches to create a generalized version of the smaller dataset.

In Saucier's test, each personality trait is computed using the rating of eight unipolar adjectives, i.e, no adjective is used for more than one personality trait. Indeed, it is known, beforehand, which adjectives are used by each trait. For example, the *Extroversion* trait is computed based on four positively weighted adjectives and four negative ones. However, in the proposed architectures that make the ASAP method, all 40 adjectives are used to compute all traits, allowing the ML models to use adjectives selection/non-selection to compute several traits, thus harnessing inter-trait relationships. For instance, *bold*, one of the adjectives used by Saucier to compute *Extroversion*, shows a smaller importance in the conceived architectures when quantifying *Extroversion*. The same happens for *bashful* in *Extroversion*, *creative* in *Openness*, and *practical* in *Conscientiousness*, just to point a few. This could lead us to hypothesize that, one, the list of forty adjectives could be further reduced to a smaller set of adjectives by removing those that are shown to have a smaller importance and that, two, there are adjectives that can be used to quantify distinct personality traits, such as the case of *disorganized*, which can be used for the *Conscientiousness* and the *Agreeableness* traits. It is also interesting to note the lack of features assuming high importance when quantifying *Openness*. In fact, one of its adjectives, *ordinary*, seems to assume higher importance in the *Agreeableness* trait. Overall, Saucier's adjective-trait relations are being found and used by the conceived models.

Since the conceived ML architectures proved to be both performant and efficient when using a selection of adjectives, future research points towards a reduction to the minimum required set of adjectives that do not harm the method's accuracy, further reducing complexity and the time it takes to be performed by the subject [58].

4.3 Boredom Detection

Boredom is a subjective feeling that tends to emerge in situations where there is a lack of stimuli, being a state of relatively low arousal and dissatisfaction [233]. In other words, boredom may be a sign that an activity is not sufficiently satisfying neither motivating. Common sense tells us that, when bored, people tend to use their smartphone. Indeed, it is safe to assert that people tend to use their smartphone, if available, in situations where boredom arises. In addition, it has been reported that bored people tend to consume more, which makes them a better candidate to subscribe to promotional campaigns and engage with new content [234]. On the other hand, bored people could also make a more productive use of such idle moments. Hence, the conducted research aims to detect such feeling when using the smartphone, without any kind of biometric data and using only data from smartphones' sensors and ML models. A research question has been elicited and stands as follows, viz.

RQ4.2. Is it possible to accurately detect and predict boredom using ML models and smartphone's sensors data?

To achieve the proposed goal, it is of the utmost importance the existence of a dataset from where one could conceive and evaluate several candidate ML models. Hence, the next lines describe the conceived mobile application for data collection, with the user being queried, at specific intervals, how bored he felt during that same period. This labeling method is known as the *Experience Sampling Method*, i.e., a research method that probes participants to report on their thoughts or feelings on multiple occasions over time. The next lines will also go through the current state of the art regarding boredom detection, the collected dataset, and the results obtained from the conceived ML models.

4.3.1 State of the Art

Literature reveals that researchers have already engaged in studying ways to measure and detect boredom as well as other feelings, such as stress [235, 236], fatigue [237, 238], or happiness [239]. Initial attempts to detect boredom focused on specific data collected by sensors that must be carried at all times for continuous monitoring, representing a major limitation [240]. Recent times came, however, with promising results regarding the use of computers and smartphones to detect boredom.

In 2013, Bixler and D'Mello focused on the writing periods at a computer to predict boredom, using Decision Trees and Naive-Bayes classifiers, achieving promising results [240]. Their predictions were based on a set of values obtained with the help of a keystroke analysis, task appraisals, and personality traits. It should be noted that the Naive-Bayes classifier obtained an accuracy of 82% while the Decision Trees obtained a precision of 87% when trying to predict boredom. Still regarding computers, Guo et al. (2009) found, with the help of Support Vector Machines, that web interaction events, such as mouse movements or the number of clicks on a page, allow the prediction of whether a person is willing to be distracted, which may be indicative of boredom [241]. In 2019, Seo et al. developed ANNs to classify boredom using data from electroencephalography and galvanic skin response exams, achieving an accuracy of 79.98% [242]. A very similar study of the same authors focused only on data from electroencephalography exams, obtaining higher precision with a K-Nearest Neighbor based-model [243].

Considering smartphones, one important attempt has been made to use the smartphone's data to predict user boredom. In 2015, Pielot et al. focused, among others, on what aspects of mobile phone usage are the most indicative of boredom. The authors developed a mobile application for data collection, which run for 14 days [244]. The collected data included, among others, user's demographic and personal information. They framed the problem as a binary one, i.e., whether the user is bored or not. Three distinct classifiers were used, with Random Forests being the one showing the best accuracy (when compared to Logistic Regression and Support Vector Machines). On the other hand, the overhead between precision and recall is significant, since precision levels limited the recall to 30% [244].

Studies have also been carried out in relation to other feelings. Sano and Picard (2013) have reported good results for stress recognition using a combination of mobile devices and clothing-associated sensors, despite some restrictions, such as the limited number of subjects and data [245]. Another study, conducted by Bogomolov et al. (2013), concluded that it was possible to accurately recognize the happiness of an individual during his daily affairs, using a large set of features, obtained with the help of smartphones, such as communication data, proximity sensors information, bluetooth connection, weather data, and personal characteristics [239]. In 2016, Pimenta et al. developed ANNs to detect mental fatigue based on the user's interaction patterns with the computer (mouse and keyboard), showing that when users claim to feel mentally fatigued, they use these peripherals in a different manner [238].

4.3.2 Data Collection and Exploration

Given the absence of open data from where one could derive knowledge, we were required to gather our own dataset. For that, we conceived a mobile application for data collection, which is described in the next lines.

4.3.2.1 Mobile Application for Data Collection

To build the dataset, a mobile application was conceived and made available online. It targets Android users with, at least, Android Marshmallow (API version 23). Android Pie, API version 28, is the targeted version while Android Studio 3.5.2 was the used IDE.

Since the goal was to use only data from the smartphones' sensors, we were required to build an application that was able to query both soft and physical sensors during a period of time. We would then store the sensors' values and ask the subject how bored did he feel at that time. This allowed us to label the collected data with the corresponding boredom value. This process repeats itself over time. The conceived application consists of a set of services, broadcast receivers, and sensors listeners that query sensors periodically. The function of these components is as follows:

- *Services* are components representing the application's desire to perform long-running operations in the background, i.e., the component keeps executing even when the subject is not interacting with the application or if the screen of the smartphone is locked, for example. Since API version 26, the Android system imposes major restrictions on running background services when the application itself is not in the foreground. Each service class requires a corresponding declaration in the manifest file;

- *Broadcast Receivers* provide the ability to subscribe to particular events that may happen at any time (similarly to the publish-subscribe pattern). Such events may be raised by the Android system or any other application. When a broadcast is sent, the Android system routes broadcasts to all applications that have subscribed to it. Broadcasts can be received through manifest-declared receivers or context-registered receivers. Listing 4.3 depicts an example of a *BroadcastReceiver* to listen for changes on the audio jack of the smartphone (headset plugged/unplugged);

Listing 4.3: Creation of a *BroadcastReceiver* to listen for changes on the audio jack.

```
class AudioJackReceiver: BroadcastReceiver() {

   override fun onReceive(context: Context?, intent: Intent) {
      if (intent.action == Intent.ACTION_HEADSET_PLUG)
         FirebaseBoredom.audioJack = intent.getIntExtra("state", -1).toFloat()
   }

   fun getFilter(): IntentFilter? {
      val filter = IntentFilter()
      filter.addAction(Intent.ACTION_HEADSET_PLUG)
      filter.priority = 1000
      return filter
   }
}
```

- *Sensors Event Listeners* is an Android feature that is used to receive notifications from the sensor manager when there is new sensor data. Listing 4.4 depicts an example of a *SensorEventListener* to listen for updates on the accelerometer sensor.

Listing 4.4: Creation of a *SensorEventListener* to register the accelerometer sensor value.

```
class AccelerometerSensorListener: SensorEventListener {

   private lateinit var sensorManager: SensorManager

   override fun onSensorChanged(event: SensorEvent) {
      if (event.sensor.type == Sensor.TYPE_ACCELEROMETER) {
         FirebaseBoredom.accelerometerSensor = event.values[0]
         sensorManager.unregisterListener(this)
      }
   }

   fun setSensorManager(sensorMan: SensorManager) {
      sensorManager = sensorMan
   }

   override fun onAccuracyChanged(sensor: Sensor?, accuracy: Int) {}
}
```

In terms of structure, the conceived application consists of the following packages, viz.

1. *activity* package, where all views of the application are defined:

   a) *BoredomExplainerActivity* provides the rationale of all used sensors;

   b) *BoredomQuestionActivity* requests subjects to quantify how bored they feel at that time. As soon as the subject sets his boredom state, data are stored and committed (or cached if no internet connection is available);
   c) *InitialActivity* is the main activity and the one starting the background service for data collection;
   d) *StopBackgroundServiceActivity* allows the subject to stop, at any time, the background service.

2. *handlers* package, containing all handlers responsible for the domain's logic:
   a) *FirebaseHandler*, which includes all logic to access, write, and read from Firebase's Cloud Firestore database;
   b) *PermissionHandler*, implementing the required methods to obtain the required permissions;
   c) *ServiceHandler*, implementing all methods to start and stop the boredom service.
3. *model* package, containing all classes that represent data objects, including the *FirebaseBoredom* object;
4. *receivers* package, where the broadcast receivers are set, in particular:
   a) *AudioJackReceiver*;
   b) *BootReceiver*;
   c) *CallsReceiver*;
   d) *KeysReceiver*;
   e) and *ScreenReceiver*.
5. *sensors* package, where the sensors event listeners are set, in particular:
   a) *AccelerometerSensorListener*;
   b) *AmbientTemperatureSensorListener*;
   c) *GravitySensorListener*;
   d) *GyroscopeSensorListener*;
   e) *HumiditySensorListener*;
   f) *LightSensorListener*;
   g) *MagneticSensorListener*;
   h) *PressureSensorListener*;
   i) and *ProximitySensorListener*.

6. *services* package, where the two services used by the boredom application are set:

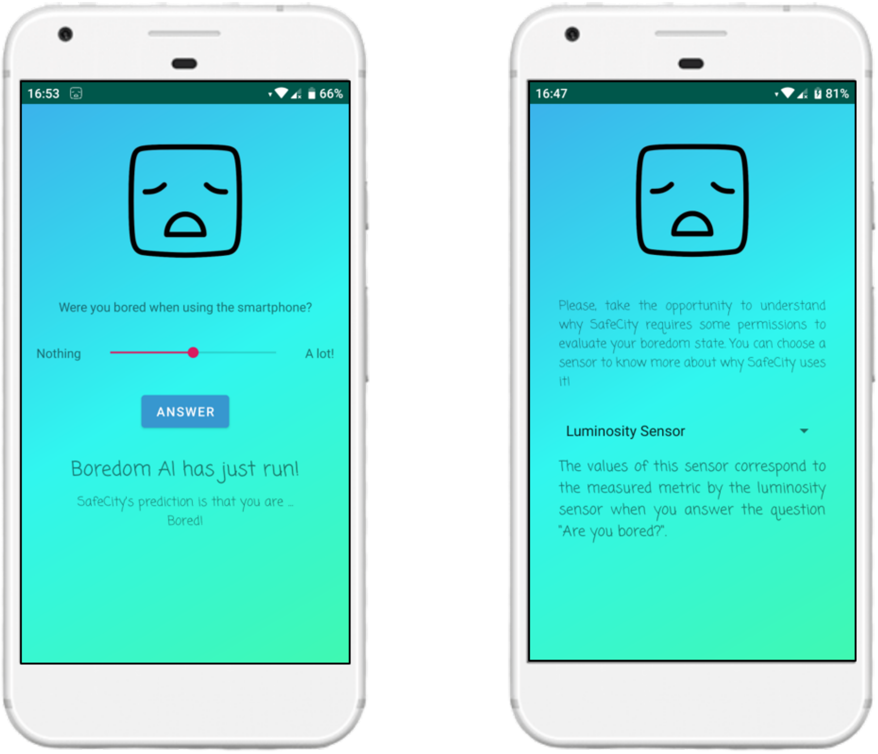
   a) The *BackgroundService* is the one responsible for starting the main background service and for registering all broadcast receivers;

   b) The *NotificationsService* is the one responsible for sensing all notifications that emerge in the subject's smartphone, classifying notifications into three distinct types (chatting, social, and other).

Android has been imposing major restrictions on background services when the application that launched the service is not in the foreground. Hence, to guarantee that the service keeps running as soon as the subject starts the boredom application, we were required to use a background service with a foreground notification. This type of service is known as a Foreground Service because, even though it runs in the background, it displays a foreground notification in the smartphone's screen, continuously. With this, the Android system considers users to be aware that a particular service is running in the background and, thus, will not kill neither consider it a candidate for killing when low on memory. Foreground services keep running even if the user closes the application.

Regarding the implemented broadcast receivers, two deserve a deeper look. The first, *BootReceiver*, is the one listening for a system's boot, being responsible for restarting the boredom service as soon as the boot is completed, thus guaranteeing that the service keeps running even if the smartphone was turned off and on. The second, *ScreenReceiver*, besides counting the number of screen activations the subject makes, also contains the timer to query subjects about their boredom state. I.e., the first time the screen is activated after starting the boredom service, a screen timer is initialized and scheduled to run every 10 minutes (*SCREEN_TIMER_MILLI_PERIOD* constant). Every 10 minutes, the screen timer resets all values, obtains a first snapshot of the sensors' values, and starts another timer (the notification timer) that is scheduled to run only once, after 3 minutes (*NOTIFICATION_TIMER_MILLI_PERIOD* constant). This second timer is responsible for launching a notification in the subject's smartphone, which is a shortcut to the *BoredomQuestionActivity* (Figure 4.12a). If the user dismisses this notification, data are discarded and timers keep executing as scheduled. However, if the user clicks on the notification, he is redirected to the referred activity, which requests him to quantify, using a slider, how bored the user is feeling. As soon as the subject saves his answer, a second snapshot of the sensors' values is obtained and data are committed or stored to be committed later, if no internet connection is available. The subject has 20 minutes to submit his answer otherwise all data are discarded and new snapshots are obtained. Data are collected if, and only if, permission was granted by the subject to access such sources. To be as less intrusive as possible, we opt not to implement the *process_outgoing_calls* neither the *read_sms* android permissions. The subject may decide, at any time, to turn on, or off, the boredom service, having also available an activity that describes all used sensors (Figure 4.12b).

It is important to emphasize that no biometric neither demographic data are collected, neither it is possible to link a record of the dataset to the person who answered it. In other words, only the sensors' values and the boredom level are stored, there is no user identifier. All data are committed to an instance



Were you bored when using the smartphone?

Boredom AI has just run!

SafeCity's prediction is that you are ... Bored!

Please, take the opportunity to understand why SafeCity requires some permissions to evaluate your boredom state. You can choose a sensor to know more about why SafeCity uses it!

Luminosity Sensor

The values of this sensor correspond to the measured metric by the luminosity sensor when you answer the question "Are you bored?".

(a) Activity asking the subject "How Bored Do You Feel?".

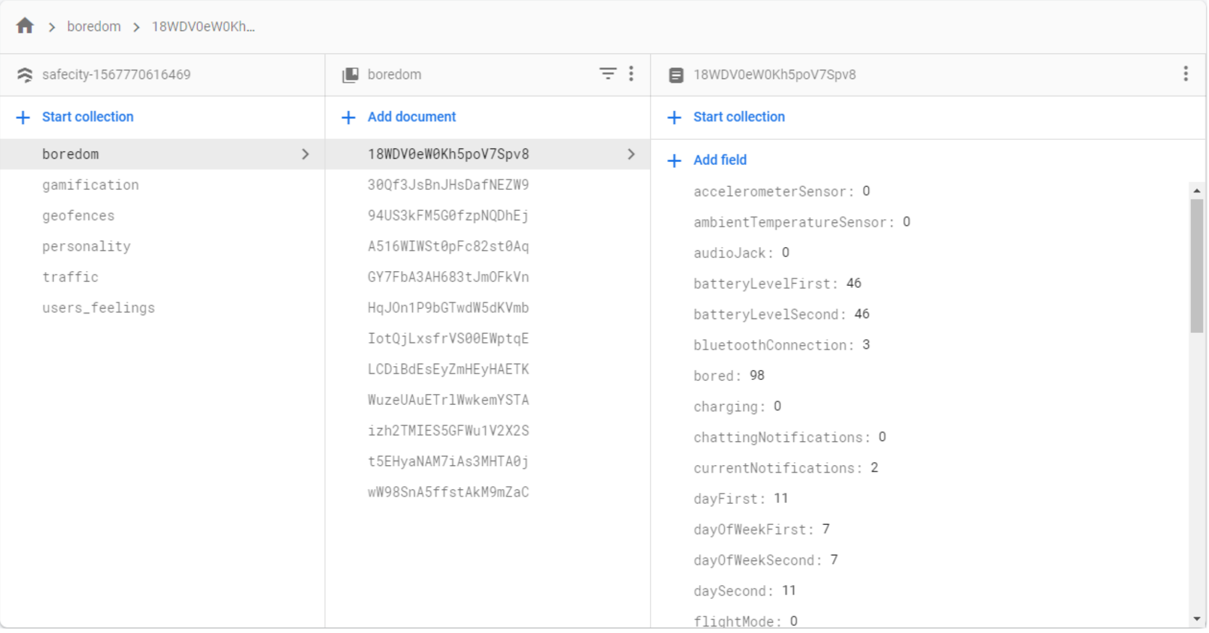
(b) Activity with the rationale of all the used sensors.

Figure 4.12: Main activities of the mobile application to build the boredom dataset.

of Firebase's Cloud Firestore database, which, similarly to geofences and the emotional map, contains a master collection (*boredom*) holding documents per committed entry. Each document is labeled with a random identifier and holds a set of fields that correspond to the features that make the boredom dataset. Figure 4.13 depicts the hierarchical data structure to store boredom entries. Listing 4.5 details the access rules to the *boredom* collection. Users are only able to create new entries and only if they are authenticated users (write permission). Except for the database administrator, no one is allowed to read or update any existing entry.

Listing 4.5: Access rules to the *boredom* collection.

```
rules_version = '2';
service cloud.firestore {
  match /databases/{database}/documents {
    match /{document=**} {
      allow read, write: if false;
    }
    match /boredom/{document=**} {
      allow create: if request.auth != null;
    }
  }
}
```



boredom
gamification
geofences
personality
traffic
users_feelings

18WDV0eW0Kh5poV7Spv8
30Qf3JsBnJHsDafNEZW9
94US3kFM5G0fzpNQDhEj
A516WIWSt0pFc82st0Aq
GY7FbA3AH683tJmOFkVn
HqJOn1P9bGTwdW5dKVmb
IotQjLxsfrVS00EWptqE
LCDiBdEsEyZmHEyHAETK
WuzeUAuETrlWwkemYSTA
izh2TMIES5GFWu1V2X2S
t5EHyaNAM7iAs3MHTA0j
wW98SnA5ffstAkM9mZaC

```
accelerometerSensor: 0
ambientTemperatureSensor: 0
audioJack: 0
batteryLevelFirst: 46
batteryLevelSecond: 46
bluetoothConnection: 3
bored: 98
charging: 0
chattingNotifications: 0
currentNotifications: 2
dayFirst: 11
dayOfWeekFirst: 7
dayOfWeekSecond: 7
daySecond: 11
flightMode: 0
```

Figure 4.13: Firebase's Cloud Firestore cloud-hosted NoSQL database holding boredom data.

4.3.2.2 Data Exploration

The dataset used in this work was collected from December 03rd 2019 to February 16th 2020. It contains a total of 1511 observations, without missing values. There are, however, failed sensors' readings filled with the value -1. Indeed, both the *ambientTemperatureSensor* and the *humiditySensor* features are always filled with -1, meaning that none of the used smartphones had available such sensors. Table 4.9 presents the 45 features that make the collected dataset. All features' values are real numbers.

With all features assuming a non-Gaussian distribution (under the Shapiro-Wilk test with $p < 0.05$),

Table 4.9: Features available in the boredom dataset.

#	Feature	#	Feature	#	Feature
1	*accelerometerSensor*	16	*gyroscopeSensor*	31	*orientation*
2	*ambientTemperatureSensor*	17	*homeButtonPress*	32	*otherNotifications*
3	*audioJack*	18	*hourFirst*	33	*outgoingCalls*
4	*batteryLevelFirst*	19	*hourSecond*	34	*pressureSensor*
5	*batteryLevelSecond*	20	*humiditySensor*	35	*proximitySensor*
6	*bluetoothConnection*	21	*incomingCalls*	36	*recentButtonPress*
7	*bored*	22	*isCharging*	37	*ringerMode*
8	*chattingNotifications*	23	*lightnessSensor*	38	*screenActivations*
9	*currentNotifications*	24	*magneticSensor*	39	*smsReceived*
10	*dayFirst*	25	*minuteFirst*	40	*socialNotifications*
11	*dayOfWeekFirst*	26	*minuteSecond*	41	*timestamp*
12	*dayOfWeekSecond*	27	*mobileDataSensor*	42	*weekend*
13	*daySecond*	28	*monthFirst*	43	*wifiSensor*
14	*flightMode*	29	*monthSecond*	44	*yearFirst*
15	*gravitySensor*	30	*notificationsRemoved*	45	*yearSecond*

the non-parametric Spearman's rank correlation coefficient was used. In this dataset there are a few pairs of highly-correlated features including the pairs *wifiSensor* and *mobileDataSensor*, *hourSecond* and *hourFirst*, *daySecond* and *dayFirst*, *dayOfWeekSecond* and *dayOfWeekFirst*, *batteryLevelSecond* and *batteryLevelFirst*, and *year* and *month*.

The correlation between the independent features and the dependent one was analyzed using the F-test, which can assess multiple coefficients simultaneously. In this test we aim to find the independent features that allow us to reject the null hypothesis that the fit of an intercept-only model and a linear model is equal, with $p < 0.05$. Eleven features were found to be statistically significant, allowing us to reject the null hypothesis for each one. Among such features, the ones with most importance are *bluetoothConnection*, *mobileDataSensor*, *wifiSensor*, *weekend*, *currentNotifications*, and *isCharging*.

The target, i.e., the *bored* feature, is, in its original form, imbalanced. Figure 4.14 depicts the cardinality of observations per *bored* values in twenty bins of equal width. The great majority, 388, fall into the first bin, which ranges from 0 to 5 (no boredom). The bin comprising bored values from 50 to 55 (somewhat bored) contains 274 observations. The third most frequent bin is the last one, from 95 to 100 (absolutely bored), with 193 observations. On the other hand, if to divide the *bored* feature into two bins of equal width, both ones would contain approximately the same amount of observations ($\approx$ 750) [43].

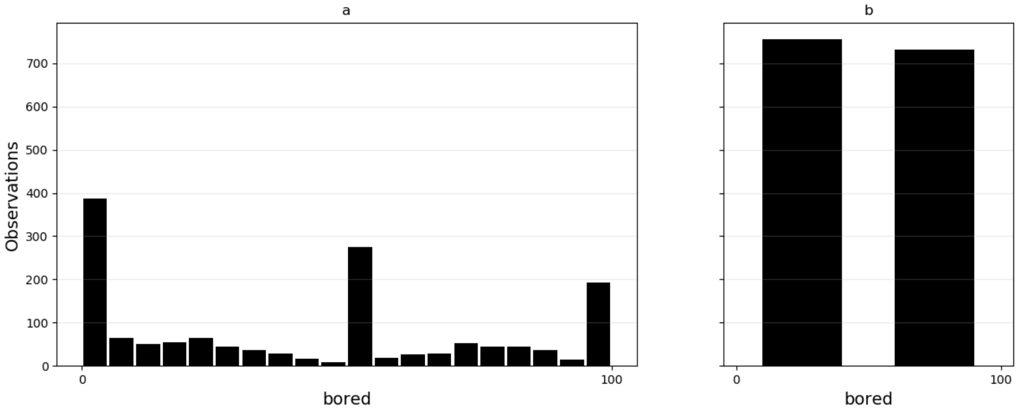


Figure 4.14: Histogram for the *bored* feature, using (a) 20 bins and (b) 2 bins of equal width.

4.3.3 Data Preparation and Pre-processing

The first step to clean the dataset was to drop the *ambientTemperatureSensor* and *humiditySensor* features. Then, 23 observations with the year as -1 were also removed. Since no variations occurred between the pairs *yearFirst* and *yearSecond*, and *monthFirst* and *monthSecond*, the first were removed and the second were renamed to *year* and *month*, respectively. The *month* feature was then incremented by one since it was in the interval $[0, 11]$. An index, based on the timestamp up to the minutes, was created, being sorted in ascending order.

As feature engineering, three new features were created. The first one, entitled as *batteryVariation*, was created based on the variation of the *batteryLevelSecond* and *batteryLevelFirst*, allowing one to understand if the level of battery dropped, remained stable, or even if the smartphone was charging. The second and third ones, *minutesInterval* and *hourInterval*, allows one to understand how many minutes and hours that particular observation encompasses, respectively.

Based on the analysis of the correlation matrix, eight features were dropped, in particular, *dayFirst*, *hourFirst*, *minuteFirst*, *year*, *month*, *otherNotifications*, *batteryLevelSecond*, and *dayOfWeekFirst*. Other features were also renamed, in particular, *dayOfWeekSecond* to *dayOfWeek*, *daySecond* to *day*, *hourSecond* to *hour*, and *minuteSecond* to *minute*.

Finally, considering the candidate models we aim to experiment, data were normalized, i.e., scaled to the range $[0, 1]$. In fact, even though Decision Trees are not affected by any monotonic transformation to the input data, Support Vector Machines and ANNs work better with normalized data. Since no categorical feature was present in the dataset, no feature encoding method was applied. Finally, the problem was framed as a binary classification one, i.e., the target feature, *bored*, was binned into ”Not Bored” and ”Bored” bins. As shown previously, with a binary classification problem the dataset gets well balanced, with 732 ”Bored” observations and 756 ”Not Bored” ones. The shape of the final dataset is of 1488 observations with 35 features, including the target one.

4.3.4 Experiments

The goal of the conducted research is to develop and tune the best possible ML model to detect and predict boredom when a person is using his smartphone. Hence, to take advantage of the dataset's characteristics, five distinct models were conceived and tuned. Such models, besides behaving well in binary classification problems, also have the ability to generalize well in smaller datasets. All candidate models were tuned as described in the following lines.

4.3.4.1 Experimental Setup

Python, version 3.7, was the used programming language for data preparation and pre-processing as well as for model development and evaluation. Knime, a free and open-source data analytics platform, was also used for the same purposes. TensorFlow, Pandas, NumPy, scikit-learn, matplotlib, and pickle were the used libraries. For increased performance, Tesla T4 GPUs were used. All hardware is made available by Google's Colaboratory.

4.3.4.2 Model Conception and Tuning

Each model was tuned in regard to a set of hyperparameters to look for the best candidate one. Five distinct models have been experimented. The Decision Tree and the Support Vector Machine had their results validated under 10-fold cross-validation, while the Random Forest, the Gradient Boosted Trees, and the MLPs experienced 5-fold cross-validation to reduce the training times. Nested cross-validation was implemented with an outer $k = 5$. Training samples were further split into training and validation, in a ratio of 9:1. Table 4.10 describes the search space for each hyperparameter of the candidate models.

Table 4.10: Hyperparameter's search space for each candidate model.

Model	**Hyperparameter**	**Search Space**
Decision Tree	Quality Measure	[Gain ratio, Gini index]
	Pruning	[No Pruning, MDL, Reduced Error]
	Minimum Number of Records	[1, 25] with a unitary step size
Random Forest	Quality Measure	[Information Gain (Ratio), Gini index]
	Number of Trees	[100, 1000] with step size of 100
	Tree Depth	[-1] & [5, 25] with step size of 2
Support Vector Machine	Kernel	[Linear, Polynomial, RBF, Sigmoid]
	C	[0.001, 0.01, 0.1, 1]
	Gamma	[0.001, 0.01, 0.1, 1, Auto, Scale]
	Degree	[1, 7] with a unitary step size
Gradient Boosted Trees	Number of Estimators	[300, 400, 500]
	Eta	[0.01, 0.001]
	Trees' Max Depth	[4, 12, 18]
	Colsample by tree	[0.2, 0.3]
Multilayer Perceptrons	Learning Rate	[0.01, 0.001]
	Number of Neurons	[16, 32, 64]
	Dense Activation	[ReLU, sigmoid]
	Batch size	[16, 32]

Random Search was the used algorithm for hyperparameter search. The candidate models were evaluated in regard to their accuracy, precision, and recall. The accuracy metric tells us the percentage of correctly classified observations. On the other hand, the precision and recall metrics allow one to understand, and measure, relevance based on true and false positives. In other words, precision tells us how many of the observations classified as ”bored” were, indeed, ”bored” observations. Recall tells us, from all ”bored” observations, how many did the model found (did the model found all the ”bored” observations? How many observations did the model incorrectly classified as ”not bored”?). Precision may be prioritized over recall if the goal is to reduce the number of false positives.

4.3.5 Results and Discussion

Table 4.11 summarizes the obtained results for each conceived model. Gradient Boosted Trees were the ones presenting the best results, achieving an accuracy higher than 70%. The best Random Forest model closely follows, presenting slightly lower values. On the opposite spectrum, MLPs were the ones showing worst performance.

The best Gradient Boosted candidate model achieved an accuracy of 71.8% with 400 estimators, 0.001 as learning rate, 12 depth levels, and 20% of sub-sampled columns (colsample). It showed a precision of 73.9% and a recall of 71.5%. In general, all Gradient Boosted candidate models showed high performance. The second best candidate model was a Random Forest, achieving its best performance with Information Gain as quality measure, 800 decision trees, and a maximal tree depth of 25 levels. It achieved an accuracy

Table 4.11: Summary results of the three best candidate models, per algorithm, for boredom detection.

Model	**Hyperparameters**				**Accuracy**
	Qual. Measure	*Pruning*		*Min. Records*	
Decision Tree	Gini index	MDL		14	**0.653**
	Gini index	No pruning		11	0.649
	Gain ratio	MDL		17	0.633
	Qual. Measure	*N° of Trees*		*Tree Depth*	
Random Forest	Information Gain	800		25	**0.684**
	Information Gain	800		-1	0.683
	Gini	400		15	0.683
	Kernel	*C*	*Gamma*	*Degree*	
Support Vector Machine	Polynomial	1	Scale	4	**0.659**
	Polynomial	0.1	1	3	0.631
	RBF	1	0.001	-	0.603
	Estimators	*Eta*	*Depth*	*Colsample*	
Gradient Boosted Trees	400	0.001	12	0.2	**0.718**
	400	0.01	18	0.3	0.696
	300	0.001	18	0.2	0.671
	Learn. Rate	*Neur.*	*Activ.*	*Batch Size*	
Multilayer Perceptrons	0.001	32	ReLU	16	**0.651**
	0.001	64	ReLU	32	0.650
	0.001	32	ReLU	32	0.614

of 68.4%, presenting an overall reduction of more than 3% when compared to the best model. In terms of precision and recall, it achieved a value of 70.9% and 70.6%, respectively. The fact that the dataset was well balanced, allowed the models to achieve interesting precision and recall values. In fact, the "Not Bored" class had a slightly better precision and recall than the "Bored" one. The better performance of these two models can be explained by the fact that both make use of several decision trees (400 and 800, respectively), making them an ensemble learning method that makes use of a simple yet powerful concept - the wisdom of crowds. For instance, a single Decision Tree achieved an overall accuracy of 65.3%, and a mean precision and recall of 65.8%, using Gini Index as quality measure, minimum description length as pruning method, and 14 observations as stopping criteria. This single tree showed an acceptable, although inferior, performance. However, it is faster to train and is computationally lighter when compared to Gradient Boosted Trees and Random Forests.

As previously demonstrated, Gradient Boosted Trees provide the ability to compute estimates of feature importance, i.e., a score that measures how useful a feature was when building the boosted trees. In this experiment, this importance was estimated using *gain* as importance type. Figure 4.15 depicts a heat-map view of the importance of each feature to predict boredom. Those showing higher importance were *bluetoothConnection*, *wifiSensor*, *lightnessSensor*, *dayOfWeek*, *audioJack*, and *ringerMode*. Interestingly, from these, only the first two were previously shown to have a higher correlation with the target

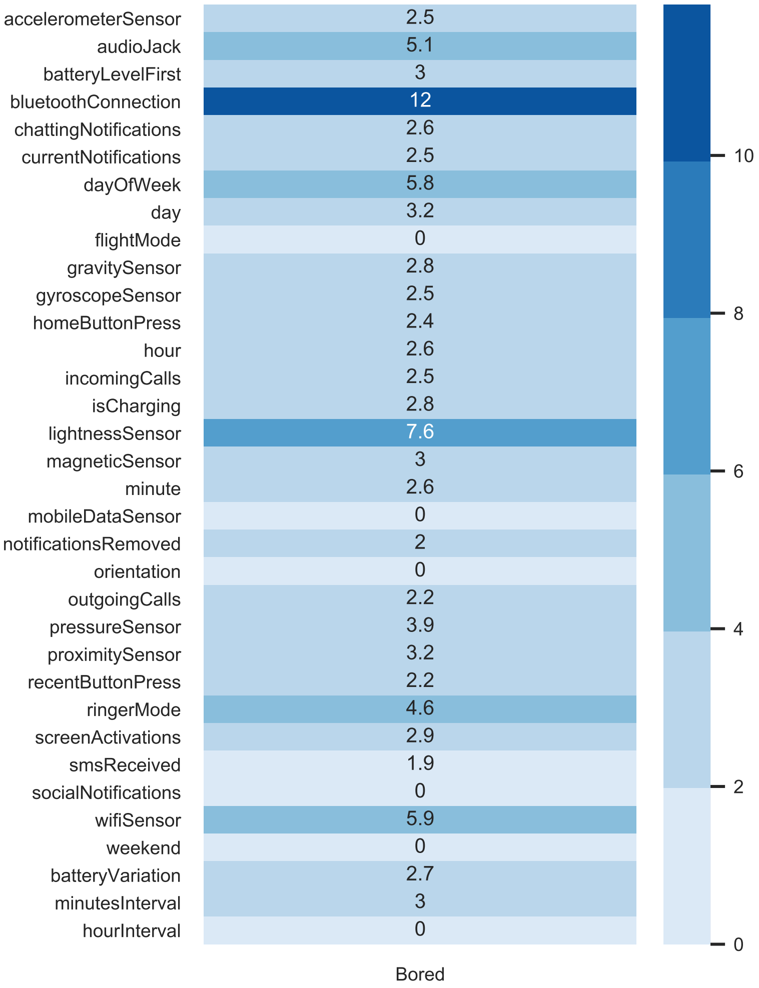


Figure 4.15: Feature importance heat-map for the best Gradient Boosted model.

feature (based on the results of the F-test performed during the data exploration process). Moreover, even though *weekend* and *mobileDataSensor* showed high correlation with the target feature, the boosted trees assigned null importance to both features. Other features also showed null importance, including the *hourInterval*, *socialNotifications*, *orientation*, and *flightMode*. Such features are good candidates to be discarded in order to reduce the model's complexity and computational cost.

Two non tree-based models have also been experimented. Both showed performances similar to those of the single Decision Tree (≈ 65%). The Support Vector Machine, which achieved an accuracy of 65.9%, a precision of 61.5%, and a recall of 66.1%, used a polynomial kernel, a fourth-degree support function, a regularization parameter of 1, and a scaled gamma of $1/(nrOfFeatures \times variance)$. Nonetheless, the models that behaved the worst were MLPs. The best MLPs, with four hidden layers, Adam as optimizer, and binary cross-entropy as loss function, showed an accuracy of 65.1%, and a precision and recall of 67% and 66%, respectively. MLPs were also computationally more expensive when compared with the remaining candidates.

Taking into account the available data and the defined restrictions (no biometric neither personal data), the achieved results show promising prospects with regard to the detection of boredom using only smartphone's sensor data, leaving aside all intrusive data such as calls, cameras, location, and messages. This may change the way we interact with smartphones, opening new ways for the recommendation of to-do tasks, books, or even ways to grow one's knowledge upon situations where there is lack of stimuli. The conceived mobile application behaved properly and is still available online in order to increase the size of the dataset. The overall results are in line with expectations, given the fact that the dataset leaves aside all kinds of biometric and intrusive data. In addition, the conceived approach only considers smartphone sensors', eliminating the need for subjects to carry specific sensors or use computers. As a direct answer to the elicited research question (RQ4.2), it can be said that it is possible to conceive ML models that are able to detect, with promising accuracy and precision, user boredom using only the smartphone's sensors as base data. This can open new avenues to improve targeted advertising and time management, among many others. The plan is now to keep expanding the dataset and re-evaluate the proposed models with more data.

