Data Analytics in an Internet of Things Edge Cloud Setting

Laura Erhan

Supervisory team
Director of Studies: *Dr. Ovidiu Bagdasar, University of Derby, UK*
1st Supervisor: *Dr. Lee Barnby, University of Derby, UK*
External Supervisor: *Prof. Antonio Liotta, Free University of Bozen-Bolzano, IT*

A submission in partial fulfilment of the requirements of the University of Derby for the award of the degree of Doctor of Philosophy

College of Science and Engineering, University of Derby, Derby, UK

November 2021

Contents

List of Figures vii

List of Tables ix

List of Abbreviations xi

Declaration xiii

Acknowledgments xv

Abstract xvii

1 Introduction 1

1.1 Motivation 1

1.2 Research questions and objectives 3

1.3 Thesis contributions and outline 5

2 Background 7

2.1 Internet of Things 7

2.2 Edge Cloud setting 8

2.2.1 Cloud computing 9

2.2.2 Edge computing 10

2.3 Data analytics 11

2.4 Summary 12

3 Smart Anomaly Detection in Sensor Systems: A Multi-Perspective View 13

3.1 Introduction 14

3.2 About anomalies 18

3.2.1 The concept of anomaly . . . 18
3.2.2 Anomalies in sensor systems . . . 19
3.2.3 Anomaly detection datasets . . . 21
3.3 Conventional techniques for anomaly detection . . . 22
3.3.1 Statistical methods . . . 22
3.3.2 Time series analysis . . . 23
3.3.3 Signal processing . . . 24
3.3.4 Spectral techniques . . . 25
3.3.5 Information theory . . . 25
3.3.6 General considerations about conventional techniques . . . 26
3.4 Data-driven techniques for anomaly detection . . . 27
3.4.1 Supervised learning . . . 27
3.4.2 Semi-supervised learning . . . 28
3.4.3 Unsupervised learning . . . 29
3.4.4 Reinforcement learning . . . 29
3.4.5 Deep learning . . . 30
3.4.6 Online vs offline detection/algorithms . . . 34
3.5 Architectural perspective . . . 35
3.5.1 Anomaly detection in the Cloud . . . 35
3.5.2 Anomaly detection in the Fog . . . 36
3.5.3 Anomaly detection at the Edge . . . 37
3.5.4 Hybrid anomaly detection models . . . 38
3.6 Open issues and challenges . . . 39
3.6.1 Miniaturization . . . 40
3.6.2 Acceleration . . . 41
3.6.3 Energy efficiency . . . 42
3.6.4 Security . . . 43
3.6.5 Sensors softwarization . . . 45
3.6.6 Architectural models . . . 46
3.6.7 Data heterogeneity . . . 47
3.7 Conclusion . . . 48

4 Embedded Real-Time Data Imputation for Environmental Intelligent Sensing 51
4.1 Introduction . . . 52
4.2 Related work . . . 54

4.3 Methodology . . . 56
4.3.1 Dataset choice and description . . . 56
4.3.2 Dataset impairment . . . 58
4.3.3 Methods chosen for the edge data imputation . . . 59
4.3.4 Experiment design . . . 61
4.4 Result analysis . . . 61
4.4.1 Non-bursty case . . . 62
4.4.2 Bursty case . . . 64
4.4.3 Time and space complexity . . . 68
4.5 Discussion . . . 70
4.6 Conclusion . . . 71

5 Improve Well-being through Urban Nature (IWUN): a Real Social Case Study in an Edge Cloud Setting 73
5.1 Introduction . . . 75
5.2 Related work . . . 77
5.2.1 Big Data in social science studies . . . 77
5.2.2 Mining objective and subjective data . . . 78
5.2.3 Investigating the connection between well-being and nature through app-based studies . . . 79
5.3 Methodology . . . 79
5.3.1 Shmapped and data collection . . . 79
5.3.2 Data cleaning and pre-processing . . . 81
5.3.3 Text analysis . . . 83
5.3.4 Image analysis . . . 85
5.3.5 Time analysis . . . 86
5.4 Dataset characterization . . . 86
5.4.1 Demographic description of the participants . . . 86
5.4.2 Insights on the interaction between the participants and Shmapped . . . 87
5.5 Features noticed by the users . . . 87
5.5.1 What do the images say? . . . 89
5.5.2 What does the text say? . . . 90
5.5.3 How do image and text correspond? . . . 95
5.6 Time spent in green spaces . . . 96
5.6.1 Top users and parks based on average time spent in green spaces . . . 96

5.6.2 Age and gender distribution in park utilization 98
5.7 Comparison between objective and subjective interaction 100
5.8 Conclusion 102

6 Conclusions and Future Work Directions **105**
6.1 Conclusions and thesis contributions 105
6.2 Future work 107

Bibliography **111**

List of Figures

3.1 High perspective on sensor networks with a focus on anomaly detection techniques (conventional vs. data-driven), and architectural models (Cloud, Fog, Edge). 17

4.1 Histogram of the ozone measurements in the dataset. 58
4.2 RMSE value in relation to impairment rate (%) for the non-bursty case. . . . 62
4.3 Non-bursty case comparison with 5%, 50%, and 85% impairment rate. 63
4.4 Non-bursty case and bursty case for the same impairment rate (20%). 64
4.5 Density plots for different scenarios. 65
4.6 Evolution of RMSE with impairment rate in the bursty case. 66
4.7 Colormap showcasing the RMSE in relation to the impairment rate and burst size. 67
4.8 Execution times (s) on laptop and RPI 4B 4GB for the non-bursty case and varying impairment rates. 68
4.9 Colormap showcasing the execution time (s) in relation to the impairment rate and burst size for kNN, missForest, and MICE data imputation. 69
4.10 Snapshot of the CPU and RAM memory usage for the non-bursty case with a 50% impairment rate on the RPI 4B (4GB of RAM) for each algorithm (1 - mean imputation, 2 - MICE imputation, 3 - missForest imputation, 4 - kNN imputation). 70

5.1 Preview of the extracted geo-fences. 81
5.2 Age distribution of the sample dataset. 87
5.3 Participants' aggregated responses to two questions. 88
5.4 Heat-map representing the density of the users' feelings and the associated grades. The scale varies from blue (medium) to red (high). 88
5.5 Top 10 labels for each category of images. 89

5.6 Clusters produced from k-means clustering (k=40) of textual observations. Legend captures 25 clusters. 92
5.7 Classification of the textual observations into the themes of Table 5.1 with the FastText algorithm. 93
5.8 Age classification of textual observations into the themes of Table 5.1. 94
5.9 Gender classification of textual observations into the themes of Table 5.1. . . 95
5.10 Endcliffe Park utilization based on the concentration of location points (green - low number, red - high number). 98
5.11 Age and gender groups interactions with top 10 most visited green spaces. The percentages are computed only on the samples in these top areas. The amount of interaction decreases from left to right. 99
5.12 Comparison of objective (green) and subjective (orange) interaction for the top 10 most visited green spaces divided for the three age groups: young (18 to 35), middle-aged (36 to 53), and senior (54 to 72). The top 10 is according to the subjective data (density of observations). The percentages are computed only on the samples in these top areas. 101
5.13 Comparison of objective (green) and subjective (orange) interaction for the top 10 most visited green spaces for the two gender groups. The top 10 is according to the subjective data (density of observations). The percentages are computed only on the samples in these top areas. 102

List of Tables

3.1 Promising research challenges. 39
3.2 Key references about techniques and architectural models. 48

4.1 Description of the dataset pollution measurements. 58
4.2 The methods used for corrupting the dataset. 59
4.3 Hardware specifications for the experimental environment. 61

5.1 Labels from training data from a study of human connection to nature[188]. . 84
5.2 Example of labelling for an image. 85
5.3 Number of labels for participant categories. 89
5.4 Example of text clustering, considering k = 40. 91
5.5 Average time spent in parks, by the user. 96
5.6 Average time spent inside green spaces, by the park. 97

List of Abbreviations

ANN	Artificial Neural Network
ARIMA	Autoregressive Integrated Moving Average
ARMA	Autoregressive Moving Average
BME	Bayesian Maximum Entropy
CART	Classification and Regression Trees
CAV	Connected Autonomous Vehicle
CNN	Convolutional Neural Network
DBN	Deep Belief Network
DBSCAN	Density-Based Spatial Clustering of Applications with Noise
DT	Decision Trees
EBGAN	Energy Based Generative Adversarial Network
ECG	Electrocardiography
EEG	Electroencephalography
GAN	Generative Adversarial Network
GMM	Gaussian Mixture Model
GNN	Graph Neural Network
GRNN	General Regression Neural Network
IIoT	Industrial Internet of Things
IoT	Internet of Things
IoV	Internet of Vehicles
IRL	Inverse Reinforcement Learning
IWUN	Improve Well-being through Urban Nature
LSTM	Long-Short Term Memory
MEC	Multi-Access Edge Computing
MICE	Multiple Imputation by Chained Equations
ML	Machine Learning

NAB	Numenta Anomaly Benchmark
NFV	Network Function Virtualization
OC-SVM	One Class Support Vector Machine
ODDS	Outlier Detection DataSets
PCA	Principal Component Analysis
PMF	Probabilistic Matrix Factorization
QoS	Quality of Service
RBM	Restricted Boltzmann Machine
RL	Reinforcement Learning
RMSE	Root Mean Square Error
RNN	Recurrent Neural Network
RPI	Raspberry Pi
RPL	Routing Protocol for Low-Power and Lossy Networks
SDN	Software Defined Networking
SGTM	Successive Geometric Transformation Models
SOM	Self Organizing Maps
SSA	Singular Spectrum Analysis
SVM	Support Vector Machine
TSA	Time Series Analysis
VARMA	Vectorized Autoregressive Moving Average
VNF	Virtual Network Function
WSN	Wireless Sensor Network

Declaration

This is to declare that the work stated in this dissertation was done by the author, and no part of the dissertation has been submitted in a dissertation form to another university or similar institution. No human or animal participation have been included in this research, and the research presented in this dissertation has been ethically approved. The author confirms that this work has not been accepted in substance for any degree and is not concurrently presented for any degree other than Doctor of Philosophy (Ph.D.) being studied at the University of Derby, and that appropriate credit has been given within the thesis where reference has been made to the work of others. Parts of this dissertation have previously appeared in the papers which I authored or co-authored.

Laura Erhan

University of Derby, 2021

Acknowledgements

Starting a Ph.D. is easy, but going forward with it and actually finishing it is not an easy task. Like many others before me, I have encountered both ups and down in my journey. I have to admit that there were moments when I was not sure that it was a feasible task. Now, while in front of this written Ph.D. thesis, I want to acknowledge that none of this would have been possible without the help, support and encouragement of a lot of people.

Firstly, I would like to thank my supervisory team. Antonio, thank you for offering me this opportunity and offering me guidance and support along the way. Ovidiu, thank you for stepping in when needed, and for motivating me. I would also like to thank Ashiq and Lee for all their contributions and support. Although not part of my supervisory team, I would also like to thank here Mario, who has contributed to part of my Ph.D. endeavours.

Another thank you goes to my work colleagues. In no particular order, Lucia, Maryleen, Enrico, Francis, Luigi, Luca, Pasquale thank you for all the discussions, laughs, and trips. They have all contributed to both my personal and professional development.

I would also like to acknowledge all the seminars, workshops, evaluations and events organized by the University of Derby and the PGR group. I had the chance to meet many people and learn a lot. This all has contributed to a nice Ph.D. journey.

Another thank you goes to my professors and colleagues from Iasi, Romania.

A special mention goes to my friends back home (in no special order): Roxana, Iustina, Elena, Ana, Andrei, George, I cannot thank you enough for all the discussion, good times, tremendous support and thumbs up that you have shown me throughout these last years.

And the biggest thank you and love goes to my family and to Dominik. They have been the ones always there for me, be it ups or downs, always checking in and voicing their support. Mom, dad, Laurentiu, thank you for everything! Dominik, thank you for being my pillar in all those tricky moments!

As I am sure I have forgotten a lot of people, I will take this line to thank everybody who has contributed to my Ph.D. journey these past years!

Abstract

Over the past years, the Internet of Things (IoT) has vastly expanded with a multitude of devices now monitoring, sensing, and acting on the surrounding environment. This in turn creates a large amount of data to be processed and analysed in order to gain insight into specific problems. The development of computationally powerful IoT devices allows for the processing pipeline to start close to the data collection points, namely at the Edge of a system, and for it to continue if needed, all the way up to the Cloud, where heavy processing can be undertaken. We investigate how machine learning techniques can be used to take advantage of the Edge by pushing computation to smaller devices such as the Raspberry Pi, and how IoT data analytics can be obtained both with the help of the Edge and the Cloud.

This thesis revolves around three main directions for IoT data analytics.

Firstly, we discuss anomaly detection, an important theoretical and practical problem, due to its broad set of application domains, ranging from data analysis to industrial automation. Herein, we review state-of-the-art methods that may be employed to detect anomalies in the specific area of sensor systems. In this context, anomaly detection is a particularly hard problem, given the need to find computational-energy-accuracy trade-offs in a constrained environment. We taxonomize methods ranging from conventional techniques (statistical methods, time-series analysis, signal processing, etc.) to data-driven techniques (supervised learning, reinforcement learning, deep learning, etc.). We also look at the impact that different architectural environments (Cloud, Fog, Edge) can have on the sensors ecosystem.

Secondly, we advocate for the use of machine learning at the sensor nodes to perform data imputation, an essential data-cleaning operation, in order to avoid the transmission of corrupted (and often unusable) data to the Cloud. Starting from a publicly available pollution dataset, we investigate how two machine learning techniques (kNN and missForest) compare against two statistical based techniques (mean and MICE) and how these can be embedded on a Raspberry Pi to perform data imputation in real-time, without affecting the data collection process. The experimental results provide details of the accuracy and execution times, while

demonstrating the accuracy and computational efficiency of edge-learning methods for filling in missing data values in corrupted data series.

Finally, we present a study case which is representative for smart cities and IoT analytics in an Edge Cloud setting. A field experiment aiming to better understand the interactions between citizens and urban green spaces was carried out in Sheffield, U.K., which involved 1870 participants for two different time periods (7 and 30 days). Objective (sensor information) and subjective data (direct input from the users) was collected via a smartphone app. Location data from green spaces was complemented by textual and photographic information provided by the users. With the use of data science and machine learning techniques, we identify the main features observed by the citizens through both text and images, the time that people spent in parks, as well as the top interaction areas. This allows us to gain an overview of certain patterns and the behaviour of the citizens within their surroundings and proves the capabilities of integrating technology into large-scale social studies.

Chapter 1

Introduction

The growth and development of both the Internet of Things (IoT) and the computing paradigms over the past years provide us with the possibility of better monitoring, interpreting, and improving the surrounding world. In this Ph.D. thesis, we investigate how the world of IoT and machine learning techniques intersect in an Edge Cloud setting while considering three main topics, namely anomaly detection, data imputation, and data analytics for a real and representative smart city case study. This chapter presents the motivation that fuelled this work, the contributions we made to the research field, as well as an outline of this thesis.

1.1 Motivation

Throughout the past years, the number of connected Internet of Things (IoT) devices has been continuously increasing. According to IoT Analytics [108], 12.3 billion IoT devices will be active by the end of 2021, with the number reaching 27.1 billion by 2025. These devices are already used in a multitude of scenarios including smart cities (e.g. smart traffic systems, pollution monitoring, surveillance systems), smart healthcare (e.g. medical monitoring of patients), smart homes (e.g. smart appliances), and Industry 4.0 (e.g. smart machines, predictive maintenance). A challenge arising is to make sense of and use all the generated data. Traditionally, data was sent up to the cloud to be stored, processed, and analysed. However, this is no longer feasible given the ever-increasing number of devices and the associated generated data.

One important thing to note is that beside the increasing number of IoT devices, there have also been major hardware developments. As a result, part of the IoT world can handle performing different degrees of computation locally, at the edge of the network. Moreover,

a switch from the well-established cloud computing paradigm to that of edge computing was made possible. Pushing computation to the edge allows for better use of the existing infrastructure. Raw data (generated by the sensors etc.) which can often be unreliable or corrupted does not need to be transmitted further up the processing pipeline, but can instead be (at least) pre-processed at the edge. Some IoT edge devices can handle even heavier computations, which allows for obtaining results for less complex tasks directly at the edge of the system. As a result, only already processed data, if not partial results are being sent further up, all the way to the cloud, where heavier processing can be undertaken. This results in reduced bandwidth usage, better response times, and avoiding network bottlenecks. It is important to highlight that we do not encourage the elimination of the cloud, but rather the use of the available computing resources starting from the IoT sensing device, with the involvement of the next computational layer only when needed. In this way, the cloud rather handles the complex tasks that cannot be taken care of at lower levels.

IoT systems monitor the surrounding world through a variety of sensors, as well as through direct input from different actors (participants). Therefore, a variety of (raw) data are captured and need to be analysed to gain insight in regard to the specific application scenarios. The field of machine learning has grown over the past years, and it is gaining more and more importance as it can help us sieve through massive amounts of data by automatically detecting patterns, predicting future values, and taking decisions in uncertain conditions. Machine learning techniques are used along the more traditional statistical techniques to make sense of the IoT data. It is important to note that despite the view that machine learning implies heavy processing, part of the techniques is also suitable for performing at the edge. Their memory requirements or complexity can be fairly low and do not require that the execution happens only on more powerful devices such as PCs, laptops, or the cloud.

In this thesis, we investigate how data analytics can be carried out within the context of the IoT in an Edge Cloud setting. Particularly, we focus on anomaly detection and data imputation within an edge environment, as well as on a real case study within an edge cloud environment. We use existing machine learning and statistical techniques in a new context. We showcase the power of employing various techniques at the edge; we introduce a new taxonomy for smart anomaly detection in sensor systems; we investigate how data imputation can be embedded in constrained environments for environmental intelligent sensing; we demonstrate how data science techniques can be applied in a real social study.

Throughout the remainder of this chapter, we will highlight the research questions and the objectives behind this thesis, as well as the contributions of this work and the thesis outline.

1.2 Research questions and objectives

Previously, we presented the motivation behind this work, but we have not explicitly highlighted the problem we tackle as part of this thesis. Despite the growth of IoT, the emergence of the paradigm of edge computing, along with the hardware developments of current IoT devices, there is still limited interest at how IoT data analytics can be performed whilst taking advantage of the edge of the network/system. In this thesis, we want to showcase how even more advanced techniques (machine learning based) can be used within a variety of IoT scenarios, as well as in computationally-constrained environments such as the RPI.

The work in this thesis spans across three different directions, whilst pertaining to data analytics in an IoT Edge Cloud Setting. The specific research questions we answer in this thesis include:

1. What is the state of the art for smart anomaly detection in the context of sensor systems?
2. Is real-time data imputation possible and efficient at the edge?
3. How can the use of data science and machine learning techniques enhance the data analysis in social studies?

The corresponding objectives are:

1. Proposing a novel taxonomy for smart anomaly detection in sensor systems while considering both conventional and data-driven techniques, as well as different architectural environments; Providing a multi perspective view of the state of the art.
2. Developing a real-time IoT-based platform to evaluate machine learning techniques (kNN and missForest) against statistical techniques (mean and MICE) for real-time embedded data imputation (on a Raspberry Pi), while considering both bursty and non-bursty missing data in an environmental IoT scenario.
3. Showcasing how applying data science and machine learning techniques to the data from a real social field experiment can complement the traditional analysis and provide new insight.

The novelty and originality that this thesis brings to the research field is focused on a synthesis of the state of the art and novel applications of existing methods and knowledge.

We propose a new taxonomy and review of the state of the art for anomaly detection in sensor systems (1, Chapter 3). We apply, compare, and evaluate existing data imputation

techniques within a new context and scenario, namely at the edge (on a Raspberry Pi) for the case of bursty and non-bursty missing data from an environmental scenario (2, Chapter 4). We make use of existing data science and machine learning techniques within a new scenario, namely a real social case study, to obtain insights (3, Chapter 5). Furthermore, most of the work presented in Chapters 3, 4, 5 is already published. The corresponding list of publications is:

- Journal paper: **L. Erhan**, M. Ndubuaku, M. Di Mauro, W. Song, M. Chen, G. Fortino, O. Bagdasar, and A. Liotta. "Smart anomaly detection in sensor systems: A multi-perspective review". In: *Information Fusion* 67 (2021), pp. 64–79. DOI: 10.1016/j.inffus.2020.10.001.

- Conference paper: **L. Erhan**, M. Di Mauro, O. Bagdasar, and A. Liotta, "Critical comparison of data imputation techniques at IoT edge". In: *Intelligent Distributed Computing XIV*. 2022.

- Journal paper: **L. Erhan**, M. Di Mauro, A. Anjum, O. Bagdasar, W. Song, and A. Liotta. "Embedded Data Imputation for Environmental Intelligent Sensing: A Case Study". In: *Sensors* 21.23 (2021), p. 7774. DOI: 10.3390/s21237774.

- Conference paper: E. Ferrara, A. Liotta, **L. Erhan**, M. Ndubuaku, D. Giusto, M. Richardson, D. Sheffield, and K. McEwan. "A pilot study mapping citizens' interaction with urban nature". In: *2018 IEEE 16th Intl. Conf. on Dependable, Autonomic and Secure Computing, 16th Intl. Conf. on Pervasive Intelligence and Computing, 4th Intl. Conf. on Big Data Intelligence and Computing and Cyber Science and Technology Congress (DASC/PiCom/DataCom/CyberSciTech)*. 2018, pp. 836–841. DOI: 10.1109/DASC/PiCom/DataCom/CyberSciTec.2018.00-21.

- Conference paper: E. Ferrara, A. Liotta, M. Ndubuaku, **L. Erhan**, D. Giusto, M. Richardson, D. Sheffield, and K. McEwan. "A demographic analysis of urban nature utilization". In: *2018 10th Computer Science and Electronic Engineering (CEEC)*. 2018, pp. 136–141. DOI: 10.1109/CEEC.2018.8674206.

- Journal paper: **L. Erhan**, M. Ndubuaku, E. Ferrara, M. Richardson, D. Sheffield, F. J. Ferguson, P.Brindley, and A. Liotta. "Analyzing objective and subjective data in social sciences: Implications for smart cities". In: *IEEE Access* 7 (2019), pp. 19890–19906. DOI: 10.1109/ACCESS.2019.2897217.

1.3 Thesis contributions and outline

This thesis is organized as follows: Chapter 2 introduces the reader to some background information relevant to the topics that the thesis covers, as well as related work. Chapter 3 offers a multi perspective view over the topic of smart anomaly detection in sensor systems. Chapter 4 discusses embedded real-time data imputation for environmental intelligent sensing. Chapter 5 presents a real social case study representative for an IoT Edge Cloud setting, together with the performed data analytics. Chapter 6 concludes this thesis and discusses possible directions for future work.

In this thesis, our aim is to demonstrate how a variety of data analytics can be performed in an IoT Edge Cloud setting. Our main contributions include:

- a new taxonomy for anomaly detection in sensor systems in terms of methods (ranging from conventional to data-driven techniques) together with a review of the state of the art, including the impact of architectural environments on the sensors ecosystem (Chapter 3);
- showcasing how real-time machine learning based data imputation for an environmental scenario can be performed at the edge (on a Raspberry Pi), as well as an evaluation of two machine learning techniques against two statistical based techniques for the task at hand (Chapter 4);
- demonstrating how data science and machine learning techniques can enhance and complement the data analytics process in a real social case study (Chapter 5).

In the following, we detail the main contributions highlighted above, for each corresponding chapter. Each of the chapters aims at being self-contained. Therefore, these could be read independently of the other content presented in this thesis.

Chapter 3 discusses anomaly detection, an important theoretical and practical problem, due to its broad set of application domains, ranging from data analysis to industrial automation. Herein, we review state-of-the-art methods that may be employed to detect anomalies in the specific area of sensor systems. In this context, anomaly detection is a particularly hard problem, given the need to find computational-energy-accuracy trade-offs in a constrained environment. We taxonomize methods ranging from conventional techniques (statistical methods, time-series analysis, signal processing, etc.) to data-driven techniques (supervised learning, reinforcement learning, deep learning, etc.). We also look at the impact that different architectural environments (Cloud, Fog, Edge) can have on the sensors ecosystem.

In Chapter 4, we advocate for the use of machine learning at the sensor nodes to perform data imputation, an essential data-cleaning operation, in order to avoid the transmission of corrupted (and often unusable) data to the Cloud. Starting from a publicly available pollution dataset, we investigate how two machine learning techniques (kNN and missForest) compare against two statistical based techniques (mean and MICE) and how these can be embedded on a Raspberry Pi to perform data imputation in real-time, without affecting the data collection process. The experimental results provide details of the accuracy and execution times, while demonstrating the accuracy and computational efficiency of edge-learning methods for filling in missing data values in corrupted data series.

Chapter 5 presents a case study which is representative of smart cities and IoT analytics in an Edge Cloud setting. A field experiment aiming to better understand the interactions between citizens and urban green spaces was carried out in Sheffield, U.K., which involved 1870 participants for two different time periods (7 and 30 days). Objective (sensor information) and subjective data (direct input from the users) was collected via a smartphone app. Location data from green spaces was complemented by textual and photographic information provided by the users. With the use of data science and machine learning techniques, we identify the main features observed by the citizens through both text and images, the time that people spent in parks, as well as the top interaction areas. This allows us to gain an overview of certain patterns and the behaviour of the citizens within their surroundings. Furthermore, it proves the capabilities of integrating technology into large-scale social studies.

Chapter 2

Background

This chapter provides a brief theoretical background relevant to the topics presented in this thesis. We discuss in greater detail the Internet of Things, and the edge and cloud computing paradigms. Furthermore, we cover different types of data analytics methods that are encountered throughout this study.

2.1 Internet of Things

The Internet of Things (IoT) is present everywhere in today's world including the industry, the academic world, the health sector, the public infrastructure, and the consumer products. A large palette of devices ranging from simple sensors to microcontrollers to smart appliances to smartphones can be considered part of the IoT. It started being a topic in the research world in 2010, and it truly took off in 2013 [132]. In 2008, the European Commission published a workshop report claiming that in the next 20 years the IoT will become a reality *"with omnipresent smart devices wirelessly communicating over hybrid and ad-hoc networks of devices, sensors, and actuators working in synergy to improve the quality of our lives and consistently reducing the ecological impact of mankind on the planet"* [107]. With a few years still left to go, there are many scenarios and situations within smart cities, industry, etc. that can be accurately described by the aforementioned statement.

The Internet of Things is a paradigm that encompasses a variety of smart, interconnected devices that can collect data and interact with each other to accomplish common goals. It is an important source of heterogeneous data, covering a multitude of application areas. Actually, IoT systems currently generate data at the zettabyte (i.e. a trillion gigabyte) level [187]. This poses important challenges for both the network infrastructure and the processing pipeline.

As a result, solutions including pre-processing at the edge of the network, new architecture models, new processing paradigms, and new lightweight algorithms are being proposed to mitigate the big IoT data challenge. Other significant challenges within the IoT include security and privacy, trust management, interoperability and standardization, ethical and legal concerns, scalability and reliability, along with Quality of Service (QoS) [128], [171].

A visualization for the schematic architecture of the IoT is given by Molaei et al. as part of their paper, a comprehensive review of the IoT and its impact within the mining industry [165]. The authors identify four main layers, namely the perception layer consisting of sensors and actuators, the network layer comprising internet gateways, the Edge IoT concerned with analytics and preprocessing, and the Cloud application layer which doubles as a data center.

This section only aims to provide the reader with a brief overview of IoT in terms of definition, description, and challenges. For further information, we recommend the following sources: Atzori et al. provide in 2010 one of the earlier comprehensive IoT survey studies identifying the many visions, enabling technologies, possible applications, as well as open issues for the IoT [17]. Ge et al. discuss in [87] how big data technologies are employed across main IoT domains such as healthcare, energy, smart cities, industry, etc., whilst providing an extensive review of the research pertaining to both IoT and big data. In his book [132], Perry Lea focuses on both the theoretical, and practical aspects of implementing edge and IoT systems for industry. Finally, in [31], the authors provide a roadmap for the IoT services. They identify the major criteria defining IoT services, namely smartization, augmentation, and contextualization. Furthermore, they suggest future research directions, which also include designing and engineering scalable and robust IoT-based solutions. The report was published in September 2021 and provides an accurate snapshot of the current state of the art and challenges.

In this thesis, the IoT serves as our source of data, as well as the general scenario (application domain) for which we perform different data analytics.

2.2 Edge Cloud setting

Traditionally, cloud computing has been the enabler of data analytics by allowing for storage and processing via advanced techniques of large amounts of data, including the realm of IoT systems. However, triggered by the advancements of IoT devices, their hardware capabilities, and the development of the infrastructure, alongside the drawbacks of cloud computing, new computing paradigms have emerged. We are witnessing a transition from the traditionally

cloud-based systems to a more edge-oriented approach enabled by the rise of edge computing, and/or fog computing. Depending on requirements, current systems and frameworks can be developed as to take advantage of all the computational power available between the IoT data collection points found at the Edge of a system (i.e. sensors, microcontrollers), and the full-fledged systems available at the Cloud level. We consider this to be an Edge Cloud setting. In the remainder of this section, we aim to provide the reader with a brief introduction and overview of the cloud and edge computing paradigms.

2.2.1 Cloud computing

Cloud computing represents the most computationally powerful asset commonly available. The Cloud is a centralized infrastructure system that offers access to a plethora of resources, including servers, databases, computers, and software services. Usually, access to it is on a pay-per-use model, that is to say that customers pay only for the resources that their systems or applications actually use. In [157], the National Institute of Standards and Technology (NIST) provides the standard definition of cloud computing, alongside its five characteristics (on-demand self-service, broad network access, resource pooling, rapid elasticity, measured service), three service models (Software as a Service, Platform as a Service, and Infrastructure as a Service), and four deployment models (private cloud, community cloud, public cloud, and hybrid cloud).

Cloud computing has contributed to the development of the IoT by allowing for storage and computation of the massive IoT data, along with dedicated infrastructure and data analytics tools. Examples of Cloud IoT services include *Amazon Web Services (AWS) IoT*, *Microsoft Azure IoT*, *Google Cloud IoT Solutions*. However, there exist inherent disadvantages of the cloud computing model for the case of the ever-growing IoT. Li et al. highlight that the bandwidth limitation, as well as the global centralization of data processing and storage inadvertently triggered different degrees of latency, data transmission overhead, less mobility support, and lack of context awareness [136]. These issues along with an increasing need for quick information access, real-time response times, and efficient data processing encouraged the development of edge-oriented paradigms, especially for the case of IoT.

Herein, we provide for interested readers a few literature pointers tackling the topic of cloud computing, whilst taking into account the IoT or edge-oriented paradigms: Botta et al. discuss in their 2015 survey [30] the topic of integration of cloud computing and IoT. They present both fields, alongside the motivation for integrating the two, applications, platforms, challenges, and open issues. Elazhary provides a comprehensive review of how IoT and

cloud computing intersect while considering a variety of edge-oriented emerging paradigms [57]. Authors in [16] discuss and evaluate performance metrics for cloud, fog, and edge computing. The work in [181] provides a comparison of the cloud, fog, and edge computing paradigms from a software engineering point of view. After introducing all three concepts, the authors compare functional requirements such as scalability, interoperability, etc., and discuss applications with different time considerations (real-time, near real-time, and non-real-time).

2.2.2 Edge computing

Edge computing is a fairly new distributed computing paradigm that advocates for the processing and storage of the data as close as possible to the data source itself. It emerged in response to the disadvantages of the centralized model of cloud computing. By taking advantage of the computational power available at the edge of the network, response time, latency issues, data transmission, and bandwidth usage can be reduced (in comparison to the traditional centralized model). Gartner estimates that by 2025, 75% of enterprise-generated data will be processed outside a traditional centralized data centre or cloud, as opposed to 10% in 2018 [160]. The growth and expansion of the edge computing paradigm are a direct result of the growth and development of the IoT. The scale and complexity of the IoT generated data have outpaced network and infrastructure capabilities, allowing edge computing to offer a more efficient alternative with data being processed and analysed close to the data source [105].

A standard definition of edge computing does not exist, but its main characteristics do include being close to the data source, and being a distributed computing framework. In the literature, edge computing can be often found alongside fog computing. Mahmud et al. define fog computing as a distributed computing paradigm that serves as an intermediate layer between the cloud and the IoT devices [146]. Some researchers place fog computing in between the cloud and edge computing (with the edge layer comprising the IoT devices), while others consider it part of edge computing. Kalyani et Collier [117] place edge computing at the network edge and edge devices level, with a focus on the IoT level, whilst fog computing is located near the edge and the network core, with a focus on the infrastructure level. Both paradigms have limited computational resources, with those pertaining to the edge being more limited.

In the context of this thesis, we consider the edge as encompassing the variety of IoT devices, the edge of the network, and in the immediate vicinity of the data source. Where we do discuss the concept of fog, we consider it to be in between the edge and the cloud, with a closer link to the cloud.

It is important to note that taking advantage of the Edge, in terms of computational power, does not imply that the Cloud should not be involved in the computation anymore. Ideally, for each type of application and system, the computational needs shall be split in accordance with the requirements, all throughout the Edge Cloud processing pipeline, without involving the Cloud unless actually needed.

For a reader interested in knowing more about the topic briefly presented in this section, we suggest the following reads: Yu et al. discuss in their survey [236] edge computing for the IoT, while looking at integration, advantages, and challenges. In [23], the authors investigate the moving to the *Edge-Cloud-of-Things*, i.e. presenting the move from the centralized cloud platforms to decentralized platforms. The analysis includes the state of the art of different edge paradigms, their role, as well as challenges and research directions. Li et al. [136] review edge-oriented paradigms, while also focusing on architecture and resource management. [69] compares a full-cloud and edge-cloud architecture for a study case concerned with deep learning based anomaly detection.

Throughout this thesis, a great emphasis is placed on the edge computing paradigm. We focus on how anomaly detection and data imputation can be carried out at the edge, close to the data source, and in sensor systems. However, we do acknowledge the easiness, convenience, and importance of also using cloud applications and more powerful resources for the social case study (IWUN, Chapter 5).

2.3 Data analytics

Massive amounts of data are collected in a plethora of IoT situations. However, the collected data is often raw, messy, and without any meaning by itself. Data analytics is required in order to make sense of the collected data and gain valuable insight into the situation at hand.

Data analytics is a general term used to represent different types of data analysis. In this thesis, we are concerned with data analytics for the IoT, in an Edge Cloud setting. There are three main directions that are explored within this study, namely smart anomaly detection, data imputation, and data analytics pertaining to a real social case study. These directions are often encountered in an IoT scenario, and they pertain to current research topics in the field.

When it comes to data analytics in an IoT Edge Cloud setting, we need to also consider the available techniques. On one hand, there are the more traditional methods, often inspired by statistics; on the other hand, there is an increase in the use of those from the machine learning realm. Throughout this thesis, we cover both types, with a focus on the machine learning

techniques, and investigating the potential of employing them in the edge environment, rather than the traditional cloud, or equally computationally powerful devices. The use of machine learning techniques at the edge is a step towards edge intelligence. Data is kept close to where it is generated, and it is analysed and processed with the help of powerful techniques. Machine learning often outperforms classical methods because it is a better fit for uncovering patterns in large, unorganized amounts of data. More details about machine learning can be found in Section 3.4. Even though we cover the topic of anomaly detection, part of the information is not dependent on the application scenario.

For those interested in machine learning and data analytics for the IoT, we suggest the following read: [2] offers an in-depth overview of the convergence of machine learning and IoT, along with a critical review on data processing and knowledge discovery for the IoT. Furthermore, the authors propose a framework on how machines could autonomously manage knowledge, as well as future research opportunities.

2.4 Summary

This section provided the reader with a wider context of where our work fits. We discussed key concepts such as IoT, edge and cloud computing, as well as data analytics. We have also provided a few fundamental references for the interested reader.

The concepts discussed herein fit within the title of this thesis, namely *Data Analytics in an Internet of Things Edge Cloud Setting*, which aims at setting the scene and context of this work.

Chapter 3

Smart Anomaly Detection in Sensor Systems: A Multi-Perspective View

Anomaly detection is concerned with identifying data patterns that deviate remarkably from the expected behaviour. This is an important practical research problem, due to its broad set of application domains, from data analysis to e-health, cybersecurity, predictive maintenance, fault prevention, and industrial automation. Herein, we review state-of-the-art methods that may be employed to detect anomalies in the specific area of sensor systems, which poses hard challenges in terms of information fusion, data volumes, data speed, and network/energy efficiency, to mention but the most pressing ones. In this context, anomaly detection is a particularly hard problem, given the need to find computing-energy-accuracy trade-offs in a constrained environment. We taxonomize methods ranging from conventional techniques (statistical methods, time-series analysis, signal processing, etc.) to data-driven techniques (supervised learning, reinforcement learning, deep learning, etc.). We also look at the impact that different architectural environments (Cloud, Fog, Edge) can have on the sensors ecosystem. The review points to the most promising intelligent-sensing methods, and pinpoints a set of interesting open issues and challenges.

This chapter is based on our paper, "*Smart anomaly detection in sensor systems: A multi-perspective review*" [59] published in the *Information Fusion* journal. Despite being published only in March 2021, it was available online starting October 2020, and the paper currently has 23 citations (November 2021).

3.1 Introduction

Thanks to the hyper-connectivity of the Internet of Things (IoT), electronic sensors and sensor systems have become major generators of data, currently reaching yearly rates on the zettabyte (i.e., a trillion gigabyte) scale [187]. This ever-increasing amount of data has reached the big-data sphere [87], not only for its sheer volume but, especially, in terms of variety, velocity, veracity, and variability. Thus, relying merely on cloud-assisted computing for the analysis of sensor data would pose too much of a burden on the network. In this context, a vast body of researchers has been investigating a range of methods for detecting anomalies in sensor systems, a critical building block in IoT networks and systems.

Our aim is to review methods, ranging from conventional techniques (statistical methods, time-series analysis, signal processing, etc.) to data-driven techniques (supervised learning, reinforcement learning, deep learning, etc.). Moreover, we look at the impact that different architectural deployments (Cloud, Fog, Edge) could have on the sensors world. The review takes into account the most interesting smart sensing methods, and identifies a set of appealing open issues and challenges.

Anomaly detection [41] is a much broader problem, going well beyond the sensor systems that we scrutinize herein, and dating back many years in the research panorama. It pertains to a vast number of application domains, each one with its peculiarity and constraints. Prominent examples, beyond the general fields of data analysis and artificial intelligence, are cybersecurity, predictive maintenance, fault prevention, automation, and e-health, to mention but a few.

More generally, anomaly detection is concerned with identifying data patterns that deviate remarkably from the expected behaviour. This is crucial in the process of finding out important information about the system functioning, detecting abnormalities that are often rare or difficult to model or, otherwise, to predict [4]. Timely identification of anomalies is crucial to tackling a number of underlying problems that, if undetected, may lead to costly consequences. Examples are: spotting stolen credit cards; preventing systems failure; or anticipating cancer occurrence.

Anomaly detection has conventionally been tackled from the statistical viewpoint. The prominence of machine learning (ML) has, however, opened new possibilities for the detection of outliers, particularly thanks to the availability of vast amounts of data to train sophisticated learning models. This is an attractive proposition, particularly in domains such as the IoT, whereby new data patterns make it difficult to use static models [17, 73].

Sensor systems play an important role in modern interconnected digital infrastructures, for instance in environmental monitoring, smart cities, factory automation, autonomous

transportation, and intelligent buildings. Typically, multiple sensors, including heterogeneous ones, come together to form a system, whereby neighbouring devices can communicate with each other and transfer data to a cloud infrastructure for further elaboration [74]. Given the rich palette of sensing devices, the type of data that flows through the system can vary greatly, in format, shape (in time and space), and semantics. That is why the process of separating out normal from abnormal sensed data is a particularly challenging one.

In the context of IoT applications, sensors are the principal source of big data, hence anomaly detection at the edge can be a powerful means to address the inevitable data communication bottlenecks. An example is offered by an analysis by Cisco [49], which estimated that a smart city with 1 million inhabitants generates daily data rates at the tune of 180 Petabytes. The real challenge is to efficiently extract useful information out of abundant but mostly useless raw data, which can lead to considerably limited transmission (and subsequent storage and processing) of sensor data from the network edge to the data centres.

To this end, machine learning methods are being investigated as a promising way to automate the process of collecting data (at the sensors) for the purpose of analysis, first locally and, then, in the cloud. This fits with the definition of machine learning (ML) by Murphy [169], as a set of methods that can automatically detect patterns in data, and then use the uncovered patterns to predict future data or to perform other kinds of decision-making under uncertainty. This notwithstanding, such methods still need to be refined and improved to tackle the vast amount of data produced by sensor systems [183].

Another peculiarity of sensor data, compared to other data-intensive systems, is the tendency to shift from offline data processes to real-time (or near real-time) requirements, whereby a timely elaboration of the raw data into usable information becomes a crucial necessity. Yet, not only industrial IoT processes but also other mission-critical domains (such as smart cities or disaster recovery applications) rely on anomaly detection (along with other data processing functionality). This criticality has led to a range of studies on how to best distribute data processes between the system's edge (sensor nodes and aggregators) and the cloud infrastructure. As a response, a broad range of studies has looked at the problem of shifting some processes from the data-intensive cloud facilities to edge devices and sensors, which are severely constrained in terms of computing and energy budget [12].

In fact, in this chapter, we argue that this architectural shift from cloud-centred to cloud-assisted processing (as illustrated in Section 3.5) pertains not only to conventional anomaly detection models (Section 3.3) but also to the emerging data-driven (and even machine learning) algorithms (Section 3.4). This trend has also been documented in [236], whereby the focus was on comparing IoT deployments on different architectural models, based on cloud and

edge computing, respectively. Similarly, authors in [174] suggest that anomaly detection at the edge of the network can act as an important building block for edge intelligence, including pre-processing, filtering, and data compression tasks. In turn, edge computing allows for scaling up IoT systems and communications.

An interesting body of surveys has appeared in the literature, touching aspects that are relevant or complementary to this work. It is worth mentioning a selection of papers that have provided a snapshot of the state of the art at different points in time, particularly when machine learning was still not sufficiently mature for edge computing. A 2004 survey by Hodge and Austin, provides a general overview of methodologies for outlier detection, both statistical and machine learning based [99]. A broad overview of anomaly techniques spanning diverse research areas and application domains is given in [41]. Neither of the works is specifically targeting sensor systems, which is our focus. Also, important recent developments in lightweight machine learning methods, in connection with fog and edge computing, could have not been captured.

Useful, background information relevant to this chapter can be found in [192], which presents anomaly detection techniques for evolving data, namely data streams and evolving graphs. Techniques for discrete sequences have been introduced in [40]. A perspective on intrusion detection in IoT systems is given in [24].

An interesting set of review papers by Markou and Singh provide a comprehensive background on novelty detection techniques, looking specifically at neural networks [150] and statistical approaches [149], with an updated review in [179]. Those papers focus on presenting the reader with a wide range of techniques and in-depth explanations, alongside their strengths and weaknesses. While their emphasis is on computational – intensive methods, we focus on the sensor systems issues and constraints, looking at advancements in lightweight machine learning and architectural paradigms that allow for anomaly detection closer to the network edge.

More closely related to our work is the paper by Xie et al. [230], offering a review of anomaly detection techniques, specifically for Wireless Sensor Networks (WSNs). The authors introduce key design principles to be considered for anomaly detection in WSNs, including WSNs security and node failure topics. Interestingly, they introduce a number of exemplary cases for each category and research strand, specifically for WSNs, along with two types of WSN architectures, namely flat and hierarchical ones. In our review, we consider architectural approaches and computational models that had not reached maturity back in 2011, looking at 1) the broader sensor systems ecosystem; 2) Fog, Cloud, Edge, and distributed computing for sensor systems; and 3) lightweight ML for constrained devices.

In this chapter, we provide an up-to-date snapshot of anomaly detection techniques, with a specific focus on methods suitable for sensor systems and cloud-assisted sensing. This involves both data-intensive methods, suited for cloud computing, and lightweight methods, aimed at edge and in-node computing. The context of our review is exemplified in Figure 3.1, where we depict the software architectural elements (bottom) and a taxonomy of the methods considered for sensor systems (top).

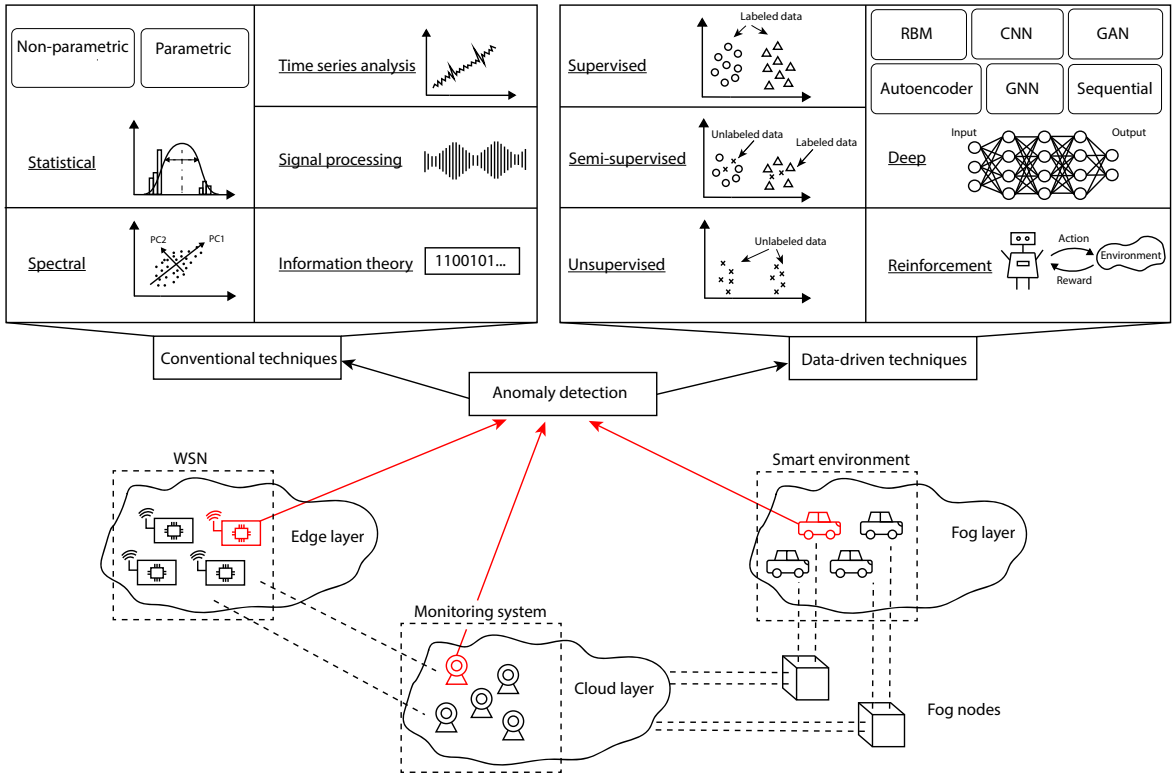


Figure 3.1: High perspective on sensor networks with a focus on anomaly detection techniques (conventional vs. data-driven), and architectural models (Cloud, Fog, Edge).

The chapter is organized as follows. Section 3.2 provides an introduction to anomaly detection and places it in the particular context of sensor systems. Our taxonomy considers conventional, more established techniques for anomaly detection in Section 3.3. We then consider data-driven methods for anomaly detection (Section 3.4), looking closely at the options that machine learning offers for sensor systems. We then switch to an architectural viewpoint (Section 3.5), discussing options to deploy anomaly detection processes in the Cloud, in the Fog, or at the Edge. We finalize with a glimpse of what the future may hold for the field (Section 3.6), discussing interesting research issues and challenges.

This is a particularly prolific investigation domain, where both the software architectures

and the computations methods are evolving, and there is a trend to push intelligence toward the edge, close to where the data is actually generated. We discuss the miniaturization and acceleration of data-intensive methods, energy efficiency, hierarchical learning models, and data heterogeneity and fusion.

3.2 About anomalies

This section provides a general introduction to anomaly detection. In the second part, we put anomalies in the particular context of sensor systems.

3.2.1 The concept of anomaly

Anomalies are also known as outliers, abnormalities, or deviants. Hawkins [98] defines an outlier as *"an observation, which deviates so much from other observations as to arouse suspicions that it was generated by a different mechanism."* An important consideration in anomaly detection is the type. The classification includes the following categories [41]:

- point anomalies: when an individual data point is different from the rest of the data;
- contextual anomalies: when a data instance is anomalous in a specific context only, meaning that in all other situations it would be perceived as normal;
- collective anomalies: when a group of related data points is anomalous compared to the dataset; the individual data points could represent normality, while it is their actual sequence that represents an anomaly.

Another important aspect to consider is the type of input data, as this requires certain techniques and poses a number of challenges. Some details to consider are dimensionality (unidimensional, or multidimensional); the number of attributes (univariate, or multivariate); and any existing relationship between the data instances. It is common for the data points to be related to each other, as is the case for sequence data, spatial data, and graph data [41]. For sequence data, also known as temporal data, the data points are linearly ordered, representing an ordered series of events. Common examples include time series (both discrete and continuous), genome sequences, and so on. Spatial data consists of points that are related in space. Data can be both spatial and temporal, as is the case for images and video, and it is referred to as spatio-temporal. Other types of data which pose difficulties for anomaly detection include:

- Evolving data: the environment that one monitors to detect anomalies can be non-stationary, meaning that the data changes in time, as the characteristics of the system dynamically change. As a result, the models and algorithms need to be adaptive and account for the changes which occurred. Classic stationary algorithms do not work as the underlying data distribution changes constantly.
- Streaming data: it is a sequence of data points that is mainly generated by real-time data sources. It is characterized by a continuous flow and a high number of data instances over a short period of time. It poses multiple technical challenges, such as storing and processing the data on the go or online. Furthermore, the data can also evolve and change throughout time.
- Correlated data: in certain situations, when multiple data sources are used to monitor one particular system, it is often likely that the generated data streams are correlated. In these cases, for detecting anomalies, it is worth analysing also the combination of the data streams as certain data instances may not be anomalous by themselves, but they may indicate an anomaly when looked at all-together. The data instances that can be related to each other can also be part of the same stream, as is the case with sequence data. In the latter case, in order to detect anomalies, one needs to analyse a sequence of data instances of a certain length.
- Heterogeneous data: in IoT systems, it is often the case that the collected data is heterogeneous. Generally speaking, analysing such data poses challenges as in how to combine and interpret the amount of information provided.
- Contaminated data: it refers to situations in which the data source is affected by noise from the environment, or in which there are missing values in the data, for instance as a result of different hardware or software malfunctions. The challenge arises from the fact that it is difficult to distinguish between true anomalies and different degrees of contamination. Furthermore, the contamination could strongly affect anomaly detection.
- Big data: the challenge when working with big data comes from the overwhelming size of the data, as captured by the five V's: volume, velocity, variety, veracity, and value.

3.2.2 Anomalies in sensor systems

Besides the aspects highlighted previously, the application domain and the specific scenario play an important role in choosing a technique to detect anomalies. Looking at sensor

systems, common scenarios include Wireless Sensor Networks (WSNs), Smart Cities, Smart Environments, and, generally, IoT systems. Common sources of anomalies include [27]:

- Environment: when the state of the environment changes, e.g. a new component is introduced, a highly unusual event occurs (such as natural disasters).
- System: the anomalies in this category are generated by the fact that a component of the system malfunctions or breaks down. The part of the system affected can be anything from a sensor to a cluster of nodes. The anomalies can be generated by hardware faults or limitations.
- Communication: these types of anomalies occur in systems that make use of different communication technologies (often wireless). They are caused by loss or delays of the communication packages. Part of these anomalies can be managed or avoided by making use of communication protocols in the applications.
- Various attacks: these anomalies are introduced into the system by a malevolent party. They are created by different means, ranging from physical intervention on the system to loading communication traffic.

Furthermore, besides the anomaly classification presented in 3.2.1, for sensor systems we can also encounter the following categorization which is based on the faults noticed in a real deployment [170], [201], [234]:

- Spike: a peak of short duration in the recorded value. Usually, this distinctively deviates from the other measurements.
- Noise: there is an increase in the variance of a number of successive data samples. The real values can be strongly affected.
- Constant: a sensor reports as a measurement a constant value, indifferent to the actual conditions.
- Drift: an offset is recorded in the measurements of the sensor values.

At this aim, the authors in [170] present a more detailed taxonomy of sensor faults, with more categories, as well as possible causes and implications. In [37], Cauteruccio et al. propose three different anomaly taxonomies and formalize eight types of anomalies in the context of developing a framework for anomaly detection and classification within multiple IoT scenarios.

In sensor systems, most data to be analysed is collected by the sensors connected to a processing unit. Classic sensors are more likely to collect data in the form of time series. Nevertheless, cameras also act as sensors, collecting images and videos. These are grouped under the category of spatio-temporal data. Anomalies are usually identified as unusual local changes in the spatial or temporal values. For spatial data (images), the surrounding neighbours of a data point play an important role in differentiating normal from abnormal data [206].

By looking at the research areas that have proposed various techniques for anomaly detection, we could characterize two main categories, namely: conventional techniques and data-driven techniques. In the first category, we have included techniques belonging to research fields with a long history in the topic, such as methods from statistics and from signal processing. In the data-driven category, we have included techniques based on machine learning and data mining. These research fields have become particularly active in the past few years, due to new technological advancements and current opportunities. In the following sections, we will present these two categories, with emphasis on the data-driven methods. It is important to note that these two categories are not mutually exclusive; there are in fact solutions and algorithms that combine different techniques from both areas to address anomaly detection problems. Furthermore, there exist other categorizations of the methods used for outlier detection. For example, in [41] the authors group the methods in the following important categories: classification based, clustering based, nearest neighbour based, statistical, information theoretic, and spectral.

3.2.3 Anomaly detection datasets

When coping with anomaly detection problems, it is crucial to validate the effectiveness of a proposed technique onto real data. Indeed, since anomalies can occur unpredictably and at any time, it can be difficult to generate them artificially. At this aim, industry and academia make available some datasets containing a number of anomalies "fingerprints" captured on the field. Here, we make a brief excursus about the most credited anomaly detection datasets, which include both real and artificially generated data and cover multiple application domains.

The Bot-IoT dataset [124] contains a collection of normal and anomalous network traffic generated within a real IoT scenario at UNSW Canberra Cyber Lab. It is specifically exploited to validate techniques aimed at discriminating regular from anomalous flows within a sensors network.

Outlier Detection DataSets (ODDS) [202] provides access to a collection of datasets (with

ground truth if available) from different domains including time series for event detection, crowded video scenes, and opinion fraud detection data from online review systems.