4.4 Fully Informed VRUs - Knowledge Representation and Reasoning

A classic approach to citizen sensing focuses on the use of surveys to auscultate a person about his/her opinion on a particular topic. With this in consideration, this study aims to set a novel mathematical approach to quantify answers based on a thermodynamics' view to knowledge representation and dataset construction. In particular, the next lines present a meta-analytic study focused on proposing a method to KRR centered on logic programming, grounded on surveys that were developed to capture key components affecting pedestrian walking and crossing behaviors, attitudes, motivations, and habits. The elicited research question may be set as follows, viz.

RQ4.3. Is it possible to quantify people's qualitative responses to surveys using a thermodynamics approach to knowledge representation?

To answer this research question, a set of data from existing surveys were gathered and used to illustrate the proposed approach to KRR. The next lines will also go through the collected data, the proposed approach, and the obtained results.

4.4.1 State of the Art

KRR aims at the understanding of the information's complexity and the associated inference mechanisms [246]. Indeed, automated reasoning capabilities enables a system to fill in the blanks when one is dealing with incomplete information, where data gaps are common, i.e., although KRR has been grounded on a symbolic logic in vector spaces, the fundamentals and the attributes of the logical functions go from discrete to continuous, allowing for the representation or handling of unknown, vague, or even self-contradictory information. Such fact stands for a key distinction of the proposed approach. Otherwise, it would be only symbolic logic where the data items would remain essentially discrete, and therefore no added value would be attained.

Over the years, KRR has been proposed using a logic programming language, namely in terms of the model and the proof theoretical approach to computing, being this last one used in this work [247–252]. Therefore, a program is to be here understood as a set of predicates' extensions as depicted in Listing 4.6.

Listing 4.6: Predicates' extensions.

```
{
   ¬ p ⇐ not p, not $except_p$
   p ⇐ $p_1$, ..., $p_n$, not $q_1$, ..., not $q_m$
   ?($p_1$, ..., $p_n$, not $q_1$, ..., not $q_m$) (n,m ≥ 0)
   $except_{p1}$, ..., $except_{pj}$ (0 ≤ j ≤ k, being k an integer number)
} :: $scoring_{value}$
```

Surveys, on the other hand, are usually used to obtain a snapshot of the attitudes, behaviors, thoughts, opinions, and comments of the targeted population about a particular topic of interest. Surveys are, in fact, useful to describe the characteristics of a population. Several studies have been carried out where a series of surveys are implemented to obtain important insights on community attitudes to road safety [253], the impact of driver distraction [254], or the implications of unlicensed driving to road safety [255], just to name a few. Other studies focused on VRUs and on evaluating their behavior at crosswalks [256], on understanding pedestrian's red-time crossing behavior [257], or perceive VRUs behavior in low, middle, and high-income countries [258]. Understanding the behaviors of VRUs on the road may, on the one hand, allow the enhancement of their safety and, on the other, may aid those that design and plan pedestrian crossings and all other environmental features that make VRUs safer.

Such works typically apply statistical-based methods to quantify the results of surveys, applying descriptive statistics, clustering, or principal component analysis. It is, however, challenging to go from a qualitative to a quantitative result. Hence, the next lines present a novel approach to knowledge representation and dataset construction based on a thermodynamics' view as a process of energy devaluation, which facilitates the process of handling qualitative data. Per se, thermodynamics is a branch of physics that deals with the relation of heat and temperature with energy and work [259]. To understand the foundations of the proposed approach, consider the first two laws of thermodynamics. The first describes the conservation of energy, stating that, for an isolated system, the total amount of energy is constant, i.e., energy can be transformed but cannot be created nor destroyed. The second law describes entropy, a property that quantifies the state of order of a system and how it can evolve [260]. These properties fit our vision of a data item when KRR practices are understood as a process of energy devaluation. Indeed, our approach understands a data item to be, in a given moment, at an entropic state with attained energy that can be decomposed and consumed in the sense of devaluation but never consumed in the sense of destruction [252]. When decomposed, a data item may be seen in the form, viz.

- *exergy*, also called available work, is the energy that can be used after a transfer operation, i.e., the generated entropy. It is given, in Figure 4.16, as the darker areas;

- *vagueness*, which describes energy values that may, or may not, have been transferred and consumed. It is given, in Figure 4.16, as gray painted areas; and
- *anergy*, stands for the potential that has not yet been transferred, being, therefore, available. It is given, in Figure 4.16, as white painted areas.

Question	Scale									
	(5)	*(4)*	*(3)*	*(2)*	*(1)*	*(2)*	*(3)*	*(4)*	*(5)*	*Vagueness*
1.	×	×								
2.						×	×			
3.					×					
4.										×

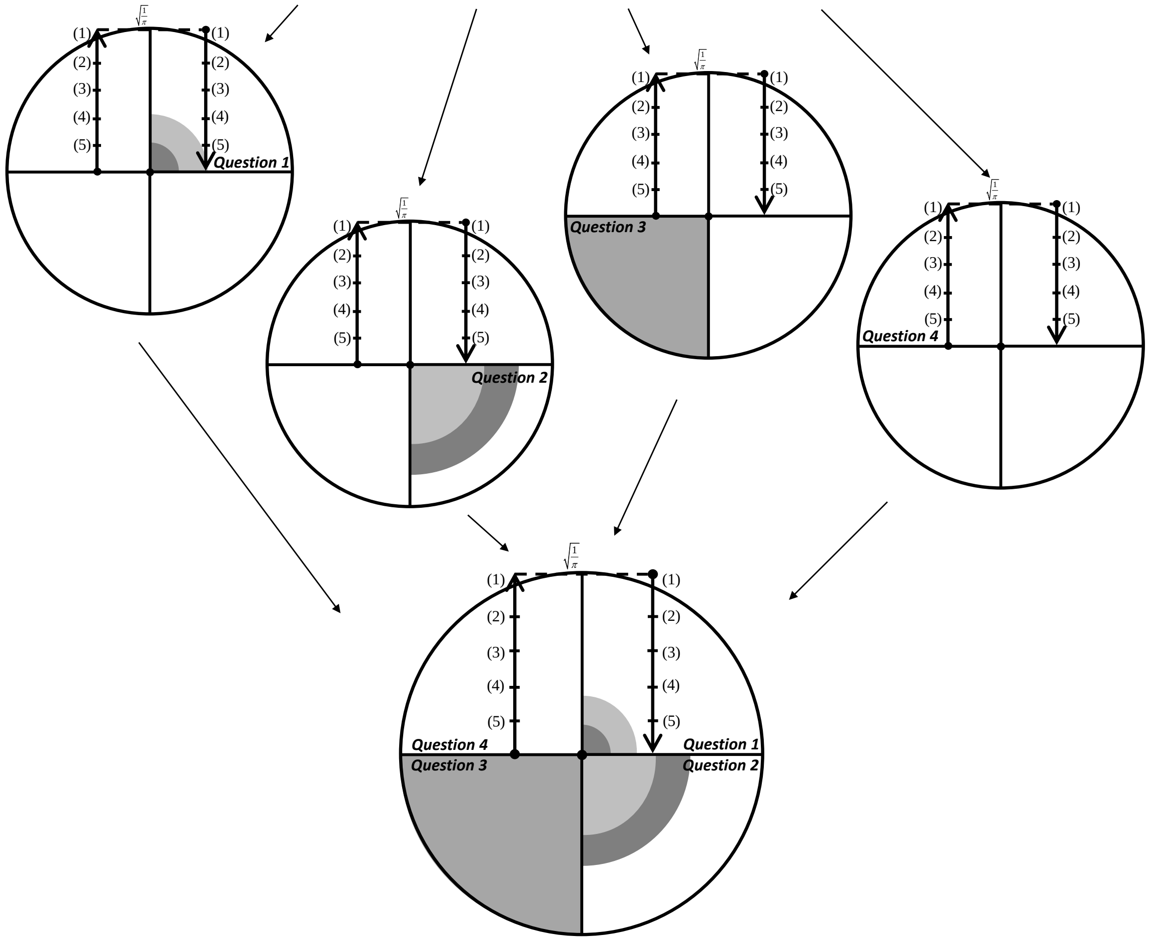


Figure 4.16: An assessment of the attained energy with respect to the answers to a generic survey.

In order to create a comprehensible process, the related energy values are displayed graphically in Figure 4.16. This figure depicts a table that represents the answers of a single person to a four-question survey as well as the different forms of energy referred to above, i.e., *exergy*, *vagueness*, and *anergy*. For instance, the answer to the first question is both (5) and (4). In fact, the used scale can assume any form. To facilitate the explanation process, consider the following scale, viz.

Extremely Agree (5), *Strongly Agree* (4), *Agree* (3), *Disagree* (2), *Strongly Disagree* (1), *Disagree* (2), *Agree* (3), *Strongly Agree* (4), *Extremely Agree* (5)

Considering this scale, the answer to the first question is extremely agree and strongly agree, in this order, which means that most of the time, the subject extremely agrees but sometimes may only strongly agree. The scale assumes this form as it allows the subject to select several answers for the same question but also to have a notion of how the answers tend to evolve. On the other hand, the answer to question 4 is marked as vagueness (or uncertainty), i.e., the user opted not to respond to such a question. This example serves to illustrate the proposed method based on a set of answers to a questionnaire.

The answers to each question are depicted graphically in the same figure. Considering that the markers on the axis correspond to any of the possible scale options, any system will behave better when entropy decreases, or goes from top to bottom (i.e., from extremely agree to strongly disagree), indicating that the system's performance decreases as the entropy increases, which is the case with respect to the first question. This graphical figure is here used to abstract from the mathematical foundations that make the ground on which the proposed method rests. The next lines will describe the used data as well as the entire process for KRR from a mathematical perspective.

4.4.2 Data Collection and Exploration

The data used in this work are based on the surveys prepared by Papadimitriou et al. [210], which were developed to capture key components affecting pedestrian walking and crossing behaviors, attitudes, motivations, and habits in urban areas, i.e., the human factors that characterize VRUs. The implemented surveys examine mobility characteristics and travel motivations, risks perception and value of time, attitudes towards walking and compliance to traffic rules, and opinion on drivers, among others.

The surveys included fifty-four questions using a five-point *Likert* scale (from ”strongly agree” to ”strongly disagree”), complemented with basic questions on demographics. Seventy-five young and middle-aged participants took the survey, with forty of them being males. The authors carried out a descriptive analysis of the answers to identify trends and patterns. It was found that most pedestrians have positive attitudes and behaviors, with a non-negligible proportion showing negative attitudes and actions. A principal component analysis was also performed, revealing that there may be three distinct dimensions of pedestrian behavior. While the first two are related to risk perception and taking, the third focuses on walking motivations [210].

To assess VRUs' human factors on a more comprehensive scale and to depict the proposed approach to KRR, three surveys were considered. The first, the Travel Motivation Five-Item-Questionnaire (TMQ-5), consists of five questions and was designed to assess the pedestrian's distribution of travel motivation:

Q1. I walk for the pleasure of it;

Q2. I walk because it is healthy;

Q3. In short trips, I prefer to walk;

Q4. I prefer taking public transportation than my car;

Q5. I walk because I have no other choice.

The second survey is entitled Crossing Roads Four-Item-Questionnaire (CRQ-4) and was designed to assess individual differences in the proneness to take risks when crossing a road. It is given in the form:

Q1. Crossing roads outside designated locations increases the risk of accident;

Q2. Crossing roads outside designated locations is wrong;

Q3. Crossing roads outside designated locations saves time;

Q4. Crossing roads outside designated locations is acceptable.

The last considered survey, the Self-Assessment and Identity Three-Item-Questionnaire (SAIQ-3), is set as follows:

Q1. I am more careful than other pedestrians;

Q2. I am less likely to be involved in a road crash than other pedestrians;

Q3. I am faster than other pedestrians.

4.4.3 Data Preparation and Pre-processing

All surveys were designed to assess VRUs general feelings about their motivations and behaviors. The scale used in this work is made upon the terms, viz.

Strongly Agree (4), *Agree* (3), *Disagree* (2), *Strongly Disagree* (1), *Disagree* (2), *Agree* (3), *Strongly Agree* (4)

A neutral term, which stands for the unknown or vagueness, is also included. The reason for the individual's answers is in regard to the query, viz.

As an individual, how much would you agree with each question of the TMQ-5, CRQ-4, and SAIQ-3?

Table 4.12 is given with the answers of a single pedestrian to the three surveys referred to above, i.e., the TMQ-5, CRQ-4, and SAIQ-3.

Table 4.12: Single pedestrian answers to the TMQ-5, CRQ-4, and SAIQ-3.

Survey	**Question**	**Scale**							
		(4)	*(3)*	*(2)*	*(1)*	*(2)*	*(3)*	*(4)*	*vagueness*
TMQ-5	Q1	X		X					
	Q2					X		X	
	Q3					X	X		
	Q4			X					
	Q5								X
CRQ-4	Q1						X		
	Q2					X		X	
	Q3			X					
	Q4		X	X					
SAIQ-3	Q1		X	X					
	Q2				X				
	Q3					X			

4.4.4 Experiments

To facilitate the demonstration process, and for the sake of presentation and simplicity, full computation details are provided only for the TMQ-5. The computational process for the remaining surveys is the same.

Let us consider the answers of a single pedestrian to each survey (Table 4.12). For instance, the answer to Q1 of the TMQ-5 was both strongly agree and disagree, in this order. This means that most of the time the subject strongly agrees that he walks for the pleasure of it. However, there are situations where he simply disagrees. The subject may be referring to situations where he is forced to walk when going to work, for example. On the other hand, the subject opted not to disclose his answer to Q5, which was marked as vague or unknown. Figure 4.17 describes the subject's answers to the TMQ-5 in terms of the different forms of energy referred to above, using circles with an unitary area value [252].

From this figure, it becomes possible to measure the respective entropic states as untainted energy in the form of best and worst case-scenarios, as shown in Table 4.13. This table presents the mathematical *formulae* to compute *exergy*, *vagueness*, and *anergy* from qualitative answers to a survey, in particular, the five questions of TMQ-5. The remaining questions of all other surveys are calculated likewise. It is now possible to set the obtained answers as a function of its entropic state. Indeed, the collected data may be structured in terms of the extent of predicate *tmq*, *crq*, and *saiq*, as follows:

$tmq: \boldsymbol{EX}ergy, \boldsymbol{VA}gueness, \boldsymbol{AN}ergy, \boldsymbol{VS}, \boldsymbol{QoI} \Rightarrow True, False$

$crq: \boldsymbol{EX}ergy, \boldsymbol{VA}gueness, \boldsymbol{AN}ergy, \boldsymbol{VS}, \boldsymbol{QoI} \Rightarrow True, False$

$saiq: \boldsymbol{EX}ergy, \boldsymbol{VA}gueness, \boldsymbol{AN}ergy, \boldsymbol{VS}, \boldsymbol{QoI} \Rightarrow True, False$

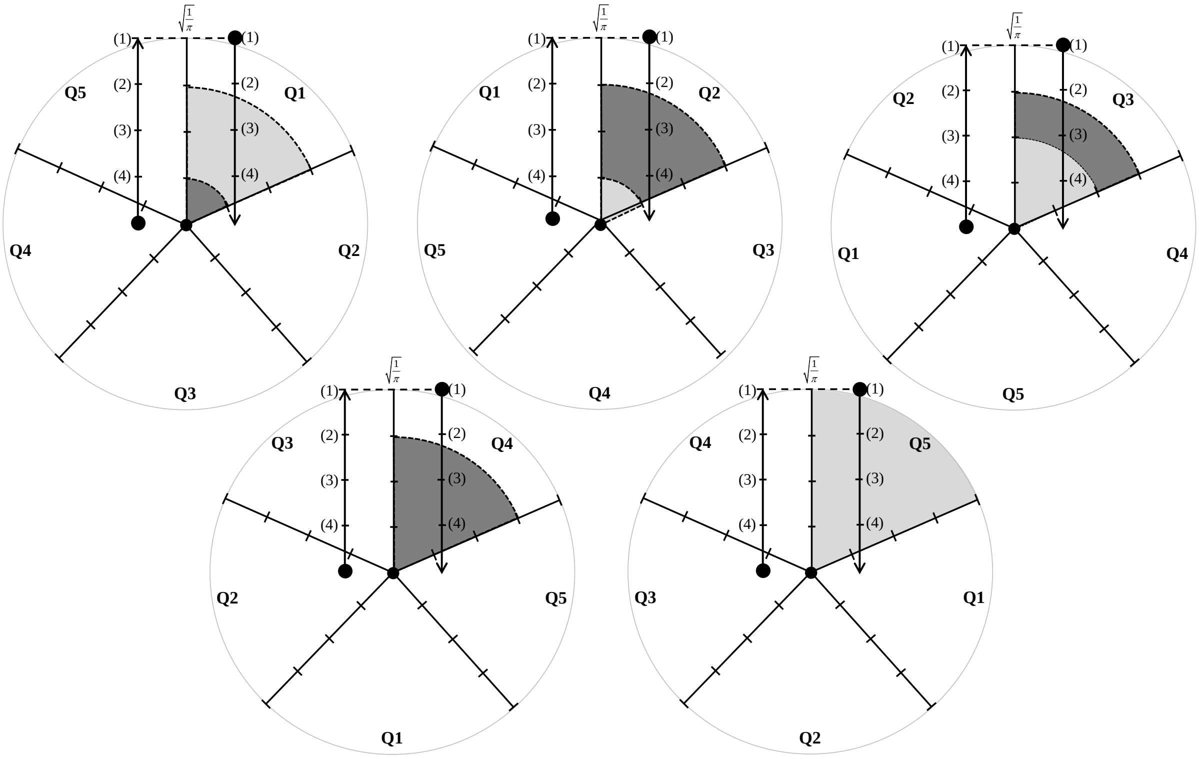


Figure 4.17: An assessment of the attained energy with respect to the answers of a pedestrian to the TMQ-5.

Where VS and QoI stand, respectively, for a VRUs safety value and the Quality-of-Information involved in such evaluation. VS is computed in the form, viz.

$$VS = \sqrt{1 - ES^2} \tag{4.1}$$

Where ES stands for an *exergy* value. All values are further normalized to range between $[0, 1]$ (Figure 4.18).

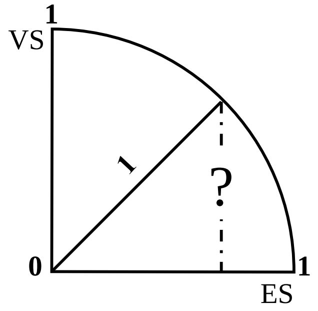


Figure 4.18: VRUs Safety evaluation.

Looking at the TMQ-5, the VS for the best-case scenario is computed as follows, viz.

$$VS_{BestCaseScenario} = \sqrt{1 - ES^2_{BestCaseScenario}} = \sqrt{1 - 0.013^2} = 0.999 \tag{4.2}$$

Table 4.13: Entropic states' evaluation for the best and worst case's scenarios for the TMQ-5.

Question	Best Case Scenario	Worst Case Scenario
Q1	$exergy_{Q1} = \frac{1}{5} \cdot \Pi \cdot r^2 \cdot \int_0^{\frac{1}{4}\sqrt{\frac{1}{\Pi}}} = 0.013$	$exergy_{Q1} = \frac{1}{5} \cdot \Pi \cdot r^2 \cdot \int_0^{\frac{3}{4}\sqrt{\frac{1}{\Pi}}} = 0.113$
	$vagueness_{Q1} = \frac{1}{5} \cdot \Pi \cdot r^2 \cdot \int_{\frac{1}{4}\sqrt{\frac{1}{\Pi}}}^{\frac{3}{4}\sqrt{\frac{1}{\Pi}}} = 0.100$	$vagueness_{Q1} = \frac{1}{5} \cdot \Pi \cdot r^2 \cdot \int_{\frac{3}{4}\sqrt{\frac{1}{\Pi}}}^{\frac{3}{4}\sqrt{\frac{1}{\Pi}}} = 0$
	$anergy_{Q1} = \frac{1}{5} \cdot \Pi \cdot r^2 \cdot \int_{\frac{1}{4}\sqrt{\frac{1}{\Pi}}}^{\sqrt{\frac{1}{\Pi}}} = 0.188$	$anergy_{Q1} = \frac{1}{5} \cdot \Pi \cdot r^2 \cdot \int_{\frac{3}{4}\sqrt{\frac{1}{\Pi}}}^{\sqrt{\frac{1}{\Pi}}} = 0.088$
Q2	$exergy_{Q2} = \frac{1}{5} \cdot \Pi \cdot r^2 \cdot \int_0^{\frac{1}{4}\sqrt{\frac{1}{\Pi}}} = 0.013$	$exergy_{Q2} = \frac{1}{5} \cdot \Pi \cdot r^2 \cdot \int_0^{\frac{3}{4}\sqrt{\frac{1}{\Pi}}} = 0.113$
	$vagueness_{Q2} = \frac{1}{5} \cdot \Pi \cdot r^2 \cdot \int_0^{\frac{1}{4}\sqrt{\frac{1}{\Pi}}} = 0.013$	$vagueness_{Q2} = \frac{1}{5} \cdot \Pi \cdot r^2 \cdot \int_{\frac{1}{4}\sqrt{\frac{1}{\Pi}}}^{\frac{1}{4}\sqrt{\frac{1}{\Pi}}} = 0$
	$anergy_{Q2} = \frac{1}{5} \cdot \Pi \cdot r^2 \cdot \int_{\frac{1}{4}\sqrt{\frac{1}{\Pi}}}^{\sqrt{\frac{1}{\Pi}}} = 0.188$	$anergy_{Q2} = \frac{1}{5} \cdot \Pi \cdot r^2 \cdot \int_{\frac{3}{4}\sqrt{\frac{1}{\Pi}}}^{\sqrt{\frac{1}{\Pi}}} = 0.088$
Q3	$exergy_{Q3} = \frac{1}{5} \cdot \Pi \cdot r^2 \cdot \int_0^{\frac{2}{4}\sqrt{\frac{1}{\Pi}}} = 0.050$	$exergy_{Q3} = \frac{1}{5} \cdot \Pi \cdot r^2 \cdot \int_0^{\frac{3}{4}\sqrt{\frac{1}{\Pi}}} = 0.113$
	$vagueness_{Q3} = \frac{1}{5} \cdot \Pi \cdot r^2 \cdot \int_0^{\frac{2}{4}\sqrt{\frac{1}{\Pi}}} = 0.050$	$vagueness_{Q3} = \frac{1}{5} \cdot \Pi \cdot r^2 \cdot \int_{\frac{3}{4}\sqrt{\frac{1}{\Pi}}}^{\frac{3}{4}\sqrt{\frac{1}{\Pi}}} = 0$
	$anergy_{Q3} = \frac{1}{5} \cdot \Pi \cdot r^2 \cdot \int_{\frac{2}{4}\sqrt{\frac{1}{\Pi}}}^{\sqrt{\frac{1}{\Pi}}} = 0.150$	$anergy_{Q3} = \frac{1}{5} \cdot \Pi \cdot r^2 \cdot \int_{\frac{3}{4}\sqrt{\frac{1}{\Pi}}}^{\sqrt{\frac{1}{\Pi}}} = 0.088$
Q4	$exergy_{Q4} = \frac{1}{5} \cdot \Pi \cdot r^2 \cdot \int_0^{\frac{3}{4}\sqrt{\frac{1}{\Pi}}} = 0.113$	$exergy_{Q4} = \frac{1}{5} \cdot \Pi \cdot r^2 \cdot \int_0^{\frac{3}{4}\sqrt{\frac{1}{\Pi}}} = 0.113$
	$vagueness_{Q4} = \frac{1}{5} \cdot \Pi \cdot r^2 \cdot \int_{\frac{3}{4}\sqrt{\frac{1}{\Pi}}}^{\frac{3}{4}\sqrt{\frac{1}{\Pi}}} = 0$	$vagueness_{Q4} = \frac{1}{5} \cdot \Pi \cdot r^2 \cdot \int_{\frac{3}{4}\sqrt{\frac{1}{\Pi}}}^{\frac{3}{4}\sqrt{\frac{1}{\Pi}}} = 0$
	$anergy_{Q4} = \frac{1}{5} \cdot \Pi \cdot r^2 \cdot \int_{\frac{3}{4}\sqrt{\frac{1}{\Pi}}}^{\sqrt{\frac{1}{\Pi}}} = 0.088$	$anergy_{Q4} = \frac{1}{5} \cdot \Pi \cdot r^2 \cdot \int_{\frac{3}{4}\sqrt{\frac{1}{\Pi}}}^{\sqrt{\frac{1}{\Pi}}} = 0.088$
Q5	$exergy_{Q5} = \frac{1}{5} \cdot \Pi \cdot r^2 \cdot \int_0^{0} = 0$	$exergy_{Q5} = \frac{1}{5} \cdot \Pi \cdot r^2 \cdot \int_0^{\sqrt{\frac{1}{\Pi}}} = 0.200$
	$vagueness_{Q5} = \frac{1}{5} \cdot \Pi \cdot r^2 \cdot \int_0^{\sqrt{\frac{1}{\Pi}}} = 0.200$	$vagueness_{Q5} = \frac{1}{5} \cdot \Pi \cdot r^2 \cdot \int_{\sqrt{\frac{1}{\Pi}}}^{\sqrt{\frac{1}{\Pi}}} = 0$
	$anergy_{Q5} = \frac{1}{5} \cdot \Pi \cdot r^2 \cdot \int_0^{\sqrt{\frac{1}{\Pi}}} = 0.200$	$anergy_{Q5} = \frac{1}{5} \cdot \Pi \cdot r^2 \cdot \int_{\sqrt{\frac{1}{\Pi}}}^{\sqrt{\frac{1}{\Pi}}} = 0$

On the other hand, the Quality-of-Information for the best-case scenario is computed as follows, viz.

$$QoI_{BestCaseScenario} = 1 - ES_{BestCaseScenario} = 1 - 0.013 = 0.987 \tag{4.3}$$

Table 4.14 presents the predicates extents' values obtained for each answer to the TMQ-5, CRQ-4, and SAIQ-3 surveys.

Figure 4.19 provides a graphical description of the energy values achieved for the five questions of TMQ-5 in regard to the computed *exergy*, *vagueness*, and *anergy*, respectively.

Table 4.14: Predicate's extension obtained from the answers of a single pedestrian to the questions of the TMQ-5, CRQ-4, and SAIQ-3. BCS stands for best-case scenario. WCS stands for worst-case scenario.

Survey	**EX**^{BCS}	**VA**^{BCS}	**AN**^{BCS}	**VS**^{BCS}	**QoI**^{BCS}	**EX**^{WCS}	**VA**^{WCS}	**AN**^{WCS}	**VS**^{WCS}	**QoI**^{WCS}
TMQ-5 Q1	0.013	0.100	0.188	0.999	0.987	0.113	0	0.088	0.993	0.887
TMQ-5 Q2	0.013	0.013	0.188	0.999	0.987	0.113	0	0.088	0.993	0.887
TMQ-5 Q3	0.050	0.050	0.150	0.999	0.995	0.113	0	0.088	0.993	0.887
TMQ-5 Q4	0.113	0	0.088	0.993	0.887	0.113	0	0.088	0.993	0.887
TMQ-5 Q5	0	0.200	0.200	1	1	0.200	0	0	0.979	0.800
CRQ-4 Q1	0.063	0	0.188	0.998	0.937	0.063	0	0.188	0.998	0.937
CRQ-4 Q2	0.016	0.016	0.234	0.999	0.984	0.141	0	0.109	0.990	0.859
CRQ-4 Q3	0.141	0	0.109	0.990	0.859	0.141	0	0.109	0.990	0.859
CRQ-4 Q4	0.063	0.063	0.188	0.998	0.937	0.141	0	0.109	0.990	0.859
SAIQ-3 Q1	0.083	0.104	0.250	0.996	0.916	0.188	0	0.146	0.982	0.812
SAIQ-3 Q2	0.333	0	0	0.942	0.667	0.333	0	0	0.942	0.667
SAIQ-3 Q3	0.188	0	0.146	0.982	0.812	0.188	0	0.146	0.982	0.812

Finally, considering the best-case scenario, the extent to the tmq, crq, and $saiq$'s predicates can be set as a logical program, viz.

$\neg\, tmq(EX, VA, AN, VS, QoI) \Leftarrow not\ tmq(EX, VA, AN, VS, QoI),\ not\ except_{tmq}(EX, VA, AN, VS, QoI).$
$tmq(0.013, 0.100, 0.188, 0.999, 0.987).$
$tmq(0.013, 0.013, 0.188, 0.999, 0.987).$
$tmq(0.050, 0.050, 0.150, 0.999, 0.995).$
$tmq(0.113, 0, 0.088, 0.993, 0.887).$
$tmq(0, 0.200, 0.200, 1, 1).$

$\neg\, crq(EX, VA, AN, VS, QoI) \Leftarrow not\ crq(EX, VA, AN, VS, QoI),\ not\ except_{crq}(EX, VA, AN, VS, QoI).$
$crq(0.063, 0, 0.188, 0.998, 0.937).$
$crq(0.016, 0.016, 0.234, 0.999, 0.984).$
$crq(0.141, 0, 0.109, 0.990, 0.859).$
$crq(0.063, 0.063, 0.188, 0.998, 0.937).$

$\neg\, saiq(EX, VA, AN, VS, QoI) \Leftarrow not\ saiq(EX, VA, AN, VS, QoI),\ not\ except_{saiq}(EX, VA, AN, VS, QoI).$
$saiq(0.084, 0.104, 0.250, 0.996, 0.916).$
$saiq(0.333, 0, 0, 0.942, 0.667).$
$saiq(0.188, 0, 0.146, 0.982, 0.812).$

One can then take advantage of such predicates to define formal reasoning procedures over a body of explicit knowledge. For instance, these data can be used to train ANNs in order to get on the fly evaluations

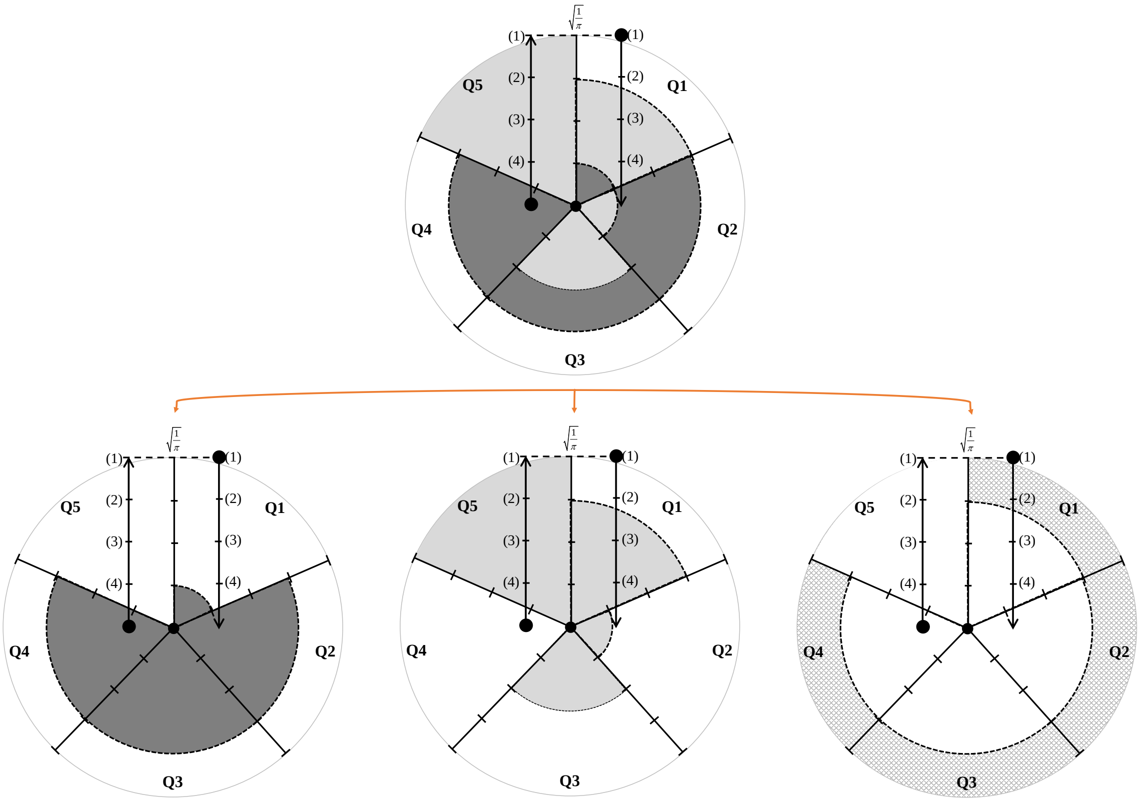


Figure 4.19: A graphical representation of the energy (in terms of *exergy*, *vagueness*, and *anergy*) achieved for the TMQ-5.

of VRUs' safety and the corresponding quality-of-information. Figure 4.20 depicts an abstract view of an ANNs' architecture to compute such features.

4.4.5 Results and Discussion

This meta-analytic study focused on the human factors that characterize VRUs and how their perception of such factors may contribute to their safety, under a well-defined setting. In other words, taking, as input to ANNs, a pedestrian's answers in terms of the extensions of the predicates referred to above, the network will return a predictive evaluation of the subject's safety. This can then be used by municipalities or the industry to propose new measures to increase and improve the performance of cities, thereby improving road safety. It is also worth highlighting the ability of the proposed method to handle unknown data as well as uncertainty in the subject's responses using a thermodynamics' view as a process of energy devaluation. It also allows one to build a quantitative dataset from a set of qualitative responses to surveys. This sets the answer to the elicited research question (RQ4.3).

As future work, and considering how social factors may shape the pedestrian perception of the city's environment, the intention is to look at ways to figure out the environment according to the pedestrian view. For that, new surveys must be conceived and distributed among a new and wider population. More studies are also expected to emerge to validate the theoretical and mathematical foundations of the proposed method.

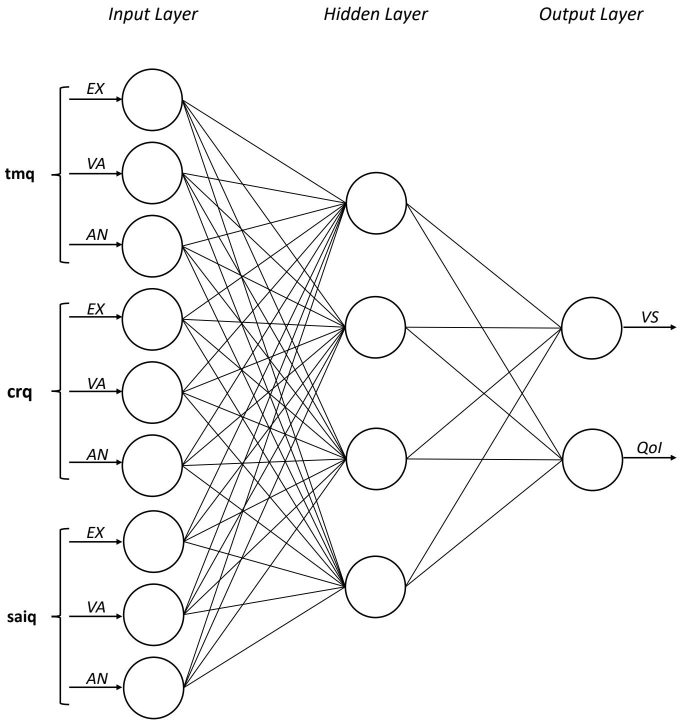


Figure 4.20: An abstract view of the ANN's architecture.

4.5 Summary

The studies and experiments detailed in this chapter aim to fulfill Task 2, focusing on the third C of what we call the 3-C, i.e., *City*, *Crowd*, and ***Citizen*** sensing. The goal is to establish the ability to sense individuals in order to include each person as an active, proactive, and reactive element of the city's ecosystem. To achieve such a goal we are required to promote the citizen sensor, offering citizens the ability to sense themselves, others, and the environment where they stand.

Three distinct experiments were carried out. The first focused on conceiving, tuning, and evaluating several candidate ML models for personality assessment, where the subject, instead of rating forty adjectives, is only required to select a set of adjectives that describe him more accurately. Two distinct architectures were conceived. One focuses on classifying the bin (low/average/high) of each personality trait, while the other provides an exact score for each trait. While the latter may be used to obtain a detailed score, the former can be useful in cases where the general qualification of each trait is more important than its specific score. The conceived models reduce the complexity and the time it takes to perform a test, making it more attractive and easier to implement by technological platforms.

A second experiment was conducted to conceive and evaluate distinct ML and DL models to predict boredom when using a smartphone. To be as less intrusive as possible, the conceived models only use features that are made available by the smartphone's sensors, discarding any type of biometric data. A mobile application was developed for data collection. The best candidate model presented promising results in regard to boredom detection in smartphones, allowing the conception of applications that are able to perceive when a user is bored on his smartphone.

Finally, a third experiment was focused on presenting a meta-analytic study for KRR centered on logic programming, which establishes a formal logical inference engine that can be complemented with, for example, an ANNs line to computation. This study was grounded on surveys that were developed to capture key components affecting pedestrian walking and crossing behaviors, attitudes, motivations, and habits. The proposed method was able to successfully handle unknown data as well as uncertainty in the subject's responses, allowing one to build a quantitative dataset from a set of qualitative responses to surveys.

To aid proving the research hypothesis and validate the conceived solutions, the conducted experiments focused on providing an answer to ***Research Question 2****: With the use of Affective Computing techniques and mobile devices, is it possible to accurately measure boredom and the personality of subjects in real-time with a non-invasive, non-intrusive approach?*. In fact, the conducted experiments open promising prospects in regard to the quantification of boredom when using the smartphone. The best candidate model achieved an overall accuracy higher than 70%, which, given the limitations of the dataset, can be interpreted as an encouraging result. On the other hand, we have now the ability to assess one's personality using a simple "game of words" where subjects are only required to select a set of adjectives that describe them the most. This opens new avenues to easily and quickly quantify personality traits without the need to conduct full personality assessment tests. The work performed in this chapter closes **Task 2** as well as the **3-C**.

The work here displayed gave origin to multiple scientific papers ([43, 58, 252, 261–263]) as well as to the participation in several scientific events. It was performed in collaboration with researchers from the ALGORITMI Centre, Braga, Portugal; the Research Group on Agent-Based, Social, and Interdisciplinary Applications, from the Complutense University of Madrid, Spain; the BISITE Research Group, from the University of Salamanca, Spain; the Chemistry Department of the University of Évora, Portugal; and the Polytechnic Institute of Viana do Castelo, Portugal.

Chapter 5

KnowLedger - A Multi-Agent System Blockchain for Smart Cities Data

Smart cities, to emerge as such, are required to have the ability to create knowledge and reason upon data describing its environment. However, this introduces novel problems related to the management of inordinate amounts of data of different types and sources. It is important to guarantee that such data may be committed by any actor, being, at the same time, freely available to anyone that aims to derive knowledge from it.

5.1 Rationale

During the last decades, data has been assuming an increased importance in the technological progress of our society. In its essence, data can refer to an individual, a group of individuals, an environment, a city, and so many other distinct sources that no lines would be enough. Formally, the Cambridge English Dictionary defines data as "*information, especially facts or numbers, collected to be examined and considered and used to help with making decisions*". While data and information are considered to be two different levels, it is in fact true that data are the source of information, with the goal being to use it to make, or help making, decisions. Better decisions.

With the progress of society in mind, one goal that has been growing in importance is related to the promotion of Smart Cities, i.e., cities that, per se, have the ability to create knowledge and reason based on data describing its environment. A crucial step towards such goal relates to the development of new sources of data from where one could derive knowledge and support the decision making process. Additionally, the availability of data, in particular historical data, assumes an increased importance for contemporary AI, in particular for ML. It should be noted that even though several platforms offer, freely, recent data, it is extremely hard to get access to historical data. Many times, significant fees are requested or the data are simply not available because no one is collecting it.

Two important points remain to be declared, i.e., how to store data in a way that it may be available to all interested parties, and how to promote accurate and reliable data. The relevance of these points is easily explained: firstly, data gathered about the environment is not linked to any individual and, therefore, should be available to every party who is interested in using it without the need to collect it repeatedly; secondly, anyone who desires to provide new data, whether it concerns the person in itself or any other aspect of the environment, should be capable of doing so in a manner that it becomes available to the community; and, thirdly, as soon as everyone gets allowed to contribute, mechanisms to control the quality, in terms of accuracy and reliability, of the provided data must arise. Hence, an immutable, distributed, decentralized solution, where everyone and everything could contribute to, is easier to adopt since centralized solutions may raise data integrity and trust issues. This comes from the observation that peer-to-peer solutions tend to guarantee equal rights to all actors. Considering all the characteristics of the problem, blockchain emerges a viable solution. Even though blockchains by themselves do not provide direct incentives to promote accurate and reliable data, they provide the possibility of tracking, by key, the actors that committed such data.

With all this in mind, this study presents *KnowLedger*, a MAS blockchain to store structured data. A set of key features are worth highlighting. Firstly, a protocol, entitled as PoC, was established to award those who commit accurate and reliable data. In a perfect world, data would be, indeed, accurate and the PoC would become redundant. However, there is always the possibility of malicious actors spamming *KnowLedger* with incorrect data. Hence, the PoC emerged to detect such situations and provide a mechanism that allows the network to detect and act to prevent such actors from cluttering the blockchain. In addition, it should be noted that *KnowLedger* is not a distributed storage solution or file system. Instead, it is a platform for well-structured, pre-processed data. Secondly, the PoC enables a scoring environment for a system comprised of a multitude of agents, where each one is encouraged to provide accurate data to maximize their PoC score. In addition, besides collecting data, intelligent agents are also responsible for building, maintaining, and operating the ledger. And, thirdly, *KnowLedger* follows a multi-chain architecture to maintain independent sets of data, with childchains representing geographical locations and datachains isolating different data types.

In regard to the problem described in the previous lines, the following research questions were elicited:

RQ5.1. Is blockchain a viable solution to store structured data?

RQ5.2. Can intelligent agents work, and maintain, a blockchain?

RQ5.3. Is it beneficial and logically viable to use a multi-chain architecture comprising a childchain for each city?

RQ5.4. Is it viable to implement a metric to successfully reward those who commit accurate and reliable data?

5.2 State of the Art

Blockchain is, in its core, a data structure, a technology, and a protocol, all at the same time. It was first proposed in 1991, by Haber and Stornetta [264], as a method for secure time-stamping of digital documents, to eradicate the possibility of adulterate documents. However, it only achieved great success with the emergence of the Bitcoin [67]. As the name implies, a blockchain is essentially a chain of blocks that contain data, with the first block of the chain being called the genesis block. Blocks may have different structures but, fundamentally, each block contains an index, a hash, a timestamp, the hash of the previous block, a data structure, and a nonce, with all these elements being part of a block's hash. To create the chain, each block contains the hash of the previous block and, so, as newer blocks are chained after it, the work to change a block would include redoing all the blocks after it, in order to recalculate all hashes of all subsequent blocks.

Some more terms require definition. First of all, let us define the meaning of mining the blockchain, i.e., the act of producing new blocks. One should bear in mind that a blockchain aims to achieve consensus over a decentralized network to guarantee that everyone agrees in the order of transactions. Hence, in simple terms, the greater the number of blocks the stronger the blockchain. Using bitcoin as example, the network is expected to produce, roughly, one block every ten minutes. This creates persistence and updates to always represent the latest state of the ledger. It is also crucial to ensure the validity of blocks and, consequently, of its transactions. It is also important to note that, for cryptocurrency-based blockchains, the order in which miners include transactions into the block is, for obvious reasons, of the highest importance. Still related to mining, the Proof-of-Work (PoW) is used to make a block's creation computationally hard in order to prevent malicious miners from redoing the entire blockchain in their favor. Using SHA256, an unpredictable pseudo-random function, the only way to create a valid hash is just by trial and error, with the nonce being used and updated to create new hashes [68]. The miner that finds the correct hash receives a pre-established value of some token or currency. New concepts have emerged to compete with PoW in order to provide consensus mechanisms and avoid double spend issues. Examples are the Proof-of-Stake [265], Proof-of-Capacity [265], or Proof-of-Importance [266]. Indeed, this last approach, used by NEM's blockchain [266], maintains peer reputations using the Eigentrust++ algorithm [267]. This approach, however, focuses more on trust between peers based on their interactions than the values they supply in the form of transactions.

Over time, it becomes heavy to store all blocks and each transaction of a blockchain (Bitcoin's blockchain size is near 280 gigabytes by May 2020). Hence, an important scalability feature consists in storing the root hash of a data structure entitled as Merkle tree [268], which, in turn, stores all the transactions present in that given block. Such a tree is a binary one, with a large number of leaf nodes at the bottom of the tree containing the corresponding data. Each intermediate node is the hash of its two children, with the single root node (the top of the tree) being, again, the hash of its two children. Obviously, the Merkle Tree hash, represented by the single root node, is part of a block's hash. Hence, if a malicious actor swaps a fake transaction in an old block this will cause a change in the corresponding node in the Merkle Tree, which will cause a change in all hashes of the tree, including the root one, which, in turn, will lead to a completely different hash of the block. This also allows for new protocols to emerge, as it is the case of the Simplified Payment Verification, which allows one to download the block headers and the branches that are only associated with the relevant transactions [68].

Soon researchers understood the applicability of this technology in different domains. One example is the effective development of smart contracts by Ethereum [68]. In Ethereum's case, the blockchain has a built-in Turing-complete programming language, allowing one to write any smart contract or transaction type that can be mathematically defined. Essentially, a smart contract designates a cryptographic tool that contains a certain value and only unlocks upon certain conditions being met. In fact, the term "contract" should not be read from the perspective of something that is to be fulfilled. Instead, it is to be understood as an autonomous piece of software that executes when called by a transaction/message. Other key aspects of Ethereum's blockchain include *gas*, the fundamental unit of computation, and the *startgas* and *gasprice* fields, which are used to prevent infinite loops and computational wastage [68]. Decentralized applications, also known as *dapps*, on the other hand, are applications built on a blockchain that combines a smart contract and a user interface, with its backend code running on a decentralized peer-to-peer network. However, current *dapps* still carry significant risks related to maintenance (harder to maintain and update since anything that is published to the blockchain is difficult to modify), performance overhead (scaling is hard), network congestion, user experience, and centralization (solutions built on top of a base layer might end up as a centralized service, eliminating all the main advantages of a blockchain) [269].

The approach proposed in this study focuses on blockchains for structured pre-processed data collected by sensors, people, institutions, or by intelligent agents. Data about the environment, the traffic flow, the weather, or a Smart City. Indeed, regarding the network itself, blockchains enjoy desirable characteristics to organize the collected data due to their immutable and referential nature. They effectively track all actors that inscribe data into the blockchain, coupling each identity to the data it published. They suffer, however, from deficient performance in querying. Hence, with this problem in mind, Muzammal et al. [270] proposed a hybrid system, entitled as ChainSQL, comprised of a blockchain and a distributed database. They opted to store transactions in the blockchain with the actual data being stored in the database. However, such a system, even though with its merits, seems to add unnecessary complexity and computational weight to a system that should be straightforward. A different approach to the problem could focus on the use of explicit indexing structures or even the implementation of a multi-chain architecture. On the other hand, in [271], the authors address the issue of data metadata, proposing a decentralized cloud data provenance architecture using blockchain. The goal is to provide tamper-proof records, enhancing privacy and availability of data. The referred study uses a blockchain to store information that includes the file name, the owner, and the action the file suffered (removal, read, or write, among others). It, however, lacks on providing enough details on the conception of the blockchain and the used protocol. More examples focused on data provenance using blockchain and smart contracts can be found in [272, 273].

It is worth noting that the number of studies that address the use of blockchain in association with MAS is limited. In fact, in 2018, Calvaresi et al. performed a systematic literature review on solutions reconciling both MAS and blockchain, identifying that most approaches are just now emerging, being mainly focused on e-governance and trust management [274]. Moreover, from those few that are now expanding this area, most of them have only produced conceptual work [274]. Examples can be found in the works of Ferrer [275] and Shermin [276]. The few prototypes that have been developed are mainly based on plug-and-play blockchains or already developed frameworks. For example, Kiyomoto et al. designed and conceived a platform for anonymized dataset trading, with peers and consensus-based blockchain mechanisms [277]. In their platform, each peer acts as a data broker and data receiver, or as a verifier for the blockchain in a data transfer transaction. The prototype was built over an open-source blockchain library, namely Hyperledger Fabric version 0.7.0. Another example is the one of Kvaternik et al. who demonstrated a simulated scenario describing the implementation of a blockchain-based transactive energy system [278]. They opted to deploy Smart Contracts as an Ethereum contract on a private blockchain. A different study focused on the notion of smart contracts to deploy fully decentralized, trustworthy coordination [279]. Such study provides a conceptual and technical view on the feasibility of implementing a LINDA-like tuple-based coordination services on top of Ethereum, i.e., blockchain-based coordination within MAS. The goal is, again, to take advantage of blockchain features such as trust, secured communications, data consistency, ordering of events, accountability, and identity management [279]. The same authors, in [280], further extend their study, discussing autonomy over smart contracts, which are to be understood as software agents. The authors highlight and discuss current limitations of state-of-art blockchains in supporting MAS design and implementation, which goes in accordance to our findings, being this the main reason why $KnowLedger$ was designed and conceived from scratch.

To conclude, it is worth noting some conclusions from the literature, particularly the one stating that existing smart contracts and $dapps$ have serious limitations and peculiarities, increasing the difficulty of implementing domain logic. Indeed, interesting conclusions emerge from the analysis of studies focused on the domains of MAS and blockchains. The first is that there is a serious lack of detail on the implementation of the intelligent agents. The second is that the option tends to go towards the use of libraries such as Hyperledger, Tendermint, or Ethereum, creating inflexibility and endless dependency.

5.3 Theoretical Foundations

The goal of this study is to conduct experiments and build a prototype, $KnowLedger$, to understand the effectiveness of blockchain as a solution to store structured data for Smart Cities. The next lines describe the ground truth on which the proposed solution rests.

5.3.1 The Problem

During these last decades, the amount and diversity of data that are being collected by individuals, the academia, and companies have grown exponentially. Indeed, new sources of data from where one could derive information are constantly emerging. Such data can then be used to produce knowledge and support the decision-maker. Therefore, the availability of data, in particular historical data, assumes an increased importance.

A data collection process may take a long time to complete. Indeed, it may never be completed. Therefore, it is usual to see applications and software artifacts constantly querying sensors and APIs in open data streams scenarios. Indeed, currently, there are available many services and platforms that offer, freely, recent data. There is, however, a major constraint when the goal is to access and collect historical features. Many times significant fees are requested. In other cases, data are simply not available because no one is storing it. As an example, to develop ML models for traffic flow forecasting one is required to have data for, at least, an entire year or, in the worst case, a few months. Two options emerge. The first is to pay, usually high fees, to have access to such data. The second is to develop a software artifact, as *The Collector*, which queries sensors regularly and, then, wait the necessary amount of time until enough historical data has been gathered.

Smart cities, to emerge as such, are required to have the ability to create knowledge and reason upon data describing its environment. From temperature and pollution data to traffic flow, data may be originated from a multitude of sources, distributed through several devices and sensors, usually referred to as the IoT. It may even be the case that the same data are being collected and stored by different entities. This could be prevented with a platform from where anyone could commit and read data. Indeed, on the one hand, historical data would be available to anyone who desired to use it without the need to have collected or paid for it. On the other, anyone would be able to contribute and make available structured data to the community.

5.3.2 Philosophy

First of all, it becomes necessary to put into words the four key principles being followed by *KnowLedger*:

- *Simplicity*: *KnowLedger* is expected to be as simple as possible and easily understandable to any developer who desires to contribute to the platform. The platform is being developed using Kotlin as programming language. New features and optimizations are not expected to add significant complexity unless being unquestionably needed;
- *Openness*: on the one hand, *KnowLedger* is an open-source platform, with its code being available on GitHub[1] under a MIT license. Anyone may contribute with bug-fixing, optimizations, or new features. On the other, *KnowLedger* is completely open to anyone who desires to commit or read data from it. A set of public and private keys is the only requirement;
- *Decorum*: to encourage accurate and reliable data, *KnowLedger* makes use of mechanisms to empower good actors and identify malicious ones. Hence, those who contribute to the platform are expected to provide accurate data and not spam *KnowLedger* with incorrect data;

[1] github.com/brunofmf/mas-blockchain-main

- *Non-censorship*: no actor should see denied the possibility of reading data. In addition, no actor should see denied, by its peers, the possibility of committing data, except for those showing malicious behavior, which will be ignoring the *Decorum* principle and will see their PoC reduced.

5.3.3 The Blockchain and its Multi-Chain Architecture

A blockchain is, in its essence, an immutable public ledger that is distributed among networked nodes. The ledger holds a perpetual footprint of interactions and transactions. A permission-less blockchain allows anyone to get involved on the network, having no central authority to edit and control the ledger, shut down the network, or change the used protocols, being, instead, focused on consensus protocols.

In Smart City scenarios, considering the existence of high-velocity data streams, we focused our attention on the implementation of a multi-chain architecture for scalable performance. In its essence, each *childchain* is tied to a particular city, or geographical location, and works as an aggregator of multiple inner parallel ones, entitled as *datachains*, each one specific to a single type of data. These datachains promote the possibility of working only the desired chain(s) of data, thus reducing the computational load of the entire solution. Structurally, each datachain is similar to existing cryptocurrency-based blockchains being tightly integrated with the main parent chain, which holds a higher level validation of its children. An even bigger aggregator blockchain is built on top of these by coupling multiple of these blockchains as childchains of the main one. In this way, it would be possible to ensure no two childchains ever represent the same geographical location nor to instantiate further nested childchains.

Cross-chains transactions are unnecessary and do not happen at either level since each childchain is deeply connected to a specific geographical location and all inner datachains must be wholly independent of each other. Conceptually, each childchain is tied to a single geographical location to have a clear segregation between locations, allowing a better management of the computational load of *KnowLedger*. The goal is to have perpendicular chains to the main one and not parallel ones to a primary blockchain, as in the case of sidechains or pegged sidechains, which are independent to the main one thus needing their own network security and block processing features, bringing an additional cost when handling and maintaining them. Figure 5.1 provides a pictorial view of the conceived blockchain.

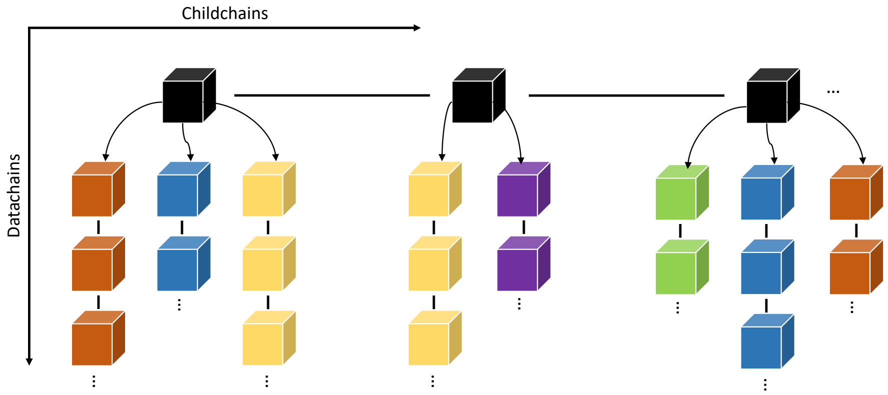


Figure 5.1: *KnowLedger*'s blockchain representation.

Finally, data stored in *KnowLedger* are available to all actors that participate in the blockchain. No actor is required to commit data to the blockchain neither to mine its blocks. It only has to be part of the network as an actor that periodically runs a synchronization behavior to have the most up-to-date version of the blockchain, its blocks, and each transaction of data within each block, thus strengthening the blockchain.

5.3.4 The Multi-Agent System

Few studies have already engaged on studying the combination of the blockchain technology with AI [274, 281]. As stated before, solutions reconciling both MAS and the blockchain technology are now emerging, being focused in the use of platforms such as *Hyperledger*, *Tendermint*, or Ethereum, creating inflexibility and dependency.

KnowLedger's proposal of a MAS blockchain is related to the ability of having multiple intelligent agents interacting with each other and with the blockchain. Each agent perceives the environment it is working on and takes action to maximize its reward. In *KnowLedger*'s case, agents try to maximize the confidence, or scoring, implemented by *KnowLedger*'s PoC, which is detailed in later lines. The intelligent agents also carry important characteristics that help strengthen *KnowLedger*, in particular, their autonomy and local view of the problem. The first relates to the fact that agents are autonomous and self-aware in regard to the task in hands. The latter comprises the fact that each agent is not responsible for the entire workload and has, instead, a specific task to complete, with the whole MAS in itself comprising a set of intelligent agents that are able to tackle the bigger picture and manifest self-organization.

Intelligent agents are to be understood as artifacts of software that have the ability to work autonomously, sharing ontologies, and interaction protocols [282] that, when coupled with blockchains, lead towards the conception of fully decentralized, immutable, autonomous systems.

5.3.5 Proof-of-Confidence

As soon as everyone is allowed to contribute, mechanisms to control the quality of the committed data, in terms of value and reliability, must arise. First and foremost, it is important to clarify what is being, indeed, committed by the agents. *KnowLedger* is not a distributed storage or file system such as the *Interplanetary File System*. Instead, it is a platform for well-structured, pre-processed data. Therefore, any agent participating in *KnowLedger* must comply with the following scoring contract:

- Any data that populates *KnowLedger* must allow for comparison against a previous transaction of the same type. As such, data must be categorized by type (such as temperature, humidity, or traffic data) and this category must allow partial ordering;

- The comparison must result in a ratio of the previous value and the current value. These values depend on the type of data under measurement;

- Data are geographically tagged either directly or indirectly, through the corresponding childchain.

The scoring approach followed by *KnowLedger* ensures block mining equitably to all participating agents, while, at the same time, promotes decorum from all actors. The PoC is calculated by taking into consideration relative measurements, with an agent's goal being to maximize this score, i.e., maximize the confidence others have on him by providing accurate and reliable data. Accuracy and reliability may be set in terms of data that tends to follow a pattern when committed by several actors during a period of time. This comes from the observation that physical data tends toward homogeneity for reduced time-frames. Taking, for example, temperature readings, it is highly probable that within a small region, and considering a time-frame of a few seconds to minutes, temperature readings will not differ significantly. In addition, instead of maintaining a complex scoring system, such as PageRank-based algorithms [283], a simple short-term memory formula was developed to ease resource usage in score attribution. All the mathematical foundations of *KnowLedger*'s PoC are provided in subsequent lines.

PoC enables a scoring environment where agents compete for the highest score. These scores are attributed to the agents that supply the transactions through the block's coinbase transaction. Indeed, in *KnowLedger* each transaction is recorded such that it references the latest known total score for such agent prior to the new block [284]. Thus, the main difference between scoring and traditional mined cryptocurrency comes from the fact that there is no reliance on a fixed lump-sum [67]. Instead, the coinbase is incrementally constructed, adding scores to each agent that adds transactions to the block.

5.4 Design and Conception

The next lines describe the design and conception of *KnowLedger*, a MAS blockchain to store structured data collected from different actors. The developed blockchain and MAS are also explained in detail. Data used in this study were made available by *The Collector*.

5.4.1 KnowLedger's Multi-Agent System

The few literature focused on coupling MAS with blockchains mainly targets the application of the latter in the framework of the former [274]. However, the opposite should not be dismissed, i.e., the application of a MAS for the benefit of blockchains. One could, for example, take advantage of intelligent agents not only for data collection but also for mining the blocks and optimize their work based on the committed data and the respective reward.

For simplicity reasons, the MAS is described before the blockchain itself. Hence, let us consider a MAS comprising two main autonomous entities, with distinct behaviors. On the one hand, we have *Slave Agents*, which are responsible for collecting and processing the data that are gathered from heterogeneous sources (*data capture* behavior). Architecturally, each slave agent manages its own data sources, being able to instantiate new slave agents that are, for example, responsible for a specific data type. Each slave agent should manage the timings and priorities that are given to each source in order to maximize the quality of data. Data representation is completely configurable, provided that it complies with the established scoring contract. These data will then originate transactions to be committed into the blockchain, i.e., transactions that will be inserted into the block under construction and that is to be added to the blockchain. This is where the second entity emerges, the *Ledger Agents*, which are the ones responsible for building and maintaining the ledger. They receive processed data from slave ones, create transactions, and then mine the blocks (*mining* behavior). To ensure that transactions are not lost, the agent that generates them must keep a transaction pool, ordered by transaction date, where it tracks which transactions it issued and are not yet present in a committed block. Moreover, to guarantee that the transactions will eventually populate a block, there must be a periodic diffusion of all not yet committed transactions (*synchronization* behavior). Block headers are also propagated between agents when mining results in a nonce that fulfills the difficulty requirements of the block.

Agents are completely autonomous, both when performing their work as well as when communicating with their peers. In addition, obvious benefits arise from having the workload distributed over distinct entities. Slave agents can focus on capturing data, processing it, and maximizing not only the number of sources but mainly the quality of the captured information. There is no need for such agents to be aware of the blockchain. Indeed, only ledger agents are required to keep a copy of the ledger. These agents are the ones responsible for handling the ledger and are exclusively focused on such task. To interact with the blockchain, all agents must have a cryptographically secure identity.

Figure 5.2 depicts the conceived MAS, with two distinct types of agents, slave and ledger ones, comprising a smart hub. Within a smart hub, slave agents are responsible for collecting data and sharing it with the ledger agent. In turn, this agent is responsible for having an up-to-date version of the blockchain and the transaction pool. For that, it should make use of synchronization behaviors, with well-defined ontologies, in order to get missing blocks and share new transactions of data that are to be inserted in the transaction pool of its peers. It should also make use of mining behaviors to mine new blocks.

KnowLedger focuses on PoC to favor those who supply accurate, frequent, and reliable data. However, it does not directly guarantee that malicious agents are barred from cluttering the blockchain with arbitrary data. It is, however, possible to detect and flag such agents due to their naturally low scoring, either through their cumulative score or their median score per transaction. It is also important to remember that, to interact with the blockchain, all agents must have a cryptographically secure identity. Hence, agents that exhibit continuous malicious behavior may have their keys flagged by the remaining agent peers due to their low score. Therefore, ledger agents may, or may not, implement a blacklist to keep track of the agents showing malicious behavior, i.e., agents that have a continuous low score. Each agent has autonomy in regard to the decision of implementing a blacklist. It is also left to each agent to choose if, and when, to remove the malicious agents from his blacklist. Blacklisted malicious agents should not have their transactions included in the transaction pool neither in the candidate block of any ledger agent that acknowledges them in their blacklist. Hence, it is left to the network to control the treatment of malicious agents and to decide whether to apply a temporary lock or a full-time one. The more nodes acknowledge a malicious agent the stronger the lock.

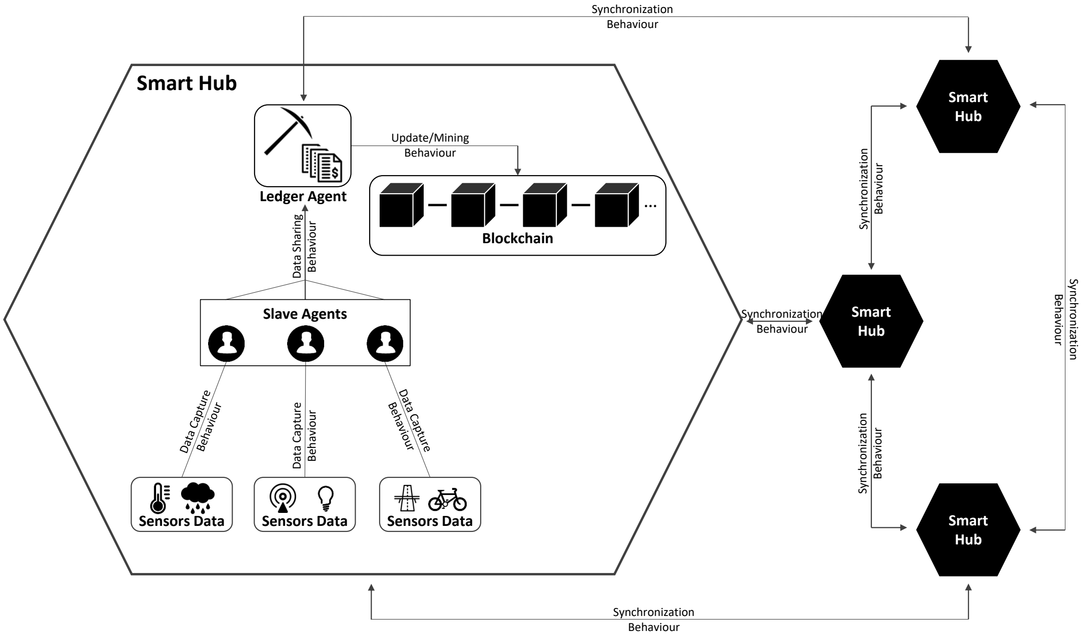


Figure 5.2: The MAS architecture, comprising Smart Hubs composed of a ledger agent and a set of slave agents.

5.4.2 KnowLedger's Blockchain

The public collection of data from arbitrary sources provided by arbitrary actors presents interesting challenges. On one hand, there is the need to have an immutable, distributed, decentralized approach, without a central authority controlling all the collected data and its access. On the other hand, it is imperative to have mechanisms to control the quality of data and favor those who supply accurate and reliable data. The first challenge is easily addressed by the blockchain itself. However, blockchains by themselves do not provide incentives to promote accurate and reliable data that can be used, for example, to train and tune ML models. This deteriorates even more with the huge number of devices with unknown origins and behaviors. Nonetheless, blockchains allow the management of immutable ledgers that track the activities of its actors closely. This grants the possibility of encouraging and empowering good actors, i.e., those who supply accurate, reliable, and frequent data. Hence, as it will be demonstrated in the next lines, we conceived a PoC metric [284], which focuses on providing incentives for those who are good actors, using a view of such actors as intelligent agents, capable of autonomous interaction with the blockchain.

5.4.2.1 Block mining and validation

$KnowLedger$'s blockchain consists of a set of childchains, connected by each genesis block. It is based on PoW for block mining mainly for its simplicity as well as to avoid different tiers of agents and an increased implementation complexity. It, however, presents some key differences in regard to the used PoC:

- The unit of transaction is the data collected by a slave agent, tagged with a timestamp and the agent's identification;

- There is no complex conditional scripting over any transaction;
- The value of a transaction is translated into a score, which makes up the transaction outputs, as opposed to a coin-based system;
- The cumulative score of an agent is monotonically increasing;
- The cumulative score provides a way to measure the importance and overall contribution of an agent to the blockchain;
- The median score per transaction provides a way to measure the confidence on an agent.

The block structure is depicted in Figure 5.3. It is important to note the presence of a block header that, similarly to other blockchains, consists of metadata at the top of a block of transactions. It contains the hash of the previous block, a nonce, a timestamp, the Merkle's Tree root hash, and its own hash. Additionally, and uniquely to *KnowLedger*, it further details three hashes: the ledger hash, which encodes the blockchain the header pertains to; the chain hash, which encodes the datachain being referenced; and the tag, which is an encoding of the data schema.

A block, besides its header, contains a set of transactions. The first one is the coinbase, being there where agents that contributed with transactions are identified to claim the reward. This special transaction also contains the block's difficulty and the block's height to ensure there is a historical record of the block's mining embedded within it and hashed with it. The remaining transactions, besides containing the data and its timestamp, contain the identification of the agent that committed the transaction. Data representation is left completely customizable, provided that there is a clear distinction between each type of data. The size of such data must be calculable, as the block size is fixed. Currently, the block size is of

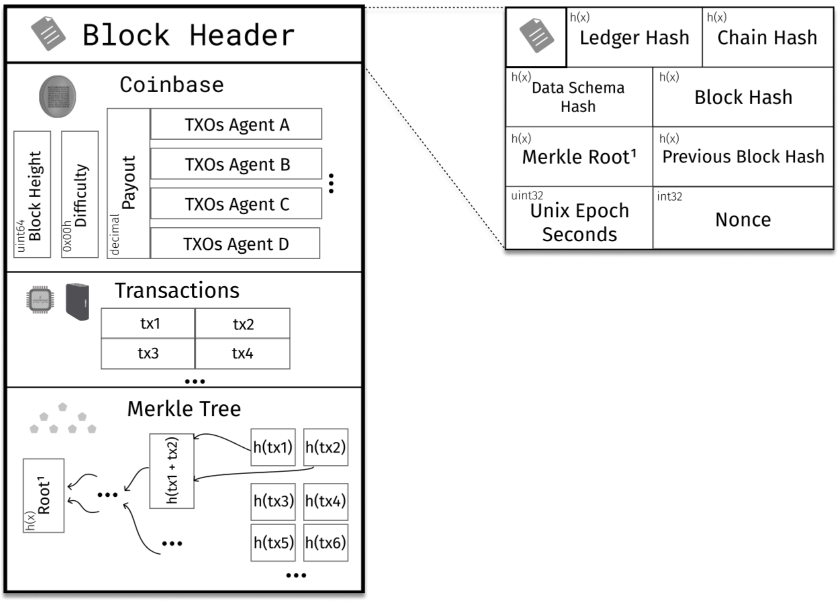


Figure 5.3: *KnowLedger*'s block structure.

two megabytes. Blocks are considered full when they are populated with enough transactions to achieve either a fixed amount of ceiling or fill the maximum allocated block size. A fixed amount of transactions help avoid extended delays in generating blocks. It should be noted that *KnowLedger* transactions, differently from other blockchains, do not require a recipient since nothing is being exchanged. Data are just being committed and, for that, a set of confidence is given as reward.

Two basic assumptions are crucial, with the first being that data exists at a specific point in time while the second is that data are tied to a geographical location. Hence, all data are required to be tagged by the agent with the time at which it was gathered and the corresponding geographic coordinates. This allows *KnowLedger* to avoid implementing additional logic to address clock synchronization issues. The decision was to take advantage of the fact that all data exists at a specific point in time to provide a comparison metric. In addition, to have some flexibility, and since data can come with some delay, a two-hour window is given when validating the candidate block's timestamp. The values here presented are candidate values, found through experimentation and testing.

To conclude, the basic block validation in *KnowLedger* is as follows:

1. Confirm that the previous block exists and that the referenced hash is correct;
2. Confirm that the timestamp of the candidate block is acceptable. This is done by ensuring the timestamp is higher than the median of the previous 11 blocks and less than 2 hours in the future of the node's clock;
3. Confirm that the first transaction is the coinbase;
4. Confirm that the number of transactions or the block's maximum size was achieved;
5. Confirm that difficulty, height, and Merkle tree root are correct;
6. Confirm that the PoW on the block is valid. This implies checking the hash of the resulting block is above the difficulty threshold;
7. Confirm the validation of each transaction. This requires ensuring the integrity of the hash and the correct signature of data;
8. Confirm the validation of the coinbase. This is done by recalculating each transaction's score according to the view of the blockchain, and ensuring that all scoring, both cumulative and for individual transactions, matches;
9. If every prior validation is confirmed, the block is considered valid.

5.4.2.2 Consensus, Proof-of-Work, and Difficulty recalculation

In its essence, consensus consists of a group decision-making process in which the members agree to support a decision that is in the best interest of the group, even if not the preferred of each individual. In blockchain terms, consensus is a dynamic way of reaching an agreement. Within *KnowLedger*'s blockchain, PoW and the developed PoC are used to confirm blockchain consensus. To interact with the Blockchain, intelligent agents are required to have a cryptographically secure identity based on an asymmetric public and private key-pair. The public identity is the one being referenced and tracked by the blockchain, with the data supplied by an agent being signed with its private key.

In terms of synchronization between agents, and as explained before, only ledger agents are required to have knowledge about the blockchain. The initial behavior of such agent, as soon as it connects to the network, should be to synchronize with other running agents through sequential transfer, starting from the latest committed block. The agent then begins executing the succeeding behaviors, including receiving and sending transactions from its peers and from its slave agents, and mine the candidate block. As soon as the candidate block has been mined, it is shared over the network, validated by peer agents, and the process restarts for a new candidate block. The block headers are propagated between agents when mining results in a nonce that fulfills the difficulty requirements of the block. When tight synchronization of transactions is ensured, only the block header needs to be propagated, as the same already validated transactions will be present in each agent's current block and only the nonce and timestamp differ in the header. Synchronization follows well-defined ontologies, one of the strengths of MAS.