The Numenta Anomaly Benchmark (NAB) [131], [5] offers time series datasets concerned with real-time anomaly detection in streaming data generated by sensor systems.

Yahoo Webscope [129] includes a labeled anomaly detection dataset released by Yahoo Labs, consisting of time series (both real and synthetic) with tagged anomaly points.

UCI Machine Learning Repository [55] comprises several hundred datasets (from video surveillance to automotive domains) explicitly designed to evaluate machine learning techniques, including the ones allowing anomaly detection.

Again, the ELKI outlier [58] dataset offers a collection of data whose details are available in a dedicated study [35], where the authors focus on the analysis of unsupervised outlier detection algorithms.

Finally, the UCSD anomaly detection dataset [218] is a collection of images acquired by a camera overlooking pedestrian walkways; anomalies include the presence of bikers, skaters, small carts, and people walking across a walkway or in the grass that surrounds it.

3.3 Conventional techniques for anomaly detection

This section provides an overview of the most conventional anomaly detection techniques developed over the years. There is a long history of statistical methods that can be used to identify outliers. We should, however, note that even the most recent machine learning algorithms are grounded on their statistical counterparts [246]. Furthermore, traditional probabilistic techniques are successfully being used by machine learning methods, e.g. Bayesian networks and Bayesian classifiers [101]. In the following, we will consider a selection of conventional techniques, highlighting strengths and limitations, which lead to the more recent data-intensive approaches.

3.3.1 Statistical methods

With statistical methods, it is assumed that the data points in a system are generated according to some statistical model. Any deviation from the expected model is regarded as an anomaly. In order to detect anomalies, statistical inference tests are applied. The types of techniques in this category can be parametric or non-parametric. For the parametric category, the underlying distribution of the data is known, and the parameters are estimated based on the data. Examples

of parametric methods include those based on the gaussian model, the regression model, or a mixture of parametric distributions [41].

On the other hand, in the non-parametric case, the parameters of the underlying distribution are not known; they are determined from the existing data. Examples include methods based on histograms or on the kernel function. The work of [99] also considers the case of proximity based techniques, such as kNN, as well as that of semi-parametric methods, where local kernel models are applied instead of a single global distribution model. The research in [149] provides an extended review of the types of statistical techniques used for novelty detection. It includes parametric approaches, such as probabilistic and gaussian mixture modeling, hidden Markov models, hypothesis testing, and non-parametric ones, such as kNN based, Parzen density estimation, string matching, and clustering. Again, the authors in [179] provide an updated review of novelty detection, including probabilistic approaches.

Statistical techniques have the advantage of being explainable as well as interpretable, especially when the distribution of the underlying data is known. In addition, some of these techniques, such as histogram-based or those which model single exponential distributions, can be easy to implement, or computationally efficient. However, kernel based techniques and models with complex distribution induce a higher computational complexity.

A limitation of the statistical approaches is that testing for outliers assumes that a specific distribution characterizes the recorded data points. This turns out not to be the case for most of the high dimensional data. Another challenge is that it may not be straightforward to choose the best statistics test for detecting the anomalies. Furthermore, histogram based techniques are not suited for multivariate data, as they do not take into account the interaction between the attributes of the data. When dealing with high dimensional data, machine learning techniques (Section 3.4) tend to perform better, as fewer assumptions are made about the actual data values.

3.3.2 Time series analysis

It is often the case that the data in sensor systems is generated as a time series. Time series analysis (TSA) is concerned with methods that analyse the characteristics of the data, to extract useful statistics from them. Time series forecasting is often used to predict future expected values of the data to be recorded. The difference between the actual value and the expected one can be used for highlighting possible anomalies.

Other common methods include cross-correlation analysis, autoregressive moving average (ARMA), autoregressive integrated moving average (ARIMA), Kalman filtering, etc. Authors

in [164] observe that the following time series models are often adopted in anomaly detection engines for the IoT: autoregressive models, symbolic TSA, seasonal trend loss decomposition, as well as combinations between machine learning techniques and TSA. In [168], the authors advance a proposal about fusing statistical models (ARIMA) and deep learning models (convolutional neural networks) for unsupervised anomaly detection. This approach takes advantage of the strengths of both techniques and outperforms the state-of-the-art anomaly detection methods on a public dataset (Yahoo Webscope). The ablation and comparative study carried out by the authors shows that the fusion of the two techniques performs better than the individual components.

Authors in [198] suggest a residual-based approach for detecting faults in large-scale sensor networks by the use of vectorized autoregressive moving average models (VARMA) and multivariate orthogonal space transformations. Three models are being proposed and compared against other existing techniques (state-of-the-art, linear, non-linear). One model (NFIR) together with the orthogonal transformations is likely to detect faults in the proposed problem. The authors' approach outperforms several state-of-the-art solutions.

Time series analysis has the undoubted advantage of being simple and effective since the pertinent outcomes can often be interpreted intuitively, especially when dealing with additive outliers. Unfortunately, it works well mainly for tracking "moderate" anomalous events. Unsatisfactory results are often achieved in cases when the anomalies are generated by more "dramatic" changes.

3.3.3 Signal processing

Signal processing deals with analysing different types of signals, including sounds, images, outcomes of monitored physical processes, data streams from sensors, etc. Signal processing techniques are often adopted when signals are affected by noise, thus, specific de-noising mechanisms are put in place. This may reveal possible anomalies also in the presence of noise.

Different transforms can be used to aid in anomaly detection, such as the Fourier transform and the wavelet-based ones. For instance, [185] presents a wavelet-based approach for generating symbols for anomaly detection. The symbols are generated from the wavelet coefficients of the time series data. The authors show that the choice of an appropriate wavelet basis and scale greatly improves computational efficiency for real-time applications. The proposed technique is validated experimentally for a number of cases, as well as compared to another partitioning technique (symbolic false nearest neighbour).

In [32], authors analyse how temporal aggregation in random packet sampling alters the

properties of the signal by introducing noise and aliasing that, in turn, affect the anomaly detection systems. They propose replacing the aggregation step with a specifically designed low-pass filter. As a result, aliasing is prevented and the new solution performs better in regard to the misdetection and detection rates, even when considering a low sampling rate of the packets.

As occurs along most statistical methods, signal processing techniques exhibit a good detection rate when tackling the so-called "zero days" anomalies, namely, abnormal events that have never occurred before the current observation. By contrast, these techniques often rely on unrealistic assumptions (e.g. quasi-stationarity) of processes such as noise, which may detrimentally affect performance.

3.3.4 Spectral techniques

The spectral approach deals with dimensionality reduction. It is assumed that the data can be embedded into a lower dimensional subspace where the normal and anomalous instances greatly differ. Techniques can be based on Principal Component Analysis (PCA), a popular technique for projecting data into a lower dimensional space. Authors in [104] propose a framework for network-wide anomaly detection based on distributed tracking and PCA. In [56], a proposal concerning a spectral anomaly detection method for WSNs by developing a graph-based filtering framework is advanced. The graphs are chosen as to include structural (proximity) information about the measured data. The authors show that standard PCA-based methods are a special case of their proposal. Their technique outperforms other state-of-the-art methods in global and distributed scenarios.

The reduction of dimensionality characteristic of the spectral techniques represents an advantage when dealing with high dimensional data. Furthermore, it is suitable to be used as a preprocessing step before applying other types of anomaly detection techniques to the subspace. The drawbacks of spectral techniques often include high computational complexity, as well as not being applicable unless the anomalies can be separated from the normal conditions in a lower dimensional embedding of the data [41].

3.3.5 Information theory

Methods in this category are aimed at analysing the information content of a dataset by employing different information theoretic measures such as Kolmogorov complexity, entropy, relative entropy, etc. The assumption behind these techniques is that anomalies distort the

information content of the data instances in an otherwise normal dataset. Usually, the metrics are first computed using the whole dataset; then, a subset of points needs to be found. If this subset is eliminated, it induces the largest difference in value for the chosen metric. Authors in [135] propose several measures for anomaly detection, namely: entropy, conditional entropy, relative conditional entropy, information gain, and information cost. These measures are used in three use cases to illustrate their utility. Furthermore, in [14], the information bottleneck formalization for outlier detection is exploited.

The work in [41] outlines the advantages and disadvantages of information theoretic methods. In particular, the specific information theoretic measure chosen for anomaly detection influences the performance of the various algorithms. It is often the case that anomalies are detected by the chosen measure only if there are numerous anomalous data. Furthermore, when dealing with naturally ordered data, the information theoretic methods depend on determining an optimal substructure size to break the data. However, these techniques do not assume an underlying statistical distribution of the data and can be used in an unsupervised fashion.

3.3.6 General considerations about conventional techniques

Conventional techniques for anomaly detection have been well assessed across the years, being also supported by robust and well established literature. In particular, these techniques allow for a rigorous quantification of an outlier, which may be detected through either of the following methods: 1) a deviation from an expected underlying model (statistical methods); 2) an abnormal metric value (e.g. the Euclidean distance) in measuring the difference between the expected and the predicted value (time series); 3) a frequency change highlighted in the Fourier transformed domain (signal processing); 4) a reconstruction error between the input and its projection on the eigenvectors space (PCA - spectral techniques); 5) a distortion in the information content (information theory).

All the aforementioned techniques mainly require a good knowledge of the so-called "ground truth", so as to derive a well quantifiable measure of the anomaly. Unfortunately, in many real-world scenarios where data models are extremely time-variant, the complexity of such techniques is not rewarded by satisfactory performance. This calls for the introduction of data-driven techniques which, at the expense of a less strict formalization, allow for more flexible adjustments, in line with the high dynamism of the Big Data paradigm.

3.4 Data-driven techniques for anomaly detection

Data-driven techniques typically refer to learning-based methods where the absence of a robust underlying mathematical model is compensated by the availability of large amounts of data, from which one can "learn" useful information. Machine learning is a vast research field with many application areas. Usually, it is classified into three main categories, namely supervised learning, unsupervised learning, and reinforcement learning. Nevertheless, there also exist other combinations, such as semi-supervised learning. Furthermore, with the technological advancements, also deep learning is on the rise. Many machine learning techniques are often getting a deep approach or are combined with deep learning.

Similarly, Hodge and Austin [99] discern three fundamental approaches to the problem of outlier detection, namely:

- Supervised: both the normality and abnormality are modelled; it requires labelled data for each of the categories;
- Unsupervised: identifying anomalies with no prior knowledge of the data;
- Semi-supervised: only normality is modelled; anomalies are identified by the fact that they are not within the normal threshold; it is also known as novelty detection or novelty recognition.

We can see that these three approaches present some overlap with the main categories of machine learning, as they share common characteristics. In the next subsections, we aim at providing the reader with a quick background of the main data-driven techniques, as well as discussing how these are being used for anomaly detection in sensor systems. We look at the categories of machine learning techniques corresponding to the approaches outlined by Hodge and Austin [99], adding new ones based on reinforcement learning and on deep learning.

3.4.1 Supervised learning

Supervised learning refers to machine learning techniques that train a model using a set of examples with a target output (labels). Supervised learning for anomaly detection has peculiar challenges when compared to other supervised learning applications. In practical cases, the rare samples are usually fewer in the training instances.

There are modifications to algorithms for such class–imbalanced scenarios, to increase the impact of the rare instances onto the models. One popular approach is to make use of

cost-sensitive learning. The training data could be relabelled using the costs, as is the case with Metacost [54]. In Metacost, the instances that have a reasonable probability of being in another class are relabelled to that class. In most cases, it is the normal instances that get relabelled.

Another type of cost-sensitive learning is the weighting method, which applies a weight on the training instance that represents its misclassification cost. Such modifications have been made to common classification algorithms like the proximity-based classifiers [148], Support Vector Machines (SVMs) [3], decision trees[213], [223], and rule-based classifiers [113].

Supervised techniques exhibit great robustness since the "ground truth" is represented by pre-labeled data. Unfortunately, in many real systems, such information is only partially available or not available at all. The introduction of semi-supervised and unsupervised methods fills this gap.

3.4.2 Semi-supervised learning

Semi–supervised learning generally refers to machine learning techniques that employ a small amount of labelled data and a large size of unlabelled data. Semi-supervised approaches can also be referred to as machine learning techniques that train a classifier with 'normal' sensor data, such that the anomalies can be constructed and evolve dynamically [111]. Though no explicit labels are used, this type of training dataset can be obtained in the real world through some form of labelling or separation of the normal data from a large pool of normal and abnormal data.

Deep learning techniques like autoencoders, and restricted Boltzmann [70], have been applied for this learning task. In the case of an autoencoder, the model is trained with the normal data, and it learns to reconstruct the input at a very small reconstruction error score. At detection, anomalies can be introduced. These anomalies would be reconstructed at a higher error score since they were not seen by the model during the training phase. A threshold is defined to capture the anomalous data.

Chong and Tay have applied this approach for the detection of anomalies in video data using, a spatio-temporal autoencoder [48]. The challenge with this technique, aside from the practical limitations of obtaining normal datasets, is that the optimal threshold can be challenging to define when obtained experimentally. One Class SVM (OC-SVM) is a type of semi-supervised SVM that does not require labels of the anomaly. It was applied in [82] for the detection of attacks in smart city sensor networks.

Although semi-supervised learning is the best choice to employ when only a few labelled

data are available, some limitations arise from assumptions in connection to the use of unlabelled data. These are based on relationships between labels and the unlabelled data distributions, whereby bad assumptions may lead to poor performance.

3.4.3 Unsupervised learning

Unsupervised learning refers to learning from data where the desired output is not available. A major challenge to deploying an anomaly detection system in sensor networks is the need for labelling data for use by learning algorithms. With unsupervised learning algorithms, we can build detection models without the need for manual labelling, thus reducing the deployment cost. Unsupervised learning algorithms are based on the assumption that the anomalies are rare and significantly different from the normal instances [134].

One of the most popular methods is based on clustering, which uses a similarity measure to cluster data instances. Anomalies are identified as data instances that do not belong to clusters or which have clusters significantly smaller than the others. In [186], the authors proposed a global outlier detection technique for anomaly detection in sensor nodes using clustering.

Other unsupervised learning algorithms are based on probabilistic modelling algorithms, whereby the anomalies are detected by estimating the likelihood of each data instance. An example is offered in the work of [219], where a Bayesian network was used as a form of unsupervised learning of the temporal and spatio-temporal data of a gas monitoring sensor network.

The supervised, semi-supervised, and unsupervised techniques work well in many data-driven situations. A general drawback of such methods is that they do not behave proactively when changes occur, unless external guidance is provided. At this aim, more recent methods, such as reinforcement learning and deep learning, have been developed, both based on the possibility to learn autonomously.

3.4.4 Reinforcement learning

Sutton and Barto [208] explore a computation approach to learning from interaction, namely reinforcement learning (RL). This approach is the closest to how humans learn, and focuses on mapping situations to actions in order to maximize a goal through a numerical reward signal.

The main actors are a software agent, which can take actions, and the environment, whose state is affected by the actions that the agent takes. The action to be taken is chosen according to a policy that defines the behaviour of the agent. After each action, the agent receives a

reward according to a reward function. The purpose of the agent is to maximize the total reward in the long run.

Besides the reward function, which defines short-term desirability, an important element is the value function, which defines long-term desirability, namely the desirability of the possible states to follow.

What differentiates reinforcement learning from the other types of learning is its independence from supervision and focus on decision-making in pursuit of a defined goal.

Reinforcement learning is often used in IoT scenarios with multiagent settings. It can be used in order to create a system that adapts to the environment. For example, Chincoli et al. have proposed a protocol for controlling the transmission power by means of reinforcement learning in a multiagent system in IoT [46].

For reinforcement learning to be employed in the field of anomaly detection, more attention needs to be paid to how the problem can be formulated. Nevertheless, there exists some work in the area. Servin et al. introduced distributed RL in a hierarchical architecture of network sensor agents [199]. The system needs to interpret the signals and the states from a team of agents to give an alarm in the case of an intrusion (intrusion detection). Oh et al. have looked at anomaly detection for sequential data [172]. The authors use Inverse Reinforcement Learning (IRL) to infer the reward function by looking at the sequence of actions taken by a target agent. In this way, outliers in trajectory data are identified. In [103] the authors suggest a time series anomaly detector based on a Recurrent Neural Network and RL. The RL method is used for the self-learning process.

While reinforcement learning is based on the concept of dynamically learning by tuning actions with the aim of maximizing a reward, novel deep-based approaches help to find patterns useful to make predictions on new data. Such operations are possible thanks to the presence of various (deep) layers, characterizing the underlying artificial neural networks (ANN), so as to emulate a human brain.

3.4.5 Deep learning

Deep learning [91] emerged from the traditional ANNs. In terms of architecture, what makes deep learning different is that the hidden layers are more in depth (deep) than the traditional ANNs, which have few hidden layers (shallow network). The additional layers enable deep learning to learn from massive data.

Another difference is that, in the more conventional machine learning, the features are manually extracted from the input, before being fed into the network for the learning process.

In deep learning, the features of the input are automatically learnt within the multiple layers.

Deep learning can be used in either a supervised or unsupervised manner. Recent works have emerged that deployed deep learning for anomaly detection in sensor systems. Various types of deep learning networks have been proposed such as the Convolutional Neural Network (CNN), autoencoders, Restricted Boltzmann machine, and the Recurrent Neural Network (RNN).

Convolutional neural networks (CNNs) work by exploiting three main concepts which are parameter sharing, sparse connectivity, and equivariant representation [194]. CNNs have been proven to be effective to extract features from samples and have been extensively applied to image, speech and text processing. The convolution layer is usually followed by the pooling and fully-connected layers for classification or regression tasks. CNNs can be used to autonomously learn useful features for an anomaly detection task [110]. Chen et al. [45] propose the use of CNN for real-time anomaly detection in multi-sensor signals (for the flash welding process). CNNs are used in order to learn the recurrence dynamics from the recurrence plots derived from the collected data. The proposed method was evaluated in both simulation studies and a real-world case study.

Autoencoders are a special type of deep learning network that learns the latent space representations of data and tries to reconstruct the input data from the representations [22]. The autoencoder aims to learn discriminative feature representations through a process of encoding and decoding. To enable the network to learn useful features, a bottleneck is introduced at the latent space to compress the data. In [48] a spatio-temporal autoencoder is used to capture anomalies in camera networks. Luo et al. [143] proposed a distributed anomaly detection technique based on autoencoders to detect spikes and bursts recorded in temperature and humidity sensors.

Even though autoencoders are an effective learning and detection technique, their performance can degenerate in the presence of noisy training data. Autoencoders can be used for anomaly detection when connected to a classification layer, as seen in the work by [111]. Authors applied a self-taught learning algorithm to solve anomaly detection for unforeseen and unpredictable scenarios in two stages. In the first stage, the features of an unlabelled data set are learnt by sparse autoencoders. The next step was feeding the features to a classifier like NB-Tree, Random Tree, or J48 trained on a labelled dataset. The labelled and unlabelled data must have relevance among them, even though they may come from different distributions.

Sequential networks, such as the Recurrent Neural Networks (RNNs), work by using temporal correlations between neurons [140]. Recently, the Long-Short Term Memory (LSTM) was added to the RNN to serve as a memory unit during gradient descent [194]. This addresses

RNN's limited capacity to capture context with increase in the time steps. RNN and LSTM have been demonstrated to perform well in detecting anomalies in multivariate time series sensor data [39]. Goh et al. [89] designed a time series predictor using RNN and used a Cumulative Sum method to identify the anomalies in a cyberphysical system. Their method was able to identify which sensor was attacked.

Chauhan et al. [43] proposed using LSTM to capture anomalies in Electrocardiography (ECG) signals. The LSTM was first used to design a predictive model of ECG signals. The probability distribution of the predicted errors was used to indicate whether a signal was normal or abnormal. One advantage of LSTM over other techniques is that data can be fed into the network without the need for pre-processing [43].

Graph neural networks (GNNs) are fit for the situation where the data is represented in the form of graphs. One of their main advantages is the ability to model complex patterns, such as high-dimensional sensor data and the relationships between different sensor nodes. Authors in [227] categorize GNNs into autoencoders, recurrent, convolutional, and spatial temporal graph neural networks, while also providing a comprehensive survey. Main challenges in applying GNNs to anomaly detection for sensor systems include modelling the problem considering graphs and obtaining the graph input data.

In their work [50], Deng and Hooi carry out anomaly detection in multivariate time series data for the case of water treatment plants. Their proposed method aims at learning the relationships between sensors and mapping them as a graph, as well as identifying and explaining deviations from the learned patterns. To this end, the authors employ four main components, namely sensor embedding, graph structure learning, graph attention based forecasting, and graph deviation scoring.

Authors in [226] discuss the use of GNNs for anomaly detection in the Industrial Internet of Things (IIoT) setting, while considering the following scenarios: smart transportation, smart energy, and smart factories. For each of the scenarios, they mention public datasets and GNN-based solutions for the task of anomaly detection. Furthermore, they identify current challenges and open issues such as reliability, explainability, high computational needs (for training complex GNN models), and modelling the data as a graph (deciding which part is modelled as a node, and which as an edge).

Generative adversarial networks (GANs), as introduced in [92], are among the most recent deep learning networks. A GAN works by training a generative network to generate fake samples, and then tries to fool the discriminator network until it can no longer differentiate between fake and real samples.

GANs have been applied to identify anomalies in high dimensional and complex data

from sensor systems in [177], [137]. Patel et al. [177] introduced GANs for continuous real-time safety in learning-based control systems. They designed a controller focused anomaly detection in the form of an energy based GAN (EBGAN). The EBGAN network distinguishes between proper and anomalous actuator commands. However, when there are few anomalies, the more traditional machine learning techniques, like K-Nearest Neighbours, tend to perform better [39].

Restricted Boltzmann machines (RBMs) are bipartite, fully-connected networks, having visible and hidden layers organized in an undirected graph [71]. Authors in [70] applied RBM for network anomaly detection.

When RBMs are stacked upon one another in multiple layers, the outcome is the Deep Belief Network (DBN). The capabilities of DBN have been explored in intrusion detection systems to detect attacks in the network [11].

DBNs can scale to large datasets and can improve interpretability, as demonstrated by Wulsin et al. in [228]. Authors applied DBN for electroencephalography (EEG) anomaly detection, using it to separate rare and highly variable events, such as the malfunction of the brain.

DBNs can be used in unsupervised learning as a pre-training method to train the parameters of the deep neural network; after that, a supervised approach is used for classification. This was the approach employed in [118] to detect anomalies and enhance the security of in-vehicular networks.

Hybrid models. There have been efforts to combine different models to produce better results in anomaly detection. When a CNN serving as an encoder is combined with an LSTM network functioning as decoder, the result is a model effective for reconstructing image frames and detecting anomalies in data. Malhotra et al. [147] combine LSTM and autoencoder for multi sensor anomaly detection. This works by reconstructing the normal behaviour of time series, and using the reconstruction errors to detect anomalies. The system captures anomalies that are caused by external factors or variables, which are not captured by the numerous sensors monitoring a system.

Zhou and Zhang [244] combined sparse autoencoders and recurrent neural networks. The autoencoder was used for feature extraction, while the RNN was trained with a sequence of temporal features to predict the subsequent ones.

Zhou et al. [243] designed a de-noising autoencoder using an anomaly regularizing penalty based on L_1 or $L_{2,1}$ norms to remove outliers from image datasets.

Munawar et al. [167] combined CNN and RBM to extract features which when combined with a prediction system allow for learning to detect irregularities in video of industrial robot

surveillance systems.

To sum up, deep learning techniques exhibit a high degree of adaptability but have a not negligible time complexity. This latter is due to the presence of many hidden layers (characterizing the deep approach) which require high training times. For this reason, within a sensor network, deep learning cannot be performed directly on board of a sensor but requires nodes with dedicated computational resources typically located at the edge, the fog, or in the cloud (see Section 3.5 for details about these architectures).

3.4.6 Online vs offline detection/algorithms

Techniques such as deep learning pose the problem of *where* to process the huge amount of data. Another big issue is connected to the *when* such data is processed.

There exists an increasing demand for accurate anomaly detection in streaming data and real-time systems. This is encouraged by the growth of the IoT and complex sensor based systems, which increase the availability of streaming data. The development of both hardware, software, and architectural resources makes the handling of streaming data and of strict requirements of real-time applications possible. A consequence of this situation is the shift from the more traditional offline processing and analysis of data to an online approach. Offline processing is concerned with analysing the complete dataset, meaning that all the required data was collected, and is fully available. This approach usually allows for high–complexity techniques to be employed, as there are no time, or computational constraints (assuming the computational part happens in powerful environments such as the cloud).

Contrarily, in online processing, there are limited computational resources, and time constraints for obtaining a result. Furthermore, the incoming data needs to be analysed as it arrives, and any further processing happens in an online manner. Most techniques can be adapted to the online scenario by using short-term memory (when the prediction is dependent only on a small number of previous measurements), windowing, regular updating of the model, etc. [4].

Multiple examples of online sequential anomaly detection methods can be encountered in sensor systems and fields such as systems diagnosis and intrusion detection. [234] proposes an online anomaly detection algorithm for sensor data, namely segmented sequence analysis that leverages temporal and spatial correlations to build a piece wise linear model of the data. [231] introduces an online non-parametric Bayesian method, OLAD, for detecting anomalies in streaming time series. [5] uses Hierarchical Temporal Memory to develop an anomaly detection technique suited for unsupervised online anomaly detection.

The employment of online or offline algorithms must be done in accordance with the requirements of the system. The online approach is preferable when dealing with processing on the go on streaming data, real-time or near real-time requirements. Offline algorithms allow for carrying out more complex tasks on powerful resources when more data is available and when an immediate response is not necessary. The trade-offs between needed computational power, processing time, performance, response time and not only need to be taken into account when designing and employing these algorithms.

The techniques discussed till now highlighted pros and cons of various approaches from a methodological point of view. In fact, the effectiveness of a specific algorithm also depends on the architectural context of the application. At this aim, in the next section, we discuss the implications that three modern architectural models (Cloud, Fog, Edge) have on sensor systems.

3.5 Architectural perspective

Sensor networks (and their ability to perform anomaly detection) can highly depend on the architectural model adopted to deploy them. We can distinguish the following models, as detailed next: the Cloud model; the Fog model; the Edge model; and hybrid combinations of other ones.

3.5.1 Anomaly detection in the Cloud

In the Cloud model [64], the information gathered by the sensors is processed in a virtually centralized environment, characterized by considerable computational resources. This would, for instance, be the case of a complex camera-based monitoring system, where the anomaly detection task results from a complex image-elaboration procedure [217].

Cloud-centred anomaly detection makes use of the virtually unlimited computational capabilities available in the cloud, for the analysis of the data collected from sensor nodes. In this case, a major challenge derives from the actual "quality" of the data to be scrutinized. Sari et al. have looked at data security and storage issues, considering the analysis of sensor data that includes vast amounts of misconfigured sensor network traffic [194].

Due to the inherent complexities and large-scale nature of virtual clouds, the infrastructure itself is prone to software and hardware failures. Song et al. have looked at how to detect hardware and communication failures [77]. They aimed at understanding complex, system–wide phenomena, to improve system and resource availability.

There has been work on intrusion detection of networks in the cloud environment, aimed at protecting sensor data and cloud resources from malicious activities. Authors in [176] deployed an anomaly detector in a cloud environment for network intrusion detection. The detector was trained with a hybrid algorithm, based on both Fuzzy C-Means clustering and Artificial Neural Networks (ANNs). The Fuzzy C-Means was used to divide the large dataset into clusters to improve the learning capability of the neural network.