To ensure that transactions will eventually populate a block, there must be a periodic diffusion of those that are not yet committed. No restrictions are applied as to the time that transactions are issued and the score is not deducted at any point. Due to this lack of restrictions, a transaction is never invalidated (as would be the case for insufficient funds or cancellation). As long as the agent stores its transactions, data gathered are guaranteed to never be lost and will eventually be recorded in $KnowLedger$, later in time.

Similarly to the case of the Bitcoin network, difficulty recalculation takes place every 2048 blocks with the intent being to achieve a fixed rate of block throughput in each datachain within childchains. As such, a median block time is calculated, taking into account the previous 2048 blocks to the one that triggers recalculation. This block time is then compared to the childchains' configured desired block time, and the difficulty is adjusted to compensate for any deviation, by either lowering or raising it.

5.4.2.3 Proof-of-Confidence

As data transactions do not codify any semantics of expenditure, as in typical cryptocurrencies, they are never invalidated by cancellations, lack of funds, or other expiration of any kind. In fact, committing data that is older than some existing data of the same category in a small geographical radius is likely to happen. To tackle the problem of scoring data for such situations, the observation that environmental data tends toward homogeneity for reduced time-frames becomes key. Hence, instead of maintaining a complex scoring system, a simple short-term memory formula can be used.

The PoC is thus calculated entirely via relative measurements, namely ratios between data values and timestamps, restricted to a geographic area. This describes an ever increasingly populated timeline of data, which, as it tends towards density, can be pushed into accurately modeling the underlying models' fluctuations. By scoring data according to its relation to its closest match in time, future or past, as well as within a reasonably sized geographical radius (configurable for any childchain, to ensure data pertains to the same area in space), data, which naturally presents patterns of frequent spikes, stable curves or nearly constant values, can be scored.

The formula to calculate the score of a given transaction is thus broken into three different components, being based on a comparison between the transaction to be added, subscript *a*, and the transaction closest in time to it, subscript *c*, present in the blockchain or in the candidate block, of the same type:

1. The first component extracts a relative measurement between the timestamps (*t*), noted as δ_t:

$$\delta_t = \frac{|t_c - t_a|}{t_c} \tag{5.1}$$

2. The second component extracts a relative measurement between the values (*v*) present in the data, noted as δ_v, for each value in the set of v_c and v_a:

$$\delta_v = \frac{|\sum v_{ci} - \sum v_{aj}|}{\sum v_{ci}}, i \in v_c, j \in v_a \tag{5.2}$$

3. The final component is a small additive base component, which guarantees a base incentive to provide data (noted as $base$).

These three components are then composed in order to emphasize and prioritize either δ_t, such that frequency and reliability of the data are valued more highly; δ_v, such that stricter homogeneity of values are valued more highly; and the $base$ component, which incentives data collection regardless of quality. At the time of writing, the default PoC formula is one that values δ_t over δ_v, with a small $base$ component, effectively prioritizing a dense timeline of data. Its formula is as follows:

$$\frac{(\delta_t \cdot base_t)^2 \cdot \frac{\delta_v \cdot base_v}{4} + base}{divisor} \tag{5.3}$$

Where,

$$base_t = 4,\ base_v = 2,\ base = \frac{1}{2},\ divisor = 50000 \tag{5.4}$$

This last formula is configurable for a given instance of a childchain. It is also worth highlighting that the use of timestamps is not managed by the agents at the transaction level, as no trust chain can be reliably established for the underlying sensors or data collector agents without imposing very restrictive penalties in hardware-cost and certificate management. As such, trust is achieved by comparing and penalizing scores by comparison to neighboring points in time and locale, deterring agents from contributing data that are not inline with others.

5.4.3 API

Advocating for open-source artifacts, all the produced software was published, in GitHub, under a MIT License. Hence, even though anyone may conceive and develop their own agents, we took the liberty of publishing the code of the intelligent agents that were used in the experiments performed in this study. JADE was the selected framework, with FIPA standards being followed. Kotlin programming language was the one chosen for the development of intelligent agents. Hence, in the project's repository, one may find a set of behaviors and actions. A set of APIs are also released by default, as well as the ontologies for block and transaction messaging. The container is also instantiated. The blockchain in itself is provided to the agents as a completely stand-alone library, being independent and unaware of the agent's platform. It is also written in Kotlin and is available in the same repository.

5.5 Experiments

The case study aimed at creating a running environment for *KnowLedger* and was conceived, in its essence, as a conceptual evaluation of the proposed solution in order to facilitate an answer to the elicited research questions. A major goal of the conducted experiments was to understand if a blockchain was a valid solution to store structured data and if a multi-chain architecture would be a viable approach to the problem in hands. Moreover, it also intended to evaluate the interaction and implementation of a MAS to work and maintain the blockchain as well as the use of the PoC to award those who commit accurate data.

The case study focused on three distinct Portuguese cities: Braga, Guimarães, and Porto. Hence, logically, *KnowLedger*'s blockchain contains three childchains. As soon as a childchain is instantiated for a specific city by a ledger agent, that same chain becomes available for all other agents that instantiate that same childchain. With the due differences, one may consider it to follow a singleton pattern. All chains are maintained by ledger agents. These agents are the ones responsible for instantiating the childchain and having, at all times, its up-to-date version. For that, they are required to make use of synchronization behaviors to communicate with their peers, and receive/send new blocks and transactions. Indeed, ledger agents are responsible for creating new transactions from data received from slave agents.

All behaviors were implemented as JADE behaviors, namely as simple, ticker, and cyclic ones. For the synchronization of ledger agents, the main behaviors are the *GetMissingBlocks*, which is self-explanatory, and the *ReceiveMessage* and *SendMessage*, which are used to communicate about newly created transactions or mined blocks. For the unidirectional communication between slave and ledger agents, the first implements the *DataCaptureAndSharing* ticker behavior, which executes periodically, while the second implements the *SlaveMessageReceiver* cyclic behavior. Indeed, slave agents only have one behavior and it is the one referred to above, i.e., *DataCaptureAndSharing*. This behavior calls *The Collector*, obtains data, processes it, and shares it with its ledger agent. On the other hand, as soon as the ledger agent receives data, it creates the respective transaction and, as soon as the candidate block is considered full, it starts mining it using the *Mining* behavior. As soon as the block is mined, it is removed from the queue, added to the childchain, and shared with peers. All this process keeps going repeatedly.

For this case study, the MAS consisted of three Smart Hubs, each one working a childchain of a different city. Each Smart Hub consisted of one ledger agent and one, or more, slave ones. Braga's Smart Hub was the one containing two slave agents, with one being responsible for collecting data about the UV index of the city with the other collecting all current traffic incidents in Braga. Porto and Guimaraes' Smart Hubs only contained one slave agent each that was responsible for collecting the UV index of the respective city. The MAS consisted of three ledger and four slave agents. Each agent had the specific geographic coordinates that were related to the city being sensed, with data being collected every 10 seconds. Figure 5.4 details each Smart Hub. It should be noted that more Smart Hubs could easily join the desired childchain, and start working and maintaining it. Each Smart Hub can opt to join several childchains and commit the corresponding blocks to the respective chain. This is the case when a ledger agent receives data from slave agents working on distinct geographical regions.

Data representation for the UV index and traffic incidents were added to the blockchain so that it could comply with the scoring contract and, thus, accepted by the blockchain. The UV index was parameterized with latitude, longitude, date, unit, and value. On the other hand, traffic incident is a more complex object, containing, among others, the magnitude of the delay caused by the incident, the locations where the traffic due to the incident starts and ends, the category, and the delay in seconds. These classes implement the *BlockChainData* interface, which, in turn, is part of *PhysicalData*, the main class in which to store blockchain data. Indeed, *PhysicalData* is what is stored in each transaction. It was also necessary to define application-specific ontologies, including actions, concepts, and predicates to ensure proper communication between the agents. Data converters were also implemented to *convertToJadePhysicalData* and to *convertFromJadePhysicalData*, among others. Agents were then able to register the proper language and ontologies.

Each Smart Hub runs inside a container. Hence, three distinct containers, plus the main container,

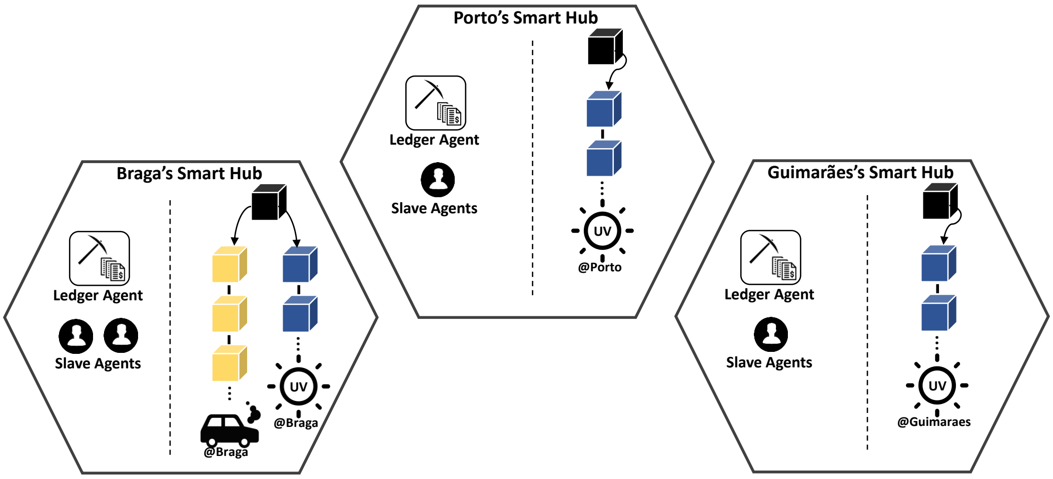
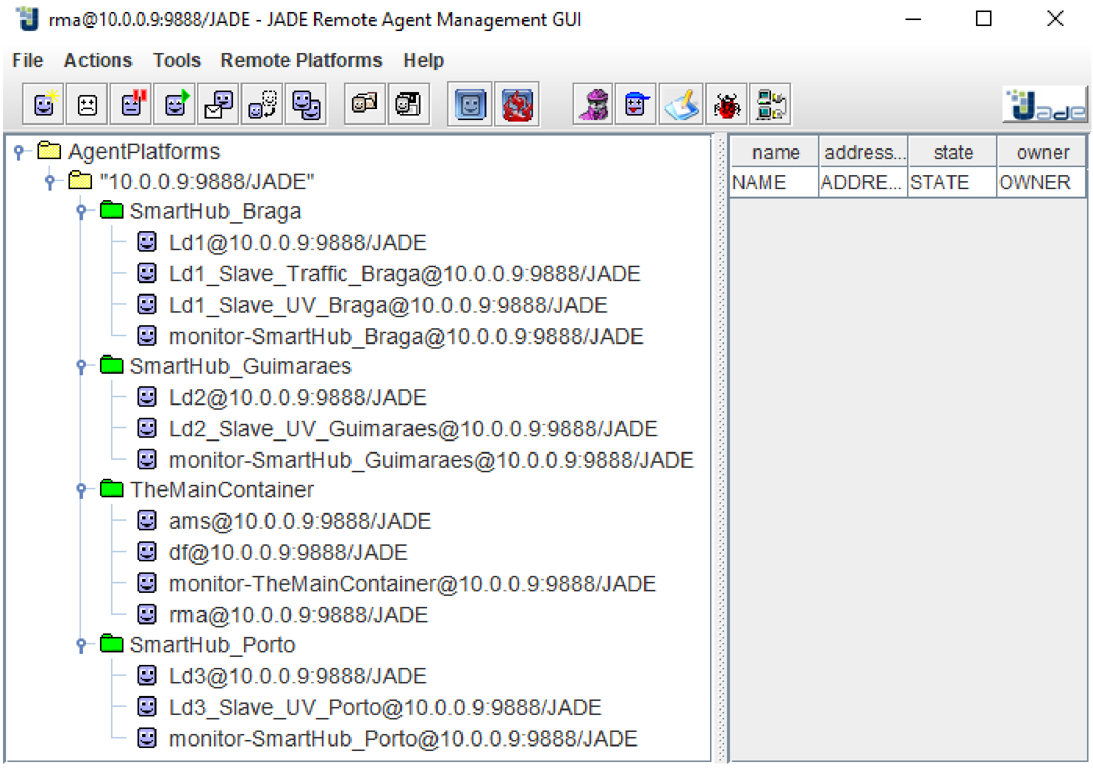


Figure 5.4: Implemented Smart Hubs.

were initialized. All containers are distributed over processes running in the same host. Each container is entitled as SmartHub_<*CITY*>, with <*CITY*> being a placeholder for the respective value (SmartHub_Braga, for example). Ledger agents are named as Ld<*NUMBER*>, with <*NUMBER*> being a placeholder for an unitary increasing number (Ld1, for example). On the other hand, slave agents indicate, in their name, the corresponding ledger agent, the sensed parameter, and the city (Ld1_Slave_UV_Braga). Figure 5.5 depicts the view of each Smart Hub from JADE Remote Agent Management GUI.



- AgentPlatforms
  - "10.0.0.9:9888/JADE"
    - SmartHub_Braga
      - Ld1@10.0.0.9:9888/JADE
      - Ld1_Slave_Traffic_Braga@10.0.0.9:9888/JADE
      - Ld1_Slave_UV_Braga@10.0.0.9:9888/JADE
      - monitor-SmartHub_Braga@10.0.0.9:9888/JADE
    - SmartHub_Guimaraes
      - Ld2@10.0.0.9:9888/JADE
      - Ld2_Slave_UV_Guimaraes@10.0.0.9:9888/JADE
      - monitor-SmartHub_Guimaraes@10.0.0.9:9888/JADE
    - TheMainContainer
      - ams@10.0.0.9:9888/JADE
      - df@10.0.0.9:9888/JADE
      - monitor-TheMainContainer@10.0.0.9:9888/JADE
      - rma@10.0.0.9:9888/JADE
    - SmartHub_Porto
      - Ld3@10.0.0.9:9888/JADE
      - Ld3_Slave_UV_Porto@10.0.0.9:9888/JADE
      - monitor-SmartHub_Porto@10.0.0.9:9888/JADE

name	address...	state	owner
NAME	ADDRE...	STATE	OWNER

Figure 5.5: Smart Hubs and intelligent agents' view from JADE Remote Agent Management GUI.

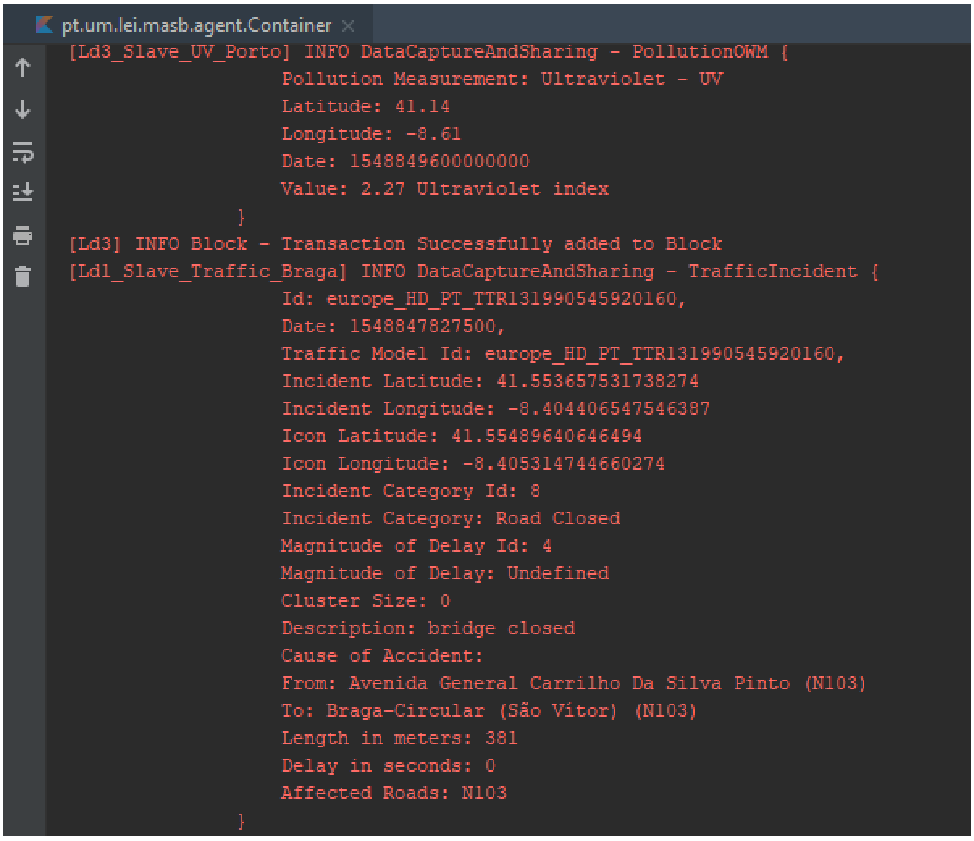
As soon as the agents went live, they immediately started executing, autonomously, the expected behaviors upon each scenario. Data were collected, transactions were created and, as soon as the candidate block became full, the mining started and synchronization happened. It is also worth mentioning that the blockchain, and its childchains, had, prior to execution, their blocks limited to five transactions. This number was defined for demonstration purposes. All code, diagrams, parameters, and implementation details of this experiment are available in *KnowLedger*'s repository.

5.6 Results and Discussion

Concerning the conceived MAS and the implemented Smart Hubs, slave agents had their focus set on collecting data, being completely abstracted from the concept of blockchain, childchain, blocks, or transactions. Considering that the PoC metric is publicly known, they could work on maximizing the collected data and the obtained score. Indeed, the PoC was shown to effectively maintain short-term memory, comparing transactions of the same data type that are the closest in time. Moreover, as soon as a poorly scored reading was confirmed by its succeeding, higher scores were given due to the stabilization of readings. The current PoC formula promotes frequency over value, prioritizing a dense timeline of data.

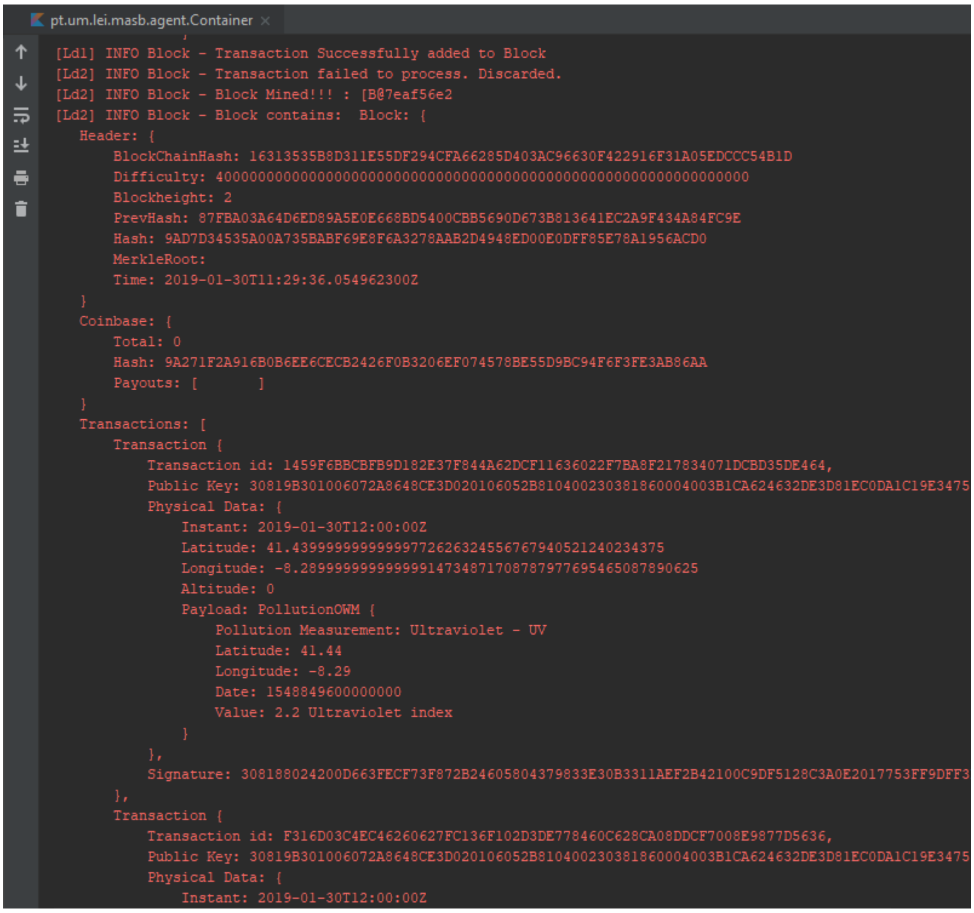
Figure 5.6 shows a set of log messages from the slave agents *Ld1_Slave_Traffic_Braga* and *Ld3_Slave_UV_Porto*. With self-describing names, the first one is collecting traffic incidents data for the city of Braga using the *DataCaptureAndSharing* behavior, while the second one, using the same behavior, is collecting UV measurements for the city of Porto. Figure 5.6 also shows that both data types are geographically tagged, which allows the validation that data are being committed to the correct childchain. Each slave agent collects the respective data every 10 seconds, logs it, and then sends it to its ledger agent.

Ledger agents have an up-to-date version of the desired childchain after instantiating it. They are then able to start accepting data from their slave agents. Ledger agents, upon the reception of new data, build a transaction and attempt to insert it on the candidate block. This happens until the candidate block is full, i.e., until the block had its five transactions slots filled. Then, the mining behavior starts. As depicted in Figure 5.7, ledger agents log all transactions they add, or fail to add, to a block. Such failures can happen if, for example, the geographic coordinates do not match the instantiated childchain. However, since in this case study all slave agents were providing correct geographically-tagged data, the transaction was likely to be discarded because the candidate block was full and not yet mined. This happened because the ledger agents were only committing new transactions after mining the full candidate block. This is easily solved



```
[Ld3_Slave_UV_Porto] INFO DataCaptureAndSharing - PollutionOWM {
                   Pollution Measurement: Ultraviolet - UV
                   Latitude: 41.14
                   Longitude: -8.61
                   Date: 1548849600000000
                   Value: 2.27 Ultraviolet index
               }
[Ld3] INFO Block - Transaction Successfully added to Block
[Ld1_Slave_Traffic_Braga] INFO DataCaptureAndSharing - TrafficIncident {
                   Id: europe_HD_PT_TTR131990545920160,
                   Date: 1548847827500,
                   Traffic Model Id: europe_HD_PT_TTR131990545920160,
                   Incident Latitude: 41.553657531738274
                   Incident Longitude: -8.404406547546387
                   Icon Latitude: 41.55489640646494
                   Icon Longitude: -8.405314744660274
                   Incident Category Id: 8
                   Incident Category: Road Closed
                   Magnitude of Delay Id: 4
                   Magnitude of Delay: Undefined
                   Cluster Size: 0
                   Description: bridge closed
                   Cause of Accident:
                   From: Avenida General Carrilho Da Silva Pinto (N103)
                   To: Braga-Circular (São Vítor) (N103)
                   Length in meters: 381
                   Delay in seconds: 0
                   Affected Roads: N103
               }
```

Figure 5.6: Log messages from Porto and Braga's slave agents responsible for collecting UV data and traffic incidents, respectively.



```
[Ld1] INFO Block - Transaction Successfully added to Block
[Ld2] INFO Block - Transaction failed to process. Discarded.
[Ld2] INFO Block - Block Mined!!! : [B@7eaf56e2
[Ld2] INFO Block - Block contains:  Block: {
   Header: {
       BlockChainHash: 16313535B8D311E55DF294CFA66285D403AC96630F422916F31A05EDCCC54B1D
       Difficulty: 4000000000000000000000000000000000000000000000000000000000000000
       Blockheight: 2
       PrevHash: 87FBA03A64D6ED89A5E0E668BD5400CBB5690D673B813641EC2A9F434A84FC9E
       Hash: 9AD7D34535A00A735BABF69E8F6A3278AAB2D4948ED00E0DFF85E78A1956ACD0
       MerkleRoot:
       Time: 2019-01-30T11:29:36.054962300Z
   }
   Coinbase: {
       Total: 0
       Hash: 9A271F2A916B0B6EE6CECB2426F0B3206EF074578BE55D9BC94F6F3FE3AB86AA
       Payouts: [       ]
   }
   Transactions: [
       Transaction {
           Transaction id: 1459F6BBCBFB9D182E37F844A62DCF11636022F7BA8F217834071DCBD35DE464,
           Public Key: 30819B301006072A8648CE3D020106052B810400230381860004003B1CA624632DE3D81EC0DA1C19E3475
           Physical Data: {
               Instant: 2019-01-30T12:00:00Z
               Latitude: 41.43999999999999772626324556767940521240234375
               Longitude: -8.2899999999999991473487170878977695465087890625
               Altitude: 0
               Payload: PollutionOWM {
                   Pollution Measurement: Ultraviolet - UV
                   Latitude: 41.44
                   Longitude: -8.29
                   Date: 1548849600000000
                   Value: 2.2 Ultraviolet index
               }
           },
           Signature: 308188024200D663FECF73F872B24605804379833E30B3311AEF2B42100C9DF5128C3A0E2017753FF9DFF3
       },
       Transaction {
           Transaction id: F316D03C4EC46260627FC136F102D3DE778460C628CA08DDCF7008E9877D5636,
           Public Key: 30819B301006072A8648CE3D020106052B810400230381860004003B1CA624632DE3D81EC0DA1C19E3475
           Physical Data: {
               Instant: 2019-01-30T12:00:00Z
```

Figure 5.7: Log messages from the ledger agent working the city of Braga (Ld1), when inserting a transaction, and Guimarães (Ld2), when successfully mining a block.

by allowing the *SlaveMessageReceiver* behavior to accept data while a candidate block is being mined by the mining behavior.

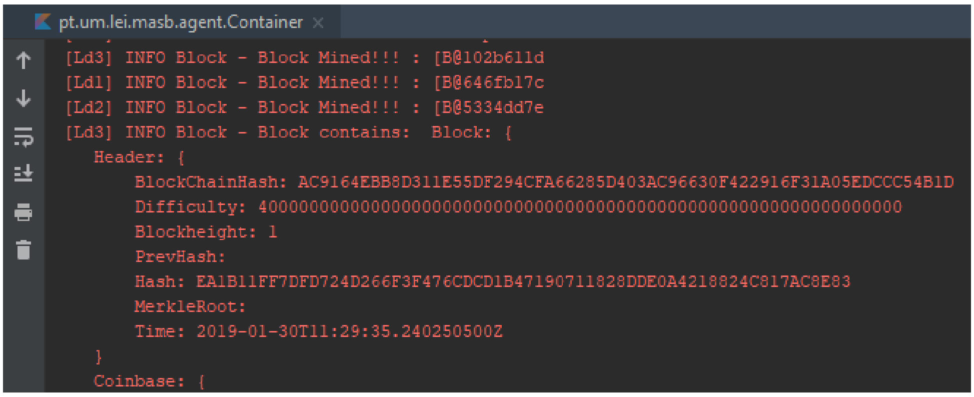
Figure 5.7 shows multiple scenarios. It starts with a message from Ld1 (Braga's Smart Hub), where the agent states it was able to successfully add a transaction to its candidate block. Such a message is followed by a message from Ld2 (Guimaraes' Smart Hub), with the agent notifying it was unable to process a transaction and, hence, it would be discarded. As explained previously, this happened because the candidate block was full and not yet mined. In other words, agent Ld2 was mining a block at that time. As soon as a valid hash is found by the agent, the block is considered mined and disseminated among peers. Figure 5.7 also depicts a log message from Ld2, where the agent communicates it has mined a block. The block itself is logged after being mined and validated, presenting a set of information such as its hash, timestamp, height, and the entire list of transactions, among others.

The JADE framework also revealed its utility mainly due to its simple but yet rich environment. It allowed the conception of a set of crucial behaviors that allow a ledger agent to easily work with, and maintain, a blockchain while, at the same time, being able to receive data from its slave ones. Moreover, a framework such as JADE significantly eases the synchronization protocol among peers. Regarding the slave agents, they were sending data directly to its ledger, with an alternative being to broadcast the data to all those that, inside the same hub, are interested in subscribing their feeds. There are however a few drawbacks mainly related to the conception of ontologies, which proved to be somehow delicate.

The childchains also proved to be easily instantiated by any ledger agent. Moreover, the fact that each city had its own childchain allowed agents to reduce the computational weight of the blockchain by instantiating only the desired ones. At any moment, a ledger agent may opt to instantiate new childchains and start working on it. Lookup is logically faster because it is performed inside the childchain, allowing one to access a city historic data of any parameter without the need to go through all other cities and/or parameters. In fact, the calculation of the PoC also benefits from this faster lookup, as well as the validation of blocks transmitted through the network. Blocks and transactions also accomplished their job, even in the presence of concurrent mining and synchronization (Figure 5.8). To provide long-term storage support, an embedded dataset management system was implemented as well as a primitive cache, being included in *KnowLedger*'s blockchain. OrientDB, a distributed NoSQL database with a graph engine, was the choice for its capability of working in a schema-hybrid mode, allowing certain data properties to be defined.

To increase the adoption of the platform, and to encourage more participants, the award that is given to the agent in exchange for its accurate data are expected to be tradable, i.e., a city could opt to provide discounts on local products based in, or in exchange for, the confidence of an user. Hence, not only the city is encouraging more users to provide data but also users feel encouraged to participate and provide accurate and frequent data to receive confidence that can be later applied to obtain discounts.

The obtained results show promising prospects towards the use of MAS blockchains to store structured data. As direct answers to the elicited research questions, it has been confirmed that (RQ5.1) blockchain emerges as a viable solution to store Smart Cities' data due to its immutable, distributed, decentralized nature. Coupled with a MAS, it is able to offer an environment where one is encouraged to provide accurate, reliable, and frequent data. Moreover, without the need for a centralized authority, everyone has access to the entire chain of data without fees or restrictions. Intelligent agents also showed to be able to autonomously maintain a simplified multi-chain blockchain with just a small set of behaviors (RQ5.2). The multi-chain architecture (RQ5.3) proved to be logically viable and feasible, bringing benefits not only to the system in itself but also for those working and maintaining it, guaranteeing a logical separation of data. Finally, (RQ5.4) the scoring environment provided incentives for those who commit structured data that, per the implemented PoC, had the frequency of commits valued higher than the homogeneity of values,



```
[Ld3] INFO Block - Block Mined!!! : [B@102b611d
[Ld1] INFO Block - Block Mined!!! : [B@646fb17c
[Ld2] INFO Block - Block Mined!!! : [B@5334dd7e
[Ld3] INFO Block - Block contains:  Block: {
   Header: {
       BlockChainHash: AC9164EBB8D311E55DF294CFA66285D403AC96630F422916F31A05EDCCC54B1D
       Difficulty: 400000000000000000000000000000000000000000000000000000000000000
       Blockheight: 1
       PrevHash:
       Hash: EA1B11FF7DFD724D266F3F476CDCD1B47190711828DDE0A4218824C817AC8E83
       MerkleRoot:
       Time: 2019-01-30T11:29:35.240250500Z
   }
   Coinbase: {
```

Figure 5.8: Concurrent mining and synchronization for three ledger agents.

with a small base value to encourage data collection.

5.7 Summary

The study and experiments detailed in this chapter provide the answer to **Task 4**. In particular, the goal was to understand the viability of coupling two distinct technologies, MAS and blockchains, for the benefit of Smart Cities through means of data collection and availability. In fact, Smart Cities introduce novel problems related to the management of inordinate amounts of data of different types and sources. In addition, as soon as everyone gets allowed to contribute, mechanisms to control data quality must arise.

KnowLedger was designed and conceived from scratch, without any dependency to third-party blockchains, being, to the best of our knowledge, unique in regard to the manner it merges MAS and blockchains, to its multi-chain architecture, and to the used protocol to ensure confidence on the actors and the committed data. Hence, it is worth highlighting that this case study aimed to evaluate, conceptually, the proposed solution in order to facilitate an answer to the elicited research questions. In addition, the PoC protocol was established to award those who commit accurate and reliable data, providing a scoring environment for a system comprised of a multitude of autonomous agents. Besides collecting data, intelligent agents are also responsible for block mining, allowing them to optimize their work based on the committed data and the respective reward. The implemented multi-chain architecture separates cities and data types, using childchains and datachains to reduce querying time and the computational load of *KnowLedger*.

To prove the research hypothesis and validate the conceived solutions' specimens, the conducted study aims to address ***Research Question 4****: Is blockchain a viable solution (in terms of efficiency and performance) to store structured data? Can intelligent agents work, and maintain, a blockchain?*. First and foremost, intelligent agents, living within a MAS, were able to successfully implement ontologies, behaviors, and maintain the blockchain. The blockchain in itself presents a viable approach to store data in a way that it becomes freely available to the community, incentivizing user engagement through tokens that are given and that may be exchanged by discounts in products and/or services of a city.

It is worth highlighting that the conception of solutions like the one proposed in this chapter is a never ending process, i.e., it requires monumental work-forces to constantly build, improve, and enhance the platform. *KnowLedger*'s repository already contains several hundreds of thousands of lines of code that represent several months of conception, thinking, and coding. It is, however, far from being complete. In fact, *KnowLedger* still requires many days, months, and probably years of conception and coding. This is to say that future work is vast and spread along several axis. We aim to, with great preponderance, evaluate *KnowLedger*'s performance in regard to data throughput, scalability, and an increased number of agents and data sources. The goal is to evaluate *KnowLedger*'s performance and behavior when facing a superlative number of data streams, where the quantity and velocity of input data are significant. Load tests are also essential to validate the database performance and scalability. In addition, rigorous tests, such as code reviews and penetration tests, will continue to happen. In turn, the metric to award those who supply good and frequent data seems now stable, with an interesting feature including dynamic adjustment based on data fluctuation and frequency. Blacklist implementation is still to be tested and evaluated. It is also expected the development of a web page to inspect the blockchain, its childchains, its blocks, and each committed transaction so that results become easier to assess and share.

The work presented in this chapter gave origin to a scientific paper published in the proceedings of the 3rd International Conference on Cyber-Technologies and Cyber-Systems ([284]) as well as to another one, published in the highly-ranked *Artificial Intelligence Review* journal ([285]).

Chapter 6

Conclusions and Future Work

Long and intensive years of research culminated in the work described in this thesis. The goal has been, at all times, to further extend the frontiers of knowledge, thus promoting the technological progress of our society. The conducted experiments focused on three rather large concepts, i.e., Smart Cities, VRUs, and the IoP, in order to increase the safety of those more vulnerable on the road while pushing for smarter cities and for an environment where people are active and proactive actors.

VRUs' vulnerability arises from several directions. No monolithic solution will, singularly, settle out all problems nor will it eliminate all fatalities. Instead, the difference is made through the sum of several small contributions. Our contribution is grounded on the perception of people, i.e., VRUs, as active, re-active, and participative actors of the city ecosystem. From this premise emerged what we call the 3-C, corresponding to *City*, *Crowd*, and *Citizen* sensing. The 3-C defined the guidelines and goals for the conducted research, establishing the research path with precision and clarity. This, complemented with the adopted research methodology, allowed us to organize a logical sequence of standardized procedures to build scientific knowledge, producing a replicable, precise, falsifiable, and parsimonious process.

We organized the 3-C into three ordered layers. The first, *City Sensing*, focused on creating new means to sense the city's environment, thus allowing cities to understand and act on problematic situations where the human being is the weaker part. Based on our vision of an architecture for the IoP, a MAS was conceived and made responsible for the entire non-mobile data collection process and for the full ML pipeline (Automated ML). Then, we focused on two case studies, i.e., traffic flow and UV index forecasting. While the first allows VRUs to know and avoid highly congested hours, the second allows the same users to adapt their behavior in situations of increased risk. In fact, the most effective strategy to control skin cancer is to avoid sun exposure during peak UV radiation levels. For these case studies, state of the art time series forecasting models were conceived, tuned, and evaluated under different settings and conditions to look for the one with the best performance. In addition, the conducted experiments allowed one to establish the better performance of LSTMs when compared to classic statistical-based models, as well as its ability to accurately forecast multiple future timesteps using a recursive multi-variate approach.