Despite being a reference method for anomaly detection, the cloud-centric approach can be negatively affected by misconfigured traffic, due to the high volumes of incoming traffic [194]. Cloud-centric anomaly detection may not be efficient for real-time applications due to the issues of latency, bandwidth, and communication costs. As the number of sensors generating data increases, the issue of information bottleneck becomes more significant, since all raw data is required to reach the cloud, before any processing may be applied. Delay becomes even more undesirable when a feedback control loop is involved between the cloud and the physical devices, which is the case of wireless sensors and actuator networks [7].

The cloud architectural model is well coupled with resource consuming techniques such as the deep-based ones. Obviously, such augmented "power" implies more costs, thus, a reasonable trade-off should be considered. A good strategy is to leave into the cloud only the critical functions and go towards a "smoother" paradigm represented by fog computing.

3.5.2 Anomaly detection in the Fog

In the Fog model [146],[241] the information is processed on intermediate, fog nodes lying between the cloud resources and the sensor system itself. This would, for instance, be the case of a smart car environment, where an anomaly event (e.g. a generic self-driving malfunctioning) must be elaborated quickly, namely, as closely as possible to the source [220].

The need for introducing a model lying in the middle between a fully centralized and a fully decentralized approach was conceived alongside the Internet of Things concept. In practice, fog computing can be considered as an extension of cloud computing towards the sensors, with the aim of accelerating the information processing through the intermediate fog nodes.

Accordingly, some techniques have been devised to exploit the presence of such intermediate nodes, so as to also improve anomaly detection. This is the case of [207], where the authors aim at energy-efficiency by introducing the "virtual control nodes" to realize a cross-layer, clustering method directly on the sensing layer. A data reduction scheme is advanced in [51], where fog nodes are able to build a prediction model fitting the sensor data characteristics,

resulting in a data stream reduction at the source node. Again, an energy optimization problem accounting for the presence of fog nodes able to directly manipulate sensor measurements has been solved in [72].

Focused on evolved architectural schemes which exploit the power of the fog paradigm are the works proposed in [18] and [222], where an efficient car parking architecture and a virtualization scheme for physical sensors are advanced, respectively.

From an architectural viewpoint, fog computing offers good chances to efficiently handle supervised techniques. For instance, an intensive training stage could be performed in the cloud, whereas the classification stage can occur on-board of sensors. This notwithstanding, in some situations (e.g. sensors located in areas with no data connection) the sensors might not have the access to the training set elaborated within the cloud. In such cases, the sensors are compelled to perform each operation on their own, in line with the so-called edge computing model.

3.5.3 Anomaly detection at the Edge

In the Edge model [236], the information can be directly processed on board of sensors, with options for distributed or collaborative decentralized computations. This would, for instance, be the case of a WSN deployed to gather environmental parameters (e.g. temperature, CO_2, ground PH, etc.) [26]. Often, such sensors are located quite far from a stable data connection; thus, the majority of information processing is executed on the sensor itself.

Recently, several attempts have been made to develop machine learning algorithms that bring anomaly detection from the data centres closer to the sensor nodes. Software platforms like TensorFlow, Caffe, Tencent have toolboxes that enable lightweight, high-performance neural network anomaly detection processes that are suitable for edge nodes. In [197], an edge node equipped with a deep autoencoder model to detect anomalies is advanced. Authors in [9] propose a computation offloading model for the mobile edge with a focus on IoT applications. The model is based on deep reinforcement Q-learning, and it shows improvements in terms of latency, energy consumption, and execution time.

A range of lightweight machine learning methods (referred to as shallow-learning) has been adapted to run directly on sensor devices by Bosman et al. [28]. They have demonstrated the viability of learning at the edge on devices having as little as 20 kbytes of memory in non-floating point hardware. A range of anomalies may be detected directly in the sensor node, with improvements based on ensemble methods.

Bosman et al. have also explored the intrinsic value of collaborative learning, involving data

streams from neighbouring sensors [29]. Interestingly, they have shown that, by aggregating data from just a few sensor nodes, it is possible to substantially improve the accuracy of anomaly detection, still with no intervention from high-performance computing nodes.

Ultimately, edge solutions target the scalability issues arising from the cloud and (to some extent) fog solutions, which incur network bottlenecks and feedback response latency. Nevertheless, the purely edge-based solutions are limited in the type of anomalies that may be detected. More realistically, these are used as data pre-processing steps, and in combination with deeper learning methods that run in the fog and in the cloud, leading to the hybrid methods discussed next.

3.5.4 Hybrid anomaly detection models

It is also possible to consider situations where different architectural models are used in combination, resulting in hybrid models. This is the case of a sensor network where part of the information processing (referred to as soft, or lightweight) is performed on board of the sensor (referred to as in-node anomaly detection), and the remaining critical part (hard processing) is performed either in the fog, or in the cloud, or in a mixed solution.

An exemplary case is offered by Cauteruccio et al. [38], who combined the strengths of cloud and edge nodes to detect anomalies in a heterogeneous sensor network. At the edge nodes, an unsupervised neural network was deployed to detect short-term anomalies or alerts. These are used to determine which portions of the data streams should be transmitted to the cloud (along with the pre-processed alerts) for a computationally-intensive learning task. The authors show how this combination of short-term learning and long-term learning, based on a method called “multi-parametrized edit distance”, may be used to achieve anomaly detection on different timescales. Furthermore, this hybrid (hierarchical) approach is shown to outperform other stand-alone strategies.

Another exemplary case has been proposed by Luo et al. [143]. The system runs the more computationally-intensive tasks in the cloud and is able to offload specialized detection tasks to individual sensors, taking into account the capabilities of individual nodes. The whole framework is organized in a distributed computing fashion, which does not require synchronous coordination (or communication) among sensors and between sensors and cloud. This offers some sort of asynchronous, hierarchical learning framework, which has rarely been explored in the literature.

Still in the direction of distributed ML architectures, authors in [240] have pinpointed important system-level issues that are not commonly addressed in centralized learning processes.

These include consistency, fault tolerance, communication, storage, and resource management. Noteworthy frameworks for the execution of distributed ML processes, across different servers and sensor systems, are MLbase [126] and Gaia [102].

Table 3.1: Promising research challenges.

Challenge	Main issue	Research topics
Miniaturization	Hardware Limits	· ML techniques for constrained devices [28],[29] · Distributed ML techniques [126],[102] · Efficient fusion strategies [25]
Acceleration	Circuitry Limits	· FPGA-based optimization [162],[224] · Algorithm complexity reduction [224],[163],[141]
Energy Efficiency	Power Limits	· Efficient routing protocols [225], [235],[100],[78] · Smart topologies [235],[100] · Lightweight operating systems [214]
Security	Computational Limits	· Distributed security mechanisms [79],[190] · Novel keys distribution schemes [139] · False data injection [42],[1]
Sensors Softwarization	Special-Purpose Architecture	· NFV/SDN for sensors [239], [245], [109], [84] · Integration with the 5G paradigm [88], [195]
Architectural Models	Intelligence Distribution	· Hierarchical Learning [174] · Hybrid models [130],[15]
Data Heterogeneity	Aggregation	· Ad-hoc middle-wares to homogenize data [166], [36] · New fusion strategies accounting for data variability [166],[97]

3.6 Open issues and challenges

Anomaly detection in sensor systems brings about a broad range of possibilities, opening up to interesting "research", as well as "practical" realization challenges. Researchers will continue to be particularly intrigued by the promise of machine learning and hybrid software architectures, and the realization of hierarchical learning methods in a mixed Cloud-Fog-Edge setting. A pivotal issue is that, at its full extent, anomaly detection requires computational power, in neat contrast with the substantial hardware and software constraints of sensors. Next, we highlight promising trends and research challenges, as depicted in Table 3.1.

3.6.1 Miniaturization

Classic anomaly detection techniques are not designed to run in sensor systems, mainly due to their limited hardware capability, in terms of CPU, memory, and connectivity. Some sensor nodes are even further constrained by their inability to compute floating-point operations, which Bosman et al. have demonstrated to cause machine learning algorithms to become inaccurate and unstable [28]. They have also provided practical guidelines as to how to miniaturize and adapt a number of shallow-learning algorithms to stabilize.

Limited accuracy arising from memory limitations has been addressed, to some extent, through ensemble learning [28] and collaborative learning (involving data streams from neighbouring sensors) [29]. Yet, the miniaturization of machine learning algorithms to fit constrained devices remains a largely unsolved problem. This issue poses particularly hard hurdles when combined with the energy limitations of edge devices, which are typically battery operated or even rely on fairly limited energy-harvesting techniques [210].

Another promising research direction is "distributed machine learning", where ML algorithms are re-designed to run in a multi-node environment. In this case, important issues include parallel data management, fault tolerance, and communication/synchronization among nodes. Distributed machine learning systems like MLbase [126] and Gaia [102] have been developed to run ML algorithms across different servers and over large-scale sensor systems. Moreover, authors in [25] proposed a system design for federated learning, which is a kind of distributed machine learning approach, to enable model training using decentralized data from devices such as mobile phones.

Both the distributed and the decentralized ML approaches present potential alternatives to centralized ML, particularly in the context of hybrid computing models. At the same time, new data-fusion strategies are required to consider the heterogeneity of data sources and handle the "missing data" problem [151], which is particularly problematic in sensor systems, due to unreliable connectivity and sensor faults.

All in all, miniaturization is perhaps one of the hardest challenges among those highlighted in Table 3.1 and discussed in this chapter. This is due to the variety of related open research topics discussed above (miniaturization of current ML techniques, development of new ML techniques, redesign of techniques and approaches for a distributed environment, efficient information fusion strategies while considering parallel data management, fault tolerance, node communication), as well as the lack of one size fits all approach. The miniaturization process needs to be adjusted to the hardware limits and energy constraints of the system, which in turn impacts and possibly limits the available software options (algorithms, models,

paradigms).

3.6.2 Acceleration

Besides the algorithmic and software development in machine learning, there are also important developments in the hardware that may be used. This is mostly relevant to the server side of computation (in the cloud and some fog nodes) to perform the training of anomaly detection models on high-performance computing platforms.

In the meantime, there is also progress in specialized acceleration hardware for edge computing. Several companies have invested, and recently released, chips and processors that can accelerate the execution of machine learning algorithms, often based on FPGA circuitry, to optimize some critical computations. For example, the Xilinx FPGA technology can be applied to the deep learning inference phase in order to achieve low latency, high compute performance, as well as low wattage [162]. This opens the avenue for powerful techniques to be run more closely to the sensor systems or even as part of them.

Again, Wess et al. [224] have looked at how ANNs may be effectively implemented on FPGA for anomaly detection, considering EGC data. Their focus is on optimizing ANNs for FPGAs, based on piece-wise linear approximated transfer functions and fixed point arithmetic. Furthermore, they carry out a resource trade-off analysis between the data point precision, size, latency, and accuracy of the neural network, aimed at choosing a suitable ANN configuration (i.e. the number of input and hidden-layer neurons).

However, one must be aware that issues such as battery drainage and memory limitation may still exist in edge computing. Besides acceleration based on hardware developments, considerable work is directed at addressing the acceleration of the algorithms through advanced computing methods. Mocanu et al. have proposed an ingenious method to accelerate the ML training process, using network science strategies [163]. By training sparse rather than fully-connected ANNs, they achieved a quadratic reduction in the number of parameters, at no accuracy loss. In fact, Liu et al. have demonstrated how deep learning over one million artificial neurons may be computed on commodity hardware, based on the sparse neural training strategy [141].

These developments suggest a promising research direction in ML acceleration based on bio-inspired algorithms. The sparse neural networks training approach discussed above is drawing inspiration from the natural sparsity of biological neural networks, as opposed to the traditional fully connected layers predominantly used in the research field. Enormous advantages arise from the possibility of making the existing algorithms more efficient, without

changing the hardware platforms.

3.6.3 Energy efficiency

Some sensor systems such as WNSs have to deal with energy limitations, as these are often battery operated or rely on energy–harvesting. An important research strand is the investigation of the trade-offs between algorithmic complexity and energy consumption, particularly in edge devices. These energy constraints rule out the most sophisticated ML methods in edge computing. Thus, most investigations are directed towards shallow learning methods and simple reinforcement methods for in-node learning. An example is the work on shallow learning in sensors by Bosman et al. [28], [29], as introduced in Section 3.5.3. Also, relevant is the work on Q-Learning (reinforcement learning) in sensors by Chincoli et al. [46], as introduced in Section 3.4.4.

A common understanding so far is that the computational and energy overheads introduced by shallow learning are roughly counter-balanced by comparable savings arising from data compression and communication efficiency. Nevertheless, this is a very much debated issue, certainly worthy of more methodical investigation.

Along this line of investigation, several researchers have focused on energy and communication efficiency that may be pursued through intelligent networks methods. The general issue is that communications incur significant energy overheads, often overshadowing the other neat energy-consumption contributors. What is more, in wireless networks, and particularly WSNs, a significant portion of energy consumption derives from the sensing element, even while in standby mode [125]. Thus, WSN communication efficiency remains an open research challenge.

A vast body of work has been looking at smart communication protocols, with new approaches to routing and topology control. A review of shallow-learning algorithms for energy efficiency based on intelligent power control methods has been written by Chincoli et al. [47]. They have also demonstrated an autonomous self-learning method to minimize transmission power in WSNs, using in-node reinforcement learning [46].

Assessing the impact that individual nodes have on the overall WSN consumption is a complex, non-deterministic problem, as discussed in [125]. To this end, cooperative learning offers a promising direction to address intelligent-based network efficiency. An example that is actually not employing ML, but is using effective cooperative processes, is offered by the Routing Protocol for Low-Power and Lossy Networks (or RPL) [225]. RPL is implemented in TinyOS [214], i.e. a lightweight operating system designed to run in WSN-based environments.

The core idea is for the protocol to create new paths and/or topologies, aiming at preserving the residual power of the various sensors.

Remarkably, the authors in [235] propose an approach aimed at forbidding low-energy nodes to be forwarder nodes when the residual energy is lower than a specific threshold. Again, an interesting extension of RPL to mobility nodes (useful, for instance, within the automotive environment) is advanced in [100] where, by exploiting additional fields of RPL control packets, it is possible to advise a mobile sensor node for maintaining the energy by reducing network overheads.

Authors in [78] propose a multi-path routing protocol for WSNs which takes into account environmental data in order to assure high routing reliability in harsh environmental conditions. Furthermore, the protocol favours the choice of routes that allow for high-energy efficiency and low delivery latency.

Despite the vast body of work on energy efficiency, the use of intelligent algorithms has been limited to a few cases and is yet to find space in the standards. Cognitive networks, including in-node learning and internode collaborative learning, still have enormous research potential, particularly in the context of self-managed energy efficiency in wireless mesh networks, sensor networks, and mobile ad-hoc networks.

3.6.4 Security

Due to their limited resources, sensors cannot afford the luxury of hosting sophisticated protection mechanisms onboard; thus, they represent a very attractive prey for cyber-criminals. From a malicious perspective, compromising a sensor might have a double purpose. On one hand, an attacker can be interested in stealing data or manipulating signals managed by a sensor through the so-called denial of service attacks [180, 153]. It is the case of medical devices (e.g. insulin pumps, pacemakers, etc.) that, if compromised, can provoke irreversible damage to the patients [237]. On the other hand, a sensor can be deceitfully exploited as a propagating vector of a cyber-threat, whose avalanche effect globally amplifies the final malign result [154, 155]. This is the case of the recent botnet Mirai [121], based on an infectious propagation mechanism aimed at poisoning IP-cameras systems. Such security breaches unavoidably impact also the sensors' anomaly detection functionalities. Along these lines, many security frameworks and procedures are being proposed to tackle the aforementioned issues.

For instance, distributed security mechanisms applying the blockchain concepts to the sensors systems world have been proposed in [79] and [190]. Precisely, in the former work,

a blockchain-based learning framework exploits the "collective intelligence" of a set of connected autonomous vehicles (CAVs) aimed at improving the collective learning operation, where two approaches emerge: a centralized one, where each CAV uploads the sensing data into the cloud so that the model training is performed in a central server; a distributed one, where, according to a federated learning strategy, training data are distributed among learners, and the model training is performed locally. The latter work, instead, proposes a Safety-as-a-Service framework, where distributed security and confidentiality mechanisms relying on blockchain principles are applied to the realm of Industrial IoT.

An efficient random key distribution scheme (SRKD) to counter the replication attacks is instead proposed in [139]. Being specifically conceived for wireless sensor systems, such a scheme exhibits storage and communication overhead lower than its classic counterparts.

Another prominent problem in constrained environments concerns the presence of malicious sensors which may inject anomalous data to corrupt the estimates at the fusion center (the so-called false data-injection attack). Accordingly, the authors in [42] propose a detector to reveal time varying injection attacks within cyber-physical systems. The same family of attacks has been faced in [1], where the authors introduce a method to reveal false data injection attacks against a hydraulic sensor-based system by exploiting the autoencoders.

On another note, when looking at security within sensor systems and the IoT paradigm, one needs to also consider the in-network anomaly detection. Within the systems themselves, potential security risk needs to be assessed and identified based on network traffic data and behaviour. For example, the work in [85] investigates the increase of security in IoT-enabled applications. The authors introduce a real-time multi-stage anomaly detection scheme that tries to cover the gaps with the traditional Density-Based Spatial Clustering of Applications with Noise (DBSCAN) algorithm. In [86] an ensemble based anomaly detection technique is proposed to identify the malicious behaviour of nodes in the cloud environment. RBM and Unscented Kalman Filter are used for feature selection and optimization. The Artificial Bee Colony-based Fuzzy C-means is afterwards used to partition the datasets into relevant clusters. These clusters are then used to build a normal/abnormal profile of behaviour and to classify the occurring events as normal or anomalous.

At present, many efforts are being devoted to implementing security mechanisms directly on board of a sensor, so that each device could react to cyber threats without the need of being supervised by a central server. Unluckily, due to the currently limited technology, such an approach seems still far from being used in real scenarios, but it would be supposed to replace the classic security-centric approach in the next few years, so as to make the sensors ever more independent and autonomous from a centralized architecture.

3.6.5 Sensors softwarization

The emerging NFV (Network Function Virtualization) and SDN (Software Defined Networking) paradigms play a leading role within the process of network "softwarization", especially in modern 5G infrastructures. NFV allows decoupling the underlying physical infrastructure (e.g. CPU, power supplies, etc.) of a network element from its software logic, whereas SDN allows redefining the control plane of a network by enriching the routing and forwarding strategies. The combination of the two paradigms goes in the direction of improving flexibility and maintainability, and allows effective cost sharing. Although the softwarization process mainly involves classic network nodes (e.g. routers, firewalls, etc.), a recent interest in adopting it within the sensor-based environments has emerged.

For instance, the authors in [239] introduce a virtualized framework simulating a real IoT deployment, where virtual IoT honey nets are used to distract possible intruders from the real targets. The key idea is to transform the physical model into a common interoperable data model and, in turn, translate it into a software-based setting composed of Virtual Network Functions (VNFs). The overall process is governed by NFV/SDN security policies, designed to dynamically handle operations such as filtering, dropping, and diverting.

A combination of NFV and SDN is investigated in [245] to automate network processes within an Internet-of-Vehicles (IoV) ecosystem. More specifically, NFV is exploited to offload IoV tasks towards the edge and cloud nodes, whereas SDN is utilized to smartly configure routing and forwarding paths across the IoV network.

The work in [109] proposes an SDN-based framework to be exploited in IoT environments. In particular, two intelligent SDN controllers are advanced: the first one has the capability of estimating the packet flows within a specific sensing area through a partial recurrent spike neural network, and the second one exploits an ANN to select the cluster head and its members in the considered sensing area.

Garg et al. [84] bring forward an SDN–based real-time anomaly detection framework for social multimedia traffic. The framework consists of two components, namely the anomaly detection module which leverages the advantages of improved RBM and SVM, and the end-to-end data delivery module of social media traffic. The latter comprises an SDN-assisted multi-objective flow routing scheme, which allows for the trade-off between latency, bandwidth, and energy consumption utilization.

As key enablers of 5G network infrastructures, NFV/SDN concepts have been profitably exploited in [88], where a softwarized 5G network has been designed to support the implementation of the so-called tactile internet and to provide mission-critical IoT services.

Again, the work in [195] focuses on a 5G-IoT architecture based on the multi-access edge computing (MEC) paradigm. In particular, the authors discuss the VNF life-cycle management through ad-hoc scheduling and allocation strategies allowing to dynamically instantiate, scale, migrate, and destroy the VNFs.

Actually, sensors softwarization still has a long way to go, since virtualization technologies have been explicitly conceived for general-purpose architectures, whereas sensors often rely on very customized hardware (e.g. FPGA), which requires an additional effort to be managed.

3.6.6 Architectural models

When designing sensors systems, pinpointing a set of anomaly detection methods that can best fit the underlying architectural deployment is a non-trivial task. An important challenge is to identify techniques that are flexible enough as to be able to adjust to architectural changes – for instance, from cloud to fog or *vice versa*. It is equally crucial to dynamically manage the resources in a way that best employs the underlying infrastructure. An example would be to automatically transfer the most demanding operations to the core network when operating over the cloud. Task offloading has been investigated extensively in the literature [66]. Yet, its application to IoT and cloud-assisted sensing still poses considerable open issues, given the variety and heterogeneity of frameworks and the significant constraints of edge devices.

Accordingly, techniques such as hierarchical and distributed learning, which rely on a *divide et impera* approach, could be attractive when used in tandem with the novel paradigms of cloud, fog, and edge computing. A prominent area of investigation is aimed at better understanding how, and to what extent, it is possible to utilize the computing cycles available at the edge to reduce network latency and bottlenecks, while at the same time respecting the energy limitations of edge devices.

An example where machine learning may be used in a mixed architectural fashion is given by authors in [174]. They illustrate the benefits of a hierarchical system versus a centralized one, corroborated by a use case involving cognitive transmission power control.

Distributed learning is explored in [130], where the authors experiment with a fog architecture in an Industrial IoT context. The sensor motes learn a model from the incoming data, and periodically transmit updated parameters of the model to the fog layer. This information is then sent to the cloud, where the data can be further analysed and visualized. The results show high accuracy for simulating the original data, while minimizing the number of packets sent over the wireless link and the energy consumption.

Likewise, the authors in [15] introduce a design framework for smart audio sensors which

can record and pre-process the raw audio streams. The extracted features are transmitted to the edge layer, where anomaly detection algorithms executed as microservices are able to detect anomalies in real-time by analysing the received features.

These are just a few examples, showing the potential and the difficulties involved in the use of mixed architectural models, under constrained hardware platforms.

3.6.7 Data heterogeneity

Although not directly connected to a specific anomaly detection technique or a particular architectural model, dealing with the data heterogeneity can be a very challenging task across the sensor systems world. This issue typically emerges when one is interested in extracting a particular "meaning" from a bulk of data having different shapes (e.g. temperature measurements and movement alerts), or coming from different sensor ecosystems (e.g. self-driving cars and smart cameras).

This is typically the realm of data fusion [158], which has recently turned its attention to smart methods in IoT settings [53], opening an avenue for new research issues. An important problem is the identification of anomalies that are not visible in individual (homogeneous) sensor data streams, and become evident only when heterogeneous streams (typically from neighbouring sensors) are combined. Particularly difficult is the fusion of streams that operate on different time scales, different ranges, and have different patterns.

Valuable examples are provided by healthcare applications, where data are generated from multiple sensors and sources, ranging from ECG reports (time series data) to blood test reports (key value pairs), to x-rays (images). How to fuse these data into a single patient record to detect anomalies in the patient's health record can be challenging.

Accordingly, there is a growing interest in using hybrid machine learning models to tackle heterogeneous sources [166]. The local models can be employed to learn specific complexities of sensor data, and the models are aggregated to a global model in edge and cloud environments [36]. Similarly, several types of complex data are emerging with increasing heterogeneous sensors and applications. Anomalies will be detected in such complex data as time-series, sequential patterns, and biological sequences; graphs and networks; spatio-temporal data, including geospatial data, moving-object data, cyber-physical system data, and multimedia data [97].

Table 3.2: Key references about techniques and architectural models.

Conventional techniques				
Time series analysis	[164],[168],[198]	Statistical	Non-parametric	[41],[99],[149],[179]
Signal processing	[185],[32]		Parametric	[41],[99],[149],[179]
Information theory	[135],[14],[41]	Spectral		[104],[56],[41]
Data-driven techniques				
Supervised	[54],[148],[3],[213],[223],[113]	Deep	RBM	[71],[70],[11],[228],[118]
Semi-supervised	[111],[70],[48],[82]		CNN	[194],[110]
Unsupervised	[134],[186],[219]		GAN	[92],[177],[137],[39]
			Autoencoder	[22],[48],[143],[111]
Reinforcement	[208],[46],[199],[172],[103]		Sequential Nets.	[140],[194],[39],[89],[43]
			Hybrid models	[147],[244],[243],[167]
Architectural perspective				
Cloud	[64],[194],[77],[176],[217]			
Fog	[146],[241],[207],[51],[72],[18],[222]			
Edge	[236],[26],[197],[9],[28],[29]			
Hybrid	[38],[143],[240],[126],[102]			

3.7 Conclusion

In this review, we are focused on capturing the state of the art of the anomaly detection problem across the sensor systems world. We have explored this timely problem along two main directions, as depicted in Figure 3.1. First, we looked at two broad sets of techniques. The more conventional ones typically exhibit robust mathematical formalism but are not always able to fully capture the complexity of real-world systems in deterministic models. Then, our emphasis shifted toward data-driven techniques that, by relying on machine learning concepts, aim at overcoming the non-linearity of sensor systems. In this case, however, the system is treated as a black box. Thus, the more realistic view achieved on the data behaviour sacrifices the formalism robustness and creates the issues of data explainability and interpretability.

Going beyond the individual anomaly detection methods, we explored the orthogonal strand of the architectural models that may be adopted for detecting anomalies in sensor systems. The most commonly used ones include: 1) Cloud-assisted sensing, which follows the (virtually) centralized computing paradigm; 2) Fog in sensor systems, which follows the partially centralized paradigm; and 3) Edge sensing, the fully decentralized paradigm.

It turns out that reality is often a mix & match of different techniques (algorithms), models (deterministic vs predictive), architectures (Cloud, Fog, Edge), which makes our taxonomy very relevant to those interested in the complex area of anomaly detection in sensor systems. To facilitate the navigation of the most relevant works, we have clustered key papers in Table

3.2, to be read in combination with the taxonomy introduced in Figure 3.1.

Our study sparks an interesting set of open research questions, as discussed in Section 3.6, spanning from the miniaturization of the actual algorithms (to fit in constrained devices) to the acceleration of the processes (to scale up data analysis). We also discuss the issues of energy efficiency (particularly sensitive in devices), architectural models (to decentralize processes), and data heterogeneity (to improve accuracy by fusing data). Again, key references in relation to research challenges are summarized in Table 3.1.

Our review recognized the important role that machine learning is playing in anomaly detection in sensor systems, identifying a range of important challenges that go beyond the development of suitable algorithms and lay at the intersection between computing (learning models), communications (efficiency), and engineering (constraints).

Chapter 4

Embedded Real-Time Data Imputation for Environmental Intelligent Sensing

Recent developments in cloud computing and the Internet of Things have enabled smart environments in terms of both monitoring and actuation. Unfortunately, this is often resulting in unsustainable cloud-based solutions whereby, in the interest of simplicity, a wealth of raw (unprocessed) data is pushed from sensor nodes to the cloud. Herein, we advocate the use of machine learning at the sensor nodes to perform essential data-cleaning operations, to avoid the transmission of corrupted (often unusable) data to the cloud. Starting from a public pollution dataset, we investigate how two machine learning techniques (kNN and missForest) can be embedded on Raspberry Pi to perform data imputation in real-time, without impacting the data collection process. Our experimental results demonstrate the accuracy and computational efficiency of edge-learning methods in filling in missing data values in corrupted data series. We find that kNN and missForest correctly impute up to 40% *of randomly distributed missing values, with a density distribution of values that is indistinguishable from the benchmark. We also show a trade-off analysis for the case of bursty missing values, with recoverable blocks of up to 100 samples. Computation times are shorter than sampling periods, allowing for real-time data imputation at the edge.*

This chapter is based on two of our papers, namely the conference paper "*Critical comparison of data imputation techniques at IoT Edge*" [62], part of the 14th International Symposium on Intelligent Distributed Computing (IDC 2021), and the journal paper "*Embedded Real-Time Data Imputation for Environmental Intelligent Sensing*" [61], which is now published in the *Sensors* journal. Code and data related to this chapter are available on a GitHub repository [60].

4.1 Introduction

Smart environments find themselves at the intersection of the Internet of Things (IoT) and cloud computing and are capable of gathering information on the surroundings (monitoring), as well as manipulating it in order to accommodate certain conditions (actuation) [6]. Examples include, but are not limited to, smart cities, smart homes, smart grids, smart industry, smart health, or smart transportation. A challenge that arises is the management of the big IoT data generated by these types of systems [87]. It is getting noteworthy attention across application domains ranging from Industrial IoT (IIoT) [44], to green energy [229], healthcare [211], Industry 4.0 [173], and many other domains [59]. Using only cloud-assisted computing may soon prove to be unsustainable, given the costs associated with storing, transmitting, and processing the sheer amount of raw (unprocessed) data produced by the sensor nodes. Therefore, we advocate for the use of solutions involving edge computing, a paradigm proposed for solving IoT and localized computation needs [236],[196],[52]. In this way, part of the processing can be done at the edge, close to the data source, which in turn results in cost savings related to data transmission, latency, and bandwidth usage among other benefits.

Among the plethora of data management issues, the missing data imputation emerges as one of the most critical problems. The following taxonomy of missing data management techniques is suggested in [83]: $i)$ missing data deletion; $ii)$ estimation of the missing data on the basis of modelling the known distribution (e.g. Gaussian Mixture Models, Expectation-Maximization); $iii)$ imputation or estimation of missing data through machine learning techniques.

In this work, we mainly focus on the environmental edge data imputation using machine learning (ML) approaches. Actually, applying machine learning techniques to the missing data problem is not new in the literature. For instance, in [8], the authors exploit artificial neural network models to impute data (in particular, anomalies) through time series data. Decision trees (DT) and CART algorithms are applied in [209] and [138], respectively, to solve the missing data issues. Other examples include Support Vector Machines (SVM) and Self Organizing Maps (SOM) which are exploited in [81] and in [115], respectively. In all the aforementioned works, the ML-based techniques are basically stressed to evaluate their accuracy, but the possibility of applying them directly on board the sensors has not been investigated.

Conversely, we propose the use of machine learning techniques at the sensor nodes, close to the data source, for cleaning incoming data by imputing missing values. This approach is motivated by the goals of minimizing network and energy costs necessary for data transmission,

as well as increasing the reliability of the collected data.

As opposed to the classic approach of collecting the data from the sensors and then forwarding it to the cloud, we propose embedding pre-processing at the edge of the network. The growth of the IoT and the availability of computationally powerful devices, which can be deployed within the IoT environment, allow us to perform computationally intensive tasks at the edge, in order to increase data reliability and usability. In turn, this prevents the transmission of corrupted and unusable data further up the processing pipeline, including all the way up to the cloud. Additionally, further processing, depending on the application scenario, should be carried out at the edge after the data imputation process as to minimize the transmission of data across the network. Furthermore, additional checks can be performed at the edge to decide when and if the data should be transmitted to the next processing layer. One example for further processing at the edge, while making use of the edge imputed data, could be incorporating federated learning [156] on either the same edge device, or a neighbouring dedicated edge device. In this way, the new reliable data (as opposed to the raw, potentially contaminated and missing data) can serve as training data for the local training of different machine learning models.

As proof of concept for embedded real-time data imputation at the edge, we investigate and evaluate the performance of two representative machine learning techniques, kNN and missForest, on the board of an IoT device, namely the Raspberry Pi 4B (RPI 4B). We start with an artificially generated public pollution dataset, which serves as the benchmark. The dataset is corrupted to different degrees to account for two scenarios of missing data, namely random non-bursty missing data and random bursty missing data. kNN and missForest, two commonly used machine learning techniques, are benchmarked against two popular statistical based techniques, namely the mean imputation and multiple imputation by chained equation - MICE, for the task of filling in the missing data whilst taking into account the following metrics: root mean square error (RMSE), density distribution, execution time, and RAM and CPU utilization. Our experimental evaluation is carried out on the board of a constrained environment, namely the RPI 4B, and on a laptop, to verify the computational trade-offs under limited computing conditions.

We find that kNN and missForest outperform the statistical based techniques and can correctly impute up to 40% of randomly distributed missing values, if we are to consider an RMSE of up to 10 as acceptable, given the environmental dataset we use. Furthermore, they are able to recover blocks of up to 100 missing samples in the bursty case scenario before their performance drops to that of mean imputation and MICE. Additionally, we provide a trade-off analysis for the bursty case, considering the chosen algorithm, dataset impairment

rate, and burst size. We also discuss time and space complexity for the different scenarios in terms of execution time and RAM and CPU utilization for the RPI 4B. The resulting execution times are shorter than the sampling period of the considered environmental IoT scenario, thus allowing us to show that real-time data imputation can be achieved at the edge on board constrained devices. This encourages us to continue investigating how to take advantage of edge computing in order to optimize existing processing pipelines within the IoT, as well as to build on top of existing smart intelligent sensor systems.

In this work, we attempt to provide an answer to the question of "where" performing the data imputation techniques (e.g., closer to the cloud or to devices). Such a question does not admit a unique response, since it strongly depends on the particular application and/or context. For instance, performing data imputation within the cloud could be attractive since we have virtually infinite computational resources; thus, we could neglect issues related to the time/space complexity of specific techniques. On the other hand, the cloud being "far" from the devices, some local correlations among measurements could be lost, and the data imputation could be inaccurate. The aforementioned benefits and drawbacks are reversed if we decide to perform data imputation techniques directly on board the devices. In our analysis, we try to quantify such differences through an experimental comparison with data imputation techniques applied close to the cloud (e.g., on a fixed and powerful platform) and close to the edge (namely on board the Raspberry Pi).

The chapter is organized as follows: Section 4.2 places our work within the literature; Section 4.3 mainly focuses on the methodology and the chosen dataset, the scenarios designed for impairing the dataset, the techniques adopted for the edge data imputation, as well as the cases considered for the experimental work; Section 4.4 presents the results and findings for the considered scenarios, while Section 4.5 discusses important considerations as part of the undertaken experimental work; Section 4.6 concludes the present chapter and outlines directions for future work.

4.2 Related work

The missing data imputation problem in IoT has received significant attention in recent years. Precisely, two main tracks of research emerge: the first one is focused on a cooperative approach, where one tries to identify space/time correlations among data acquired by the IoT devices, so that measurements missing from a sensor can be replaced by measurements from other correlated sensors. The second one focuses on specific techniques/algorithms, which

typically borrow crucial concepts from statistics and/or machine learning.

In line with the former track, the work in [122] advances a double layered clustered scheme along with a consensus-based framework aimed at substituting missing values from the sensors measurements. In particular, the nodes located at the edge perform the data imputation. Such an imputation is evaluated by considering the correlation between the measurements affected by missing values coming from a given sensor (e.g. sensor A), and the "healthy" measurements collected by other sensors located in the proximity of the sensor A.

A similar approach relying on the spatio-temporal correlations has been adopted in [152], where a comparison among several data imputation techniques has been proposed. Yet, the authors in [76] propose a data imputation strategy that relies on the concept of "group opinion". For instance, metrics like the Mahalanobis distance and the cosine similarity are combined to evaluate the data replacement proposed by a group of peer devices. Finally, the group and the local opinions are aggregated through a weighting mechanism to propose the best replacement.

A similar concept based on the opinion of a group of devices is exploited in [75], where a continuous correlation detection methodology is applied in real-time to streams of data coming from IoT devices. The temporal-spatial correlation jointly with a kNN algorithm is exploited in [175], where the spatial correlation of sensor data is described through a linear regression model, and where information from multiple neighbouring nodes is used to estimate the missing data jointly, rather than independently.

In line with the second research track, in [65] incomplete sensed data across the IoT world is managed through the probabilistic matrix factorization (PMF) method along with the usage of a K-means algorithm to measure the similarity among neighbouring sensors.

In [90] the missing values imputation within sensor-based measurements is performed through the Bayesian Maximum Entropy (BME) technique. The performance of the BME technique seems to outperform the PMF in terms of accuracy, time efficiency, and robustness.

The authors in [142] face the missing data problem in IoT systems by focusing on the common mode failure problem, where a single event can lead to the loss of data from a large number of sensors, simultaneously. They advance a specific technique to deal with the large gaps in univariate time-series data, along with an iterative framework nicknamed *Itr-MS-STLecImp* which acts according to two steps: gap segmentation and gap reconstruction.

A Gaussian Mixture Model (GMM) to handle missing values in IoT systems has been advanced in [233]. In particular, the authors proposed the recovery of 21 missing temperature sensor values from a set of 220 observations.

The complex neural-based approach used in [215] relies on a combination between the General Regression Neural Networks (GRNNs) and the Successive Geometric Transformation

Models (SGTMs) to solve the problem of completing missing data from IoT devices.

Authors in [123] specifically tackle the problem of data reconstruction across wireless sensor networks, where data loss is commonly due to noise, unreliable links, or collisions. To address such problems, they propose an algorithm dubbed *ESTICS* which exploits advanced concepts of the compressive sensing theory to reconstruct the massive missing data.

Finally, the Singular Spectrum Analysis (SSA) to maximize the accuracy of data imputation in IoT-based surveillance environments is advanced in [178], where a non-parametric spectral estimation along with spatial-temporal correlations of time-series data from IoT devices are exploited together.

While all the works discussed earlier present interesting and innovative techniques or frameworks to manage the problem of missing data, some important differences arise with respect to our proposal. First, in many works (see, e.g. [76, 75, 90, 142, 233, 123]), despite the data coming from the IoT, the analysed imputation algorithms actually run on standard computer architectures (e.g. PCs, laptops, etc). Also, the importance of tackling the missing data problem as close as possible to the devices is evident (e.g. at the edge of the network [232]), thus our assessments are carried out straight on board of the devices. Second, many existing experiments (see, e.g. [152, 76, 75, 142]) are performed by looking at simplistic data missing models, while we here consider the problem of *bursty missing values*, which often arises when a sensor becomes unavailable for a certain (finite) period of time (a common situation for environmental sensors). Finally, compared to other works (see, e.g. [122, 152, 215, 123]) which only account for one performance analysis (for instance, the imputation method accuracy), in our approach, we complement this analysis with a time assessment, which can be critical within real-time environmental data.

4.3 Methodology

In this section, we explain the environmental dataset (Section 4.3.1), the data corruption scenarios (Section 4.3.2), the techniques chosen for performing the data imputation (Section 4.3.3), and the design of the experiments (Section 4.3.4).

4.3.1 Dataset choice and description

In order to analyse and carry out the experiments for edge data imputation, we chose an artificially generated IoT dataset, namely *Pollution Measurements (generated data)*, part of the *City Pulse Smart City Datasets* [10]. The dataset was created so as to complement the

traffic level data (real dataset of vehicle traffic observed between two points for a set duration of time) in the city of Aarhus, Denmark.

The synthesized pollution data consists of observations for the air quality index, which simulate a pollution sensor attached to each traffic sensor in the traffic dataset. The values are generated every 5 minutes using a pre-selected pattern[1]: each sensor measurement (e.g. ozone level) is initially assigned a value between 25 and 100; every 5 minutes, the values will be updated as follows: if the previous value was outside the 20 to 210 range, then a random integer between 1 and 10 is added or subtracted, as to keep the new value in the desired interval or as close as possible to it; for all other cases a random integer between -5 and 5 is added to the previous value.

For this work, from the aforementioned dataset, we chose the data corresponding to only one sensor. The dataset consists of $17,568$ samples. Each sample has associated a set of environmental measurements (e.g. ozone level, carbon monoxide, etc.) along with location data (longitude and latitude), and timestamp. For the simplicity of the analysis, we dropped from the dataset the location data, as well as the timestamp (the order of the samples assures the time continuity of the samples), and retained only the pollution measurements. A description of the dataset is given in Table 4.1, where std stands for standard deviation, whereas 25%, 50%, 75% represent the first quartile, the second quartile (median), and the third quartile, respectively. For simplicity, we impaired only the ozone values, and thus applied data imputation to the time series only. Figure 4.1 depicts the histogram of the original (unimpaired) ozone measurements.

The distribution of the chosen dataset makes the task of data imputation easier if compared to highly dynamic/irregular data scenarios. However, this is also due to the high correlation among subsequent data points. Every new data sample depends on the previous value and varies by a value of at most 5 or 10 as explained above. As a result, the data has a regular behaviour, without sudden changes. If the data distribution was the same, but there was little or no correlation between consecutive data points, the data imputation task would be more difficult. Furthermore, if the data distribution was gaussian, the data imputation process would be easier to handle in comparison to our scenario because mean imputation could recover missing data with acceptable accuracy.

As mentioned before, we chose to impair only the ozone values. Choosing to impair one of the other pollution measurements, instead of the ozone values, would have produced similar results. This is due to the fact that the mechanism to generate the data is similar, resulting in

[1] `http://iot.ee.surrey.ac.uk:8080/datasets/pollution/readme.txt`

data samples with an analogous behaviour (highly regular and correlated), which is visible in Table 4.1. This correlation among different pollution measurements is maintained even for a real dataset, as opposed to an artificially generated one, given the generally high correlation among pollution measurements. Impairing data from multiple pollution measurement columns at the same time would make the task of data imputation more difficult, and the results would vary as the problem at hand changes from univariate imputation to multivariate data imputation.

Table 4.1: Description of the dataset pollution measurements.

	Ozone	Particullate matter	Carbon monoxide	Sulfure dioxide	Nitrogen dioxide
count	**17568**	17568	17568	17568	17568
mean	**92.42**	106.12	100.54	131.66	159.18
std	**46.18**	52.01	49.66	50.51	43.43
min	**15**	15	15	15	18
25%	**54**	60	56	99	134
50%	**87**	107	99	131	173
75%	**127**	146	138	177	193
max	**215**	215	215	215	215

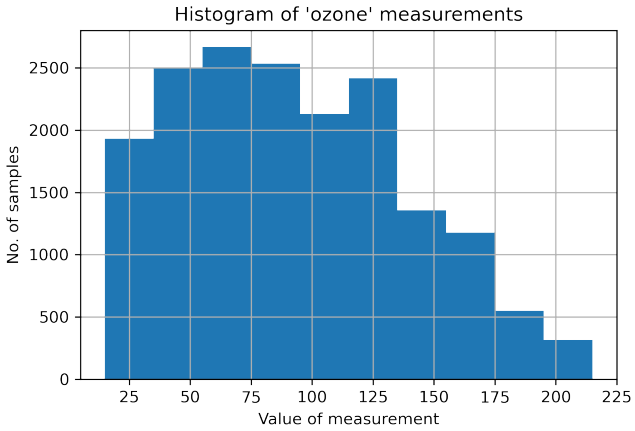


Figure 4.1: Histogram of the ozone measurements in the dataset.

4.3.2 Dataset impairment

In its original form, the dataset used in this work can be considered curated and with no missing values. In order to test and compare different methods for data imputation at the edge,

Table 4.2: The methods used for corrupting the dataset.

Type of introduced error (random)	Description
Non-bursty case	we randomly select the individual data points to be invalidated from the dataset, in order to reach the desired dataset impairment level ●●●●●●●●●●●●●●●●●●●● 20 data points, desired impairment rate of 20% (4 points to be invalidated) ●- original data point, ●- invalidated data point (N/A)
Bursty case	we randomly select bursts of a given size of data points to be invalidated from the dataset, in order to reach the desired dataset impairment level ●●●●●●●●●●●●●●●●●●●● 20 data points, desired impairment rate of 30% with burst size = 3 (6 points to be invalidated as part of 2 bursts of size = 3 data points) ●- original data point, ●- invalidated data point (N/A)

we impaired the dataset by introducing missing data in two different ways, which mimic two scenarios, as described in Table 4.2: the *non-bursty* scenario and the *bursty* scenario. On one hand, for the non-bursty case, individual data points are randomly chosen to be invalidated within the dataset. For the bursty case, bursts of a given size of data points are randomly chosen to be invalidated. The impairment level is expressed in terms of corrupted data points. In Table 4.2 we also provide a visual example of the way in which the data points are invalidated in the two cases. These types of errors are often encountered in an IoT setting.

4.3.3 Methods chosen for the edge data imputation

In this work, we want to analyse and compare well-known and readily available techniques in order to showcase the simplicity and ease of deployment as part of pushing the data processing to the edge. We chose two machine learning based techniques (kNN and missForest), evaluating them against two statistical based methods (mean and MICE). In this way, we want to highlight that machine learning based techniques can achieve a good performance level in an edge environment without special modifications. The criteria used for choosing these four techniques included them being well-known and commonly used for the task of data imputation, being readily available in terms of available implementations and easiness of use, and being suitable for running on devices with modest computational resources.

Mean imputation is a simple and popular approach, where missing values are replaced with the mean of the considered variable. In our case, the missing data for the ozone measurements are replaced with the mean of the observed ozone values. However, one must be aware that often enough this method is not producing good enough results as it changes the standard deviation, and it does not account for the relationship among the variables.

Multiple imputation by chained equations (MICE) data imputation is a robust, statistical, principled, multiple imputation technique. It works by making multiple predictions for each missing value. The procedure fills in the missing data through an iterative series of predictive models, as explained in [184]. Azur et al. provide a comprehensive analysis and describe the chained equation approach to multiple imputation in [19], as well as an overview of the steps used by the MICE algorithm for convergence. In this work, we use the Python library function *impyute.imputation.cs.mice* that differs from the implementation proposed by Buuren et al. in [33] in two aspects, namely stopping criterion and variable to regress on[2]. We apply the technique to the whole dataset (5 columns, as described in Table 4.1).

missForest data imputation is an iterative imputation method based on a random forest and has been introduced in [204]. It works by averaging over a number of different decision trees (unpruned classification or regression trees). In this work, we use the *missForest* method, part of the *missingpy* Python libraryThe default values for the parameters of the method were used. The number of trees in the forest (*n_estimators*) is set to 100. The algorithm runs iteratively until the stopping criterion is met, or the maximum numbers of iterations (default value set to 10) is reached. The former happens as soon as the difference between the imputed arrays over successive iterations increases for the first time. We apply the technique to the whole dataset (consisting of the 5 columns, as described in Table 4.1).

kNN data imputation works by filling in missing data points based on the values of its closest k neighbours, identified through the usage of the euclidean distance [216]. In this work, we use the *KNNImputer* method, part of the *sklearn.impute* Python library. We apply the technique to the whole dataset (comprising the 5 columns, as described in Table 4.1). We chose a k value of 3 in order to keep the search for neighbours to a minimum. This can be further optimized by analysing the impact of the k value over the performance in relation to the time and space complexity.

[2] `https://impyute.readthedocs.io/en/latest/_modules/impyute/imputation/cs/mice.html`

4.3.4 Experiment design

The experiments used to evaluate and compare the performance of the chosen techniques for the data imputation correspond to the impairment methods described in Table 4.2.

For the *non-bursty case*, we compare the performance of the data imputation methods in the context of an impairment rate varying from 1% to 99%. A step of 5% is used for the impairment rates between 5% to 95%, and a step of 1% between 95% and 99%.

For the *bursty case*, the methods consider a burst size varying from 5 to 200 with a step of 5. We include also the non-bursty case scenario (with burst size 1) for comparison. The impairment rate is kept within the 1% to 25% range.

An important aspect of our experiment design is given by the environment used. In order to showcase the possible use of these data imputation techniques at the edge, within the IoT ecosystem, we make use of a Raspberry Pi 4B with 4 GB of RAM [212]. The RPI 4B is used to run the experiments and collect metrics for the execution time and memory usage. The experiments are also carried out on a laptop for comparison purposes. The technical specifications for the laptop and the Raspberry Pi 4B are highlighted in Table 4.3. Part of the experiments, which does not concern measuring execution times, is run exclusively on the laptop. Moreover, all graphics presented in this work were generated on the laptop.

Table 4.3: Hardware specifications for the experimental environment.

Hardware specifications	**RPI 4B**	**Laptop**
RAM	4 GB	16 GB
CPU	Broadcom BCM2711, quad-core Cortex-A72 (ARM v8) 64-bit SoC @ 1.5GHz	Intel® Core™ i7-8850H CPU @ 2.60 GHz

The setup of the RPI 4B for this work included installing the *Raspbian GNU/Linux 10 (buster)* operating system on the device. The code for the experiments was written and executed both on the laptop and the Raspberry Pi with *Python3.7*.

4.4 Result analysis

In this section, we present and discuss the results of the experiments. We group the results based on the two scenarios: the non-bursty case and the bursty case. We compare the two machine learning based techniques with the statistical based methods for the task of data

imputation by considering the root mean square error (RMSE), execution times, and the data distribution. Furthermore, we discuss the CPU and RAM usage for the RPI 4B.

4.4.1 Non-bursty case

Figure 4.2 depicts the evolution of the RMSE in relation to the impairment rate, which varies from 1% to 99%. As expected, the mean imputation performs the worst, as it does not consider anything else except the mean of the dataset. MICE imputation performs slightly better in comparison to mean. However, both techniques experience rapid and steady growth with the increasing amount of missing data. On the other hand, missForest and kNN experience a worsening of performance with the increase of the impairment level at a much slower rate. Only after an impairment rate of 75%, there is a more rapid increase, with the two methods eventually catching up to the other two. Additionally, considering a desirable RMSE of at most 10, mean and MICE imputation can handle an impairment level of at most 5%; however, missForest and kNN can handle missing data of up to 40 and 50%, respectively. The choice of a RMSE value of at most 10 was based on discussions with the supervisory team taking into account a small value, the chosen environmental dataset, the results, and the wish to best highlight the differences between the four techniques.

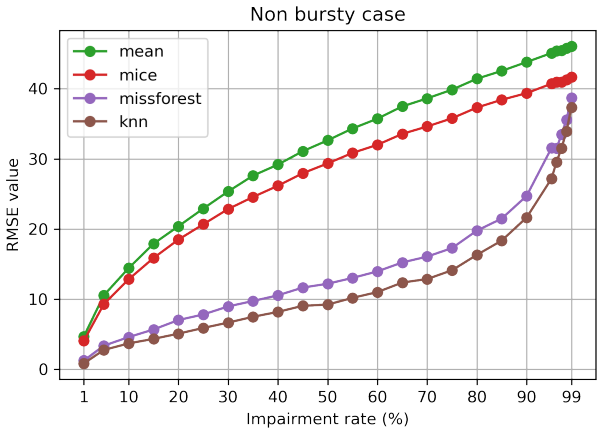


Figure 4.2: RMSE value in relation to impairment rate ($\%$) for the non-bursty case.

Figure 4.3 shows the data distribution by means of density plots, histograms, and box plots of the datasets, accounting for the original, contaminated, and after the imputation state, for 3 different impairment levels. For plotting the density, the Python library function *pandas.DataFrame.plot* was used. The function argument *kind* was set to *density*,

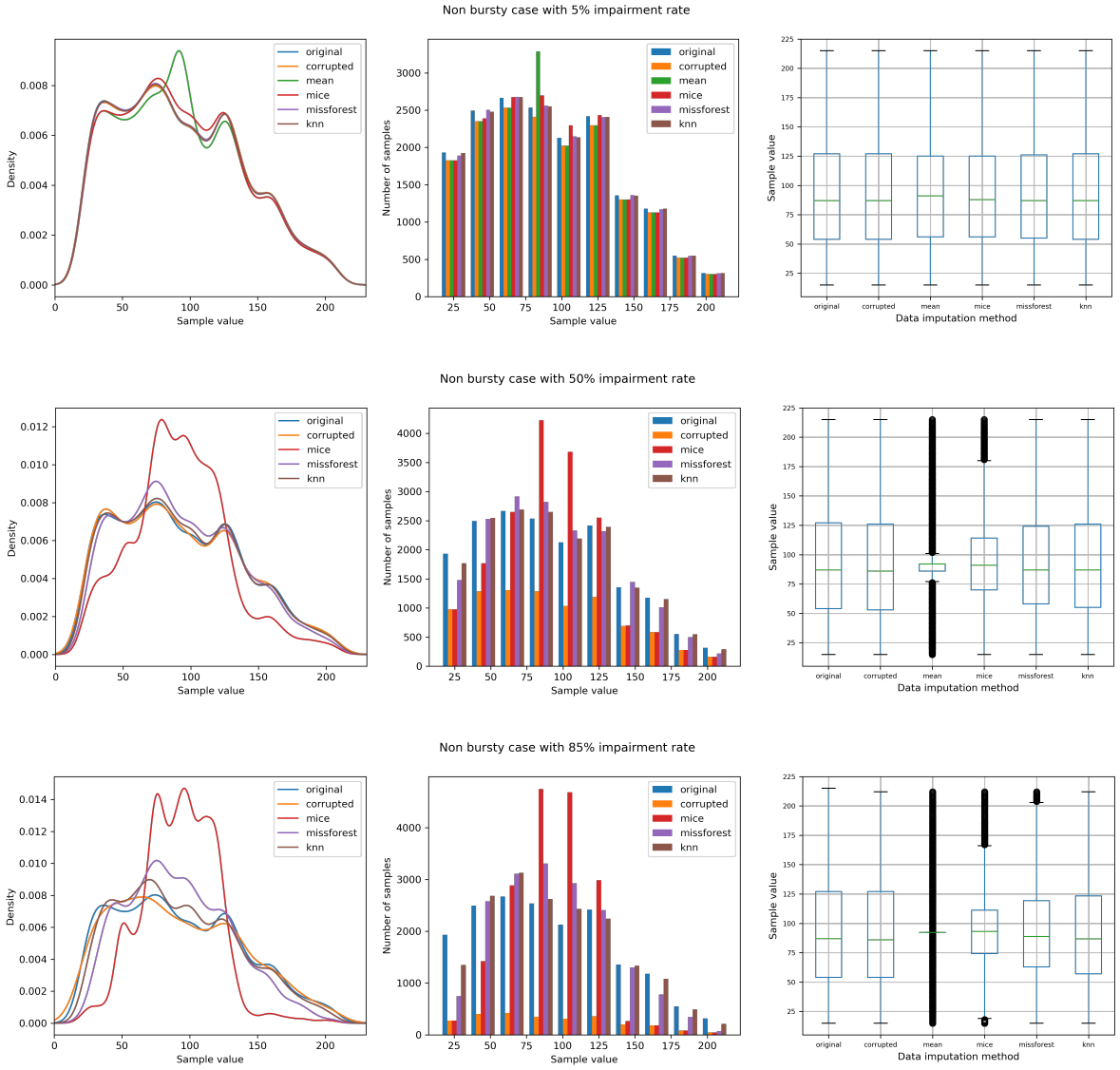


Figure 4.3: Non-bursty case comparison with 5%, 50%, and 85% impairment rate.

which corresponds to a kernel density estimation plot. The Python library functions *matplotlib.axes.Axes.hist* and *pandas.DataFrame.boxplot* were used for plotting the histograms and box plots. Starting from the top (5% of missing data), most techniques stay close to the original dataset. However, one can already notice the disadvantages of the mean imputation (the increase of distribution around the mean value). From the other plots, the line representing the mean imputed dataset is not included, to avoid skewing the plot. On the other hand, the behaviour of the mean imputation method is shown in the box plots, where the poor performance emerges especially for growing impairment rates.