The second C, *Crowd Sensing*, focused on conceiving means to sense large groups of people without requiring them to do, install, wear, or carry anything. This passive crowd sensing approach was successfully achieved using ESP8266 ESP-12E NodeMCU Amica boards, which were programmed to capture WiFi probe requests emitted by people's devices when looking for APs. The ESP8266 boards, programmed for crowd sensing, gave origin to what we call the "Smart Scanners". These sensors have a reduced cost (2.4€ per board) and a set of impressive features for its size and weight, being able to properly sense crowds both in outdoor and indoor scenarios. The outdoor case study focused on important points of interest for VRUs, which include pedestrians' crossings, roads, and sidewalks. The Smart Scanners are also able to unscramble the captured MAC addresses, which, to some extent, can describe the sensed population in regard to their devices. To complement the second C, an active sensing approach to crowd sensing focused on designing and conceiving a mobile application for geofencing as well as an emotional map to quantify people's feelings. While the former focuses on alerting VRUs when they are near high traffic, high pollution, crowed, or dangerous zones, the latter depicts an emotional map using a heat-map approach based on the feelings of the citizens towards their city.

The third C, *Citizen Sensing*, on the other hand, focused on sensing a person in order to give depth to the individual. More than considering VRUs as "flat things", without personality neither emotions, we conceived a new test, the ASAP one, based on a ML approach to personality assessment. This test reduces the time required by typical personality assessment tests, being easier to implement by technological platforms when the goal is to obtain an approximation to the subject's personality. To successfully achieve this goal, we were required to conceive and make available a web platform for data collection. In fact, this third C also required us to conceive a mobile application that was responsible for building a dataset from where one could experiment several ML and DL models for boredom prediction. The goal was to detect user boredom when using the smartphone, as people tend to use their smartphone in situations where boredom arises. The best candidate model presented a promising performance, in terms of accuracy and precision, allowing the conception of applications that are able to perceive when a user is bored. We then engaged in a meta-analytic study focused on the human factors that characterize VRUs. The proposed method, based on a thermodynamics' view as a process of energy devaluation for knowledge representation and dataset construction, is able to build a quantitative dataset from a set of qualitative responses to surveys as well as to characterize one's perception of the queried factors.

Considering the research domains of this PhD work, we were also interested in studying and understanding how to address novel problems, introduced by Smart Cities, related to the management of inordinate amounts of data of different types and sources. We aimed to conceive a solution where data could be committed by any actor, being, at the same time, freely available to anyone that aims to derive knowledge from it. It should be noted that even though several platforms offer recent data, it is extremely hard to get access to historical data. Many times significant fees are requested or data are simply not available. Hence, we have coupled two distinct technologies, MAS and blockchains, to produce a platform, *KnowLedger*, that makes use of a multi-chain architecture to store structured data. We have also conceived a new protocol, entitled as PoC, to award those who commit accurate and reliable data. Due to the complexity and time it takes to build a fully-functional platform, we have presented a case study that is, in its essence, a conceptual evaluation of the theoretical ground on where the proposed solution rests.

Finally, to aggregate all the conducted experiments and the produced prototypes, a collaborative platform was designed, conceived, and made available online[1] being entitled as *SafeCity*. It is a platform for safer and smarter cities, making use of solutions' specimens produced as result of all this work. This platform is detailed in the next section, which also depicts the main practical results. Afterwards, a new section describes the main quantitative results, including, but not limited to, scientific dissemination, awards, participation and organization of scientific events, lectures, and supervision of students.

6.1 Practical Results

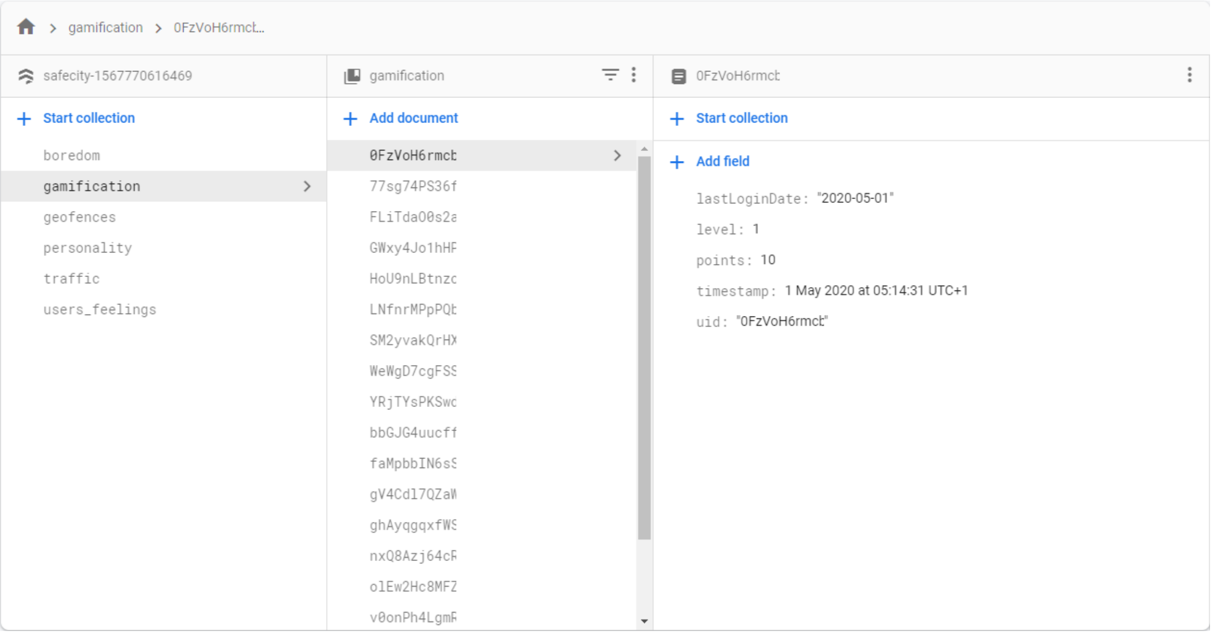
The next lines describe the main practical results achieved during the implementation of the work plan to validate the research hypothesis. It presents the conceived platform, the attained accomplishments, and the main contributions to the state of the art.

6.1.1 SafeCity - A Platform for Safer and Smarter Cities

A collaborative platform, entitled as *SafeCity*, aggregates the solutions' specimens that were conceived during the conducted research, working as a front-end of information to VRUs. It has a modular architecture, with each feature being implemented by a service that must define, at least, its Android activity, the corresponding handler, and the data model. *SafeCity* mobile application targets Android users with, at least, Android Marshmallow (API version 23). Android Pie, API version 28, is the targeted version. It uses Firebase's Cloud Firestore. Figure 6.1 depicts the NoSQL database holding several collections, including the ones that have been described in previous chapters (the same rules apply for database access).

SafeCity is available online, at Google's Play Store. After installing the application, the user is asked to register or to login into the platform. The user has available two options. He can opt to register using his Google or Facebook account. Since the user is on an Android-based smartphone, he will always have a Google account that can be used to register. Otherwise, Facebook credentials can also be used (Figure 6.2a). It is worth noting that *SafeCity* uses Firebase Authentication SDK as authentication backend service (Figure 6.3). This SDK integrates federated identity providers, leverages industry standards like OAuth 2.0, and tightly integrates with other Firebase services. It also provides access to user's basic profile information (photo, user email, and name), which is used by *SafeCity* to fill the user's data in the profile activity (Figure 6.2c). It is at the profile activity where the user can logout, check his current points endorsed by the gamification engine (as explained in subsequent lines), and where the user can turn on, or off, all the AI engines used by *SafeCity* (boredom, personality, and traffic AI engines). Figure 6.2b

[1] https://play.google.com/store/apps/details?id=com.plugable.safecity



gamification > 0FzVoH6rmct...

safecity-1567770616469

+ Start collection

boredom
gamification
geofences
personality
traffic
users_feelings

gamification

+ Add document

0FzVoH6rmct
77sg74PS36f
FLiTda08s2a
GWxy4Jo1hHF
HoU9nLBtnzc
LNfnrMPpPQt
SM2yvakQrHX
WeWgD7cgFSS
YRjTYsPKSwc
bbGJG4uucff
faMpbbIN6sS
gV4Cd17QZaW
ghAyqgqxfWS
nxQ8Azj64cF
olEw2Hc8MFZ
v0onPh4LgmF

0FzVoH6rmct

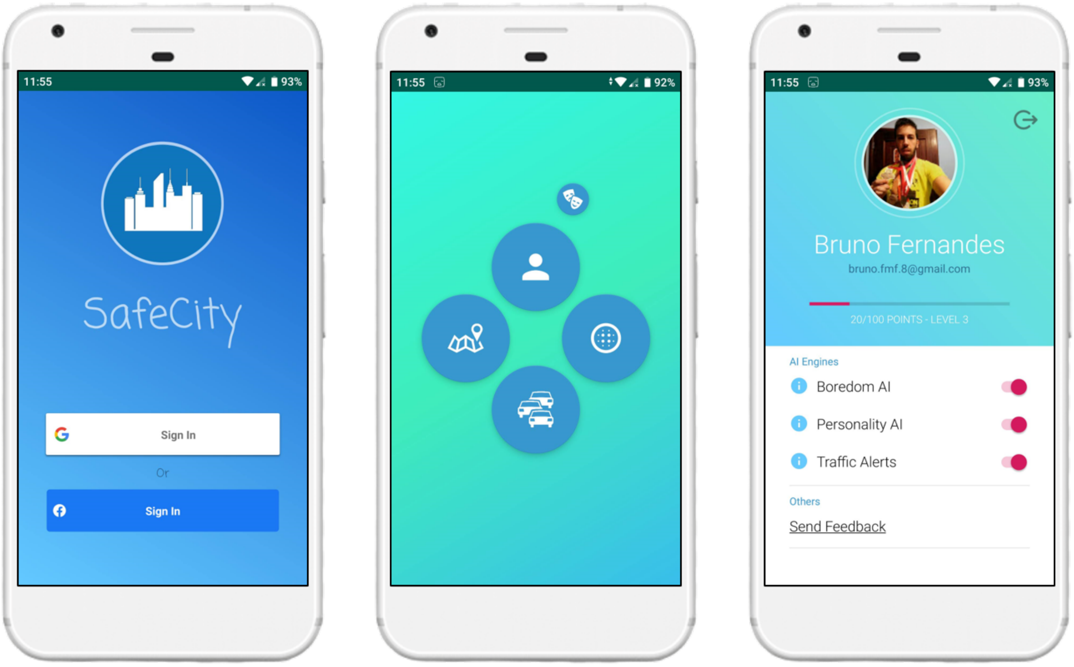
+ Start collection

+ Add field

lastLoginDate: "2020-05-01"
level: 1
points: 10
timestamp: 1 May 2020 at 05:14:31 UTC+1
uid: "0FzVoH6rmct"

Figure 6.1: Firebase's Cloud Firestore cloud-hosted NoSQL database supporting *SafeCity*.

depicts the main activity of the platform, i.e., the activity that is opened when the user signs in. It allows the users to navigate to their profile, or to other activities that implement the existing services (geofencing, traffic, emotional map, and personality services). The platform implements two locales, *pt* and *en*, which are loaded depending on smartphone's default language. User preferences are also stored so that they can be loaded by *SafeCity* each time the user starts the application, or if he signs in after signing out. This is achieved through means of the native Android's Preference library.

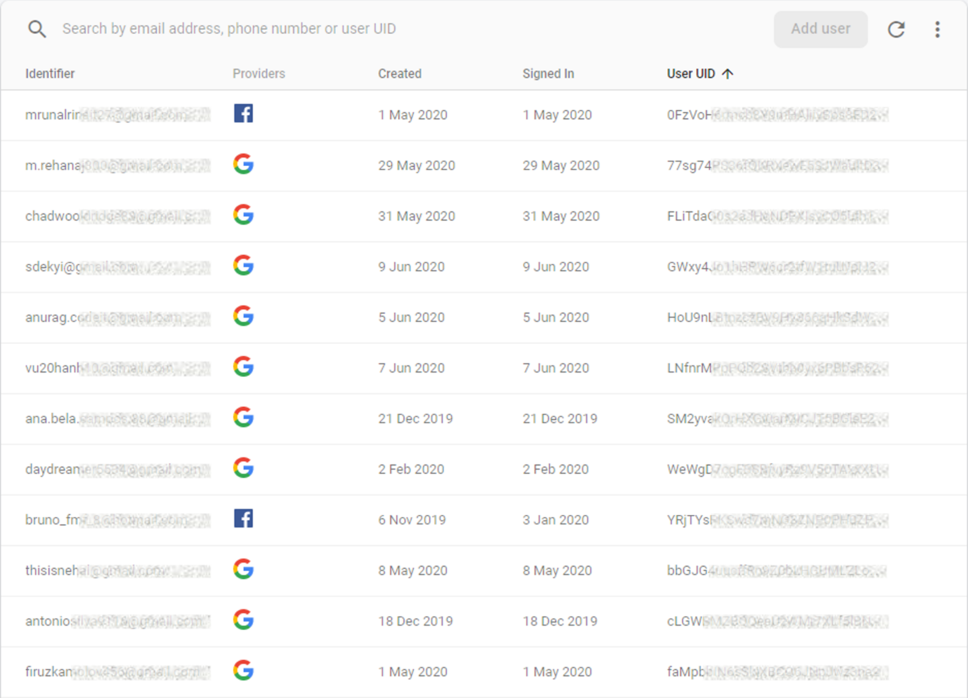


(a) Login and Register activity.

(b) Main activity after login.

(c) Profile activity.

Figure 6.2: *SafeCity* views when signing in.

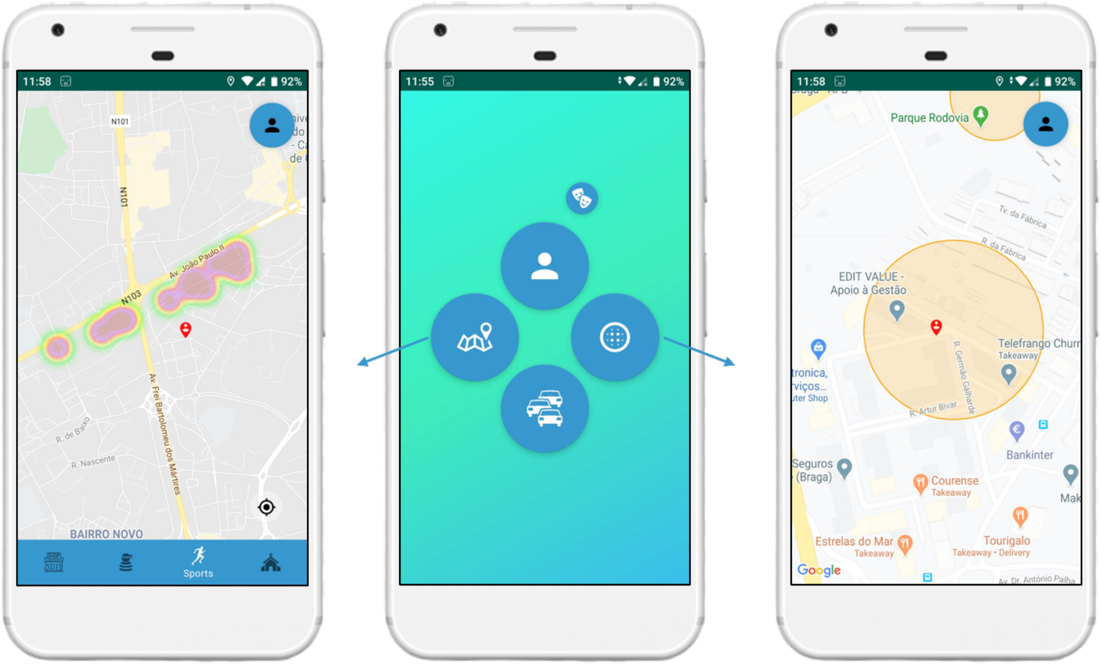


Search by email address, phone number or user UID

Identifier	Providers	Created	Signed In	User UID ↑
mrunalrin[redacted]	f	1 May 2020	1 May 2020	0FzVoH[redacted]
m.rehana[redacted]	G	29 May 2020	29 May 2020	77sg74[redacted]
chadwoo[redacted]	G	31 May 2020	31 May 2020	FLiTdaG[redacted]
sdekyi@g[redacted]	G	9 Jun 2020	9 Jun 2020	GWxy4J[redacted]
anurag.c[redacted]	G	5 Jun 2020	5 Jun 2020	HoU9nL[redacted]
vu20hanh[redacted]	G	7 Jun 2020	7 Jun 2020	LNfnrM[redacted]
ana.bela.[redacted]	G	21 Dec 2019	21 Dec 2019	SM2yva[redacted]
daydream[redacted]	G	2 Feb 2020	2 Feb 2020	WeWgD[redacted]
bruno_fm[redacted]	f	6 Nov 2019	3 Jan 2020	YRjTYs[redacted]
thisisneha[redacted]	G	8 May 2020	8 May 2020	bbGJG4[redacted]
antonio[redacted]	G	18 Dec 2019	18 Dec 2019	cLGW[redacted]
firuzkan[redacted]	G	1 May 2020	1 May 2020	faMpb[redacted]

Figure 6.3: Firebase Authentication backend service showing some of *SafeCity*'s registered users.

SafeCity implements the previously described emotional map (Figure 6.4a) as well as the geofencing application (Figure 6.4c). The underlying structure is the same, i.e., users can add and delete feelings from the map (minimum distance of 200m between feelings), which presents the feelings of the user and the community (as described in Chapter 3.3.4). Geofences, on the other hand, characterize dangerous zones of the city, with users being warned as soon as they enter/exit them (as presented in Chapter 3.3.3).



(a) Emotional map activity.

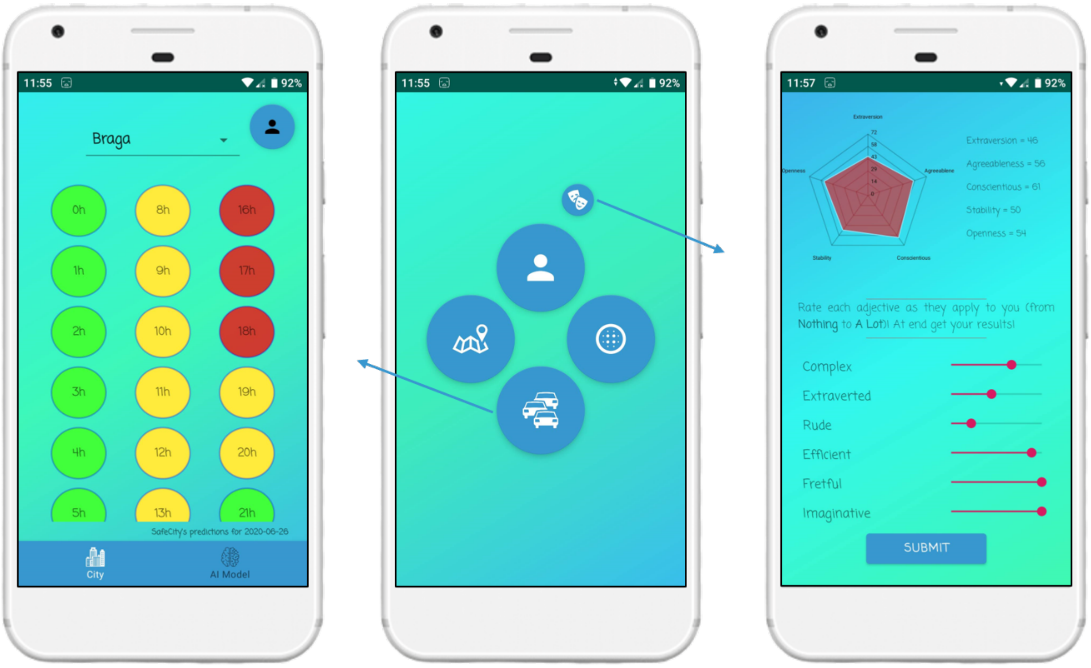
(b) Main activity after login.

(c) Geofencing activity.

Figure 6.4: *SafeCity* views for the emotional map and geofencing services.

SafeCity implements two more services, one for traffic flow forecasting and another for personality assessment. The former reads forecasts made by the *ML Architect* which, as described in Chapter 2.3.2, holds all non-mobile ML models and uses real-time data gathered by *The Collector* to obtain forecasts. *SafeCity* reads such forecasts and presents them to the user, using colors to distinguish between high, medium, and low traffic hours (Figure 6.5a). With this, VRUs can now adapt their behavior and, for example, opt for a low-traffic hour to go cycling or go for a walk. Forecasts are available for the cities of Braga, Guimarães, and Porto. Each city has its own traffic forecasting model as detailed in Chapter 2.4. In addition, *SafeCity* implements a traffic alert service, which runs every morning as soon as the user unlocks the smartphone's screen, to warn users about high traffic hours for their city for the day. On the other hand, the personality assessment activity, as depicted in Figure 6.5c, allows users to perform and obtain an estimate of their personality traits (as described in Chapter 4.2).

Among other views, three more are worth describing. Figure 6.6a presents the notification raised by the traffic alert service, warning the user about intensive traffic flow at the city of Braga during the afternoon. It also presents the foreground notification of the boredom service, which is collecting the required data in the background to predict user boredom. The user can, at any time, stop this service. It behaves as described in Chapter 4.3, using the best Decision Tree model to make on-device inference, i.e., a Decision Tree is used by *SafeCity*'s boredom service. Although Decision Trees were not the best candidate model, it was the one deployed for one main reason: the model's size, in terms of memory usage, is significantly lower when compared to forest-based models (from a few kilobytes to several hundred megabytes). In fact, the deployment of forest-based models for on-device inference suffers from this major drawback. On the other hand, the loss in accuracy, from Decision Trees to forest-based models, is not significant. Another



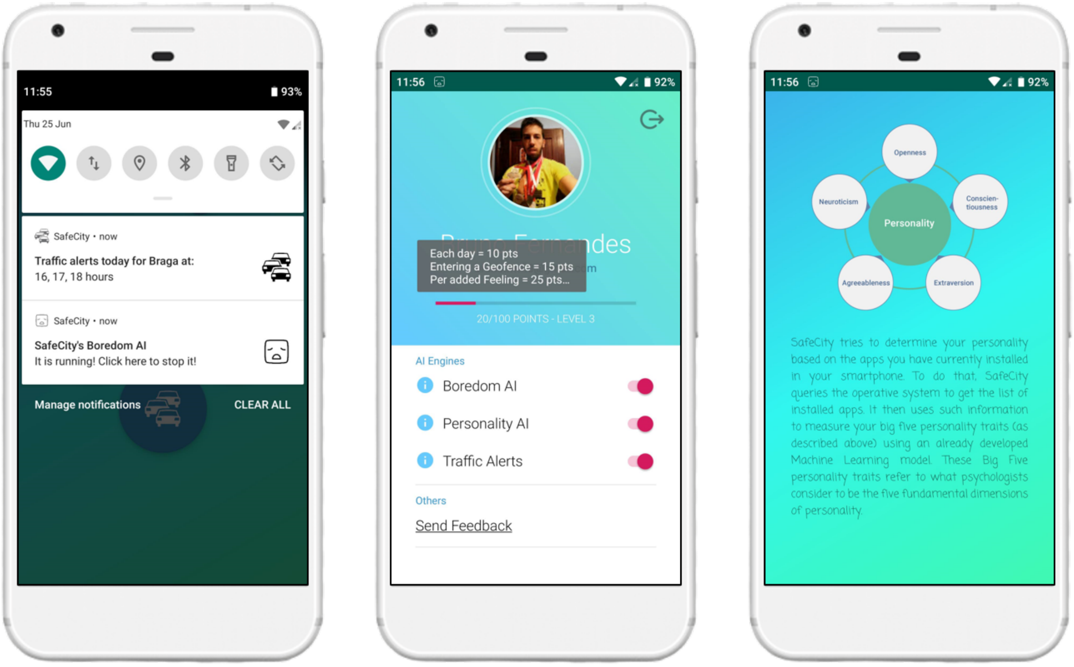
SafeCity's predictions for 2020-06-26

Extraversion = 46
Agreeableness = 56
Conscientious = 61
Stability = 50
Openness = 54

Rate each adjective as they apply to you (from Nothing to A Lot)! At end get your results!

(a) Traffic forecasting activity. (b) Main activity after login. (c) Personality assessment activity.

Figure 6.5: *SafeCity* views for the traffic forecasting and personality assessment services.



SafeCity • now

Traffic alerts today for Braga at:
16, 17, 18 hours

SafeCity • now

SafeCity's Boredom AI
It is running! Click here to stop it!

Each day = 10 pts
Entering a Geofence = 15 pts
Per added Feeling = 25 pts...

SafeCity tries to determine your personality based on the apps you have currently installed in your smartphone. To do that, SafeCity queries the operative system to get the list of installed apps. It then uses such information to measure your big five personality traits (as described above) using an already developed Machine Learning model. These Big Five personality traits refer to what psychologists consider to be the five fundamental dimensions of personality.

(a) SafeCity's notifications. (b) Gamification description. (c) Personality data explained.

Figure 6.6: *SafeCity* additional views.

interesting aspect of this deployment phase is related to the fact that even though trained Decision Trees are fairly simple models (ultimately, a set of *ifs*), deploying them on smartphones is a demanding task that is not fully described anywhere. The procedure to deploy such models may be summarized as follows:

1. Fit the tree-based model and create a Predictive Model Markup Language (PMML) file, which consists of a standardized XML-based predictive model interchange format;

2. Convert the PMML file into a serialized object file;

3. Load the serialized file into the mobile application using the *JPMML-Android* library;

4. Create an *evaluator* object and validate the loaded model;

5. Prepare the input map (data must be pre-processed according to the model's expected input);

6. Feed the model and obtain the prediction.

Figure 6.6b presents yet another feature of *SafeCity*, i.e., a gamification engine to support and promote user engagement. This gamification engine is part of a service that applies game design elements and game principles in a non-gaming context. It consists of a set of levels and points, which are to be given for the successful accomplishment of specific tasks. The goal is to have points that can be exchanged for specific discounts in cities' services and products. A long click in the user's current points shows which activities award points. Points must be collected to level up. There is currently no limit on the number of levels. To level up an user must collect 100 points. Points are awarded upon the following logic:

- 10 points per login per day;
- 15 points upon entering/exiting a geofence;
- 25 points per added feeling to the emotional map.

Figure 6.6c depicts a feature of *SafeCity* that is not yet fully implemented, i.e., *SafeCity* asks for user's permission to obtain the full list of applications the user has installed on his smartphone. The goal is to determine one's personality traits based only on the applications installed on the smartphone. More data are still required for *SafeCity* to create a production-level model that accurately achieves such a goal.

SafeCity aims to be a platform for safer and smarter cities. It may be used by anyone who desires to do so, even though its target audience is those more vulnerable on the road. The collaborative aspect of *SafeCity* is related to the community that uses it and about everyone's opinion on specific categories and certain zones of a city. The platform here described is the result of several years of research, acting as the answer to the last research question, ***Research Question 5****: Can a collaborative platform allow VRUs to have a better perception of their surroundings and become active actors of the city's ecosystem?*. The goal of *SafeCity* is precisely to make available a set of information about the current and future status of the environment where one stands. Community knowledge and forecasts allow VRUs to adapt their behavior upon dangerous situations, dangerous hours, and dangerous zones of a city. *SafeCity* closes **Task 5**, the final task of this PhD thesis.

SafeCity, besides originating a scientific paper on practical demonstrations of MAS ([209]), was also distinguished with the prestigious 1st prize of the **IBM Award of Scientific Excellence** at the 18th International Conference on Practical Applications of Agents and Multi-Agents Systems (PAAMS'20), held at the University of L'Aquila, in Italy.

6.1.2 Accomplishments and Hypothesis Validation

An hypothesis was elicited when devising the work plan. It went through several fields, being predicted that the use of features from the fields of AmI, Affective Computing, IoP, MAS, and ML would make it easier for people, in particular VRUs, to participate and have access to a set of information about the current and future status of their city. The answer to this hypothesis is set in terms of the answers to the elicited research questions. In fact, given its array of features, the research hypothesis was broken down into five main exploratory, evaluation, and action research questions, which aid validate the conducted research. In addition, each main research question was further split into smaller specific research questions, which were validated in the corresponding chapter in this PhD thesis. The conceived platform, *SafeCity*, makes use of multiple solutions' specimens produced throughout this PhD work.

The first research question was addressed in Chapter 2 of this thesis, where a MAS was designed and conceived, being made of software agents that autonomously gather city's data and that are responsible for fulfilling the entire ML pipeline, i.e., pre-process and prepare data, invoke a pre-trained model, and compile and execute it to obtain the desired output. This MAS, which goes from data collection to model conception, gave origin to *The Collector* and the *ML Architect*. *SafeCity* makes use of both to present actionable forecasts to its users, allowing them to be more aware of their surroundings. This research question had its validity settled by means of the produced and deployed prototypes, and their performance, which is assessed by the number of scientific papers, published and reviewed by scientific peers, that were based on data collected by the conceived MAS. In addition, a Best Application Paper award was also received in regard to the ability to forecast important environmental parameters, such as the UV index.

The second research question was addressed in Chapter 4, where we present the conducted research in regard to the Citizen Sensor and Affective Computing. The goal was to use non-invasive, non-intrusive approaches to measure the personality of subjects as well as the presence of boredom when using a smartphone. We want to give depth to the individual and consider him/her more than just a smartphone holder. The work realized under the umbrella of this research question allowed us to set a new method to assess personality using a simple ”game of words” where subjects are only required to select a set of adjectives that describe them the most. This opens new avenues to easily and quickly quantify personality traits without the need to conduct full personality assessment tests. ML models were also conceived to predict when a subject is bored, making use of smartphone's sensor data without the need for any intrusive features such as access to calls, messages, or biometric data. *SafeCity* makes use of the best decision tree for on-device inference of boredom. The performance of the conceived models, which show promising prospects in regard to the inference of personality and boredom, validate this research question, which was further validated by the scientific community in the form of the published scientific papers.

The third research question was addressed in Chapter 3 of this thesis, where we present a new sensor, the *Smart Scanner*, for passive crowd sensing, which requires no participation from people. With an interesting accuracy, the sensor is able to sense the density of people in indoor and outdoor environments, being also possible to unscramble the captured MAC addresses. This passive crowd sensing approach can open new ways to sense pedestrians' crossings and streets, allowing, for example, the conception of new methods to evolve current pandemic approaches to confinement by providing estimates on the number of people in certain zones during confinement periods. To complement this study, two active crowd sensing approaches were also conceived. *SafeCity* implements these two approaches, i.e., geofencing and the emotional map. The validity of this research question is assessed through means of the produced prototypes, being also validated by the published scientific papers, which were reviewed by scientific peers.

The fourth research question was addressed in Chapter 5, focusing on studying a novel approach to data storage in Smart Cities. The premise was based on the fact that data collected about Smart Cities' environments should be made available to all. This, combined with the need for an immutable, distributed, decentralized approach, put the spotlight on the blockchain technology. We went further and conceived a MAS blockchain, where intelligent agents are able to successfully implement ontologies, behaviors, and maintain the blockchain and its unique multi-chain architecture (required to reduce the computational load of the solution). A new protocol, the PoC, was implemented to control the quality of the data provided by the agents. This research question had its validity settled with the produced case study and the corresponding scientific publications.

Finally, the fifth and last research question was addressed in this chapter. Its answer is settled by *SafeCity*, the collaborative platform that makes use of the conducted research and the conceived prototypes to give its users a set of information about the current and future status of their city, allowing them to have a better perception of their surroundings and become active actors of the city's ecosystem. The validity of this last research question is established by the deployment of *SafeCity*, further legitimized by the scientific community. In addition, a prestigious award, i.e., the IBM Award of Scientific Excellence, was earn by *SafeCity*, which helps strengthen its validity.

With the research hypothesis being successfully established, it is worth highlighting that, throughout this work, we focused on empirical research through means of experiments conducted in the laboratory, field experiments, simulation environments, and proof-of-concepts. The followed methodology is entirely non-invasive, and preserves data and people's privacy. The goal has been, at all times, to set new solutions to improve our quality of life as human beings. Table 6.1 summarizes the location where the elicited tasks, research questions, and goals are addressed in this document.

Table 6.1: Location, in this thesis, where the elicited tasks, research questions, and goals are addressed.

Chapter	**Task**	**Research Question**	**Goal**
1	-	-	-
2	1 & 3	1	City Sensing
3	3	3	Crowd Sensing
4	2	2	Citizen Sensing
5	4	4	Data Storage
6	5	5	Collaborative Platform

6.1.3 Contributions to the State of the Art

From this PhD work, multiple contributions to the state of the art have arisen. In the fields of AmI, IoT, and sensorization, we have designed, conceived, and evaluated new sensors for passive crowd sensing. We have also proposed a geofencing approach to increase road safety as well as an emotional map that handles people's feelings in regard to certain zones and categories of a city. In fact, in the field of Affective Computing, we have proposed new methods to assess one's personality and to detect boredom when using smartphones. These new methods have been validated and accepted by the scientific community, and may open new avenues to improve the way we interact with smartphones and technological platforms. It is also worth highlighting the fact that a mobile application and a web platform have been developed for data collection since no dataset existed with the desired characteristics. We have published the collected datasets online, being available to others who desire to use such data[2].

In the field of MAS, we have contributed with an autonomous system that autonomously fulfills the entire ML pipeline. We have also contributed with important progress to blockchains, in particular, when

[2] https://github.com/brunofmf/Datasets4SocialGood

combined with MAS. In regard to the field of ML, we have conceived, tuned, and evaluated several different architectures and models for classification and time series forecasting, including Decision Trees, Random Forests, Gradient Boosted Trees, Support Vector Machines, MLPs, and LSTMs. Our experiments also clarified the better performance of LSTMs for time series forecasting when compared to statistical-based models, as well as their ability to accurately forecast traffic flow using a Recursive Multi-Step Multi-Variate approach.

From a technological and engineering point of view, several prototypes were conceived. The one deserving higher attention is *SafeCity*, the one aggregating multiple models, approaches, and techniques to increase road safety and foster Smart Cities.

6.2 Quantitative Results

To enable a quantifiable measurement process, the research plan encompassed the dissemination of the developed work in scientific conferences, journals, and prototype demonstrations. In addition, it also included the participation and organization of scientific events. The quantitative results are also evaluated in terms of the accomplished lectures, talks, awards, and supervision of students during their academic growth, both as engineers and as human beings.

6.2.1 Scientific Publications

The work described in this thesis was disseminated through the participation in several scientific events in the computer science field as well as through the publication of scientific manuscripts in international journals, book chapters, and conference proceedings.

6.2.1.1 International Journals

The work detailed in this PhD thesis has been documented in international scientific journals. The following list presents, in chronological order, published journal articles as well as those accepted to appear:

1. Fernandes, B., Silva, F., Analide, C. & Neves, J., Crowd Sensing for Urban Security in Smart Cities. *Journal of Universal Computer Science*, Vol. 24(3), pp. 302-321, 2018. doi: 10.3217/jucs-024-03-0302. Q2 - Computer Science (miscellaneous).

2. Rodrigues, M., Monteiro, V., Fernandes, B., Silva, F., Analide, C. & Santos, R., A gamification framework for getting residents closer to public institutions. *Journal of Ambient Intelligence and Humanized Computing*, Vol. 11, pp. 4569-4581, 2019. doi: 10.1007/s12652-019-01586-7. Q1 - Artificial Intelligence, Q1 - Informations Systems, Q1 - Telecommunications, Q1 - Computer Science (miscellaneous).

3. Fernandes, B., González-Briones, A., Novais, P., Calafate, M., Analide, C. & Neves, J., An Adjective Selection Personality Assessment Method using Gradient Boosting Machine Learning. *Processes*,

Vol. 8(5):618, 2020. doi: 10.3390/pr8050618. Q2 - Chemical Engineering, Q2 - Process Chemistry and Technology.

4. Fernandes, B., Silva, F., Alaiz-Moretón, H., Novais, P., Neves, J. & Analide, C., Long Short-Term Memory Networks for Traffic Flow Forecasting: Exploring Input Variables, Time Frames and Multi-Step Approaches. *INFORMATICA*, Vol. 31(4), pp. 723-749, 2020. doi: 10.15388/20-INFOR431. Q1 - Applied Mathematics, Q2 - Information Systems.

5. Fernandes, B., Diogo, A., Silva, F., Neves, J. & Analide, C., KnowLedger - A Multi-Agent System Blockchain for Smart Cities Data. *Artificial Intelligence Review*, 2021 (Accepted to appear). Q1 - Artificial Intelligence, Q1 - Language and Linguistics.