For an impairment rate of 50%, it can be noticed that missForest and kNN remain fairly close to the original dataset, with MICE not being able to recover missing data as accurately. When the corruption level reaches 85%, which can be considered an extreme condition, both missForest and kNN experience a significant drop in performance, particularly towards the interval ends; nevertheless, they recover missing data better than mean and MICE.

4.4.2 Bursty case

Figure 4.4 allows us to compare the data distributions for the same impairment rate between the non-bursty case (burst size of 1) and the bursty case with a burst size of 100. It is observed that data missing in chunks has a greater impact on the performance of the data imputation methods, particularly on the machine learning based methods, as can be noticed in the density plots.

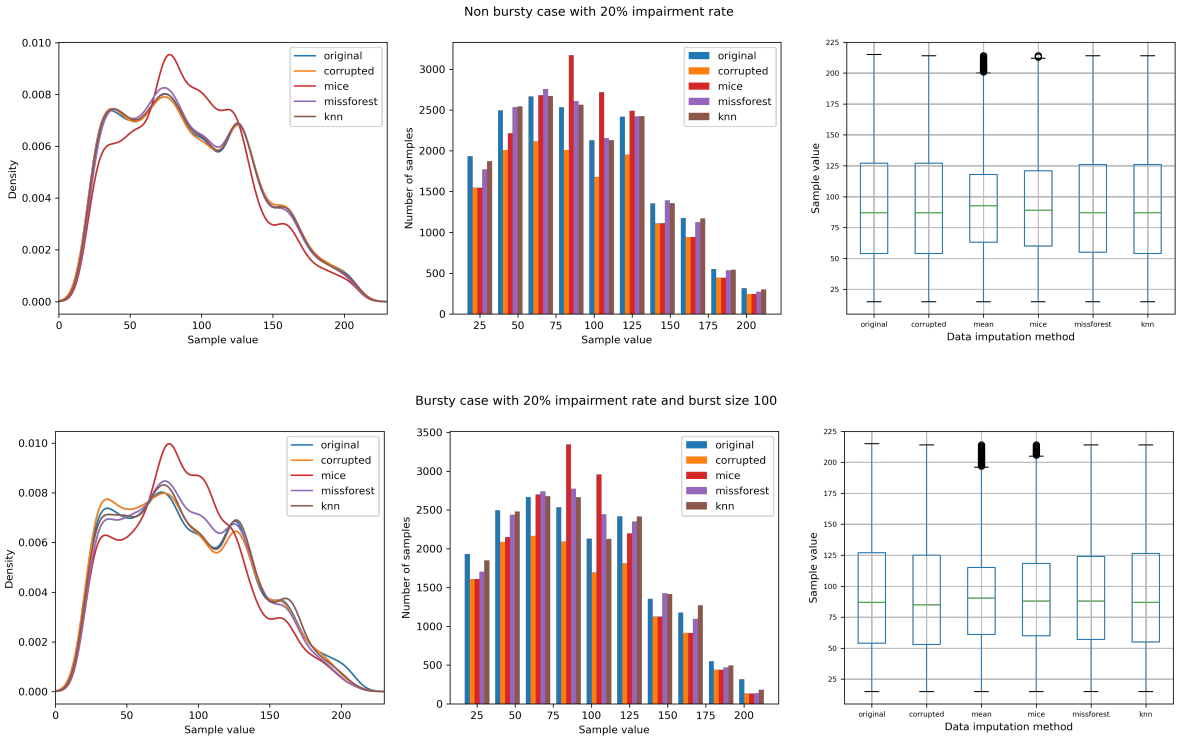


Figure 4.4: Non-bursty case and bursty case for the same impairment rate (20%).

Figure 4.5 offers a more in depth look into the performance of the four data imputation methods for the following scenarios: a fairly low contamination rate (5%) with a burst size of 25 and 100 data points, respectively, and a higher contamination rate (25%) with the same burst sizes of 25 and 100 data points. Figure 4.5 depicts the density plots and showcases

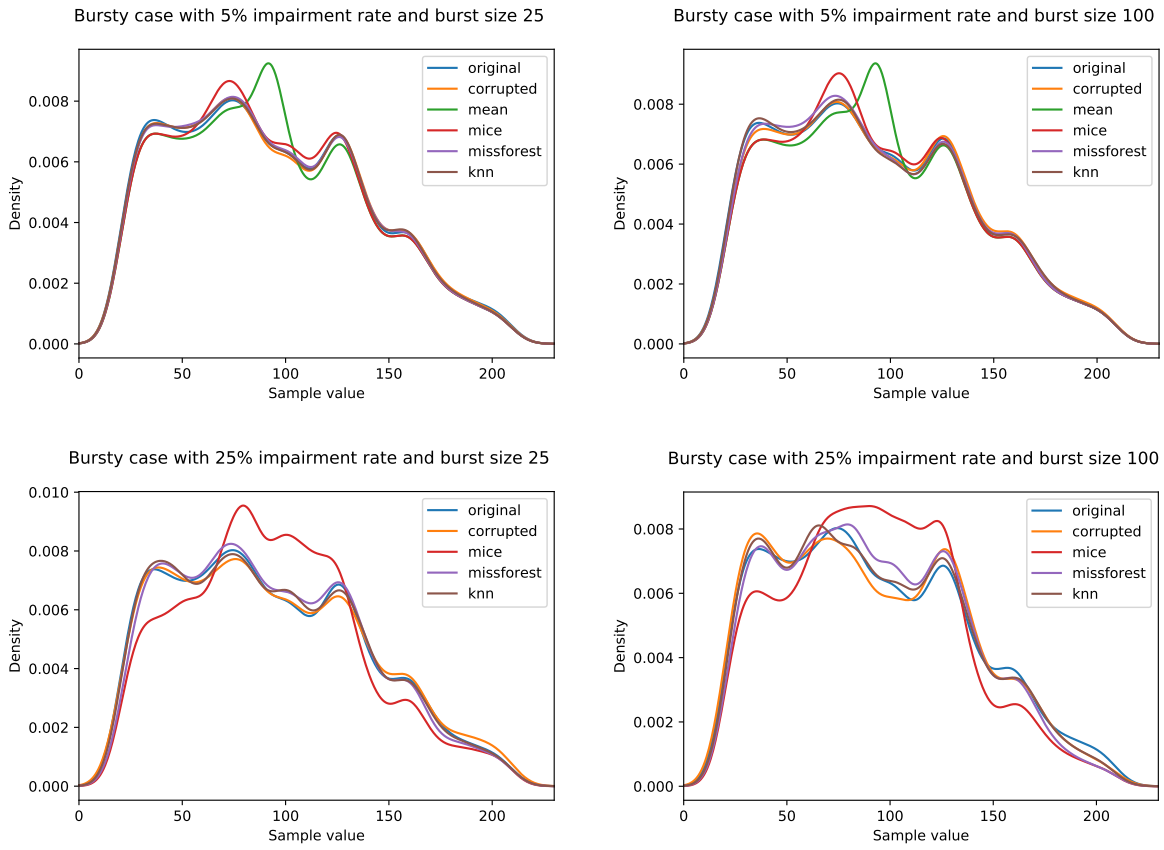


Figure 4.5: Density plots for different scenarios.

how the kNN and missForest outperform the other two techniques. For the case of 25% contamination rate, the density plot corresponding to the mean is not included. This is done as to avoid skewing the overall view caused by the decrease in performance. One interesting thing to note again in Figure 4.5 is how the increase in burst size affects the performance of the algorithms for the same contamination rate.

Figure 4.6 offers an in depth perspective of the evolution of the RMSE for each algorithm considering bursts of size varying from 1 to 200 points, and contamination rates between 1% and 25%, with a step of 5%. We can observe that the mean and MICE methods are not sensitive to the bursty case scenario, as opposed to missForest and kNN. It is easily noticeable that the two machine learning based techniques have a similar performance and outperform the statistical based methods in all the scenarios considered, especially for lower burst sizes (below 100 points). For the two, a decrease in performance can be noticed with the increase of the burst size. They only reach the performance level of the other two methods for a burst size of around 100. For an impairment rate of 1%, while kNN and missForest do outperform

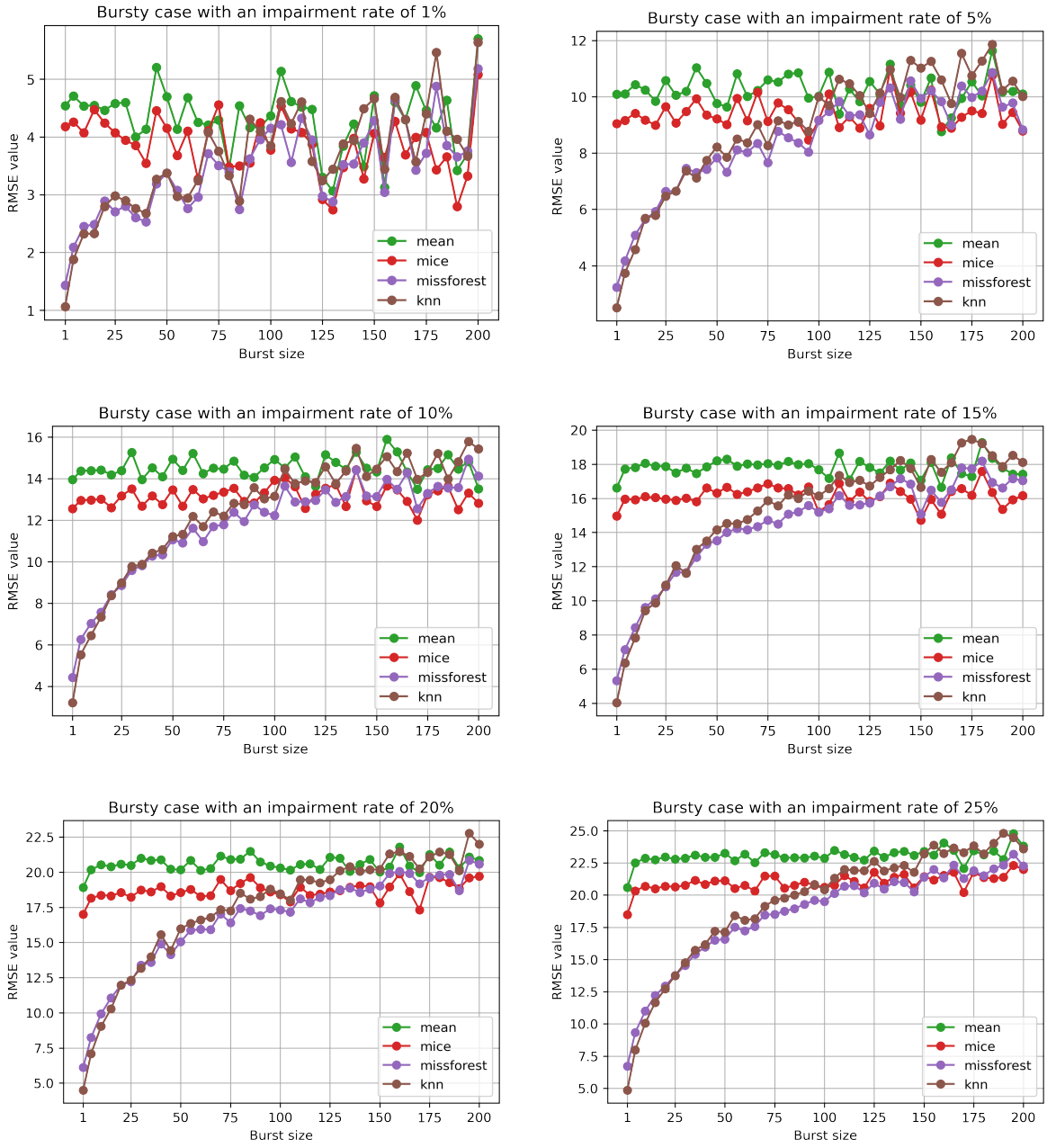


Figure 4.6: Evolution of RMSE with impairment rate in the bursty case.

mean and MICE, the difference in RMSE values is not as large as in the other scenarios with burst sizes below 75 points. As the impairment rate grows, meaning that more chunks of data are missing from the dataset, the statistical based methods are not able to accurately impute missing values, resulting in higher RMSE values. MICE does perform better than mean, which is to be expected given that the mean approach trivially imputes the mean value across all the

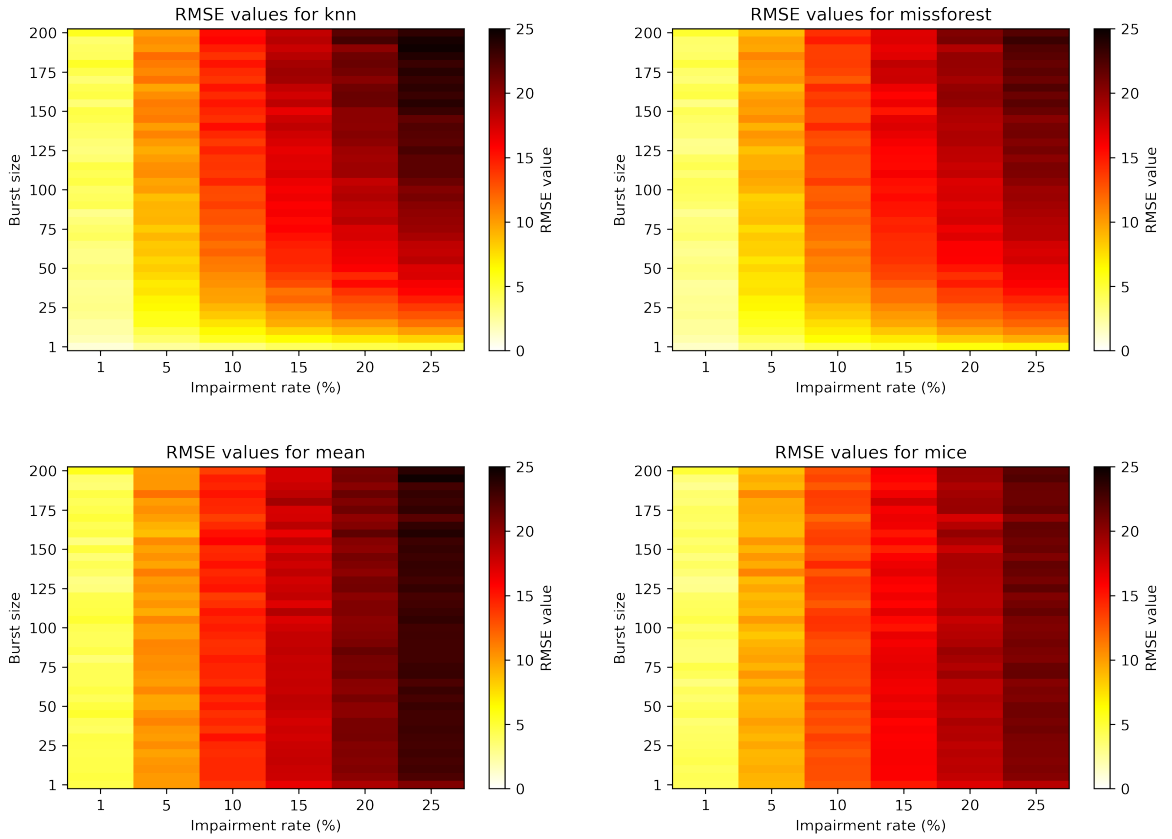


Figure 4.7: Colormap showcasing the RMSE in relation to the impairment rate and burst size.

existing data points for all the missing value slots. However, the difference is not too large, since MICE suffers from the assumption that there exists a linear relationship between the involved variables. Unluckily, such a situation is rarely met in real-world cases. Conversely, kNN and missForest make use of other information such as neighbouring values and are able to more accurately predict the missing values even at higher contamination rates, especially for lower burst sizes. By looking at the six presented scenarios, one notices that at the point of burst size being equal to 100 points, there is no longer a clear difference in RMSE values among the four techniques. Furthermore, up to a burst size of 25 points, the performance achieved by using kNN and missForest for the data imputation process is 50% higher than that of the mean and MICE methods. Cases depicting a contamination rate of more than 25% were not considered as the RMSE values were worsening, hence the data imputation process was not deemed reliable.

Figure 4.7 depicts a colormap for each of the four algorithms, while taking into account RMSE and burst size, thus, complementing Figure 4.6. Again, it is clear that the burst size

does not affect statistical based methods, but impacts the machine learning based methods. The impairment rate columns for mean and mice do not vary in colour (similar shades), as opposed to those for missForest and kNN (different colours and shades). Furthermore, for the bursty case scenario, it can be observed that missForest imputation performs slightly better than kNN. The impairment rate columns for missForest, while similar to the corresponding columns for kNN, showcase overall lighter colour shades. This implies a lower RMSE value for the majority of the impairment rate - burst size combinations. This mirrors the same behaviour previously noticed in Figure 4.6.

4.4.3 Time and space complexity

For the non-bursty case, we showcase and compare the execution time for each impairment rate while considering, as the running environment, both the laptop and the RPI 4B. This is depicted in Figure 4.8. The shown execution times correspond to the average value of 10 different runs. Mean imputation has a constant time of below 0.01 seconds for the laptop, and of approximately 0.02s for the RPI 4B. For both kNN and MICE imputation, the execution time goes up with the increase of the impairment rate. However, for missForest the execution time decreases due to the complexity drop of the dataset (higher degree of missing data), which results in less computationally intensive tasks for building the necessary decision trees. The maximum execution time for the RPI in the non-bursty case is around 80 seconds, below the sampling time window of 5 minutes. Hence, this would allow for real-time data imputation within the IoT scenario of our chosen pollution dataset.

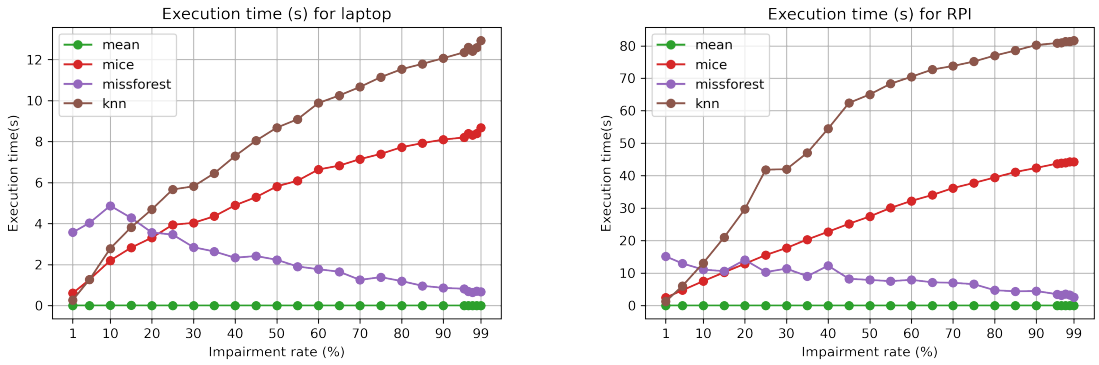


Figure 4.8: Execution times (s) on laptop and RPI 4B 4GB for the non-bursty case and varying impairment rates.

Figure 4.9 shows execution times for the RPI 4B for the bursty case, considering both the

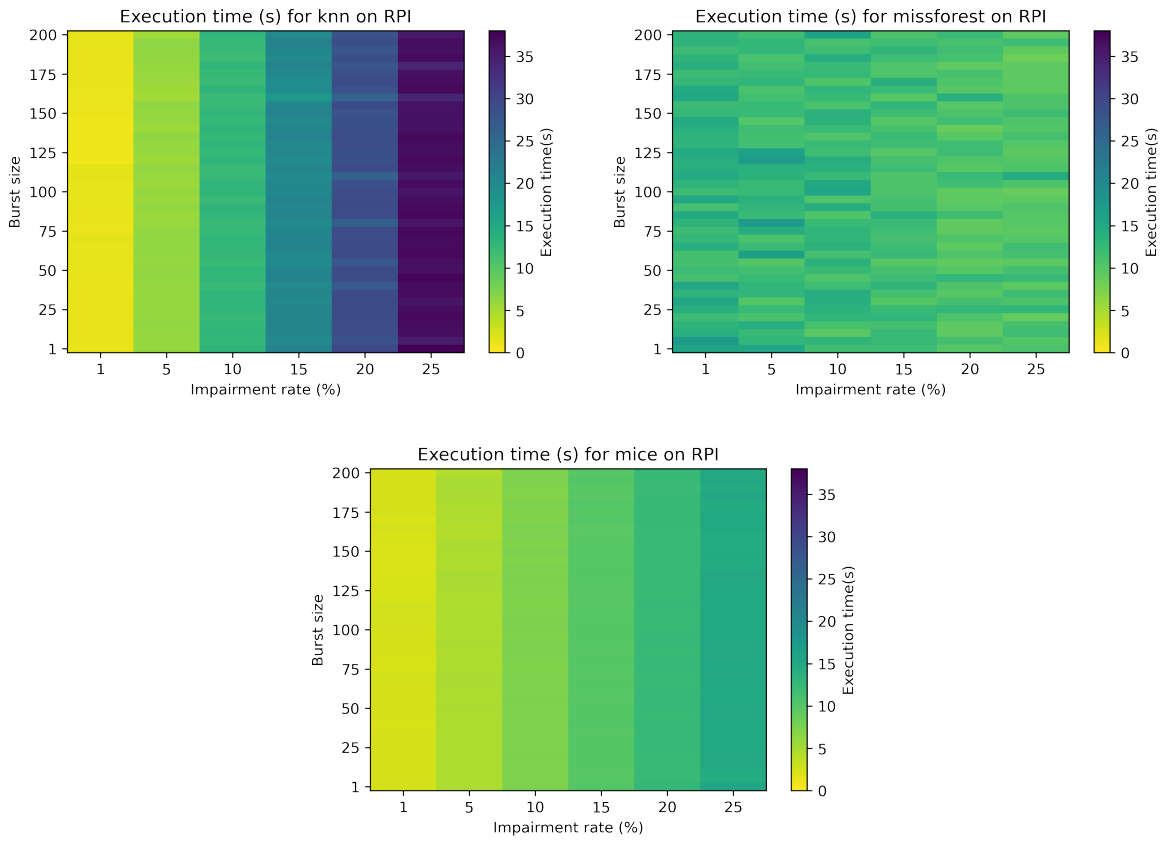


Figure 4.9: Colormap showcasing the execution time (s) in relation to the impairment rate and burst size for kNN, missForest, and MICE data imputation.

impairment rate and the burst size. The execution time for mean imputation is not depicted as it is constant as per the non-bursty case (approximately 0.02 s). Execution times for kNN and MICE are not affected by the burst size, while missForest may be just slightly affected.

Figure 4.10 provides a snapshot of the CPU and RAM usage of the RPI 4B for the non-bursty case with a 50% impairment rate. The snapshot was taken with *netdata*, a tool for real-time monitoring[3]. MissForest is the only method using all 4 cores of the RPI 4B. The other techniques use only one core at maximum, amounting to roughly 25% of the total CPU power. In terms of RAM usage, kNN imputation uses up to 3 GB of RAM for an impairment rate of 50%. kNN imputation can be carried out up to 30% impairment level also on an RPI 4B with 2 GB of RAM. The other techniques have a fairly constant RAM usage of a maximum of 0.5 GB, and, therefore, can also run on an RPI 4B with 2 GB of RAM.

[3] `https://www.netdata.cloud/`

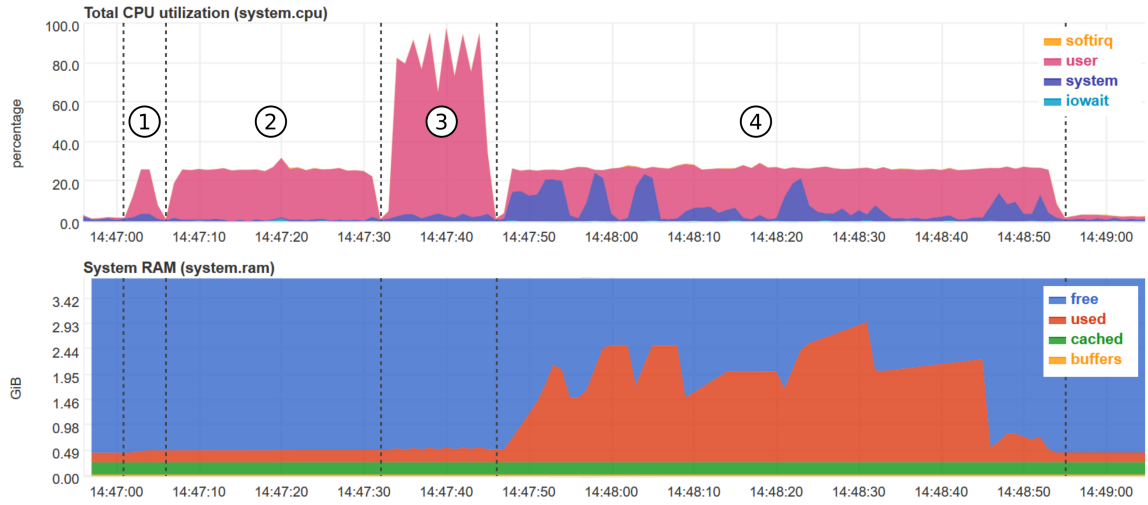


Figure 4.10: Snapshot of the CPU and RAM memory usage for the non-bursty case with a 50% impairment rate on the RPI 4B (4GB of RAM) for each algorithm (1 - mean imputation, 2 - MICE imputation, 3 - missForest imputation, 4 - kNN imputation).

4.5 Discussion

The performed experiments lead us to a number of insightful observations. In regard to performance, kNN and missForest clearly outperform the remaining algorithms, due to their particular structure. For instance, it is useful to recall that both kNN and missForest are non-parametric techniques, thus, they perform well when there is no particular assumption about the data to impute. Conversely, MICE assumes that the data generating process originates from a parametric distribution. In particular, it assumes a linear relationship among the involved variables (rarely met in real-world cases) which causes an accuracy decay. Finally, the mean imputation method is the worst in terms of accuracy, mainly due to the fact that such a technique typically causes a strong variance alteration of the mean-imputed variables.

Regarding the time complexity analysis, a known weakness of the kNN is the computational time, mainly due to the fact that the data needs to be calculated and sorted, leading to a complexity order of O(n log n). Furthermore, parallelizing kNN is not necessarily a trivial task given the centralized approach of components of the algorithm, e.g. sorting. It is important to note that parts of the kNN algorithm can be parallelized, but it requires a more in depth analysis of the implementation and selection of the components which could benefit from parallelization. Although MICE is (slightly) faster than kNN, it relies on an imputation process repeated until all the missing data are estimated, resulting in a non-negligible computation time. In contrast, missForest can benefit from a strong parallelization (given the distributed nature of

the algorithm), whereas mean imputation takes advantage from a very simple implementation, resulting in very competitive execution times.