6.2.1.2 Book Chapters

The conducted experiments, the obtained results, and the drawn conclusions were also published and validated in book chapters, as follows:

1. Fernandes, B., Neves, J. & Analide, C., Envisaging the Internet of People: an Approach to the Vulnerable Road Users Problem. In De Paz, J., Julián, V., Villarrubia, G., Marreiros, G. & Novais, P. (eds) Ambient Intelligence – Software and Applications – 8th International Symposium on Ambient Intelligence (ISAmI 2017), Advances in Intelligent Systems and Computing, Vol. 615, pp. 104-111, Springer, 2017. doi: 10.1007/978-3-319-61118-1_14.

2. Fernandes, B., Neves, J., Vicente, H. & Analide, C., Towards Road Safety – A Social Perception. In Arai, K., Kapoor, S. & Bhatia, R. (eds) Intelligent Systems and Applications (IntelliSys 2018), Advances in Intelligent Systems and Computing, Vol. 869, pp. 47-57, Springer, 2018. doi: 10.1007/978-3-030-01057-7_4.

3. Fernandes, B., Vicente, H., Ribeiro, J., Analide, C. & Neves, J., Evolutionary Computation on Road Safety. In Juez, F. et al. (eds) Hybrid Artificial Intelligent Systems (HAIS 2018), Lecture Notes in Computer Science, Vol. 10870, pp. 647-657, Springer, 2018. doi: 10.1007/978-3-319-92639-1_54.

4. Neves, J., Dias, A., Morais, A., Fonseca, F., Loreto, P., Alves, V., Araújo, A., Machado, J., Fernandes, B., Ribeiro, J., Analide, C., Ferraz, F., Neves, J. & Vicente, H., Predicative Vagueness in Lung Metastases in Soft Tissue Sarcoma Screening. In Groza, A. & Prasath, R. (eds) Mining Intelligence and Knowledge Exploration (MIKE 2018), Lecture Notes in Computer Science, Vol. 11308, pp. 80-89, Springer, 2018. doi: 10.1007/978-3-030-05918-7_8.

5. Fernandes, B., Silva, F., Alaiz-Moretón, H., Novais, P., Analide, C. & Neves, J., Traffic Flow Forecasting on Data-Scarce Environments Using ARIMA and LSTM Networks. In Rocha, Á., Adeli, H., Reis, L. & Costanzo, S. (eds) New Knowledge in Information Systems and Technologies (WorldCIST'19), Advances in Intelligent Systems and Computing, Vol. 930, pp. 273-282, Springer, 2019. doi: 10.1007/978-3-030-16181-1_26.

6. Fernandes, B., Neves, J. & Analide, C., SafeCity: A platform for Safer and Smarter Cities. In Demazeau, Y., Holvoet, T., Corchado, J. & Costantini, S. (eds) Advances in Practical Applications of Agents, Multi-Agent Systems, and Trustworthiness (PAAMS 2020), Lecture Notes in Computer Science, Vol. 12092, pp. 412-416, Springer, 2020. doi: 10.1007/978-3-030-49778-1_37.

7. Oliveira, P., Fernandes, B., Analide, C. & Novais, P., Multi-step Ultraviolet Index Forecasting using Long Short-Term Memory Networks. In Dong, Y., Herrera-Viedma, E., Matsui, K., Omatsu, S., González-Briones, A. & Rodríguez-González, S. (eds) Distributed Computing and Artificial Intelligenc (DCAI 2020), Advances in Intelligent Systems and Computing, Vol. 1237, pp. 187-197, Springer, 2020. doi: 10.1007/978-3-030-53036-5_20.

8. Oliveira, P., Fernandes, B., Aguiar, F., Alcina, M., Analide, C. & Novais, P., A Deep Learning Approach to Forecast the Influent Flow in Wastewater Treatment Plants. In Analide, C., Novais, P., Camacho, D. & Yin, H. (eds) Intelligent Data Engineering and Automated Learning (IDEAL 2020), Lecture Notes in Computer Science, Vol. 12489, pp. 362-373, Springer, 2020. doi: 10.1007/978-3-030-62362-3_32.

9. Palmeira, J., Ramos, J., Silva, R., Fernandes, B. & Analide, C., A Machine Learning Approach to Forecast the Safest Period for Outdoor Sports. In Analide, C., Novais, P., Camacho, D. & Yin, H. (eds) Intelligent Data Engineering and Automated Learning (IDEAL 2020), Lecture Notes in Computer Science, Vol. 12489, pp. 185-196, Springer, 2020. doi: 10.1007/978-3-030-62362-3_17.

6.2.1.3 Conference Proceedings

A relevant way to validate results and obtain important insights from experts is to participate and publish in scientific conferences. The next list presents all articles published in conference proceedings during the research activities of this PhD thesis:

1. Fernandes, B., Neves, J. & Analide, C., Road Safety and Vulnerable Road Users: Internet of People Insights. In Proceedings of the 6th International Conference on Smart Cities and Green ICT Systems (SMARTGREENS 2017), pp. 311-316, 2017. doi: 10.5220/0006359303110316.

2. Diogo, A., Fernandes, B., Silva, A., Faria, J., Neves, J. & Analide, C., A Multi-Agent System Blockchain for a Smart City. In the special track BlockChain for the Internet of Things (BC4IOT) of the 3rd International Conference on Cyber-Technologies and Cyber-Systems (CYBER 2018), IARIA, pp. 68-73, 2018. ISBN: 978-1-61208-683-5.

3. Rio, A., Andrades, J., Benavides, C., Fernandes, B., Silva, F., Casteleiro-Roca, J., Rodriguez, I. & Alaiz-Moretón, H., Aprendizaje de sensorizado de entornos IoT mediante BeagleBone. In XL Jornadas de Automática: libro de actas, pp. 302-308, 2019. doi: 10.17979/spudc.9788497497169.302.

4. Fernandes, B., Vicente, H., Ribeiro, J., Capita, A., Analide, C. & Neves, J., Fully Informed Vulnerable Road Users - Simpler, Maybe Better. In Proceedings of the 21st International Conference on Information Integration and Web-based Applications & Services (iiWAS2019), Association for Computing Machinery, pp. 598–602, 2019. doi: 10.1145/3366030.3366089.

5. Ribeiro, J., Fernandes, B., Analide, C., Vicente, H. & Neves, J., Full Informed Road Networks Evaluation: Simpler, Maybe Better. In Proceedings of the 73rd Research World International Conference, pp. 46-52, 2019. http://www.worldresearchlibrary.org/proceeding.php?pid=3178.

6. Fernandes, B., Campos, C., Neves, J. & Analide, C., A Machine Learning Approach to Boredom Detection using Smartphone's Sensors. In Proceedings of the 9th European Starting AI Researchers' Symposium 2020 (STAIRS) co-located with the 24th European Conference on Artificial Intelligence (ECAI 2020), Vol. 2655, 8 pages, 2020. http://ceur-ws.org/Vol-2655/paper10.pdf.

7. Silva, C., Fernandes, B., Oliveira, P. & Novais, P., Using Machine Learning to Forecast Air and Water Quality. In Proceedings of the 13th International Conference on Agents and Artificial Intelligence (ICAART 2021), 2021 (Accepted to Appear).

6.2.2 Awards

Two prestigious scientific awards were received as result of the research conducted in this PhD thesis.

The work performed on the UV domain gave origin to a scientific paper entitled as "Multi-step Ultraviolet Index Forecasting using Long Short-Term Memory Networks", being distinguished with the **Best Application Paper** award by the program committee of the 17th International Conference on Distributed Computing and Artificial Intelligence (DCAI'20), held at L'Aquila, Italy, in 2020.

SafeCity, which gave origin to a demonstration paper entitled as "SafeCity: A platform for Safer and Smarter Cities", was distinguished with the 1st prize of the **IBM Award of Scientific Excellence** at the 18th International Conference on Practical Applications of Agents and Multi-Agents Systems (PAAMS'20), held at L'Aquila, Italy, in 2020.

6.2.3 Participation in Scientific Events

The main point of contact between domain experts is at scientific events, where studies are shared and discussed with the scientific community. During the work developed in this PhD thesis, several scientific events were attended. Due to the COVID-19 pandemic, many scientific gatherings happened in an online format, during the year 2020. Those more relevant are:

- 6th International Conference on Smart Cities and Green ICT Systems (SMARTGREENS), in 2017, at Porto, Portugal;

- 8th International Symposium on Ambient Intelligence (ISAmI), in 2017, at Porto, Portugal;

- CAR2X - New R&D challenges on Intelligent Transportation Systems, in 2017, at Braga, Portugal;
- Intelligent Systems Conference (IntelliSys), in 2018, at London, United Kingdom;
- 7th World Conference on Information Systems and Technologies (WorldCIST), in 2019, at La Toja Island, Spain;
- 1st Interdisciplinary Summer School on Artificial Intelligence (ISSAI), in 2019, at Vila Nova de Cerveira, Portugal;
- 18th International Conference on Practical Applications of Agents and Multi-Agent Systems (PAAMS), in 2020, hosted at L'Aquila, Italy;
- 17th International Conference on Distributed Computing and Artificial Intelligence (DCAI), in 2020, hosted at L'Aquila, Italy;
- 21st International Conference on Intelligent Data Engineering and Automated Learning (IDEAL), in 2020, hosted at Guimarães, Portugal;
- 24th European Conference on Artificial Intelligence (ECAI), in 2020, hosted at Salamanca, Spain.

6.2.4 Organization of Scientific Events

The organization of scientific events was carried out as part of the *Program Committee* and the *Local Organizing Committee* for the following events:

- Citizen-Centric Smart Cities Services (CCSCS'2019), Rabat, Morocco, 24-27 June, 2019;
- 10th International Symposium on Ambient Intelligence (ISAmI), Ávila, Spain, 26-28 June, 2019;
- 11th International Symposium on Ambient Intelligence (ISAmI), L'Aquila, Italy, 7-9 October, 2020;
- 1st International Conference on Disruptive Technologies Tech Ethics and Artificial Intelligence (DiT-TEt), Salamanca, Spain, 4–6 November, 2020;
- Citizen-Centric Smart Cities Services (CCSCS'2020), Madrid, Spain, 20-23 June, 2020;
- 21st International Conference on Intelligent Data Engineering and Automated Learning (IDEAL), Guimarães, Portugal, 4-6 November, 2020;
- Sustainable Smart Cities and Territories International Conference (SSCt), Doha, Qatar, 27-29 April, 2021.

And as an *invited reviewer* of several International Scientific Journals, as follows:

- IEEE Internet of Things Journal (Q1 - Information Systems, Q1 - Computer Science Applications);

- IEEE Sensors Journal (Q1 - Electrical and Electronic Engineering, Q1 - Instrumentation);
- Future Generation Computer Systems (Q1 - Computer Networks and Communications, Q1 - Software);
- PeerJ (Q1 - Computer Science (miscellaneous));
- Journal of Ambient Intelligence and Humanized Computing (Q1 - Artificial Intelligence, Q1 - Informations Systems, Q1 - Telecommunications, Q1 - Computer Science (miscellaneous)).

6.2.5 Research Stay

In the context of the research work developed during the doctoral program, a research period of one month was spent in Madrid, Spain, at the Complutense University of Madrid. This stay, performed in coordination with the *RiseWise* project, aimed to discuss the problems of women with disabilities and the drawbacks that such people face, still today, in our society. For that, meetings were held with AFADIS-UCM, a Spanish non-profit association that aims to promote and carry out activities that are aimed at improving the quality of life of people with disabilities. AFADIS-UCM has been providing support to many disabled people, with the great majority being women between 30 and 40 years old. The main impairments are associated with reduced mobility, and visual and/or sensory impairments. Focusing on the vulnerability of disabled people on the road, it was interesting to note that historical heritage protection of buildings causes significant problems to people with disabilities since it is not allowed, at least in a timely manner, the installation of support infrastructures such as ramps or sidewalks. In addition, it was found that Madrid uses a sound system, in the traffic lights, to help people with visual impairments to cross the road. However, many of those are not working and are not fixed. Hence, it was discussed the inclusion of a feedback system in *SafeCity* to alert authorities about malfunctioning traffic lights as well as the inclusion of geofences depicting crowed zones and roads under maintenance/construction, in future versions of the platform.

6.2.6 Lectures

The candidate lectured, in the academic years of 2018/2019, 2019/2020, and 2020/2021, the Curricular Units of ”Similarity Based Systems”, ”Connective Systems and Classifiers” and ”Autonomous Systems” to the Masters’ Courses of Informatics Engineering, Systems Engineering, and Mathematics and Computation.

6.2.7 Invited Talks

The research conducted in this PhD thesis was also disseminated via invited talks, as described in the following list:

- *The Internet of People approach to Road Safety and Vulnerable Road Users in Smart Cities*. Invited talk performed in February 2018 to Erasmus PhD candidates;

- *A Look at Blockchain - The paradigm, the protocol and the technology*. Invited talk performed in April 2018 within the ”Basto Tech MeetUP” event;
- *The Internet of People approach to Road Safety and Vulnerable Road Users in Smart Cities*. Invited talk performed in September 2018 to PhD candidates of the Doctoral Program in Informatics;
- *The Internet of People approach to Road Safety and Vulnerable Road Users in Smart Cities*. Invited talk performed in January 2020 within the scope of the Curricular Unit ”Advanced Artificial Intelligence” of the MAPi Doctoral Program in Computer Science.

6.2.8 Supervision of Students

The candidate has successfully co-supervised several master’s students that performed their work in the context of this PhD thesis, both during their dissertation as well as in the context of curricular units that belong to the Master’s Course in Informatics Engineering. The next lines provide a summary of the work performed by each student.

6.2.8.1 Master’s Dissertations

a. Adriana Eliana Fernandes Pereira
Physical and Psychological Health Monitoring Using Mobile Devices.
Academic year: 2018/2019.
In co-supervision with Professor Cesar Analide.
Adriana’s work consisted of designing and developing a mobile application that more than keeping track of physical aspects of its users would also track their psychological health. To achieve such a goal, users were required to use the conceived app to perform Saucier’s Mini Marker test periodically. In addition, a set of biometric data were also captured and displayed to the user, with the goal being to find and understand relations between physical and psychological health. Adriana completed her master’s degree in 2019.

b. André Sousa Diogo
A Multi-Agent System Blockchain for a Smart City.
Academic year: 2018/2019.
In co-supervision with Professor Cesar Analide.
Blockchain has been arousing the interest of many distinct parties. More than a paradigm, Blockchain is a data structure, a technology, and a protocol, all at the same time. Diogo has been focused on working with a MAS Blockchain to store and handle distinct data, with software agents building and securing the Blockchain, allowing the possibility of non-monetized rewards and the use of gamification to foster mining. Diogo is concluding his master’s degree.

c. Miguel Campos Calafate Carneiro Perdigão

A Machine Learning Approach to the Big Five Personality Test.

Academic year: 2018/2019.

In co-supervision with Professor Cesar Analide.

Miguel focused on working on a reduction of Saucier's Mini Marker personality assessment test to a "game of words", where instead of rate 40 adjectives the user is only required to select those describing him the most. After the data collection process, Miguel worked on data augmentation processes. Afterwards, a set of supervised ML models were experimented. Miguel completed his master's degree in 2019.

d. Pedro Daniel Gomes Fonseca

Smart City Geofences for Vulnerable Road Users.

Academic year: 2018/2019.

In co-supervision with Professor Cesar Analide.

While performing his dissertation, Pedro experimented several possibilities to design and implement geofences, i.e., virtual perimeters for real geographical locations in which it is possible to detect user entrance or exit. The use of an API revealed to be the most appropriate way to conceive geofences in Smart Cities. The case study focused on the creation and interaction of geofences in specific points of interest for VRUs, signaling bad visibility, roadblocks, or construction areas, among others. Pedro completed his master's degree in 2019.

e. Carlos José Gomes Campos

A Machine Learning Approach to Boredom Detection in Smartphones.

Academic year: 2019/2020.

In co-supervision with Professor Cesar Analide.

Acknowledging when a user is bored or killing his time in the smartphone would allow the possibility of suggesting a better use of such idle moments, encouraging the user to go through his to-do lists, share culturally relevant material, or even notify the user with accurate and relevant ads, making a positive impact both with the user and companies (are bored people more prone to ads?). A mobile app has already been developed for data collection. Carlos is focused on improving the developed app in order to increase the number of collected features. He is also experimenting several ML models for boredom detection. Carlos completed his master's degree in 2020.

f. Carolina Oliveira da Silva

Using Machine Learning to enhance Environmental Sustainability.

Academic year: 2019/2020.

In co-supervision with Professor Paulo Novais.

In her work, Carolina is operating a set of environmental data, with the goal being to enhance environmental sustainability. Among the available data, one may find weather snapshots, pollution values for several pollutants, and data regarding the urban water cycle of several Portuguese cities. The main goal is to conceive, tune, and experiment several ML models to provide accurate forecasts regarding several environmental values. Carolina completed her master's degree in 2020.

g. David Daniel Pinto Coelho Kramer
Crowd Sensing for Urban Security in Smart Cities.
Academic year: 2019/2020.
In co-supervision with Professor Cesar Analide.
Smart Scanners have already been conceived to passively detect devices such as smartphones or smartwatches, being able to accurately estimate the density of people in a certain area. These sensors gather data about smartphones and wearable devices, including the MAC address, the RSSI, and the timestamp. The work of David focuses on using the Smart Scanners to sense certain areas and, with the obtained data, estimate the distance of a person to the sensor, implement algorithms for crowd detection and control, and conceive DL models to forecast the expected number of people at the sensed areas. David completed his master's degree in 2020.

h. Gustavo da Costa Gomes
A City's Map from People's Feelings - A Recommender System.
Academic year: 2019/2020.
In co-supervision with Professor Cesar Analide.
Gustavo's dissertation is based on the fact that each city has its own characteristics, defined by its own people. On the one hand, there may be spots that are best suited for shopping or spots that are best suited for sporting activities. On the other hand, other places may be dangerous for people, whether due to high traffic or to high crime, for instance. The goal of this work is to conceive and implement a recommender system that, using techniques such as collaborative and content-based filtering, suggests new zones of a city to the user based on his/her profile. Gustavo is currently studying for a master's degree.

i. Bruno Manuel Chaves Martins
Cloud-based IoT as a Service.
Academic year: 2020/2021.
In co-supervision with Professor Paulo Novais.
Cloud computing is a major pillar of Software as a Service strategies because only with a robust infrastructure companies are able to provide a service with the level of quality that is needed nowadays. The integration of IoT and the cloud is a challenge in itself as a whole new array of problems need to be addressed. Hence, the goals of this study are to research and evaluate different kinds of approaches to IoT and Software as a Service based on the cloud, and implement an end-to-end cloud-based IoT as a Service system using sensors that are as less intrusive as possible in work-spaces. Bruno is currently studying for a master's degree.

j. Caroline Rodrigues

Data Science in Football.

Academic year: 2020/2021.

In co-supervision with Professor Cesar Analide.

Caroline's dissertation is focused on exploring, studying, and analyzing datasets regarding the performance of Portuguese football teams in *Primeira Liga Portuguesa*. These datasets include, among others, data regarding the overall performance of a team during each season (a total of 54 features per season) as well as the performance of each individual player (48 features per player) of a team. As a summary, the goals are to explore, study, prepare, and analyze football data; conceive methods to complement the data collection process; and conceive and implement data visualization methods embedded into a dashboarding platform. Caroline is currently studying for a master's degree.

k. Helder Daniel Domingues Gonçalves

A Dataset Collector from Public APIs.

Academic year: 2020/2021.

In co-supervision with Professor Cesar Analide.

Helder's goal is to conceive a platform that allows scholars to access prominent data sources with customized filters in order to have personalized, persistent, and updated datasets. These datasets must be updated with the frequency specified by the user so as to provide the most up-to-date results possible. The conceived platform must also be able to provide dashboards specific to each data source, displaying high level metrics as well as the option to download filtered subsets of data. It is also expected the development of multiple data visualization methods embedded into the platform. Helder is currently studying for a master's degree.

l. Rui Nuno Vilaça Ribeiro

A Data Science Approach to Portuguese Road Accidents' Reports.

Academic year: 2020/2021.

In co-supervision with Professor Cesar Analide.

There are multiple ways to identify dangerous roads such as analyzing historical data, calculating accidents' rate, and analyzing roads' conditions, among many others. For national authorities, the detection of blackspots is yet another important measure, which helps identify locations with a higher number of accidents or fatalities. Hence, the goal of this dissertation is to explore, study, and analyze Portuguese road accidents' reports produced by the Portuguese *Autoridade Nacional de Segurança Rodoviária*. The goal is to recognize blackspots, discover spatial and temporal relations, and understand the most problematic (in terms of the number and severity of accidents) days and hours on the road. Rui is currently studying for a master's degree.

m. Telmo André Moreira Cardoso

A Recommender System for a Web Book Shop.

Academic year: 2020/2021.

In co-supervision with Professor Paulo Novais.

Telmo's goals are to conceive and incorporate a system for recommending books in an already existing web application for online sales in order to improve the user experience. In particular, the goal is to adopt content-based and collaborative filtering techniques; solve the cold start problem associated with recently added users; solve gray sheep problems for books that belong to the long tail; and develop a hybrid recommender system that tackles the current limitations of the existing platform. Telmo is currently studying for a master's degree.

6.2.8.2 Informatics Engineering Laboratory

a. André Diogo, António Silva, & José Faria

A Multi-Agent System BlockChain for a Smart City.

Academic year: 2017/2018.

In co-supervision with Professor Cesar Analide.

b. Diogo Rodrigues, Luís Bouça, & Hugo Gonçalves

A City's map from People's Feelings.

Academic year: 2018/2019.

In co-supervision with Professor Cesar Analide.

c. Carlos Campos, José Oliveira, & Vitor Castro

Boredom Detection Using Mobile Phone Data.

Academic year: 2018/2019.

In co-supervision with Professor Cesar Analide.

d. Daniel Vieira, Joel Morais, & Miguel Cunha

Empowering the Citizen Sensor in Smart Cities.

Academic year: 2018/2019.

In co-supervision with Professor Cesar Analide.

e. João Palmeira, José Ramos, & Rafael Silva

A Model for the Quantification of Running and Cycling Safety.

Academic year: 2019/2020.

In co-supervision with Professor Cesar Analide.

f. João Sá, Luís Peixoto, & Tiago Silva
An Odyssey on Recurrent Neural Networks for Traffic Flow Forecasting.
Academic year: 2019/2020.
In co-supervision with Professor Cesar Analide.

g. Henrique Pereira, Pedro Moreira, & Tifany Silva
An open-source Gamification Engine.
Academic year: 2019/2020.
In co-supervision with Professor Cesar Analide.

6.3 Future Work

The way to increment one's safety at Smart Cities is not monolithic. Instead, it is made of multiple contributions that, when summed, can make a difference. Since the beginning, it was clear that this PhD work would be a multidisciplinary one, encompassing several distinct fields and domains. From AmI to ML and DL, going through sensorization and blockchains, several distinct but complementary approaches were considered, designed, and experimented. Hence, it is reasonable to argue that a multitude of possible future paths lay ahead, including:

- Extend the sensorization abilities of *The Collector*, allowing the conception of state of the art models to extract further knowledge from the city's environment. This includes, but is not limited to, air quality indexes, urban water quality, solid waste management, and traffic flow patterns, among others;

- Implement crowd sensing methods to enhance urban security. In particular, given the pandemic situation we are currently experiencing (due to COVID-19), the expectation is to use the conceived crowd sensing methods to evolve pandemic approaches to confinement. The goal is to have estimates of the number of people in certain zones during confinement periods without requiring people to install, or carry, software or sensors, allowing authorities to act accordingly when there is a high number of people in a small area;

- Evolve the conceived emotional map from people's feelings. For that, one is expected to design and implement a recommender system that, using techniques such as top-N, clustering, and collaborative and content-based filtering, suggests new zones of a city to users;

- Proceed with the development of *KnowLedger*, focusing on enhancing the current solution and on performing rigorous tests. Another goal consists of implementing and testing a blacklist solution to deny the access of malicious agents to the blockchain;

- Study, compare, and benchmark the performance of several distinct DL models for time series forecasting, focusing on MLPs, RNNs, LSTMs, Gated Recurrent Units (GRUs), CNNs, and hybrid

CNNs-LSTMs models. This may open new avenues to enhance the performance of existing time series forecasting solutions;

- Further disseminate, promote, and expand *SafeCity*, allowing its users to participate and have access to a set of information about the current and future status of their city. In addition, it is expected the inclusion of a feedback system to alert authorities about problematic situations in the city's infrastructure as well as to continue the data collection process to assess personality traits based on the applications that are installed on a smartphone.

Finally, it is worth highlighting that the research work carried out in this PhD thesis gave origin to several innovative research lines, grounded on new ways to acquire non-invasive, multi-modal data for boredom detection, personality assessment, and city and crowd sensing, which can be further extended in multiple distinct directions.

Bibliography

[1] United Nations, Department of Economic and Social Affairs. *The World's Cities in 2018 — Data Booklet*. 2018. url: https://www.un.org/en/events/citiesday/assets/pdf/the_worlds_cities_in_2018_data_booklet.pdf.

[2] World Health Organization. *Global status report on road safety 2018*. World Health Organization (WHO), 2018. isbn: 9789241565684.

[3] World Health Organization. *Health statistics and information systems - Disease burden and mortality estimates*. https://www.who.int/healthinfo/global_burden_disease/estimates/en/. [Online; accessed April, 2020].

[4] L. Costa, H. Silva, J. Oliveira, and S. Fernandes. "Incorporation of Waste Plastic in Asphalt Binders to Improve their Performance in the Pavement." *International Journal of Pavement Research and Technology.* 6.4 (2013), pp. 457–464. doi: 10.6135/ijprt.org.tw/2013.6(4).457.

[5] Y. Zou and A. Tarko. "An insight into the performance of road barriers – redistribution of barrier-relevant crashes." *Accident Analysis & Prevention.* 96 (2016), pp. 152–161. doi: 10.1016/j.aap.2016.07.022.

[6] X. Ma, Z. Tao, Y. Wang, H. Yu, and Y. Wang. "Long short-term memory neural network for traffic speed prediction using remote microwave sensor data." *Transportation Research Part C: Emerging Technologies.* 54 (2015), pp. 187–197. doi: 10.1016/j.trc.2015.03.014.

[7] Z. Zhao, W. Chen, X. Wu, P. Chen, and J. Liu. "LSTM network: a deep learning approach for short-term traffic forecast." *Intelligent Transport Systems.* 11.2 (2017), pp. 68–75. doi: 10.1049/iet-its.2016.0208.

[8] B. Fernandes, F. Silva, H. Alaiz-Moretón, P. Novais, C. Analide, and J. Neves. "Traffic Flow Forecasting on Data-Scarce Environments Using ARIMA and LSTM Networks." *Advances in Intelligent Systems and Computing.* 930 (2019), pp. 273–282. doi: 10.1007/978-3-030-16181-1_26.

[9] European Commission. *Directive 2010/40/EU of the European Parliament and of the Council of 7 July 2010 on the framework for the deployment of Intelligent Transport*

Systems in the field of road transport and for interfaces with other modes of transport. [Retrieved from http://data.europa.eu/eli/dir/2010/40/2018-01-09]. Official Journal of the European Union, 2018.

[10] A. Constant and E. Lagarde. "Protecting Vulnerable Road Users from Injury." *PLoS Medicine.* 7.3 (2010). doi: 10.1371/journal.pmed.1000228.

[11] European Commission. *2018 road safety statistics: what is behind the figures?* Memo. [Retrieved from https://ec.europa.eu/commission/presscorner/detail/en/MEMO_19_1990]. European Commission, 2019.

[12] R. Weast. "Temporal factors in motor-vehicle crash deaths: Ten years later." *Journal of Safety Research.* 65 (2018), pp. 125–131. doi: 10.1016/j.jsr.2018.02.011.

[13] Y. Guo, A. Osama, and T. Sayed. "A cross-comparison of different techniques for modeling macro-level cyclist crashes." *Accident Analysis and Prevention.* 113 (2018), pp. 38–46. doi: 10.1016/j.aap.2018.01.015.

[14] G. Prati, L. Pietrantoni, and F. Fraboni. "Using data mining techniques to predict the severity of bicycle crashes." *Accident Analysis and Prevention.* 101 (2017), pp. 44–54. doi: 10.1016/j.aap.2017.01.008.

[15] K. Skouby and P. Lynggaard. "Smart home and smart city solutions enabled by 5G, IoT, AAI and CoT services." In: *International Conference on Contemporary Computing and Informatics (IC3I)*. 2014, pp. 874–878. doi: 10.1109/IC3I.2014.7019822.

[16] B. Fernandes, J. Neves, and C. Analide. "Road Safety and Vulnerable Road Users: Internet of People Insights." In: *Proceedings of the 6th International Conference on Smart Cities and Green ICT Systems (SMARTGREENS 2017)*. 2017, pp. 311–316. doi: 10.5220/0006359303110316.

[17] K. Rose, S. Eldridge, and L. Chapin. *The Internet of Things: An Overview*. The Internet Society (ISOC), 2015.

[18] E. Kosmatos, N. Tselikas, and A. Boucouvalas. "Integrating RFIDs and Smart Objects into a Unified Internet of Things Architecture." *Advances in Internet of Things.* 1 (2011), pp. 5–12. doi: 10.4236/ait.2011.11002.

[19] A. Zanella, N. Bui, A. Castellani, L. Vangelista, and M. Zorzi. "Internet of Things for Smart Cities." *IEEE Internet of Things Journals.* 1 (2014), pp. 22–32. doi: 10.1109/JIOT.2014.2306328.

[20] A. Gabbai. *Kevin Ashton Describes the Internet of Things.* http://www.smithsonianmag.com/innovation/kevin-ashton-describes-the-internet-of-things-180953749/?no-ist. [Online; accessed April, 2020].

[21] S. Madakam, R. Ramaswamy, and S. Tripathi. "Internet of Things (IoT): A Literature Review." *Journal of Computer and Communications.* 3 (2015), pp. 164–173. doi: 10.4236/jcc.2015.35021.

[22] A. Pintus, D. Carboni, A. Serra, and A. Manchinu. "Humanizing the Internet of Things - Toward a Human-centered Internet-and-web of Things." In: *Proceedings of the 11th International Conference on Web Information Systems and Technologies (WEBIST-2015).* 2015, pp. 498–503. doi: 10.5220/0005475704980503.

[23] J. Hernández-Muñoz, J. Vercher, L. Muñoz, J. Galache, M. Presser, L. H. Gómez, and J. Pettersson. "Smart Cities at the Forefront of the Future Internet." *The Future Internet Assembly.* 6656 (2011), pp. 447–462. doi: 10.1007/978-3-642-20898-0_32.

[24] J. Miranda, N. Makitalo, J. Garcia-Alonso, J. Berrocal, T. Mikkonen, C. Canal, and J. Murillo. "From the Internet of Things to the Internet of People." *IEEE Internet Computing.* 19.2 (2015), pp. 40–47. doi: 10.1109/MIC.2015.24.

[25] B. Fernandes, J. Neves, and C. Analide. "Envisaging the Internet of People: an Approach to the Vulnerable Road Users Problem." *Advances in Intelligent Systems and Computing.* 615 (2017), pp. 104–111. doi: 10.1007/978-3-319-61118-1_14.

[26] C. Harrison, B. Eckman, R. Hamilton, P. Hartswick, J. Kalagnanam, J. Paraszczak, and P. Williams. "Foundations for Smarter Cities." *IBM Journal of Research and Development.* 54.4 (2010), pp. 1–16. doi: 10.1147/JRD.2010.2048257.

[27] M. Roscia, M. Longo, and G. Lazaroiu. "Smart City by multi-agent systems." In: *International Conference on Renewable Energy Research and Applications (ICRERA).* 2013, pp. 371–376. doi: 10.1109/ICRERA.2013.6749783.

[28] L. Anthopoulos and P. Fitsilis. "Exploring architectural and organizational features in smart cities." In: *16th International Conference on Advanced Communication Technology.* 2014, pp. 190–195. doi: 10.1109/ICACT.2014.6778947.

[29] S. Allwinkle and P. Cruickshank. "Creating Smarter Cities: An Overview." *Journal of Urban Technology.* 18 (2011), pp. 1–16. doi: 10.1080/10630732.2011.601103.

[30] R. Hollands. "Will the real smart city please stand up?" *City.* 12.3 (2008), pp. 303–320. doi: 10.1080/13604810802479126.

[31] T. Nam and T. Pardo. "Conceptualizing Smart City with Dimensions of Technology, People, and Institutions." In: *Proceedings of the 12th Annual International Digital Government Research Conference: Digital Government Innovation in Challenging Times.* 2011, pp. 282–291. doi: 10.1145/2037556.2037602.

[32] R. Giffinger, C. Fertner, H. Kramar, R. Kalasek, N. Pichler-Milanovic, and E. Meijers. *Smart cities Ranking of European medium-sized cities*. Final Report. [Retrieved from www.smart-cities.eu]. Vienna University of Technology, 2007.

[33] H. Chourabi, T. Nam, S. Walker, J. Gil-Garcia, S. Mellouli, K. Nahon, T. Pardo, and H. Scholl. “Understanding Smart Cities: An Integrative Framework.” In: *45th Hawaii International Conference on System Science (HICSS)*. 2012, pp. 2289–2297. doi: 10.1109/HICSS.2012.615.

[34] J. Bergh and S. Viaene. “Key Challenges for the Smart City: Turning Ambition into Reality.” In: *48th Hawaii International Conference on System Sciences (HICSS)*. 2015, pp. 2385–2394. doi: 10.1109/HICSS.2015.642.

[35] H. Scholl, K. Barzilai-Nahon, J. Ann, O. Popova, and B. Re. “E-Commerce and E-Government: How Do They Compare? What Can They Learn from Each Other?” In: *42nd Hawaii International Conference on System Sciences (HICSS 09)*. 2009, pp. 1–10. doi: 10.1109/HICSS.2009.169.

[36] R. Harmon, E. Castro-Leon, and S. Bhide. “Smart cities and the Internet of Things.” In: *Portland International Conference on Management of Engineering and Technology (PICMET)*. 2015, pp. 485–494. doi: 10.1109/PICMET.2015.7273174.

[37] F. Silva, D. Cuevas, C. Analide, J. Neves, and J. Marques. “Sensorization and Intelligent Systems in Energetic Sustainable Environments.” *Intelligent Distributed Computing VI.* (2013), pp. 199–204. doi: 10.1007/978-3-642-32524-3_25.

[38] A. Aztiria, A. Izaguirre, and J. Augusto. “Learning patterns in ambient intelligence environments: a survey.” *Artificial Intelligence Review.* 34.1 (2010), pp. 35–51. doi: 10.1007/s10462-010-9160-3.

[39] S. Mehrotra and R. Dhande. “Smart cities and smart homes: From realization to reality.” In: *International Conference on Green Computing and Internet of Things (ICGCIoT)*. 2015, pp. 1236–1239. doi: 10.1109/ICGCIoT.2015.7380652.

[40] M. Miranda, J. Duarte, A. Abelha, J. Machado, J. Neves, and J. Neves. “Interoperability in healthcare.” In: *Proceedings of the European Simulation and Modelling Conference (ESM 2010)*. 2010, pp. 261–265. isbn: 978-90-77381-57-1.

[41] J. Cano, C. Jimenez, and S. Zoughbi. “A smart city model based on citizen-sensors.” In: *IEEE First International Smart Cities Conference (ISC2)*. 2015, pp. 1–2. doi: 10.1109/ISC2.2015.7366177.

[42] B. Erickson, P. Korfiatis, Z. Akkus, and T. Kline. “Machine Learning for Medical Imaging.” *RadioGraphics.* 37.2 (2017). doi: 10.1148/rg.2017160130.

[43] B. Fernandes, C. Campos, J. Neves, and C. Analide. "A Machine Learning Approach to Boredom Detection using Smartphone's Sensors." In: *9th European Starting AI Researchers' Symposium (STAIRS) co-located with the 24th European Conference on Artificial Intelligence (ECAI 2020)*. 2020, p. 8. url: http://ceur-ws.org/Vol-2655/paper10.pdf.

[44] J. Biamonte, P. Wittek, N. Pancotti, P. Rebentrost, N. Wiebe, and S. Lloyd. "Quantum machine learning." *Nature*. 549 (2017), pp. 195–202. doi: 10.1038/nature23474.

[45] B. Fernandes, F. Silva, H. Alaiz-Moretón, P. Novais, J. Neves, and C. Analide. "Long Short-Term Memory Networks for Traffic Flow Forecasting: Exploring Input Variables, Time Frames and Multi-Step Approaches." *INFORMATICA*. 31.4 (2020), pp. 723–749. doi: 10.15388/20-INFOR431.