Considering the non-bursty case scenario and by solely looking at the RMSE, one would designate kNN as the winner method. Taking into account also the bursty case, one notices that the differences between kNN and missForest are minimized, with missForest even slightly taking over in terms of performance. At this point, we could say that actually, both techniques are a winner, providing good performance compared to the more classic approach of mean and MICE. However, the situation significantly changes when we take into account both execution times and RAM usage, factors which are of great importance when considering edge computing and real-time performance. missForest appears to better suit the limited resources of an edge device such as the RPI 4B for impairment rates larger than 10%, as it does not require large amounts of RAM and has a faster execution time. However, for lower impairment rates, the execution times for kNN imputation are lower than the ones for missForest, and the RAM usage is not as heavy.

Another important point for deciding the best algorithm is the possible optimizations that could minimize resource consumption and decrease execution time. All techniques presented in this chapter were used without additional optimizations, in order to offer a ground truth and bare-bones evaluation. A key optimization method would be to perform data imputation on time windows, as opposed to the whole dataset. This would in turn reduce the required computations, as well as the use of computational resources. Furthermore, the accuracy and performance of the techniques should not be negatively impacted in scenarios similar to our dataset (i.e. IoT pollution measurements data), if the time window is chosen to account for enough neighbouring data and any necessary existing patterns. Time windows may allow parallelization of the other algorithms, which would lead to faster execution times.

4.6 Conclusion

In this chapter, we investigated how to deal with missing data within IoT smart environments at the edge. Taking advantage of the edge, we performed real-time data cleaning by means of data imputation, to increase the reliability and usability of data. We investigate and evaluate kNN and missForest, two machine learning based techniques, on imputing missing data from an environmental IoT setting within two scenarios (bursty and non-bursty missing data). The two are benchmarked against two statistical based methods, namely mean imputation and MICE, considering the following metrics: RMSE, density distribution, execution time, RAM, and

CPU utilization. The experimental work is carried out close to the data source (sensor nodes), on board a constrained IoT device, namely the RPI 4B. kNN and missForest outperform the other two techniques, being able to cope with impairment rates of up to 40% for the non-bursty case, as well as being able to recover blocks of up to 100 missing data samples for the bursty case before dropping to the performance level of mean and MICE.

Execution times are shorter than the sampling period of the considered environmental IoT scenario, thus, allowing us to show that real time data imputation can be achieved at the edge on board constrained devices. This encourages us to continue exploring how to take advantage of edge computing, for optimizing existing processing pipelines in the IoT, as well as to build on top of existing smart intelligent sensors.

Chapter 5

Improve Well-being through Urban Nature (IWUN): a Real Social Case Study in an Edge Cloud Setting

The accessibility of digital technologies and the Internet of Things provides the opportunity of engaging in large scale social studies and collecting ample amounts of data concerning cities and their citizens. In this chapter, we investigate how machine learning and data science techniques can be used to analyse data from social science studies in a novel and unconventional way. The focus is on maximizing insight gain for these types of studies by considering both objective (sensor information) and subjective data (direct input from participants). The IWUN (Improve Well-being through Urban Nature) project is a pilot study that aims to better understand the impact that the interaction with nature, in particular urban green spaces, has on the citizens. A field experiment was carried out in Sheffield, UK, engaging 1870 participants for two different time periods (7 and 30 days). By the use of a specially developed smartphone app, both objective and subjective data were collected. When the participants entered any of the publicly accessible green spaces, tracking was activated and their location data from within the designated areas was recorded. Furthermore, this information was complemented by observations (both textual and photographic) that users could insert either spontaneously or when prompted by the app upon entering a green area. In the context of this thesis, the IWUN project serves as a case study for performing data analytics in the IoT Edge Cloud setting. Through the use of data science and machine learning techniques, we determine, from both text and images, the main features and characteristics that the study participants observed in their interaction with nature. Moreover, we compute

and analyse dwelling time in the parks, and the top interaction areas (both by time spent and observations made). This work highlights different patterns of citizen behaviours in green spaces and demonstrates the potential of integrating the use of technology and specific techniques into large-scale social studies, which traditionally opt for other approaches to both data collection and analysis.

The IWUN project was supported by the Natural Environment Research Council, ESRC, BBSRC, AHRC & Defra [NERC grant reference number NE/N013565/1]. The app was conceptualized and designed by a team of psychology researchers at the University of Derby. The app implementation was carried out by a development team. Further details are available in the acknowledgement section of [63]. Data analysis and visualization presented herein were performed by the Data Science Research Centre, University of Derby, UK. The work in this chapter was a joint team effort from myself and my colleagues. The team of psychology researchers contributed to our work with meaningful discussions, feedback and insight into the project itself, as well as the provided data. The tasks concerning the data analysis and implementation of the different techniques were shared as follows: I performed the experimental part for the image analysis, while my colleagues from the Data Science Research Centre undertook the experimental part for the time analysis and text analysis. However, the work presented in this chapter is the result of our joint efforts, and we all contributed to the data analysis, as well as the brainstorming and the ideas surrounding the different experimental parts.

This chapter is based on our paper "*Analyzing Objective and Subjective Data in Social Sciences: Implications for Smart Cities*" [63], which is published in the *IEEE Access* journal. Two conference papers preceded the journal paper, namely "*A Pilot Study Mapping Citizens' Interaction with Urban Nature*" [67] as part of the 2018 IEEE 16th International Conference on Dependable, Autonomic and Secure Computing, 16th International Conference on Pervasive Intelligence and Computing, 4th International Conference on Big Data Intelligence and Computing and Cyber Science and Technology Congress(DASC/PiCom/DataCom/CyberSciTech), and "*A Demographic Analysis of Urban Nature Utilization*" [68] as part of the 2018 10th Computer Science and Electronic Engineering Conference (CEEC). Furthermore, recently, the University of Derby has obtained the Green Gown award[1] through the work of the Nature Connectedness group, to which we have also contributed.

[1] `https://findingnature.org.uk/2021/11/22/double-award-win/`

5.1 Introduction

The advancements in technology and the digitalization of the physical world, allow the Internet of Things (IoT) to encourage a variety of multidisciplinary studies, part of which focuses on the human interaction with cyber-physical systems [242]. This is due to the desire of harmonizing the interaction between society and the smart things. Furthermore, a paradigm focusing on the social side of IoT emerges [95]. The Internet of Things vision for a Smart City employs advanced technologies to foster the administration of cities with the aim of providing better utilization of public infrastructure, and improved quality of service to the citizens while operating at a minimal administrative budget [238]. The end goal is to create an integrated approach for managing and analysing the data to help in planning, policy, and decision-making for a smarter environment and improved quality of life for the citizens [112]. Key interventions are possible to directly influence urban health and well-being [119].

In this work, we are looking at a real-world pilot study on how data science and machine learning techniques can enable us to gain insight into social science studies. Social science studies have conventionally been based on data gathered from paper diaries, stand-alone electronic devices or self-administered forms [144], and have employed traditional methods of data analysis which are laborious, time-consuming and can limit the insight that can be achieved through the study. In one such crowdsourcing research [191], authors investigate the perception of young people about a city in a developing country using descriptive statistics. Our work is different in the sense that it employs data science tools to uncover patterns and make correlations in a way that may not be easily identified with traditional statistical tools. Furthermore, by taking advantage of the increase in technology use, besides the subjective data directly collected from the study participants, objective data can also be obtained (as recorded by sensors in the used devices).

This pilot study is concerned with better understanding the interaction of citizens with green spaces and improving well-being through engaging with urban nature. The insights from these interactions can be used to help stakeholders in planning, policy and decision-making, in addition to improvement of the citizens' experience and life quality. The study involved tracking 1,870 subjects for two different periods (7 and 30 days), covering 760 digitally geo-fenced green spaces in Sheffield, UK. To collect data, we used Shmapped, a smartphone app developed by the IWUN project [106], which allows for measuring the human experience of city living. The app serves as a dual data-collection tool for both subjective data (well-being, personal feelings, type of social interaction, the users' observations about their surroundings) and objective data (location tracking for when a user enters a digitally geo-fenced green

space in the city and activity detection). The app is also used as an intervention tool that prompts participants to notice and record the good things in their environment, using either text or/and photographs. This is theorized to improve their well-being, as research has shown links between exposure to green space and well-being [193], [144]. Moving beyond exposure, [188] and [189] outline the benefits of improving nature connectedness through noticing the good things in nature. Such research has led to much interest in the design of smart city management frameworks for improved quality of life [20], [133]. The research in this line of interest can be a major challenge due to the complex processes involved in planning, collecting and analysing vast amounts of data. Clearly, a large-scale IoT infrastructure can improve this process by automating data collection, storage, processing and analytics [119].

We are looking at a novel model of analysing the information obtained from data driven social applications in order to maximize the insight gain. Through the use of technology, particularly smartphones, we aim at complementing the traditional way of gathering data in social sciences. Moreover, it also allows the collection of objective data (sensors' information) which can open the study to new dimensions of analysis. Through information fusion, we can find new links between a citizen's interaction with the surrounding environment and the features of the city. This kind of study enables a smart city approach and allows for a better and more accurate representation of a citizen's interaction with the city because users are asked for information or interrogated about their observations and behaviour in moments of action.

This difference from the traditional way of asking people to fill out a questionnaire, allows for monitoring in the moment of interaction, collecting both subjective and objective information about the specific time. For example, when users enter a geo-fenced green space, they are prompted to answer a series of questions specific to that moment: who is accompanying them in the visit; what good things they notice about the surroundings; how would they grade this interaction etc. Simultaneously, the location and other sensor specific information (from the accelerometer) are tracked and can be used to determine the time spent in the green space, speed etc. Furthermore, this approach allows for scaling up social studies and collecting information from multiple subjects at the same time. In a smart city scenario, this can be used to monitor and improve existing infrastructure, as well as the quality of life. We use several data science and machine learning techniques in order to gain insight from the data generated by the users in Sheffield, UK. First, we clean and pre-process the raw information, and then we proceed into a further analysis of the text observations, the images taken, as well as the location points. We identify the clusters of topics in the observations, and we automatically map the observations against the categories of themes from previous research

into noticing the good things in nature [188]. We identify the features in the images taken by the users and compare the top labels with the text data. Based on the location points, we look at the time spent in the green spaces from different perspectives and compare it against the location data derived from the observations. These types of information fusion allow us to gain a better understanding of the interactions between the users and their surroundings, as well as plan the next steps for extending and improving the present work.

This chapter is organized as follows: Section 5.2 gives an overview of the related work; Section 5.3 describes the methods employed in this study; Section 5.4 presents the dataset; Section 5.5 outlines which features were noticed by the users; Section 5.6 looks at the time users spent in green spaces; in Section 5.7 we analyse the park use based on gender and age, and in Section 5.8 we present concluding remarks and indicate future research directions.

5.2 Related work

For the related work, we focus on three main directions. Firstly, we highlight how our work fits within the topic of Big Data in cities and social science studies. Afterwards, we present and discuss how social science studies can benefit from extracting information from both objective and subjective data. Finally, we take a look at other studies which make use of an app for data collection, and which investigate the connection between well-being and nature.

5.2.1 Big Data in social science studies

Often, when talking about Big Data in the context of cities, the focus seems to be on the volume attribute of Big Data. Strom [205] outlines that Big Data is tritely characterized as anything which exceeds the storage capability of an Excel spreadsheet or of a single machine. Shahrokni et al. tackle inefficiencies in waste collection routes by analysing half a million waste fractions[200]. In a different study [13], the authors investigate social textual streams contained within 8 million tweets in order to identify traffic events in the San Francisco Bay Area. Unlike these studies dealing with Big Data in cities in terms of considerable amounts of data, in our work, we tackle other inherent characteristics of Big Data which prove to be as challenging. Admittedly, the difficulty within our data is given by its variety (structured and unstructured data), exhaustivity as it attempts to capture all the population, scalability (its size can rapidly increase), relationality (the data consists of common fields which can be correlated), and messiness, traits corresponding to Big Data [120].

5.2.2 Mining objective and subjective data

Based on the method of data collection, data could be broadly classed as objective data or subjective data. From the IoT perspective, objective data can be obtained from the things in the IoT, such as sensors, GPS receivers and smartphones, while subjective data is collected directly from humans. Technology has made it easier and faster to collect objective data, and such research can boast of a large volume of data for analysis. In one such study, [80] data collected from accelerometers are used to control gaming mechanisms that encourage metabolic activities. Authors in [34] performed real-time monitoring of urban mobility (traffic conditions and movement of pedestrians) using data collected from the GPS of mobile phone users, buses and taxis. While such objective studies may perform better at collecting information faster and at a larger scale, they hardly account for the harmonious interaction between these smart objects and humans, an important element in smart cities [96]. Again, one has to deal with issues of data quality in objective data like uncertainty (sensor precision, missing readings), inconsistency and redundancy in data [182]. Subjective data presents the problem of being limited in volume, and diminishing in quality over time (people start a study with a high response at the start and then get tired - law of diminishing returns). Social networks have made it easier and faster to collect subjective data like event tweets [159]; however, they tend to be noisy, messy and get thinner when filtered down to specific interests. Even though the process of collecting subjective data may limit the volume for Big Data studies, it could make for richer, diverse and complementary analytics for smart cities [203].

We address the limitation and leverage the strength of the two using a hybrid data collection approach. On the one hand, we collect data from GPS and sensors, and on the other, we put in the human element through text and image information collected from participants. There are similar works that have employed the concept of objective and subjective data mining. In [145] the participants are asked to report their well-being at random times during the day, whilst having their location tracked. The response of participants in the app is then correlated with the GPS and weather information. In our work, we show that the text and image entries collected from participants can be harnessed in the context of smart cities to complement other modalities, such as the location data from the GPS, thus, providing a comprehensive view of the green space in the city.

5.2.3 Investigating the connection between well-being and nature through app-based studies

The rise of the IoT and the popularity of smartphones has allowed for collecting much larger sample data (both subjective and objective) automatically. Using smartphone apps to carry out social science studies returns more statistically robust findings, while being more cost-effective, and allowing for larger datasets [161]. One example is given by Mappiness [145], a social app designed as an intervention tool with the purpose of enhancing happiness, as part of one's well-being. To this end, the participants are randomly prompted throughout the day to report on their well-being, whilst having their location recorded. Similarly, Urban Mind [21] is another social app designed with the goal of evaluating in real-time the effect that exposure to green spaces has on mental well-being. To assess the latter in the urban areas, there were seven prompts a day in which the participants had to answer a particular set of questions subject to their location (indoors/outdoors).

For both apps, the data collection happened mainly indoors, as the participants only spent at most 14% of their time outdoors. Consequently, it was challenging to gather the data in green spaces, where the expressed level of happiness is actually higher. This major limitation highlighted in the two studies triggered the optimization of Shmapped for data collection by making use of geo-fences. Namely, the green spaces were structured within bounded areas and the participants were prompted to observe their surroundings upon entering one of them. As a result, the reliability of the study which evaluates one's interaction with nature is improved, given that people are solicited to describe their experiences and feelings when in green spaces.

5.3 Methodology

In this section we discuss the methodology, namely the app facilitating data collection and interaction with participants, the cleaning and pre-processing of the collected data, as well as the chosen techniques for text, image and time analysis.

5.3.1 Shmapped and data collection

Given the spread of smartphones in today's digitalized world, it is reasonable to employ apps in order to gain insight into the users' interaction with nature. For this study, the app dubbed Shmapped (Sheffield Mapped) was developed [106]. Shmapped uses a chatbot to achieve a human friendly and engaging interaction with the participants. It collects both subjective and

objective data, by means of two main tools:

- The intervention tool, prompts the users to notice something good about their environment and to translate any observations into text, image, or both. This prompt can also be snoozed and the users are reminded of it in the evening. In the latter case, we can assume that the users' comments are made retrospectively, rather than during the moment of interaction. Since this data is generated directly by the users, we consider it to be subjective. In the remainder, we will refer to this as the *observations* or the *comments data*.

- The data collection tool tracks the participants' movement, whilst they are within geo-fenced green spaces. We collect the users' GPS location and derive their activity from the device sensors. Therefore, we can discern among different users: stationary, walking, running, etc. In the remainder, we refer to this data as *objective*, given that no user intervention is involved.

The *objective data*, also referred to as *locations*, collected through the app can be split into two main categories. The first one consists of *GPS data* corresponding to each participant. Whenever a user entered a digital geo-fence (circular area comprising a green space of interest as displayed in Figure 5.1), their location and speed data were recorded, and later used to analyse the time spent in the green areas. The data used to define the geo-fences was provided by the Sheffield City Council as detailed geography of publicly accessible green spaces. Furthermore, the use of geo-fences enabled the app to be woken up from a stand-by mode and to more reliably record GPS data of interest. The second category consists of *derived data*, i.e. fields of information derived from the GPS data which characterize the user activity. From the sensor values, we can identify the users as being still, on foot, in vehicle or unknown.

The *subjective data*, also referred to as *observations*, was collected directly from the participants. When prompted, they could introduce observations regarding the "good things they noticed" in their surroundings. As the users engage with the app, they provide the following information: comments about what they noticed, pictures (optional), the reason for being in that location ('whyThere'), their companionship ('whomWith'), information about how built-up that location is ('howMuch'), and data about how they feel in that particular moment ('howFelt'). All these fields describe their experience within the green space.

The data described above were collected throughout different testing periods, considering two cases: 7 and 30 days, respectively. Besides this data, a user had to fill three other questionnaires: one at the beginning (containing demographic data as well as assessment of

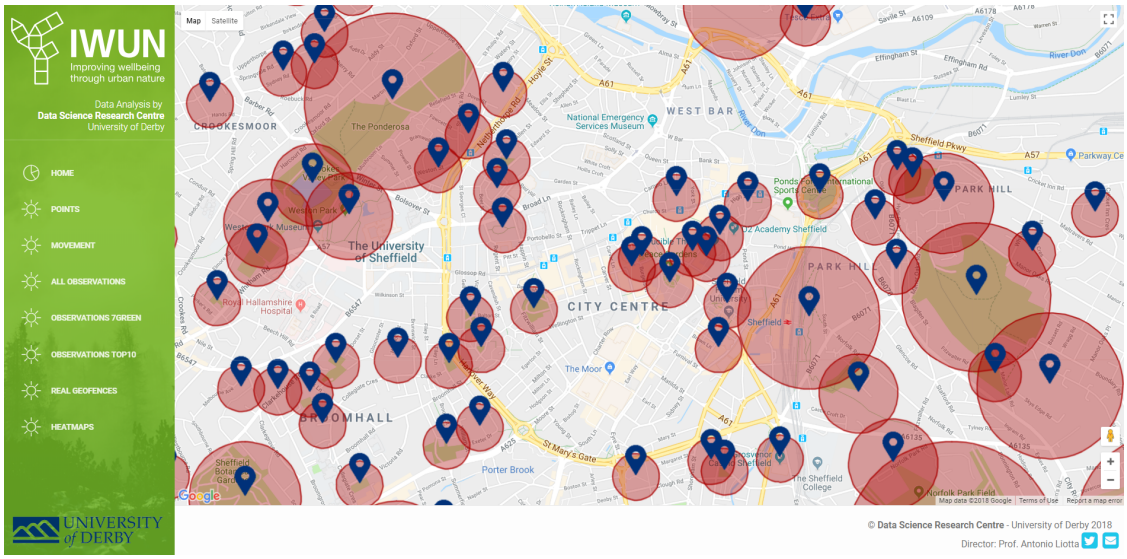


Figure 5.1: Preview of the extracted geo-fences.

individual differences and well-being); one immediately after completing the study and a third one at follow-up (1 month for the 7 days version or 3 months for the 30 days version). The last two are used to measure aspects of well-being and nature connectedness. This enabled us to track participants' well-being throughout the study.

5.3.2 Data cleaning and pre-processing

After we received the dataset and understood the aim of the study through discussion with the IWUN team, the first action point included cleaning and pre-processing the data entries. As can be expected with studies of this nature, parts of the data collected were not relevant. For example, we filtered out data corresponding to users who registered and took part in the project, but were not living in Sheffield, UK. Indeed, as the study was focused on one particular city, the few entries from other places were irrelevant for this particular analysis.

The observations collected from the participants included free answers, such as text, images, or a mix of both, as well as controlled input, such as: whom they were with, how they felt, why they went there, how built-up the environment was. The processing was intended at fusing the information by making use of text and image analysis, together with correlating the why, whom and how. The objective data consisted of information collected automatically by the app which served as a starting point for inferring time spent in different locations, along with level and type of activity. In the following, we offer details as to the cleaning and pre-processing performed for each type of data.

Users: The study was designed to split the participants into two categories, namely the green (70%) and the built (30%). Participants part of the first group reported on the good things they noticed about nature, while the second group recorded their perception of the built environment. The built category served as a control group, as designed by the team of psychology researchers. In our analysis, we split the data accordingly, while maintaining a focus on the green, as the goal was on providing insight into the interaction between citizens and their green surroundings. A number of 1870 people registered with the app, with 1290 being assigned to the green group (69%), and the remaining 580 to the built group. However, the number of uniquely identified users in our analysis was lower, given that some registered people did not go ahead with using the app and providing data. Moreover, some users were not living in Sheffield and their data was filtered out based on the postal code they provided at registration.

The **observations** encompass the images and text comments provided by the participants, and they amount to 5625 entries. It is noteworthy that, out of these, only 418 had an associated timestamp, meaning that the user recorded the entry when prompted, and not at a later time. The rest of the entries were made mainly in the evening, following the reminder given by the app. In the scenario of an entry made at a later point in time, the participant was solicited to manually input the location corresponding to the comment; however, the field was left blank more often than not. In our previous analysis, captured in [68] and [67], we focused on the entries with associated location data to determine the top green areas in terms of interaction. We did not identify a method to optimally reconnect the other observations with their location. One possibility included analysing the text comment provided and singling out unique location identifiers such as names, features etc., as well as using the location data of the day to identify the visited green areas. However, this approach would require expert knowledge of the green areas, and it would not suit, nor fit the case of multiple park visits in one day or general comments which sum up the experience with the surroundings. As a result, further analysis looked at the observations data separately from the location data. For example, the text analysis focused on classification and feature extraction. To this end, only data provided by the green group, namely 4225 entries from 718 users, was investigated. Not mixing the built and green groups is important for increasing the clustering performance, especially considering that the data used for training the classification model was concerning nature. For image analysis, 1641 images were used, out of which 1020 correspond to the green group, and 621 to the built group.

Location points: As previously mentioned and depicted in Figure 5.1, the app made use of geo-fences in order to collect location data. 949 green spaces were mapped and resulted in

760 geo-fences. Given that the geo-fences were implemented as circular areas containing the green space, location data was also collected when not in the actual green space. This data had to be filtered out, while paying attention to avoid excessive filtering. For example, GPS data from people walking on the paths bordering the green spaces were maintained. To also account for the edge case, namely people exposed to green spaces while in the close neighbourhood of the area, we select location points with an associated accuracy value of below 10 metres. The selected GPS data was used to determine dwelling time in the green spaces. To this end, we further restricted the analysis by only considering green spaces within a 5 kilometres radius circle centred in the city centre of Sheffield, UK. As a result, we analysed 539 green spaces, corresponding to 78 square kilometres and 1,184,702 location points.

5.3.3 Text analysis

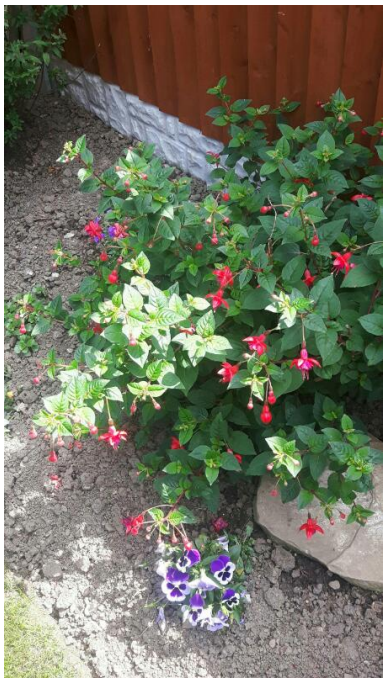
To have an initial understanding of the data, we performed an exploratory analysis to discover the key topics in the observations. We used the k-means clustering algorithm to partition the observations into chunks of related data points based on some similarity measure such as the euclidean distance, using as parameter the number of clusters, k. This number is typically determined experimentally, aiming to satisfy a given metric such as a distortion and silhouette score. An optimal number of k clusters is chosen such that it minimizes the distortion and maximizes the silhouette. We found this to be k=40, which leads to the minimum steepness of the distortion and silhouette. This means that the users' entries were divided into 40 separate clusters. Next was to map the Shmapped data against earlier studies of human connection to nature conducted by [188]. This study was conducted with 65 participants who were asked to record three good things in nature each day for five days. Using an emergent coding, the information was then hand-coded into 11 themes using content analysis, a systematic technique used to code large volumes of data [127], [221].

Table 5.1 shows the list of themes of the training data, the description and distribution in the dataset. We used the Fasttext API [114] to train a classifier with the data. Using the trained classifier, the model outputs the most likely labels for our observation data. As the training data is sparse, we train on 100,000 epochs. This is a multi-label classification problem where an input instance can be mapped to multiple output classes [94]. Hence, we extract the predicted labels and the probability. We set the threshold such that the predicted labels with a probability below 50% are discarded. Semantic analysis (as done on social tweets in [159]) does not work with our dataset because the people had been asked to notice the 'positive' things about their environment. So, most texts had positive sentiment with few outliers.

Table 5.1: Labels from training data from a study of human connection to nature[188].

S/N	Theme	Description	Example	Distr.
1	Specific part of nature	When an example of a specific plant, animal or feature of nature was given with no or very little context.	A bumble bee; Bluebell wood; Bright rainbow; Beach	100
2	Animals being active in their habitat	When animals were discussed in terms of some activity in their habitat.	Pigeons walking in a group together like a family; A buzzard being mobbed by crows; watching 2 birds dance together; Squirrels running up a tree together	109
3	Animals interacting together	Reference to animals engaging in an activity with at least one other animal, such as playing/chasing/hunting.	Pigeons walking in a group together like a family; A buzzard being mobbed by crows; watching 2 birds dance together; Squirrels running up a tree together	47
4	Sensation of nature	Items which focus on the sensations of nature: smell, sound (including bird song) or touch.	Sun on my skin; Birds tweeting in the trees; Sound of long grass in the wind; hearing the birds singing to one another	159
5	Colour	Items which had a specific emphasis on colour.	Bright pink blossom on the trees; The slug that I removed from my sage plant had quite a fetching orange belly; The grass looks very green in the rain; Green on the leaves	76
6	Effect of weather on something	When the weather has an effect on a plant or another aspect of the environment.	The breeze in the trees; Sunlight streaming in through my window; The long grass on the bank if the stream had been flattened beneath the weight of the rain drops hanging from it this morning	93
7	Growth/temporal changes	Reference to new buds, things in coming into bloom and changes associated with the seasons.	The soft new leaves emerging on our beech hedge; Purple flowers starting to bloom; Budding leaves on the trees outside my window at work; Regeneration across the seasons	124
8	Reflections on the weather	Judgement/observation on the weather, or a reflection on the dynamic weather.	How nice the weather was; dramatic hail storm this morning; The constantly changing weather, from rain to bright sunshine and back	72
9	Beauty/appreciation/wonder of a particular landscape or aspect of nature	Items which refer to beauty or a specific landscape the person appreciates. Expression of the wonder of nature, or the resilience and diversity of nature.	The beauty of a magnolia tree in someones garden; Mist shrouding the trees first thing in the morning; Cow parsley in the grass verge lining the road for miles on my way home	98
10	Good feelings	Reference to nature creating positive feelings or state of mind.	Walking by the brook at university was very peaceful; The sun was shining, walked past the park, everyone was smiling:); However I could hear the dawn chorus through my open bedroom windows and it immediately lightened my mood	40
11	Other	Statements that didn't fit into themes but didn't form a theme of their own.	A nice house made of wood. The beautiful wood texture and its functions are so great; The threat of rain in the air	20

Table 5.2: Example of labelling for an image.