[46] A. Sallab, M. Abdou, E. Perot, and S. Yogamani. "Deep reinforcement learning framework for autonomous driving." *Electronic Imaging*. 2017.19 (2017), pp. 70–76. doi: 10.2352/ISSN.2470-1173.2017.19.AVM-023.

[47] P. Rajpurkar, J. Irvin, K. Zhu, B. Yang, H. Mehta, T. Duan, D. Ding, A. Bagul, C. Langlotz, K. Shpanskaya, M. Lungren, and A. Ng. "CheXNet: Radiologist-Level Pneumonia Detection on Chest X-Rays with Deep Learning." *arXiv e-prints*., arXiv:1711.05225 (2017), arXiv:1711.05225. arXiv: 1711.05225 [cs.CV].

[48] R. Picard. *Affective Computing*. Technical Report. [Retrieved from https://affect.media.mit.edu/pdfs/95.picard.pdf]. M.I.T Media Laboratory Perceptual Computing Section, 1995.

[49] C. Barlett and C. Anderson. "Direct and indirect relations between the Big 5 personality traits and aggressive and violent behavior." *Personality and Individual Differences*. 52.8 (2012), pp. 870–875. doi: 10.1016/j.paid.2012.01.029.

[50] S. Rothmann and E. Coetzer. "The big five personality dimensions and job performance." *SA Journal of Industrial Psychology*. 29.1 (2003). doi: 10.4102/sajip.v29i1.88.

[51] T. Orzeck and E. Lung. "Big-five personality differences of cheaters and non-cheaters." *Current Psychology*. 24 (2005), pp. 274–286. doi: 10.1007/s12144-005-1028-3.

[52] A. Kazdin. *Encyclopedia of psychology*. Vol. 3. American Psychological Association, 2000. isbn: 9781557981875.

[53] Y. Mehta, N. Majumder, A. Gelbukh, and E. Cambria. "Recent trends in deep learning based personality detection." *Artificial Intelligence Review* (2019). doi: 10.1007/s10462-019-09770-z.

[54] N. Majumder, S. Poria, A. Gelbukh, and E. Cambria. "Deep Learning-Based Document Modeling for Personality Detection from Text." *IEEE Intelligent Systems*. 32.2 (2017), pp. 74–79. doi: 10.1109/MIS.2017.23.

[55] J. Yu and K. Markov. “Deep learning based personality recognition from Facebook status updates.” In: *IEEE 8th International Conference on Awareness Science and Technology (iCAST)*. 2017, pp. 383–387. doi: 10.1109/ICAwST.2017.8256484.

[56] J. Cohn and F. De la Torre. “Automated face analysis for affective computing.” *The Oxford handbook of affective computing.* (2015), pp. 131–150. doi: 10.1093/oxfordhb/9780199942237.013.020.

[57] S. Das, A. Halder, P. Bhowmik, A. Chakraborty, A. Konar, and A. Nagar. “Voice and Facial Expression Based Classification of Emotion Using Linear Support Vector Machines.” In: *2nd International Conference on Developments in eSystems Engineering*. 2009, pp. 377–384. doi: 10.1109/DeSE.2009.9.

[58] B. Fernandes, A. González-Briones, P. Novais, M. Calafate, C. Analide, and J. Neves. “An Adjective Selection Personality Assessment Method Using Gradient Boosting Machine Learning.” *Processes.* 8.5 (2020), p. 618. doi: 10.3390/pr8050618.

[59] E. Cambria. “Affective Computing and Sentiment Analysis.” *IEEE Intelligent Systems.* 31.2 (2016), pp. 102–107. doi: 10.1109/MIS.2016.31.

[60] M. Wooldridge. *An Introduction to MultiAgent Systems*. John Wiley & Sons., 2002. isbn: 9780471496915.

[61] M. Wooldridge and N. Jennings. “Intelligent agents: theory and practice.” *The Knowledge Engineering Review.* 10.2 (1995), pp. 115–152. doi: 10.1017/s0269888900008122.

[62] S. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach*. third. Pearson., 2009. isbn: 9780136042594.

[63] F. Silva, C. Analide, L. Rosa, G. Felgueiras, and C. Pimenta. “Ambient Sensorization for the Furtherance of Sustainability.” In: *4th International Symposium on Ambient Intelligence*. 2013, pp. 179–186. doi: 10.1007/978-3-319-00566-9_23.

[64] A. Costa, J. Rincon, C. Carrascosa, V. Julian, and P. Novais. “Emotions Detection on an Ambient Intelligent System using Wearable Devices.” *Future Generation Computer Systems.* 92 (2019), pp. 479–489. doi: 10.1016/j.future.2018.03.038.

[65] D. Cook, J. Augusto, and V. Jakkula. “Ambient intelligence: Technologies, applications, and opportunities.” *Pervasive and Mobile Computing.* 5.4 (2009), pp. 277–298. doi: 10.1016/j.pmcj.2009.04.001.

[66] B. Fernandes, F. Silva, C. Analide, and J. Neves. “Crowd Sensing for Urban Security in Smart Cities.” *Journal of Universal Computer Science.* 24.3 (2018), pp. 302–321. doi: 10.3217/jucs-024-03-0302.

[67] S. Nakamoto. *Bitcoin: A Peer-to-Peer Electronic Cash System*. https://bitcoin.org/bitcoin.pdf. [Online; accessed May, 2019]. 2008.

[68] Ethereum. *A Next-Generation Smart Contract and Decentralized Application Platform*. https://github.com/ethereum/wiki/wiki/White-Paper. [Online; accessed May, 2019]. 2018.

[69] X. Xu, Q. Lu, Y. Liu, L. Zhu, H. Yao, and A. Vasilakos. "Designing blockchain-based applications a case study for imported product traceability." *Future Generation Computer Systems.* 92 (2019), pp. 399–406. doi: 10.1016/j.future.2018.10.010.

[70] D. Tse, B. Zhang, Y. Yang, C. Cheng, and H. Mu. "Blockchain application in food supply information security." In: *IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)*. 2017, pp. 1357–1361. doi: 10.1109/IEEM.2017.8290114.

[71] L. Chen, W. Lee, C. Chang, K. Choo, and N. Zhang. "Blockchain based searchable encryption for electronic health record sharing." *Future Generation Computer Systems.* 95 (2019), pp. 420–429. doi: 10.1016/j.future.2019.01.018.

[72] G. Dagher, J. Mohler, M. Milojkovic, and P. Marella. "Ancile: Privacy-preserving framework for access control and interoperability of electronic health records using blockchain technology." *Sustainable Cities and Society.* 39 (2018), pp. 283–297. doi: 10.1016/j.scs.2018.02.014.

[73] A. Azaria, A. Ekblaw, T. Vieira, and A. Lippman. "MedRec: Using Blockchain for Medical Data Access and Permission Management." In: *2nd International Conference on Open and Big Data (OBD)*. 2016, pp. 25–30. doi: 10.1109/OBD.2016.11.

[74] X. Wang, X. Zha, W. Ni, R. Liu, Y. Guo, X. Niu, and K. Zheng. "Survey on blockchain for Internet of Things." *Computer Communications.* 136 (2019), pp. 10–29. doi: 10.1016/j.comcom.2019.01.006.

[75] M. Hammi, B. Hammi, P. Bellot, and A. Serhrouchni. "Bubbles of Trust: A decentralized blockchain-based authentication system for IoT." *Computers and Security.* 78 (2018), pp. 126–142. doi: 10.1016/j.cose.2018.06.004.

[76] I. Makhdoom, M. Abolhasan, H. Abbas, and W. Ni. "Blockchain's adoption in IoT: The challenges, and a way forward." *Journal of Network and Computer Applications.* 125 (2019), pp. 251–279. doi: 10.1016/j.jnca.2018.10.019.

[77] N. Mitton and H. Rivano. "On the use of city bikes to make the city even smarter." In: *International Conference on Smart Computing Workshops (SMARTCOMP Workshops)*. 2014, pp. 3–8. doi: 10.1109/SMARTCOMP-W.2014.7046675.

[78] M. Miyaji. "Data mining for safety transportation by means of using Internet survey." In: *31st IEEE International Conference on Data Engineering Workshops (ICDEW)*. 2015, pp. 119–123. doi: 10.1109/ICDEW.2015.7129561.

[79] Y. Zheng, S. Rajasegarar, and C. Leckie. “Parking availability prediction for sensor-enabled car parks in smart cities.” In: *IEEE 10th International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP)*. 2015, pp. 1–6. doi: 10.1109/ISSNIP.2015.7106902.

[80] C. Barba, M. Mateos, P. Soto, A. Mezher, and M. Igartua. “Smart city for VANETs using warning messages, traffic statistics and intelligent traffic lights.” In: *Intelligent Vehicles Symposium (IV)*. 2012, pp. 902–907. doi: 10.1109/IVS.2012.6232229.

[81] F. Silva, C. Analide, and P. Novais. “Assessing Road Traffic Expression.” *International Journal of Artificial Intelligence and Interactive Multimedia.* 3.1 (2014), pp. 20–27. doi: 10.9781/ijimai.2014.313.

[82] G. Khekare and A. Sakhare. “A smart city framework for intelligent traffic system using VANET.” In: *International Multi-Conference on Automation, Computing, Communication, Control and Compressed Sensing (iMac4s)*. 2013, pp. 302–305. doi: 10.1109/iMac4s.2013.6526427.

[83] K. Bui, O. Lee, J. Jung, and D. Camacho. “Dynamic Traffic Light Control System Based on Process Synchronization Among Connected Vehicles.” In: *7th International Symposium on Ambient Intelligence (ISAmI)*. 2016, pp. 77–85. doi: 10.1007/978-3-319-40114-0_9.

[84] I. Abbasi, B. Nazir, A. Abbasi, S. Bilal, and S. Madani. “A traffic flow-oriented routing protocol for VANETs.” *Journal on Wireless Communications and Networking.* 121 (2014), 14 pages. doi: 10.1186/1687-1499-2014-121.

[85] F. Terroso-Saenz, M. Valdes-Vela, C. Sotomayor-Martinez, R. Toledo-Moreo, and F. Gomez-Skarmeta. “A cooperative approach to traffic congestion detection with complex event processing and VANET.” *IEEE Transactions on Intelligent Transportation Systems.* 13.2 (2012), pp. 914–929. doi: 10.1109/TITS.2012.2186127.

[86] R. Hussain, F. Hussain, and S. Zeadally. “Integration of VANET and 5G Security: A review of design and implementation issues.” *Future Generation Computer Systems.* 101.1 (2019), pp. 843–864. doi: 10.1016/j.future.2019.07.006.

[87] B. Pawar and M. Dongre. “Performance Enhancement for VANET Security Using Attack-Resistant Trust (ART).” *Innovations in Electronics and Communication Engineering. Lecture Notes in Networks and Systems.* 65 (2019), pp. 219–228. doi: 10.1007/978-981-13-3765-9_23.

[88] A. Quyoom, A. Mir, and A. Sarwar. “Security Attacks and Challenges of VANETs : A Literature Survey.” *Journal of Multimedia Information System.* 7.1 (2020), pp. 45–54. doi: 10.33851/JMIS.2020.7.1.45.

[89] J. Anaya, E. Talavera, D. Gimenez, N. Gomez, F. Jimenez, and J. Naranjo. “Vulnerable Road Users Detection Using V2X Communications.” In: *IEEE 18th International Conference on Intelligent Transportation Systems*. 2015, pp. 107–112. doi: 10.1109/ITSC.2015.26.

[90] D. Boehmlaender, S. Hasirlioglu, V. Yano, C. Lauerer, T. Brandmeier, and A. Zimmer. “Advantages in Crash Severity Prediction Using Vehicle to Vehicle Communication.” In: *IEEE International Conference on Dependable Systems and Networks Workshops*. 2015, pp. 112–117. doi: 10.1109/DSN-W.2015.23.

[91] J. Anaya, P. Merdrignac, O. Shagdar, F. Nashashibi, and J. Naranjo. “Vehicle to pedestrian communications for protection of vulnerable road users.” In: *IEEE Intelligent Vehicles Symposium Proceedings*. 2014, pp. 1037–1042. doi: 10.1109/IVS.2014.6856553.

[92] V. Milanes, J. Villagra, J. Godoy, J. Simo, J. Perez, and E. Onieva. “An Intelligent V2I-Based Traffic Management System.” *IEEE Transactions on Intelligent Transportation Systems.* 13 (2012), pp. 49–58. doi: 10.1109/TITS.2011.2178839.

[93] J. Casademont, A. Calveras, D. Quiñones, M. Navarro, J. Arribas, and M. Catalan-Cid. “Cooperative-Intelligent Transport Systems for Vulnerable Road Users Safety.” In: *7th International Conference on Future Internet of Things and Cloud (FiCloud)*. 2019, pp. 141–146. doi: 10.1109/FiCloud.2019.00027.

[94] I. Soto, F. Jimenez, M. Calderon, J. Naranjo, and J. Anaya. “Reducing Unnecessary Alerts in Pedestrian Protection Systems Based on P2V Communications.” *Electronics.* 8, article number: 360 (2019). doi: 10.3390/electronics8030360.

[95] J. Park, C. Nam, J. Lee, and D. Shin. “A Pedestrian Collision Prevention Method Through P2V Communication.” *Advances in Intelligent Systems and Computing.* 841 (2019), pp. 547–553. doi: 10.1007/978-981-13-2285-3_65.

[96] W. Cho. “Safety enhancement service for vulnerable users using P2V communications.” In: *International Conference on Connected Vehicles and Expo (ICCVE)*. 2014, pp. 1002–1003. doi: 10.1109/ICCVE.2014.7297498.

[97] N. Liu, M. Liu, J. Cao, G. Chen, and W. Lou. “When Transportation Meets Communication: V2P over VANETs.” In: *IEEE 30th International Conference on Distributed Computing Systems (ICDCS)*. 2010, pp. 567–576. doi: 10.1109/ICDCS.2010.83.

[98] D. Carels, W. Vandenberghe, I. Moerman, and P. Demeester. “Architecture for vulnerable road user collision prevention system (VRU-CPS), based on local communication.” In: *18th World congress on Intelligent Transport Systems*. 2011, 10 pages.

[99] A. Sieb, K. Hubel, D. Hepperle, A. Dronov, C. Hufnagel, J. Aktun, and M. Wolfel. “Hybrid City Lighting - Improving Pedestrians’ Safety through Proactive Street Lighting.” In: *International Conference on Cyberworlds (CW)*. 2015, pp. 46–49. doi: 10.1109/CW.2015.51.

[100] D. Dey, M. Martens, C. Wang, F. Ros, and J. Terken. "Interface Concepts for Intent Communication from Autonomous Vehicles to Vulnerable Road Users." In: *Adjunct Proceedings of the 10th International Conference on Automotive User Interfaces and Interactive Vehicular Applications*. 2018, pp. 82–86. doi: 10.1145/3239092.3265946.

[101] Daimler. *Autonomous concept car smart vision EQ fortwo: Welcome to the future of car sharing*. https://media.daimler.com/marsMediaSite/en/instance/ko/Autonomous-concept-car-smart-vision-EQfortwo-Welcome-to-the-future-of-carsharing.xhtml?oid=29042725. [Online; accessed April, 2020].

[102] Ford Media Center. *Ford, Virginia Tech Go Undercover To Develop Signals That Enable Autonomous Vehicles To Communicate With People*. https://media.ford.com/content/fordmedia/fna/us/en/news/2017/09/13/ford-virginia-tech-autonomous-vehicle-human-testing.html. [Online; accessed April, 2020].

[103] Nissan News USA. *Nissan IDS Concept: Nissan's vision for the future of EVs and autonomous driving*. https://usa.nissannews.com/en-US/releases/nissan-ids-concept-nissan-s-vision-for-the-future-of-evs-and-autonomous-driving. [Online; accessed April, 2020].

[104] K. Saleh, M. Hossny, and S. Nahavandi. "Contextual Recurrent Predictive Model for Long-Term Intent Prediction of Vulnerable Road Users." *IEEE Transactions on Intelligent Transportation Systems.* (2019), 11 pages. doi: 10.1109/TITS.2019.2927770.

[105] A. Robicquet, A. Sadeghian, A. Alahi, and S. Savarese. "Learning Social Etiquette: Human Trajectory Prediction In Crowded Scenes." In: *European Conference on Computer Vision (ECCV)*. 2016, pp. 549–565. doi: 10.1007/978-3-319-46484-8_33.

[106] N. Schneider and D. Gavrila. "Pedestrian Path Prediction with Recursive Bayesian Filters: A Comparative Study." *Pattern Recognition.* (2013), pp. 174–183. doi: 10.1007/978-3-642-40602-7_18.

[107] A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, L. Fei-Fei, and S. Savarese. "Social LSTM: Human Trajectory Prediction in Crowded Spaces." In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 961–971. doi: 10.1109/CVPR.2016.110.

[108] K. Saleh, M. Hossny, and S. Nahavandi. "Intent prediction of vulnerable road users from motion trajectories using stacked LSTM network." In: *20th International Conference Intelligent Transportation Systems (ITSC)*. 2017, pp. 327–332. doi: 10.1109/ITSC.2017.8317941.

[109] H. Zhang, Y. Liu, C. Wang, R. Fu, Q. Sun, and Z. Li. "Research on a Pedestrian Crossing Intention Recognition Model Based on Natural Observation Data." *Sensors.* 20.6, article number: 1776 (2020). doi: 10.3390/s20061776.

[110] M. Goldhammer, S. Kohler, S. Zernetsch, K. Doll, B. Sick, and K. Dietmayer. “Intentions of Vulnerable Road Users Detection and Forecasting by Means of Machine Learning.” *IEEE Transactions on Intelligent Transportation Systems.* (2019), 11 pages. doi: 10.1109/TITS.2019.2923319.

[111] J. Bao, P. Liu, and S. Ukkusuri. “A spatiotemporal deep learning approach for citywide short-term crash risk prediction with multi-source data.” *Accident Analysis and Prevention.* 122 (2019), pp. 239–254. doi: 10.1016/j.aap.2018.10.015.

[112] C. Liu and A. Sharma. “Using the multivariate spatio-temporal Bayesian model to analyze traffic crashes by severity.” *Analytic Methods in Accident Research.* 17 (2018), pp. 14–31. doi: 10.1016/j.amar.2018.02.001.

[113] M. Vilaça, E. Macedo, P. Tafidis, and M. Coelho. “Multinomial logistic regression for prediction of vulnerable road users risk injuries based on spatial and temporal assessment.” *International Journal of Injury Control and Safety Promotion.* 26.4 (2019), pp. 379–390. doi: 10.1080/17457300.2019.1645185.

[114] M. Liebner, F. Klanner, and C. Stiller. “Active safety for vulnerable road users based on smartphone position data.” In: *IEEE Intelligent Vehicles Symposium (IV)*. 2013, pp. 256–261. doi: 10.1109/IVS.2013.6629479.

[115] P. Pyykonen, A. Virtanen, and A. Kyytinen. “Developing intelligent Blind Spot Detection system for Heavy Goods Vehicles.” In: *IEEE International Conference on Intelligent Computer Communication and Processing (ICCP)*. 2015, pp. 293–298. doi: 10.1109/ICCP.2015.7312674.

[116] F. Guayante, A. Diaz-Ramirez, and P. Mejia-Alvarez. “Detection of Vulnerable Road Users in Smart Cities.” In: *8th International Conference on Next Generation Mobile Apps, Services and Technologies*. 2014, pp. 307–312. doi: 10.1109/NGMAST.2014.60.

[117] H. Rohling, P. Sorowka, K. Schumann, J. Häkli, and C. Nafornita. *Deliverable D4.5 - The ARTRAC Sensor*. Deliverable. [Retrieved from http://artrac.tutech.eu/file.php/ARTRAC_Sensor.pdf-2014-10-29]. ARTRAC Consortium, 2014.

[118] ARTRAC Consortium. *Advanced Radar Tracking and Classification for Enhanced Road Safety*. http://artrac.tutech.de/index.php/page/ARTRAC-2011-01-10.html. [Online; accessed April, 2019].

[119] HIGHTS Consortium. *HIGHTS, Horizon 2020 European Union Project*. http://hights.eu/. [Online; accessed May, 2019].

[120] P. Spaanderman, M. Raffero, M. Roelleke, M. Walter, C. Gentner, F. Berens, and et al. *Deliverable D2.1 - Use cases and Application Requirements*. Deliverable. [Retrieved from http://hights.eu/download/D2.1-Use-cases-and-Application-Requirements.pdf]. HIGHTS Consortium, 2015.

[121] ICSI Consortium. *icSi Project*. http://www.ict-icsi.eu/description.html. [Online; accessed May, 2019].

[122] I. Vitali, V. Di Gregorio, G. Iovino, D. Moroni, R. Schreiner, J. Ferreira, and et al. *Deliverable D1.3.2 - System Architecture*. Deliverable. [Retrieved from http://www.ict-icsi.eu/deliverables/ICSI_D1.3.2.pdf]. ICSI Consortium, 2015.

[123] Ko-FAS. *Ko-TAG - Kooperative Sensorik*. http://ko-fas.de/28-0-Ko-TAG---Kooperative-Sensorik.html. [Online; accessed May, 2019].

[124] R. Duenkler. *Security for vulnerable road users through Ko-TAG*. https://www.iis.fraunhofer.de/en/ff/lv/lok/proj/kotag.html. [Online; accessed May, 2019].

[125] MAVEN Consortium. *Managing Automated Vehicles Enhances Network*. http://www.maven-its.eu/. [Online; accessed May, 2019].

[126] R. Blokpoel, X. Zhang, J. P□ikryl, A. Pereira, D. Wesemeyer, L. Lücken, and et al. *Deliverable D4.4 - Cooperative adaptive traffic light with automated vehicles*. Deliverable. [Retrieved from http://www.maven-its.eu/]. MAVEN Consortium, 2018.

[127] OPTIMUM Consortium. *OPTIMUM: A short description*. http://www.optimumproject.eu/about/project. [Online; accessed April, 2019].

[128] K. Thiveos. *Deliverable D6.1 – OPTIMUM Technical Architecture*. Deliverable. [Retrieved from http://www.optimumproject.eu/uploads/documents/deliverables/D6.1.pdf]. OPTIMUM Consortium, 2016.

[129] Prospect Consortium. *About the Prospect Project*. http://www.prospect-project.eu/about-prospect-project/. [Online; accessed April, 2019].

[130] M. Wisch, M. Lerner, A. Schneider, J. Juhász, G. Attila, J. Kovaceva, and et al. *Deliverable D2.1 - Accident Analysis, Naturalistic Observations and Project Implications*. Deliverable. [Retrieved from http://www.prospect-project.eu/download/public-files/public_deliverables/PROSPECT-Deliverable-D2.1-Accident-Analysis-NDS-and-Project-Implications.pdf]. Prospect Consortium, 2016.

[131] F. Fischer and M. Gkemou. *SAFE STRIP Project Presentation*. Deliverable. [Retrieved from http://safetrip.diviprojects.wpengine.com/wp-content/uploads/sites/18/2017/10/D8.1-SAFE-STRIP-Project-Presentation-v2.0.pdf]. SAFE STRIP Consortium, 2017.

[132] SAFE STRIP Consortium. *SAFE STRIP in Brief*. http://safestrip.eu/about-2/. [Online; accessed May, 2019].

[133] SAVE-U Consortium. *The SAVE-U Project*. http://www.save-u.org/. [Online; accessed May, 2019].

[134] SMART RRS Consortium. *SMART RRS - Concept and Project Objectives*. http://i3a.unizar.es/smartrrs/content.php?seccion=12. [Online; accessed April, 2019].

[135] P. Frere. "Smart Road Restraint Systems (Smart RRS): Integrating sensing technology into crash barriers." In: *IET and ITS Conference on Road Transport Information and Control (RTIC)*. 2012, pp. 1–6. doi: 10.1049/cp.2012.1556.

[136] TEAM Consortium. *TEAM Vision*. http://www.collaborative-team.eu/overview/. [Online; accessed April, 2019].

[137] A. Amditis. *TEAM Project - Tomorrow's Elastic, Adaptive Mobility*. Deliverable. [Retrieved from http://www.collaborative-team.eu/downloads/get/Mobinet_TEAM_presentation.pdf]. TEAM Consortium, 2013.

[138] TIMON Consortium. *The TIMON Project*. https://www.timon-project.eu/index.php. [Online; accessed May, 2019].

[139] Transforming Transport Consortium. *TT Project Overview*. https://transformingtransport.eu/project-overview-0. [Online; accessed April, 2019].

[140] P. Campillo, D. Cobo, and M. Schygulla. *Deliverable D5.1 – Connected Vehicles Pilots Design*. Deliverable. [Retrieved from https://transformingtransport.eu/sites/default/files/2017-08/D5.1-Connected_vehicles_pilots_design_v1.0.pdf]. Transforming Transport Consortium, 2017.

[141] G. Prati, V Puchades, and L. Pietrantoni. "Cyclists as a minority group?" *Transportation Research Part F: Traffic Psychology and Behaviour.* 47.Supplement C (2017), pp. 34–41. doi: 10.1016/j.trf.2017.04.008.

[142] XCYCLE Consortium. *XCYCLE Project*. http://www.xcycle-h2020.eu/. [Online; accessed May, 2019].

[143] G. Baldini, T. Peirce, M. Botterman, M. Talacchini, A. Pereira, M. Handte, and et al. *Internet of Things - IoT Governance, Privacy and Security Issues*. IERC - European Research Cluster on the Internet of Things, 2015.

[144] F. Boavida, A. Kliem, T. Renner, J. Riekki, C. Jouvray, M. Jacovi, and et al. "People-Centric Internet of Things - Challenges, Approach, and Enabling Technologies." In: *9th International Symposium on Intelligent Distributed Computing*. 2016, pp. 463–474. doi: 10.1007/978-3-319-25017-5_44.

[145] M. Zhang, F. Sun, and X. Cheng. "Architecture of Internet of Things and Its Key Technology Integration Based-On RFID." In: *5th International Symposium on Computational Intelligence and Design*. 2012, pp. 294–297. doi: 10.1109/ISCID.2012.81.

[146] C. Babu and B. Reddy. "Predictive data mining on Average Global Temperature using variants of ARIMA models." In: *IEEE International Conference On Advances In Engineering, Science And Management (ICAESM)*. 2012, pp. 256–260. isbn: 9788190904223.

[147] P. Cortez, M. Rocha, and J. Neves. "Evolving Time Series Forecasting ARMA Models." *Journal of Heuristics.* 10.4 (2004), pp. 415–429. doi: 10.1023/B:HEUR.0000034714.09838.1e.

[148] R. Fu, Z. Zhang, and L. Li. "Using LSTM and GRU neural network methods for traffic flow prediction." In: *31st Youth Academic Annual Conference of Chinese Association of Automation (YAC)*. 2016, pp. 324–328. doi: 10.1109/YAC.2016.7804912.

[149] G. Box and G. Jenkins. *Time Series Analysis: forecasting and control*. Holden-Day, 1976. isbn: 9780816211043.

[150] M. Van Der Voort, M. Dougherty, and S. Watson. "Combining kohonen maps with arima time series models to forecast traffic flow." *Transportation Research Part C: Emerging Technologies.* 4.5 (1996), pp. 307–318. doi: 10.1016/S0968-090X(97)82903-8.

[151] K. Li, C. Zhai, and J. Xu. "Short-term traffic flow prediction using a methodology based on ARIMA and RBF-ANN." In: *Chinese Automation Congress (CAC)*. 2017, pp. 2804–2807. doi: 10.1109/CAC.2017.8243253.

[152] B. Williams. "Multivariate Vehicular Traffic Flow Prediction: Evaluation of ARIMAX Modeling." *Transportation Research Record.* 1776.1 (2001), pp. 194–200. doi: 10.3141/1776-25.

[153] S. Hochreiter and J. Schmidhuber. "Long Short-Term Memory." *Neural Computation.* 9.8 (1997), pp. 1735–1780. doi: 10.1162/neco.1997.9.8.1735.

[154] J. Bayer, D. Wierstra, J. Togelius, and J. Schmidhuber. "Evolving memory cell structures for sequence learning." In: *International Conference on Artificial Neural Networks*. 2009, pp. 755–764. doi: 10.1007/978-3-642-04277-5_76.

[155] T. Breuel. "High Performance Text Recognition Using a Hybrid Convolutional LSTM Implementation." In: *14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*. 2017, pp. 11–16. doi: 10.1109/ICDAR.2017.12.

[156] L. Chenbin, Z. Guohua, and L. Zhihua. "News Text Classification Based on Improved Bi-LSTM-CNN." In: *9th International Conference on Information Technology in Medicine and Education (ITME)*. 2018, pp. 890–893. doi: 10.1109/ITME.2018.00199.

[157] A. Coca, D. Correa, and L. Zhao. "Computer-aided music composition with LSTM neural network and chaotic inspiration." In: *International Joint Conference on Neural Networks (IJCNN)*. 2013, pp. 1–7. doi: 10.1109/IJCNN.2013.6706747.

[158] K. Choi, G. Fazekas, and M. Sandler. "Text-based LSTM networks for Automatic Music Composition." In: *1st Conference on Computer Simulation of Musical Creativity*. 2016.

[159] V. Pham, T. Bluche, C. Kermorvant, and J. Louradour. “Dropout Improves Recurrent Neural Networks for Handwriting Recognition.” In: *14th International Conference on Frontiers in Handwriting Recognition*. 2014, pp. 285–290. doi: 10.1109/ICFHR.2014.55.

[160] R. Messina and J. Louradour. “Segmentation-free handwritten Chinese text recognition with LSTM-RNN.” In: *13th International Conference on Document Analysis and Recognition (ICDAR)*. 2015, pp. 171–175. doi: 10.1109/ICDAR.2015.7333746.

[161] A. Graves, A. Mohamed, and G. Hinton. “Speech recognition with deep recurrent neural networks.” In: *IEEE International Conference on Acoustics, Speech and Signal Processing*. 2013, pp. 6645–6649. doi: 10.1109/ICASSP.2013.6638947.

[162] H. Sak, A. Senior, and F. Beaufays. “Long Short-Term Memory Based Recurrent Neural Network Architectures for Large Vocabulary Speech Recognition.” *arXiv e-prints.*, arXiv:1402.1128 (2014), arXiv:1402.1128. arXiv: 1402.1128 [cs.NE].

[163] Y. Tian and L. Pan. “Predicting Short-Term Traffic Flow by Long Short-Term Memory Recurrent Neural Network.” In: *IEEE International Conference on Smart City/SocialCom/SustainCom (SmartCity)*. 2015, pp. 153–158. doi: 10.1109/SmartCity.2015.63.

[164] Z. Cui, R. Ke, and Z. P. Y. Wang. “Deep Bidirectional and Unidirectional LSTM Recurrent Neural Network for Network-wide Traffic Speed Prediction.” *arXiv e-prints.*, arXiv:1801.02143 (2018), arXiv:1801.02143. arXiv: 1801.02143 [cs.LG].

[165] J. Zheng, C. Xu, Z. Zhang, and X. Li. “Electric load forecasting in smart grids using Long-Short-Term-Memory based Recurrent Neural Network.” In: *51st Annual Conference on Information Sciences and Systems (CISS)*. 2017, pp. 1–6. doi: 10.1109/CISS.2017.7926112.

[166] M. Cai, M. Pipattanasomporn, and S. Rahman. “Day-ahead building-level load forecasts using deep learning vs. traditional time-series techniques.” *Applied Energy.* 236 (2019), pp. 1078–1088. doi: 10.1016/j.apenergy.2018.12.042.

[167] R. Rahimilarki, Z. Gao, N. Jin, and A. Zhang. “Time-series Deep Learning Fault Detection with the Application of Wind Turbine Benchmark.” In: *IEEE 17th International Conference on Industrial Informatics (INDIN)*. 2019, pp. 1337–1342. doi: 10.1109/INDIN41052.2019.8972237.

[168] S. Yao, S. Hu, Y. Zhao, A. Zhang, and T. Abdelzaher. “DeepSense: a Unified Deep Learning Framework for Time-Series Mobile Sensing Data Processing.” In: *International World Wide Web Conference Committee (IW3C2)*. 2017, pp. 351–360. doi: 10.1145/3038912.3052577.

[169] J. Serra, S. Pascual, and A. Karatzoglou. “Towards a universal neural network encoder for time series.” *arXiv e-prints.*, arXiv:1805.03908 (2018), arXiv:1805.03908. arXiv: 1805.03908 [cs.LG].

[170] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. Gomez, L. Kaiser, and I. Polosukhin. “Attention is All you Need.” In: *31st Conference on Neural Information Processing Systems (NIPS)*. 2017, pp. 5998–6008. url: `http://papers.nips.cc/paper/7181-attention-is-all-you-need.pdf`.

[171] D. Bahdanau, K. Cho, and Y. Bengio. “Neural Machine Translation by Jointly Learning to Align and Translate.” In: *6th International Conference on Learning Representations (ICLR)*. 2015.

[172] B. Moldan, S. Janouskova, and T. Hak. “How to understand and measure environmental sustainability: Indicators and targets.” *Ecological Indicators.* 17 (2012), pp. 4–13. doi: `10.1016/j.ecolind.2011.04.033`.

[173] B. Bilgin and A. Emadi. “Electric Motors in Electrified Transportation: A step toward achieving a sustainable and highly efficient transportation system.” *IEEE Power Electronics Magazine.* 1.2 (2014), pp. 10–17. doi: `10.1109/MPEL.2014.2312275`.

[174] A. Alam, B. Besselink, V. Turri, J. MaRtensson, and K. Johansson. “Heavy-Duty Vehicle Platooning for Sustainable Freight Transportation: A Cooperative Method to Enhance Safety and Efficiency.” *IEEE Control Systems Magazine.* 35.6 (2015), pp. 34–56. doi: `10.1109/MCS.2015.2471046`.

[175] K. Oliveira, H. Castelli, S. Montebeller, and T. Avancini. “Wireless Sensor Network for Smart Agriculture using ZigBee Protocol.” In: *IEEE First Summer School on Smart Cities (S3C)*. 2017, pp. 61–66. doi: `10.1109/S3C.2017.8501379`.

[176] G. Nagaraja, A. Soppimath, T. Soumya, and A. Abhinith. “IoT Based Smart Agriculture Management System.” In: *4th International Conference on Computational Systems and Information Technology for Sustainable Solution (CSITSS)*. 2019, pp. 1–5. doi: `10.1109/CSITSS47250.2019.9031025`.

[177] M. Nassereddine, M. Nagrial, J. Rizk, and A. Hellany. “Design of Low Cost and High Efficiency Smart PV Solar System for Sustainable Residential Home.” In: *IEEE International Multidisciplinary Conference on Engineering Technology (IMCET)*. 2018, pp. 1–6. doi: `10.1109/IMCET.2018.8603062`.

[178] A. Ghosh, J. Sarkar, and B. Das. “Sustainable Energy Recovery from Municipal Solid Waste (MSW) using Bio-reactor Landfills for Smart City Development.” In: *IEEE International Conference on Sustainable Energy Technologies and Systems (ICSETS)*. 2019, pp. 242–246. doi: `10.1109/ICSETS.2019.8745334`.

[179] A. Alberink, P. Valery, A. Russell, and A. Green. “Do forecasts of UV indexes influence people’s outdoor behaviour?” *Australian and New Zealand Journal of Public Health.* 24.5 (2000), pp. 488–491. doi: `10.1111/j.1467-842X.2000.tb00498.x`.

[180] World Health Organization, World Meteorological Organization, United Nations Environment Programme, and International Commission on Non-Ionizing Radiation Protection. *Global solar UV index : a practical guide*. [Retrieved from https://www.who.int/uv/publications/en/UVIGuide.pdf]. World Health Organization, 2002. isbn: 9241590076.

[181] L. Dodd and M. Forshaw. "Assessing the efficacy of appearance-focused interventions to prevent skin cancer: a systematic review of the literature." *Health Psychology Review.* 4.2 (2010), pp. 93–111. doi: 10.1080/17437199.2010.485393.

[182] World Health Organization. *Ultraviolet radiation (UV) - Sun protection*. https://www.who.int/uv/sun_protection/en/. [Online; accessed February, 2020]. 2020.

[183] A. Young, J. Narbutt, G. Harrison, K. Lawrence, M. Bell, C. O'Connor, and et al. "Optimal sunscreen use, during a sun holiday with a very high ultraviolet index, allows vitamin D synthesis without sunburn." *British Journal of Dermatology.* 181.5 (2019), pp. 1052–1062. doi: 10.1111/bjd.17888.

[184] F. Leccese, G. Salvadori, D. Lista, and C. Burattini. "Outdoor Workers Exposed to UV Radiation: Comparison of UV Index Forecasting Methods." In: *IEEE International Conference on Environment and Electrical Engineering and IEEE Industrial and Commercial Power Systems Europe (EEEIC/I CPS Europe)*. 2018, pp. 1–6. doi: 10.1109/EEEIC.2018.8494621.

[185] R. Deo, N. Downs, A. Parisi, J. Adamowski, and J. Quilty. "Very short-term reactive forecasting of the solar ultraviolet index using an extreme learning machine integrated with the solar zenith angle." *Environmental research.* 155 (2017), pp. 141–166. doi: 10.1016/j.envres.2017.01.035.

[186] J. Barrera, D. Hurtado, and R. Moreno. "Prediction system of erythemas for phototypes I and II, using deep-learning." *Vitae.* 22.3 (2015), pp. 189–196. doi: 10.17533/udea.vitae.v22n3a03.

[187] C. Purananunak, S. Yanavanich, C. Tongpoon, and T. Phienthrakul. "A System for Ultraviolet Monitoring, Alert, and Prediction." In: *10th International Conference on Knowledge and Smart Technology (KST)*. 2018, pp. 237–241. doi: 10.1109/KST.2018.8426146.

[188] B. Mei, R. Li, W. Cheng, J. Yu, and X. Cheng. "Ultraviolet Radiation Measurement via Smart Devices." *IEEE Internet of Things Journal.* 4.4 (2017), pp. 934–944. doi: 10.1109/JIOT.2017.2717845.

[189] P. Oliveira, B. Fernandes, C. Analide, and P. Novais. "Multi-step Ultraviolet Index Forecasting using Long Short-Term Memory Networks." In: *17th International Conference on Distributed Computing and Artificial Intelligence (DCAI 2020)*. 2020, pp. 187–197. doi: 10.1007/978-3-030-53036-5_20.

[190] A. Musa and J. Eriksson. “Tracking Unmodified Smartphones Using Wi-Fi Monitors.” In: *Proceedings of the 10th ACM Conference on Embedded Network Sensor Systems*. 2012, pp. 281–294. doi: 10.1145/2426656.2426685.

[191] J. Weppner and P. Lukowicz. “Bluetooth based collaborative crowd density estimation with mobile phones.” In: *IEEE International Conference on Pervasive Computing and Communications (PerCom)*. 2013, pp. 193–200. doi: 10.1109/PerCom.2013.6526732.

[192] L. Schauer, M. Werner, and P. Marcus. “Estimating Crowd Densities and Pedestrian Flows Using Wi-Fi and Bluetooth.” In: *Proceedings of the 11th International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services*. 2014, pp. 171–177. doi: 10.4108/icst.mobiquitous.2014.257870.

[193] S. Depatla and Y. Mostofi. “Crowd Counting Through Walls Using WiFi.” In: *IEEE International Conference on Pervasive Computing and Communications (PerCom)*. 2018, pp. 1–10. doi: 10.1109/PERCOM.2018.8444589.

[194] W. Wang, J. Chen, and X. Song. “Modeling and predicting occupancy profile in office space with a Wi-Fi probe-based Dynamic Markov Time-Window Inference approach.” *Building and Environment.* 124 (2017), pp. 130–142. doi: 10.1016/j.buildenv.2017.08.00.

[195] T. Oransirikul, R. Nishide, I. Piumarta, and H. Takada. “Measuring Bus Passenger Load by Monitoring Wi-Fi Transmissions from Mobile Devices.” *Procedia Technology.* 18 (2014), pp. 120–125. doi: 10.1016/j.protcy.2014.11.023.

[196] A. Fernández-Ares, A. Mora, M. Arenas, P. García-Sanchez, G. Romero, V. Rivas, P. Castillo, and J. Merelo. “Studying real traffic and mobility scenarios for a Smart City using a new monitoring and tracking system.” *Future Generation Computer Systems.* 76 (2017), pp. 163–179. doi: 10.1016/j.future.2016.11.021.

[197] Last Minute Engineers. *Insight Into ESP8266 NodeMCU Features & Using It With Arduino IDE*. https://lastminuteengineers.com/esp8266-nodemcu-arduino-tutorial/. [Online; accessed May, 2020]. 2020.

[198] Last Minute Engineers. *Insight Into ESP32 Features & Using It With Arduino IDE*. https://lastminuteengineers.com/esp32-arduino-ide-tutorial/. [Online; accessed May, 2020]. 2020.

[199] L. Oliveira, D. Schneider, J. De Souza, and W. Shen. “Mobile Device Detection Through WiFi Probe Request Analysis.” *IEEE Access.* 7 (2019), pp. 98579–98588. doi: 10.1109/ACCESS.2019.2925406.

[200] T. Garcia. *Burger King trolls McDonald’s with penny Whoppers to promote new app.* https://www.marketwatch.com/story/burger-king-trolls-mcdonalds-with-penny-whoppers-to-promote-new-app-2018-12-04. [Online; accessed May, 2020].

[201] O. Qayum and T. Sohail. “FenceBook a Geofencing based Advertisements Application Using Android.” *Advances in Science, Technology and Engineering Systems Journal.* 1.5 (2016), pp. 27–33. doi: 10.25046/aj010506.

[202] K. Stuart. *Realtor Open House How To: Use iBeacons and Geofencing Wisely.* https://mobilewalletmarketer.com/using-ibeacons-geofencing-real-estate-open-house-marketing/. [Online; accessed May, 2020].

[203] A. LaMarca and E. Lara. *Location Systems: An Introduction to the Technology Behind Location Awareness*. Morgan & Claypool, 2008. doi: 10.2200/S00115ED1V01Y200804MPC004.

[204] G. Cheng, Y. Guo, Y. Chen, and Y. Qin. “Designating City-Wide Collaborative Geofence Sites for Renting and Returning Dock-Less Shared Bikes.” *IEEE Access.* 7 (2019), pp. 35596–35605. doi: 10.1109/ACCESS.2019.2903521.

[205] J. Sheppard, A. McGann, M. Lanzone, and R. Swaisgood. “An autonomous GPS geofence alert system to curtail avian fatalities at wind farms.” *Animal Biotelemetry.* 3 (2015), p. 43. doi: 10.1186/s40317-015-0087-y.

[206] T. Manley. “Using GPS/Iridium Radio Collars with Geofence Technology to Monitor Management Grizzly Bears in Northwest Montana.” *Intermountain Journal of Sciences.* 23.1-4 (2017).

[207] J. Pánek. “Emotional Maps: Participatory Crowdsourcing of Citizens’ Perceptions of Their Urban Environment.” *Cartographic Perspectives.* 91 (2018), pp. 17–29. doi: 10.14714/CP91.1419.

[208] R. Mody, K. Willis, and R. Kerstein. “WiMo: location-based emotion tagging.” In: *Proceedings of the 8th International Conference on Mobile and Ubiquitous Multimedia (MUM)*. 2009, pp. 1–4. doi: 10.1145/1658550.1658564.

[209] B. Fernandes, J. Neves, and C. Analide. “SafeCity: A platform for Safer and Smarter Cities.” In: *Advances in Practical Applications of Agents, Multi-Agent Systems, and Trustworthiness (PAAMS 2020)*. Vol. 12092. 2020, pp. 412–416. doi: 10.1007/978-3-030-49778-1_37.

[210] E. Papadimitriou, S. Lassarre, and G. Yannis. “Human factors of Pedestrian Walking and Crossing Behaviour.” *Transportation Research Procedia.* 25 (2017), pp. 2002–2015. doi: 10.1016/j.trpro.2017.05.396.

[211] M. Ashton and K. Lee. “The HEXACO–60: A Short Measure of the Major Dimensions of Personality.” *Journal of Personality Assessment.* 91.4 (2009), pp. 340–345. doi: 10.1080/00223890902935878.

[212] I. Myers. *The Myers-Briggs Type Indicator: Manual (1962)*. Consulting Psychologists Press, 1962. doi: 10.1037/14404-000.

[213] D. Riso and R. Hudson. *Understanding the enneagram: The practical guide to personality types*. Houghton Mifflin Harcourt, 2000.

[214] P. Costa-Jr and R. McCrae. “Domains and facets: Hierarchical personality assessment using the Revised NEO Personality Inventory.” *Journal of personality assessment.* 64.1 (1995), pp. 21–50. doi: 10.1207/s15327752jpa6401_2.

[215] R. Goldberg. “The development of markers for the Big-Five factor structure.” *Psychological assessment.* 4.1 (1992), p. 26. doi: 10.1037/1040-3590.4.1.26.

[216] O. John and S. Srivastava. “The Big Five trait taxonomy: History, measurement, and theoretical perspectives.” *Handbook of personality: Theory and research.* 2 (1999), pp. 102–138.

[217] M. Kosinski, Y. Bachrach, P. Kohli, D. Stillwell, and T. Graepel. “Manifestations of user personality in website choice and behaviour on online social networks.” *Machine Learning.* 95 (2014), pp. 357–380. doi: 10.1007/s10994-013-5415-y.

[218] G. Saucier. “Mini-Markers: A brief version of Goldberg’s unipolar Big-Five markers.” *Journal of personality assessment.* 63.3 (1994), pp. 506–516. doi: 10.1207/s15327752jpa6303_8.

[219] C. Sumner, A. Byers, R. Boochever, and G. Park. “Predicting Dark Triad Personality Traits from Twitter Usage and a Linguistic Analysis of Tweets.” In: *11th International Conference on Machine Learning and Applications*. Vol. 2. 2012, pp. 386–393. doi: 10.1109/ICMLA.2012.218.

[220] J. Pennebaker and A. King. “Linguistic styles: Language use as an individual difference.” *Journal of Personality and Social Psychology.* 77.6 (1999), pp. 1296–1312. doi: 10.1037/0022-3514.77.6.1296.

[221] A. Cerasa, D. Lofaro, P. Cavedini, I. Martino, A. Bruni, A. Sarica, and et al. “Personality biomarkers of pathological gambling: A machine learning study.” *Journal of Neuroscience Methods.* 294 (2018), pp. 7–14. doi: 10.1016/j.jneumeth.2017.10.023.

[222] S. Levitan, Y. Levitan, G. An, M. Levine, R. Levitan, A. Rosenberg, and J. Hirschberg. “Identifying Individual Differences in Gender, Ethnicity, and Personality from Dialogue for Deception Detection.” In: *Proceedings of the 2nd Workshop on Computational Approaches to Deception Detection*. 2016, pp. 40–44. doi: 10.18653/v1/W16-0806.

[223] S. Levitan, M. Levine, J. Hirschberg, N. Cestero, G. An, and A. Rosenberg. *Individual Differences in Deception and Deception Detection*. https://www.semanticscholar.org/paper/Individual-Differences-in-Deception-and-Deception-Levitan-Levine/295332ebfb77387f4ccbacbd214edf72caf3e331. [Online; accessed March, 2020]. 2015.

[224] F. Gurpinar, H. Kaya, and A. Salah. “Combining Deep Facial and Ambient Features for First Impression Estimation.” In: *Computer Vision – ECCV 2016 Workshops*. Vol. 9915. 2016. doi: 10.1007/978-3-319-49409-8_30.

[225] Y. Güçlütürk, U. Güçlü, M. Pérez, H. Escalante, X. Baró, C. Andujar, and et al. "Visualizing Apparent Personality Analysis with Deep Residual Networks." In: *IEEE International Conference on Computer Vision Workshops (ICCVW)*. 2017, pp. 3101–3109. doi: 10.1109/ICCVW.2017.367.

[226] C. Zhang, H. Zhang, X. Wei, and J. Wu. "Deep Bimodal Regression for Apparent Personality Analysis." In: *Computer Vision – ECCV 2016 Workshops*. Vol. 9915. 2016. doi: 10.1007/978-3-319-49409-8_25.

[227] P. Wessa. *Cronbach alpha (v1.0.5) in Free Statistics Software (v1.2.1)*. https://www.wessa.net/rwasp_cronbach.wasp/. [Online; accessed May, 2020].

[228] J. Friedman. "Greedy Function Approximation: A Gradient Boosting Machine." *The Annals of Statistics.* 29.5 (2001), pp. 1189–1232.

[229] G. Cawley and N. Talbot. "On Over-fitting in Model Selection and Subsequent Selection Bias in Performance Evaluation." *The Journal of Machine Learning Research.* 11 (2010), pp. 2079–2107.

[230] W. Yu, Z. Na, Y. Fengxia, and G. Yanping. "Magnetic resonance imaging study of gray matter in schizophrenia based on XGBoost." *Journal of Integrative Neuroscience.* 17.4 (2018), pp. 331–336. doi: 10.31083/j.jin.2018.04.0410.

[231] D. Sahoo and R. Balabantaray. "Single-Sentence Compression using XGBoost." *International Journal of Information Retrieval Research.* 9.3 (2019), p. 11. doi: 10.4018/IJIRR.2019070101.

[232] J. Pesantez-Narvaez, M. Guillen, and M. Alcañiz. "Predicting Motor Insurance Claims Using Telematics Data — XGBoost versus Logistic Regression." *Risks.* 7.2 (2019), p. 16. doi: 10.3390/risks7020070.

[233] W. Mikulas and S. Vodanovich. "The essence of boredom." *The Psychological Record.* 43.1 (1993), pp. 3–12. url: https://psycnet.apa.org/record/1993-28333-001.

[234] A. Close and M. Kukar-Kinney. "Beyond buying: Motivations behind consumers' online shopping cart use." *Journal of Business Research.* 63.9-10 (2010), pp. 986–992. doi: 10.1016/j.jbusres.2009.01.022.

[235] D. Carneiro, P. Novais, J. Pego, N. Sousa, and J. Neves. "Dynamics to Assess Stress During Online Exams." In: *Hybrid Artificial Intelligent Systems (HAIS 2018)*. Vol. 9121. 2015, pp. 345–356. doi: 10.1007/978-3-319-19644-2_29.

[236] D. Carneiro, P. Novais, J. Augusto, and N. Payne. "New Methods for Stress Assessment and Monitoring at the Workplace." *IEEE Transactions on Affective Computing.* 10.2 (2019), pp. 237–254. doi: 10.1109/TAFFC.2017.2699633.

[237] A. Pimenta, D. Carneiro, P. Novais, and J. Neves. “Detection of Distraction and Fatigue in Groups through the Analysis of Interaction Patterns with Computers.” In: *Intelligent Distributed Computing VIII*. Vol. 570. 2014, pp. 29–39. doi: 10.1007/978-3-319-10422-5_5.

[238] A. Pimenta, D. Carneiro, J. Neves, and P. Novais. “A Neural Network to Classify Fatigue from Human-Computer Interaction.” *Neurocomputing.* 172 (2016), pp. 413–426. doi: 10.1016/j.neucom.2015.03.105.

[239] A. Bogomolov, B. Lepri, and F. Pianesi. “Happiness Recognition from Mobile Phone Data.” In: *Proceedings of the 2013 International Conference on Social Computing*. 2013, pp. 790–795. doi: 10.1109/SocialCom.2013.118.

[240] R. Bixler and S. D’Mello. “Detecting Boredom and Engagement during Writing with Keystroke Analysis, Task Appraisals, and Stable Traits.” In: *Proceedings of the 2013 International Conference on Intelligent User Interfaces (IUI)*. 2013, pp. 225–234. doi: 10.1145/2449396.2449426.

[241] Q. Guo, E. Agichtein, C. Clarke, and A. Ashkan. “In the Mood to Click? Towards Inferring Receptiveness to Search Advertising.” In: *Proceedings of the 2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology*. 2009, pp. 319–324. doi: 10.1109/WI-IAT.2009.368.

[242] J. Seo, T. Laine, and K. Sohn. “An Exploration of Machine Learning Methods for Robust Boredom Classification Using EEG and GSR Data.” *Sensors.* 19.20 (2019), p. 4561. doi: 10.3390/s19204561.

[243] J. Seo, T. Laine, and K. Sohn. “Machine Learning Approaches for Boredom Classification using EEG.” *Journal of Ambient Intelligence and Humanized Computing.* 10 (2019), pp. 3831–3846. doi: 10.1007/s12652-019-01196-3.

[244] M. Pielot, T. Dingler, J. Pedro, and N. Oliver. “When attention is not scarce-detecting boredom from mobile phone usage.” In: *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp)*. 2015, pp. 825–836. doi: 10.1145/2750858.2804252.

[245] A. Sano and R. Picard. “Stress Recognition using Wearable Sensors and Mobile Phones.” In: *Proceedings of the 2013 Humaine Association Conference on Affective Computing and Intelligent Interaction*. 2013, pp. 671–676. doi: 10.1109/ACII.2013.117.

[246] J. Neves, J. Machado, C. Analide, A. Abelha, and L. Brito. “The Halt Condition in Genetic Programming.” In: *Progress in Artificial Intelligence*. 2007, pp. 160–169. doi: 10.1007/978-3-540-77002-2_14.

[247] C. Analide, P. Novais, J. Machado, and J. Neves. “Quality of Knowledge in Virtual Entities.” *Encyclopedia of Communities of Practice in Information and Knowledge Management.* (2006), pp. 436–442. doi: 10.4018/978-1-59140-556-6.ch073.

[248] L. Lima, P. Novais, R. Costa, J. Cruz, and J. Neves. "Group Decision Making and Quality-of-Information in e-Health Systems." *Logic Journal of the IGPL.* 19.2 (2011), pp. 315–332. doi: 10.1093/jigpal/jzq029.

[249] J. Neves. "A logic interpreter to handle time and negation in logic data bases." In: *Proceedings of the 1984 Annual Conference of the ACM on The Fifth Generation Challenge*. Association for Computing Machinery, 1984, pp. 50–54. doi: 10.1145/800171.809603.

[250] M. Denecker and A. Kakas. "Abduction in Logic Programming." In: *Computational Logic: Logic Programming and Beyond, Essays in Honour of Robert A. Kowalski, Part I*. Springer-Verlag, 2002, pp. 402–436. doi: 10.5555/646001.675770.

[251] L. Pereira and H. Anh. "Evolution Prospection." *New Advances in Intelligent Decision Technologies.* 199 (2009), pp. 51–63. doi: 10.1007/978-3-642-00909-9_6.

[252] B. Fernandes, H. Vicente, J. Ribeiro, A. Capita, C. Analide, and J. Neves. "Fully Informed Vulnerable Road Users - Simpler, Maybe Better." In: *Proceedings of the 21st International Conference on Information Integration and Web-based Applications & Services (iiWAS2019)*. 2019, pp. 598–602. doi: 10.1145/3366030.3366089.

[253] J. Souwe, P. Gates, and B. Bishop. *Community Attitudes to Road Safety – 2017 Survey Report*. Technical Report. [Retrieved from https://www.infrastructure.gov.au/roads/safety/publications/2018/community_att_17.aspx]. 2018.

[254] S. McEvoy, M. Stevenson, and M. Woodward. "The impact of driver distraction on road safety: results from a representative survey in two Australian states." *Injury Prevention.* 12.4 (2006), pp. 242–247. doi: 10.1136/ip.2006.012336.

[255] B. Watson. *The road safety implications of unlicensed driving: A survey of unlicensed drivers*. [Retrieved from https://eprints.qut.edu.au/8946/]. Centre for Accident Research and Road Safety-Queensland, 2003.

[256] S. Deb, L. Strawderman, J. DuBien, B. Smith, D. Carruth, and T. Garrison. "Evaluating pedestrian behavior at crosswalks: Validation of a pedestrian behavior questionnaire for the U.S. population." *Accident Analysis & Prevention.* 106 (2017), pp. 191–201. doi: 10.1016/j.aap.2017.05.020.

[257] Y. Yang and J. Sun. "Study on Pedestrian Red-Time Crossing Behavior: Integrated Field Observation and Questionnaire Data." *Transportation Research Record.* 2393.1 (2013), pp. 117–124. doi: 10.3141/2393-13.

[258] R. McIlroy, K. Plant, U. Jikyong, V. Nam, B. Bunyasi, G. Kokwaro, and et al. "Vulnerable road users in low-, middle-, and high-income countries: Validation of a Pedestrian Behaviour Questionnaire." *Accident Analysis & Prevention.* 131 (2019), pp. 80–94. doi: 10.1016/j.aap.2019.05.027.

[259] T. Wenterodt and H. Herwig. “The Entropic Potential Concept: a New Way to Look at Energy Transfer Operations.” *Entropy.* 16 (2014), pp. 2071–2084. doi: 10.3390/e16042071.

[260] D. Kondepudi. *Introduction to Modern Thermodynamics*. Wiley, 2008. isbn: 9780470986493.

[261] B. Fernandes, J. Neves, H. Vicente, and C. Analide. “Towards Road Safety – A Social Perception.” In: *Intelligent Systems and Applications (IntelliSys2018)*. Vol. 869. 2018, pp. 47–57. doi: 10.1007/978-3-030-01057-7_4.

[262] B. Fernandes, H. Vicente, J. Ribeiro, C. Analide, and J. Neves. “Evolutionary Computation on Road Safety.” In: *Hybrid Artificial Intelligent Systems (HAIS 2018)*. Vol. 10870. 2018, pp. 647–657. doi: 10.1007/978-3-319-92639-1_54.

[263] J. Ribeiro, B. Fernandes, C. Analide, H. Vicente, and J. Neves. “Full Informed Road Networks Evaluation: Simpler, Maybe Better.” In: *Proceedings of 73rd Research World International Conference*. [Retrieved from http://www.worldresearchlibrary.org/proceeding.php?pid=3178]. 2019, pp. 46–52.

[264] S. Haber and W. Stornetta. “How to Time-Stamp a Digital Document.” *Journal of Cryptology.* 3 (1991), pp. 99–111. doi: 10.1007/BF00196791.

[265] A. Reyna, C. Martín, J. Chen, E. Soler, and M. Díaz. “On blockchain and its integration with IoT. Challenges and opportunities.” *Future Generation Computer Systems.* 88 (2018), pp. 173–190. doi: 10.1016/j.future.2018.05.046.

[266] NEM. *NEM: Technical Reference*. Technical Report. [Retrieved from https://nem.io/wp-content/themes/nem/files/NEM_techRef.pdf]. 2018.

[267] X. Fan, L. Liu, M. Li, and Z. Su. “EigenTrust++: Attack Resilient Trust Management.” In: *8th IEEE International Conference on Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom)*. 2012. doi: 10.4108/icst.collaboratecom.2012.250420.

[268] C. Merkle. “A Digital Signature Based on a Conventional Encryption Function.” In: *Conference on the Theory and Application of Cryptographic Techniques*. 1987, pp. 369–378. doi: 10.1007/3-540-48184-2_32.

[269] P. Gallagher. *Introduction to DAPPS*. https://ethereum.org/en/developers/docs/dapps/. [Online; accessed December, 2020]. 2020.

[270] M. Muzammal, Q. Qu, and B. Nasrulin. “Renovating blockchain with distributed databases: An open source system.” *Future Generation Computer Systems.* 90 (2019), pp. 105–117. doi: 10.1016/j.future.2018.07.042.

[271] X. Liang, S. Shetty, D. Tosh, C. Kamhoua, K. Kwiat, and L. Njilla. "ProvChain: A Blockchain-Based Data Provenance Architecture in Cloud Environment with Enhanced Privacy and Availability." In: *17th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGRID)*. 2017, pp. 468–477. doi: 10.1109/CCGRID.2017.8.

[272] A. Ramachandran and M. Kantarcioglu. "Using Blockchain and smart contracts for secure data provenance management." *arXiv e-prints.*, arXiv:1709.10000 (2017), arXiv:1709.10000. arXiv: 1709.10000 [cs.CR].

[273] R. Neisse, G. Steri, and I. Nai-Fovino. "A Blockchain-based Approach for Data Accountability and Provenance Tracking." In: *12th International Conference on Availability, Reliability and Security (ARES)*. 2017, pp. 1–10. doi: 10.1145/3098954.3098958.

[274] D. Calvaresi, A. Dubovitskaya, J. Calbimonte, K. Taveter, and M. Schumacher. "Multi-Agent Systems and Blockchain: Results from a Systematic Literature Review." In: *Advances in Practical Applications of Agents, Multi-Agent Systems, and Complexity: The PAAMS Collection*. 2018, pp. 110–126. doi: 10.1007/978-3-319-94580-4_9.

[275] E. Ferrer. "The blockchain: a new framework for robotic swarm systems." *arXiv e-prints.*, arXiv: 1608.00695 (2016), arXiv: 1608.00695. arXiv: 1608.00695 [cs.RO].

[276] V. Shermin. "Disrupting governance with blockchains and smart contracts." *Strategic Change.* 26.5 (2017), pp. 499–509. doi: 10.1002/jsc.2150.

[277] S. Kiyomoto, M. Rahman, and A. Basu. "On blockchain-based anonymized dataset distribution platform." In: *IEEE 15th International Conference on Software Engineering Research, Management and Applications (SERA)*. 2017, pp. 85–92. doi: 10.1109/SERA.2017.7965711.

[278] K. Kvaternik, A. Laszka, M. Walker, D. Schmidt, M. Sturm, M. Lehofer, and A. Dubey. "Privacy-Preserving Platform for Transactive Energy Systems." *arXiv e-prints.*, arXiv:1709.09597 (2017), arXiv:1709.09597. arXiv: 1709.09597 [cs.DC].

[279] G. Ciatto, S. Mariani, and A. Omicini. "Blockchain for Trustworthy Coordination: A First Study with LINDA and Ethereum." In: *2018 IEEE/WIC/ACM International Conference on Web Intelligence (WI)*. 2018, pp. 696–703. doi: 10.1109/WI.2018.000-9.

[280] G. Ciatto, A. Maffi, S. Mariani, and A. Omicini. "Towards Agent-Oriented Blockchains: Autonomous Smart Contracts." In: *Advances in Practical Applications of Survivable Agents and Multi-Agent Systems: The PAAMS Collection*. Vol. 11523. 2019, pp. 29–41. doi: 10.1007/978-3-030-24209-1_3.

[281] K. Salah, M. Rehman, N. Nizamuddin, and A. Al-Fuqaha. "Blockchain for AI: Review and Open Research Challenges." *IEEE Access.* 7 (2019), pp. 10127–10149. doi: 10.1109/ACCESS.2018.2890507.

[282] J. Ferber. *Multi-agent systems: an introduction to distributed artificial intelligence*. Vol. 1. Addison-Wesley Reading, 1999. isbn: 9780201360486.

[283] L. Page, S. Brin, R. Motwani, and T. Winograd. *The PageRank Citation Ranking: Bringing Order to the Web*. Technical Report 1999-66. [Retrieved from http://ilpubs.stanford.edu:8090/422/]. Stanford InfoLab, 1999.

[284] A. Diogo, B. Fernandes, A. Silva, J. Faria, J. Neves, and C. Analide. “A Multi-Agent System Blockchain for a Smart City.” In: *The Third International Conference on Cyber-Technologies and Cyber-Systems (CYBER)*. IARIA. 2018, pp. 68–73. isbn: 9781612086835.

[285] B. Fernandes, A. Diogo, F. Silva, J. Neves, and C. Analide. “KnowLedger - A Multi-Agent System Blockchain for Smart Cities Data.” *Artificial Intelligence Review (Accepted to appear).* (2021).

Appendix A

Research Methodology

This appendix illustrates the research methodology followed during the development of this Doctoral Thesis. This iterative method combines both the typical Hypothetical-Deductive scientific method with the Engineering Process. It has all the foundations of the typical Hypothetical-Deductive method, with an additional Design phase. As depicted in Figure A.1, the method starts with the identification of the research problem, followed by the definition of the hypothesis. The researcher then designs, conceives, builds, and tests solution's specimens for the problem being addressed. Afterwards, it is time to conduct studies and experiments, record observations and process those same observations, interpret results, and draw conclusions. The research methodology is fulfilled as soon as observations and results agree with the hypothesis.

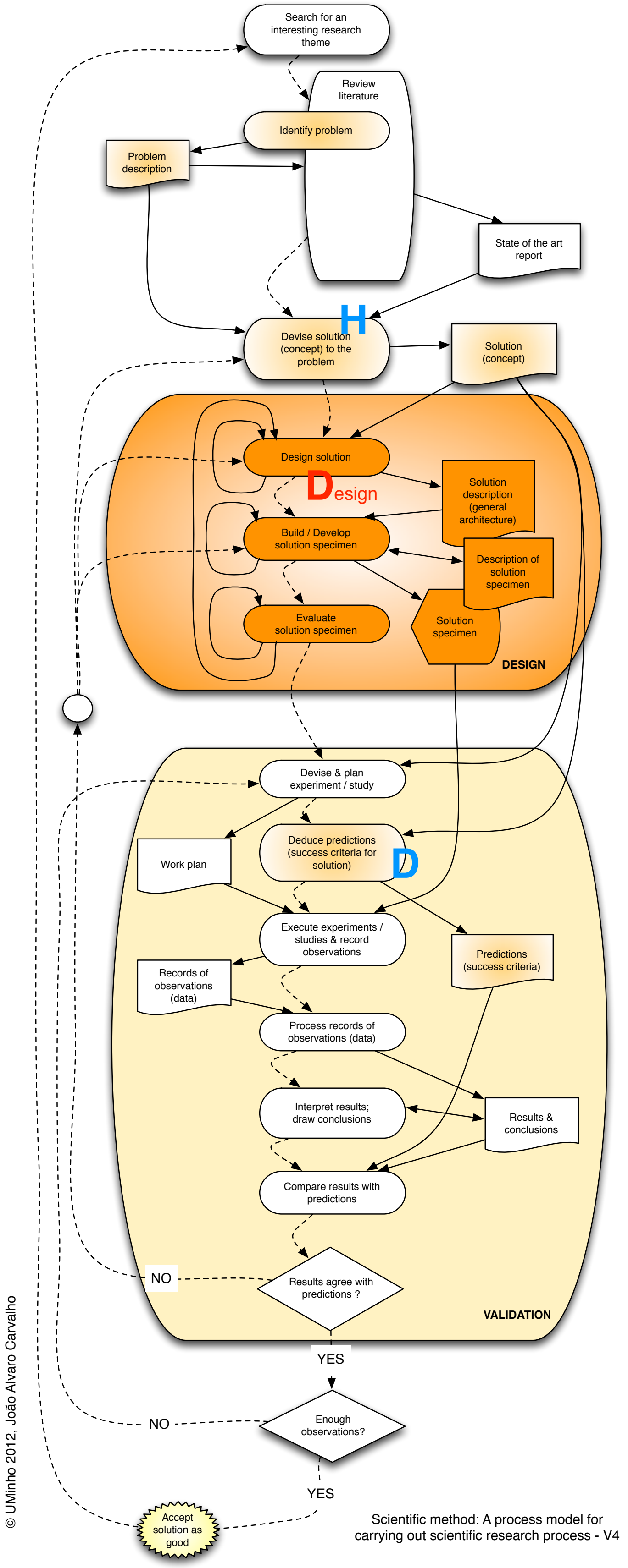

Search for an interesting research theme
Review literature
Identify problem
Problem description
State of the art report
H
Devise solution (concept) to the problem
Solution (concept)
Design solution
Design
Solution description (general architecture)
Build / Develop solution specimen
Description of solution specimen
Evaluate solution specimen
Solution specimen
DESIGN
Devise & plan experiment / study
Work plan
Deduce predictions (success criteria for solution)
D
Execute experiments / studies & record observations
Predictions (success criteria)
Records of observations (data)
Process records of observations (data)
Interpret results; draw conclusions
Results & conclusions
Compare results with predictions
NO
Results agree with predictions ?
VALIDATION
YES
NO
Enough observations?
YES
Accept solution as good
Scientific method: A process model for carrying out scientific research process - V4
© UMinho 2012, João Alvaro Carvalho

Figure A.1: A research methodology for carrying out scientific research. Copyright 2012 by University of Minho, J. A. Carvalho.

Appendix B

The Collector - Data Model

The Collector's data model consists of 9 tables. Tables prefixed with *cs* are related to crowd sensing while those prefixed with *sc* are related to city sensing. Then, those prefixed with *b* are baseline ones and contain static data such as the identification and location of the Smart Scanners for crowd sensing (*cs_probe_b_device*), the API keys for city sensing (*sc_api_b_keys*), the sensed cities (*sc_api_b_cities*), and the sensed pollution types (*sc_api_b_pollution_type*). Those prefixed with *d* are data tables and contain crowd sensing data (*cs_probe_d_data*), weather data (*sc_api_d_weather*), pollution data (*sc_api_d_pollution*), traffic flow (*sc_api_d_traffic_flow*), and traffic incidents data (*sc_api_d_traffic_incidents*). All relationships, primary and foreign keys, and column types are depicted in Figure B.1.

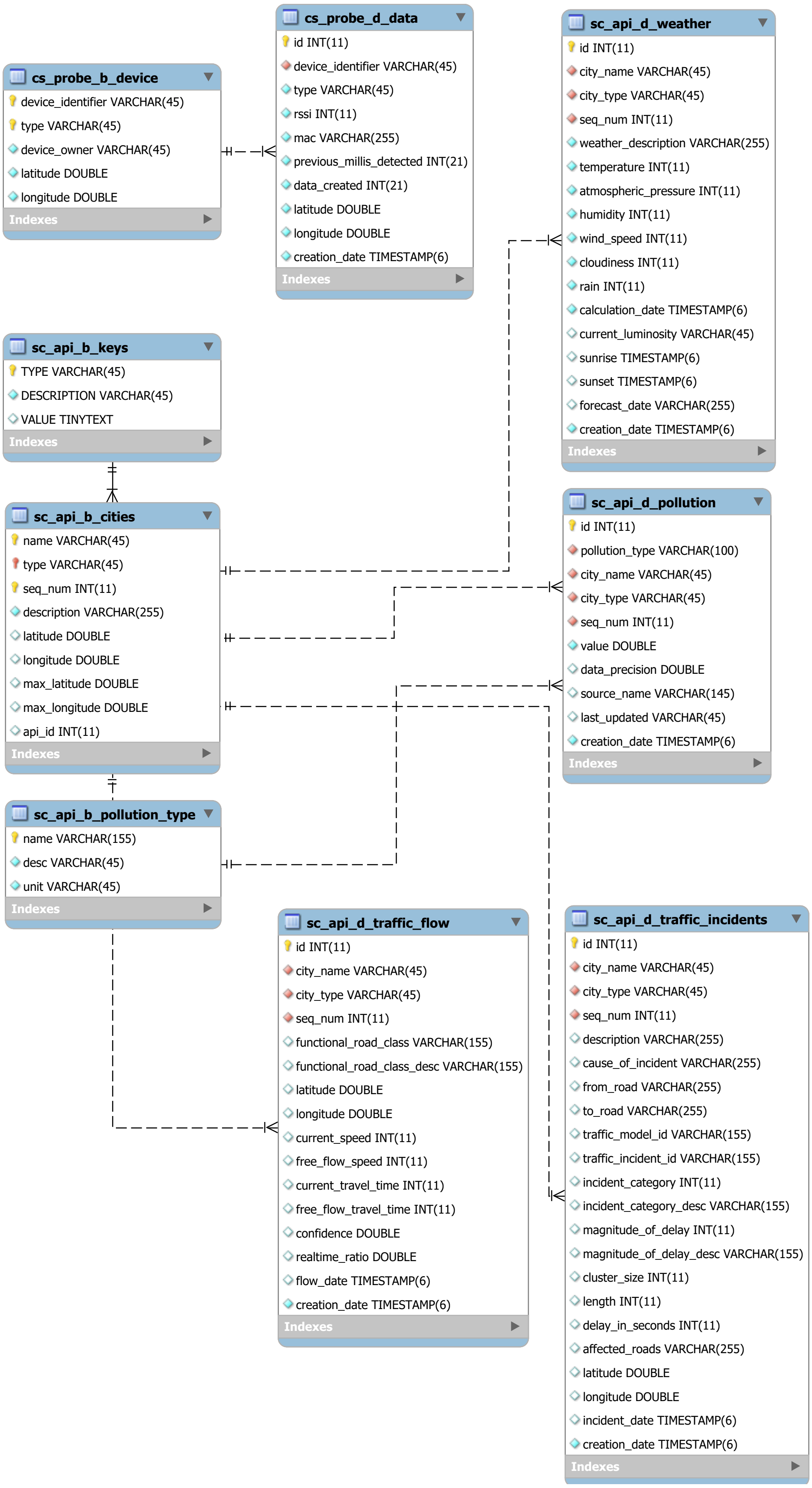


Figure B.1: *The Collector*'s data model.