Image example	Labels	Filtered labels
	plant: 0.98 flower: 0.96 flowering plant: 0.89 flora: 0.79 garden: 0.77 shrub: 0.75 annual plant: 0.69 herb: 0.67 groundcover: 0.65 yard: 0.59	plant flower flora garden shrub herb groundcover yard

5.3.4 Image analysis

When the users were prompted to insert an observation about the good things in their environment, they had the option to also take a picture. The approach undertaken for analysing them was object recognition. We used the Google Cloud Platform, namely the Google Cloud Vision API [93]. For each image, we identified a set of associated labels and their corresponding scores. An example is provided in Table 5.2. Afterwards, we carried out a frequency analysis and counted all uniquely identified labels for all images, and for each of the two groups. Furthermore, we filtered the labels in order to reduce their number and lower the amount of redundant information. This was done in the following way: for each image, its set of labels was analysed; if any of the labels contained another label, the contained label was discarded; the explanation for this action is that the 'shorter label' is the 'parent' of the composed label. Considering the example in Table 5.2, the labels 'flowering plant' and 'annual plant' would be discarded after the compression, with only the label 'plant' remaining. An additional frequency count was carried out. Further compression of the resulted labels with similar meanings (e.g. 'flower' and 'flora') would be possible with specific dictionaries of words (such as WordNet) or by manual categorization.

5.3.5 Time analysis

One point of interest in this work was to estimate the time that the participants spent in nature. For this, additional filtering of the location data was carried out as described in the following. For all the data points within green spaces, for each area and user, it was checked if two consecutive location points in a day were recorded within at most five minutes from each other. If the two have an in-between time gap of more than five minutes, it is considered that the user did not spend the time in the area. This assumption is due to the fact that while in a green space, the app should record the user's location more often. Afterwards, associated timers are increased accordingly to count the time spent in the green space, the number of visits to different areas, as well as the number of days in which the users were actually monitored. This analysis was done for all users and green areas under consideration. Subsequently, the data were further processed and used for generating multiple overviews that give an insight into total dwelling time for different parks, most popular green spaces etc.

5.4 Dataset characterization

In this section, we provide an overview of the dataset. It was compiled based on the demographic information provided by the users when registering, as well as from the controlled input entries as part of the interaction with the app throughout the study.

5.4.1 Demographic description of the participants

Figure 5.2 depicts the age distribution of the participants. The age ranges between 18 and 72. We grouped the users into three classes. This was done to establish how different categories interacted with green spaces, considering young people (age 18 to 35), middle-aged people (age 36 to 53) and senior people (age 54 to 72 years). Each class has the same age range (18 years). It can be noticed that the young people group was considerably larger, possibly due to a greater digital engagement in this category. Out of the 1782 participants, 849 were females, 489 were males, and the rest preferred to not disclose the information/other. For the analysis, we focused on the users who identified themselves as females (64.64%), or males (35.36%). Furthermore, the results for the investigations considering age or gender were normalized in order to reduce the bias and to achieve directly comparable results.

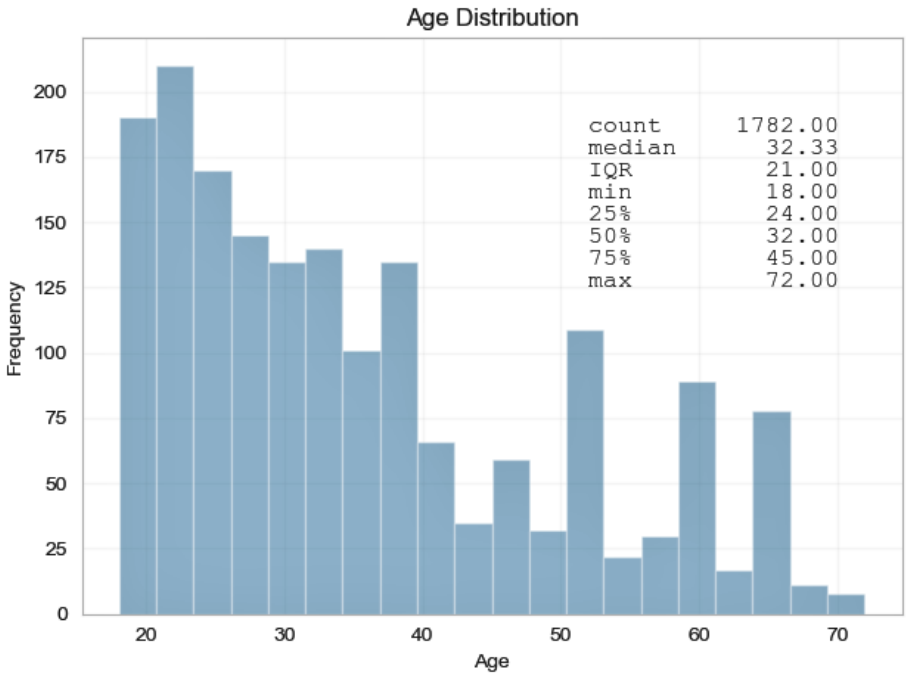


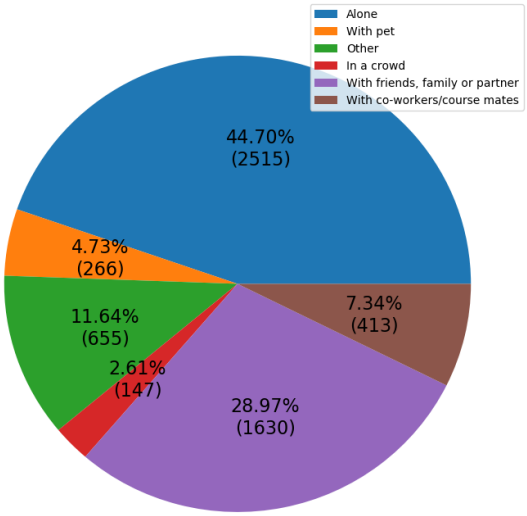
Figure 5.2: Age distribution of the sample dataset.

5.4.2 Insights on the interaction between the participants and Shmapped

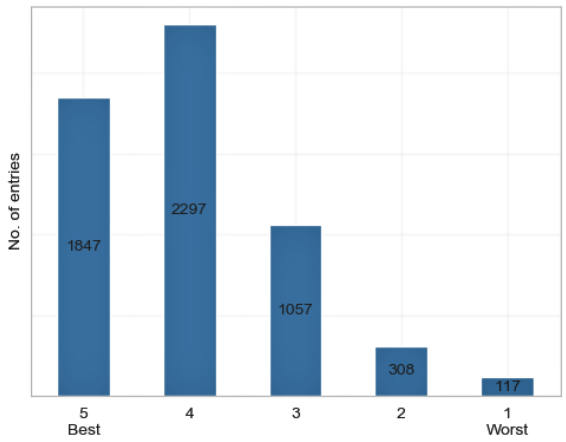
The participants in the study were also asked a few questions with a fixed answer with the daily prompt. The aggregated answers to two of them are depicted in Figure 5.3. One was related to whom they were with. The distribution of social interaction showcased in Figure 5.3a is based on 5626 entries. The vast majority were either alone or with friends and family. The *Other* group includes either a multiple option selection, or free-text comments. The most popular multiple selection was *with friends, family or partner*, and *pet*. The second question tackled giving a grade to their experience within the surrounding area in terms of their feelings. As can be observed in Figure 5.3b, the vast majority of interactions had a positive outcome. Figure 5.4 presents part of the study area as a heat map of the density of the aforementioned grades. The colour scale evolves from blue for a medium grade, to red for a high grade.

5.5 Features noticed by the users

In order to find out which elements of nature get the attention of the users, we analysed the observations data, namely the text entries and the uploaded pictures.



(a) Who are you with?



(b) How would you rate your experience on a scale from 5 (positive) to 1 (negative)?

Figure 5.3: Participants' aggregated responses to two questions.

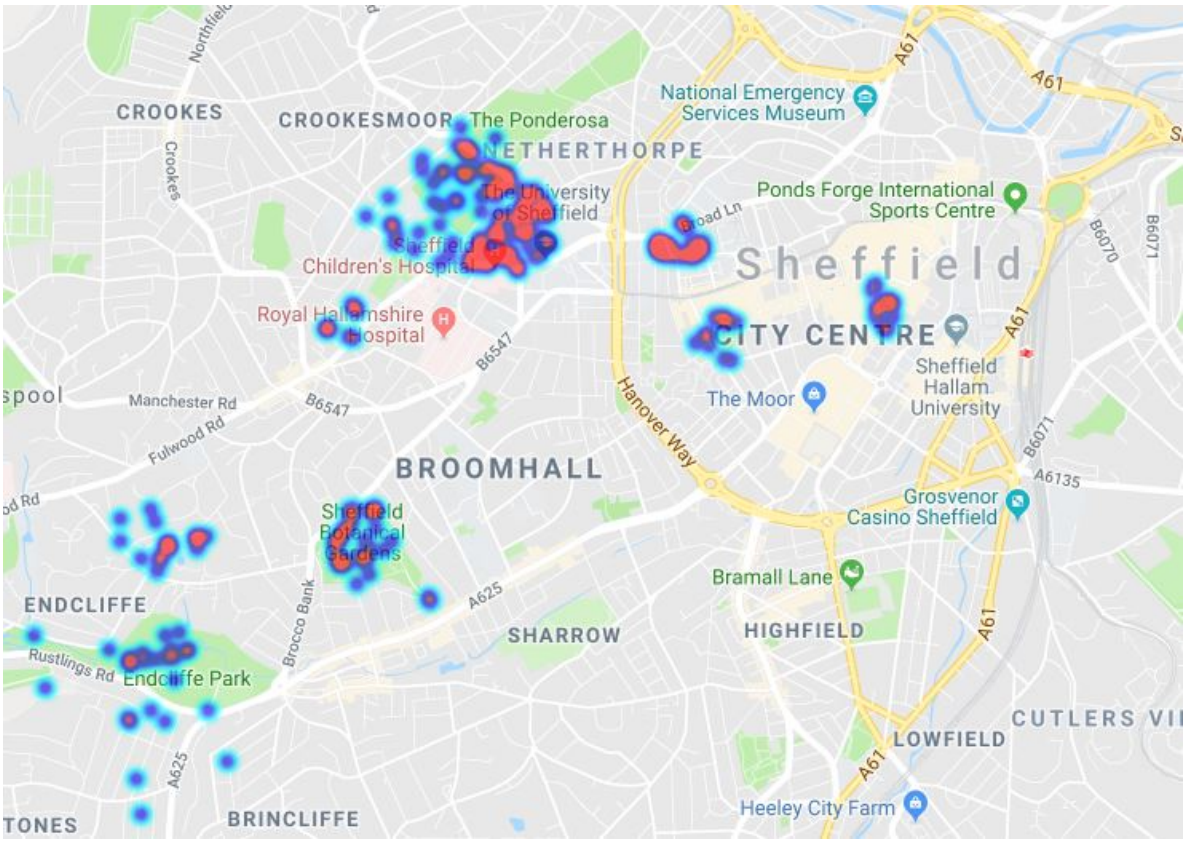


Figure 5.4: Heat-map representing the density of the users' feelings and the associated grades. The scale varies from blue (medium) to red (high).

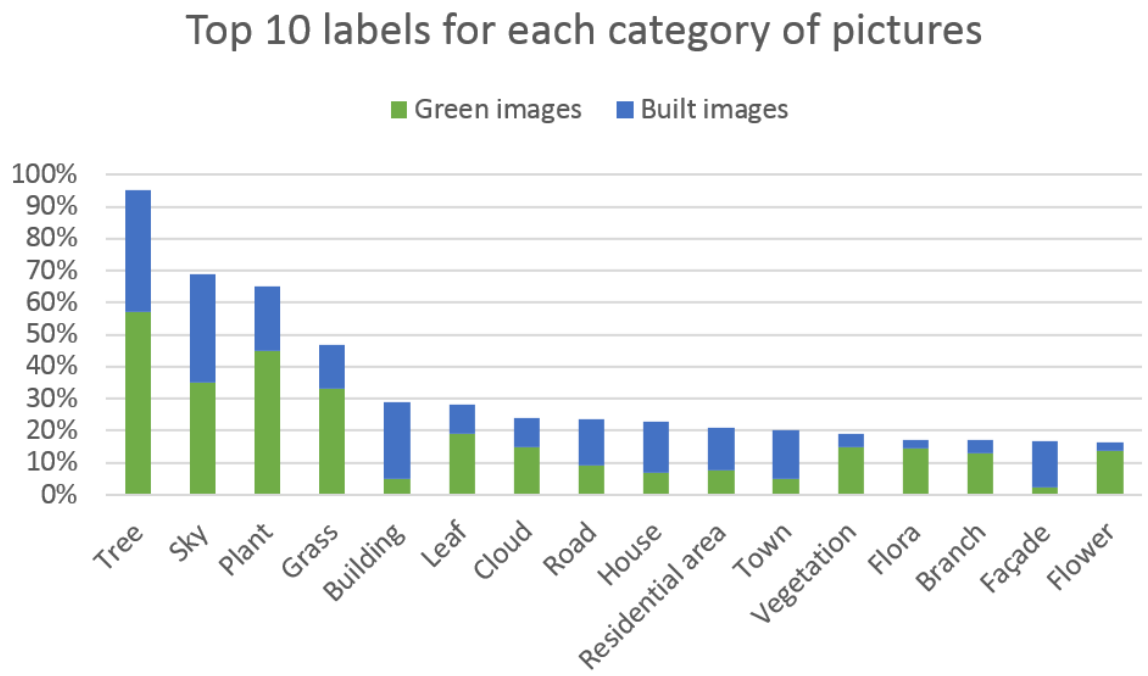


Figure 5.5: Top 10 labels for each category of images.

5.5.1 What do the images say?

For the images, we filtered the identified labels and did a count analysis as described in Section 5.3.4. Table 5.3 gives an overview of the number of labels for the two participant groups. The column *No. of labels* gives the total number of labels identified across all pictures, while the following column tells us how many of those labels are actually unique. After we apply the filtering described in Section 5.3.4, we can see that these numbers reduce.

Table 5.3: Number of labels for participant categories.

User group	**No. of labels**	**No. of unique labels**	**No. of labels after filtering**	**No. of unique labels after filtering**
Green	9610	804	8450	676
Built	5630	640	5012	530

Afterwards, the filtered unique labels were used for continuing the analysis. We chose the first ten most encountered labels for each of the two groups and looked at the overlap between them. To be able to have a fair view of the overlap, we first normalized the number of occurrences of a label by dividing this number by the total number of pictures in the category.

Figure 5.5 depicts the overlap. The x-axis identifies the labels, whereas the y-axis gives the percentage describing the presence of a label in the pictures. It can be observed that 4 of the top 10 labels are common for the two groups: *tree*, *plant*, *sky* and *grass*. As a result, the figure only has 16 labels described. For each of the labels at the top of one group, we checked if a corresponding value exists in the other group. It is interesting to note that the highest ranking label is in both cases *tree*, while the third for the green and the second for the built is *sky*. This shows that despite the group that the users belonged to, nature is salient and meaningful to people. Furthermore, trees and sky are natural elements which are the easiest to encounter in a city.

For the labels that are not in the top 10 for both groups, we can notice a differentiation based on the category, with built images containing building structures such as building, house etc., whereas the elements for green images include flora, flowers etc. The total count of the represented labels (for the top 10) in the green category is 2694, while for the built it is 1317. This represents approximately 32% of the total for the former and 26% for the latter, respectively. A better view could be obtained with the compression of synonymous labels in just a few clusters. However, this would require a dictionary for each cluster or a human expert for manual categorization of the labels.

5.5.2 What does the text say?

In order to get a first understanding of the textual observations, we use the text clustering API of [116], with k equal to 40 as explained in Section 5.3.3. The clusters obtained from a run of the algorithm are presented in Table 5.4, which contains the corresponding number of observations for each cluster and an example from the dataset. A partial visualization of the clusters using a simple technique is shown in Figure 5.6. Some observations mentioned specific parks, information which could be used for connecting the comments with the location of the citizens. Clusters 32 and 34 include observations about the parks of Sheffield, such as Weston Park, Meersbrook Park and Hillsborough Park. As mentioned throughout the chapter, some analysis pertaining to the subjective data and the green spaces could not be carried out given that only 418 observations out of 4225 were recorded at the time of the observation. The users were allowed to record their observations at the end of the day. This was supposed to be an advantage to give people flexibility and convenience and allow for more entries to be recorded. However, since most of the observations could not be tagged to a location, we could not carry out analysis specific to mapping locations with observations.

The clustering provides insight into the type of activities that people were engaging in.

Table 5.4: Example of text clustering, considering k = 40.

Cluster no.	**Dominant term**	**No. of observations**	**Example of text observation**
0, 7	Walk	186	Went for a walk to devonshire green
1	General	1324	Shepherd wheel moss
2	Love	152	Flowers are lovely
3	Heather	20	Heather covered in snow
4	Flowing	7	Fast flowing river
5	-	1	-
6	-	1	-
8	-	1	-
9	Sky	126	The sky when not fully dark
10, 27	Tree	395	I heard the wind in the trees
11	Field	52	Sheep in the fields
12	Sunset	37	The sunset when i woke up was beautiful
13	Leaves	112	Rain droplets on leaves
14	Snowdrops	19	Snowdrops are starting to appear
15	Autumn	54	I admired the autumn leaves on the trees
16	-	1	-
17	City	29	City centre greenery in the rain
18	Green	122	Green grass instead of brick or concrete greys
19	River	68	Light dappled on the river
20	Garden	192	Insect life in our garden
21	-	1	-
22	Plant	57	The plants in the court yard of Brood Cafe
23	Beautiful	91	Beautiful flat landscapes as I travelled back into York
24	-	1	-
25	Weather	48	Nice weather, breezy not rainy and not too cold
26	Flower	124	My honeysuckle flowers coming out
28	See	158	Saw a heron in flight
29	Birds	195	Loads of birds in the park
30, 38	Morning	149	Birds making noises in the morning
31	Singing	55	Birds singing in the trees
32, 34	Park	302	Nice park (weston park)
33	-	1	-
35	Peak district	41	Beautiful views over the peak district
36	Nest	9	Saw a nest of birds in a big tree
37	-	1	-
39	Duck	93	Ducks eating carrots is pretty awesome

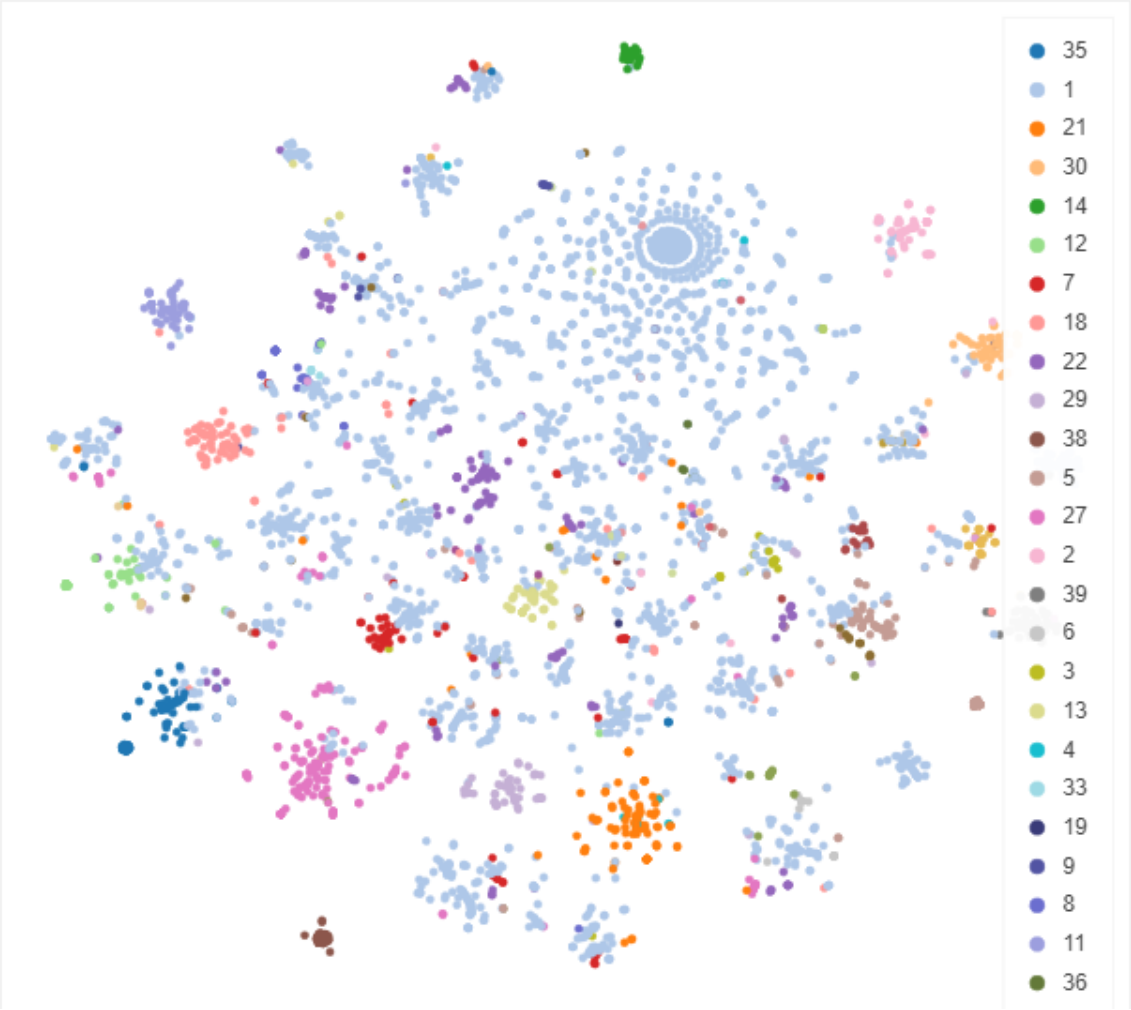


Figure 5.6: Clusters produced from k-means clustering (k=40) of textual observations. Legend captures 25 clusters.

Clusters 0 and 7 correspond to walking activities. A relation with the 'whyThere' field was noticed. For these specific text observations (Cluster 0 and 7), the answer 'Walking' was selected as the response to the question of why the people were there. Other reasons included travelling and exercising. The clusters with only one observation count contained fairly long text which could fit into multiple themes. Some other clusters give us an insight in relation to the biodiversity of the park - birds, ducks, nests, flowers etc. The category *tree* has the highest number of cluster elements for a specific dominant term and is present in two clusters (Clusters 10 and 27). It is interesting to note that in the case of the image analysis, the highest count for the extracted labels was also corresponding to *tree*.

We can see a certain level of correlation between the clusters and the themes recurring in

the [188] study and Table 5.1. There is clustering around colour, with most of the comments being about the green colour of leaves or grass. We can see the effect of weather on different elements, as well as reflections about the weather, in most of the observations containing the word 'morning'. There is also the beauty/appreciation/wonder topic in the clusters about love and beauty. Most of the clusters about animals were referring to the 'animals being active in their habitat'. Some clusters hint at the specific actions that occur naturally in the environment and which people notice: i.e. in cluster 4 ('flowing') people are mostly observing how the river is flowing, and in cluster 31 ('singing') the depicted activity is concerning birds singing on the trees.

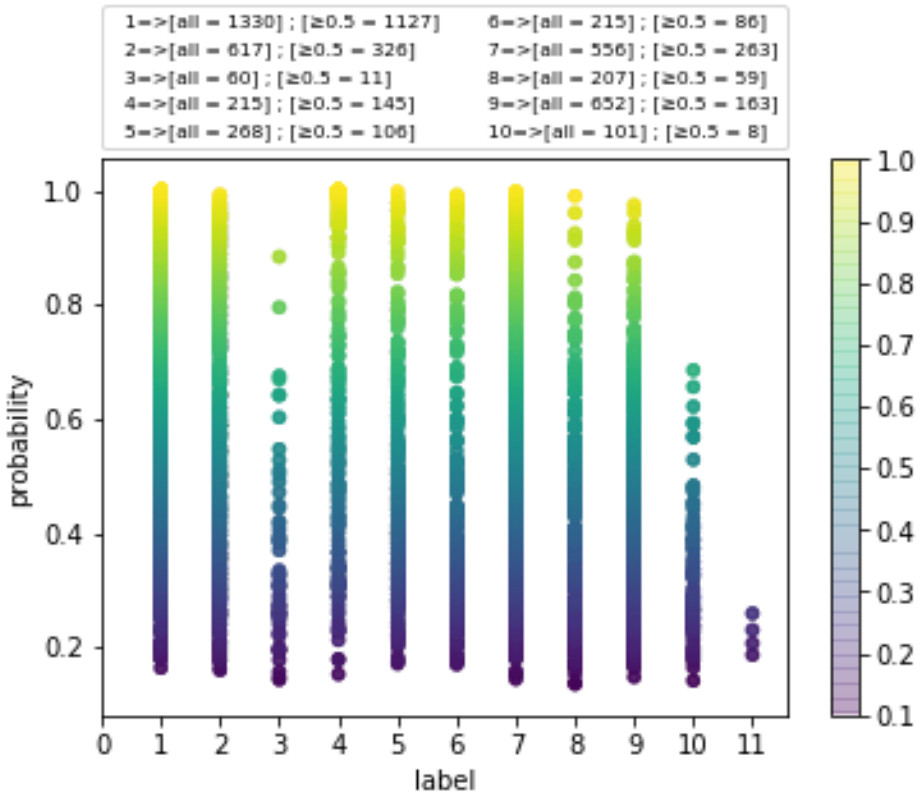


Figure 5.7: Classification of the textual observations into the themes of Table 5.1 with the FastText algorithm.

The next step for the text analysis includes the mapping of the users' comments to the themes in Table 5.1 as explained in Section 5.3.3. The results can be seen in Figure 5.7. The "specific aspect of nature" theme (label 1) happened to be the dominating theme regardless of the used threshold. For a threshold of above 50%, the "animals being active in their habitat" theme (label 2) is the second highest. In this study, the top 5 themes with a probability above

50% interestingly correspond to the top 5 themes of the study in [188] gathered by a traditional, and time-consuming approach to content analysis. The present analysis demonstrates that automated approaches to content analysis are possible. However, unlike their study which has the "sensations of nature" theme as the dominating theme, this study has the "specific aspect of nature" as the top theme.

Figure 5.8 shows the result obtained from classification for the various age groups. Themes 1 and 9 happen to be the most popular in each group, as expected from the general classification. For the younger group, there is less interest in the activity of animals in their habitat than in other age groups, as growth and temporal changes appear to be more interesting to them. In Figure 5.8, the female and male gender seem to vary only slightly, with the females scoring only a little higher for some themes like the sensations of nature, colour and beauty. In summary, understanding the good things in nature informs the design of future interventions to engage and connect people with nature for their well-being, for example, by prompting people to notice trees and birds, or adapting prompts based on gender and age. Future developments could allow real-time text analysis to vary the prompts away from aspects that are being frequently recorded, or towards those known to be associated with improvements in well-being.

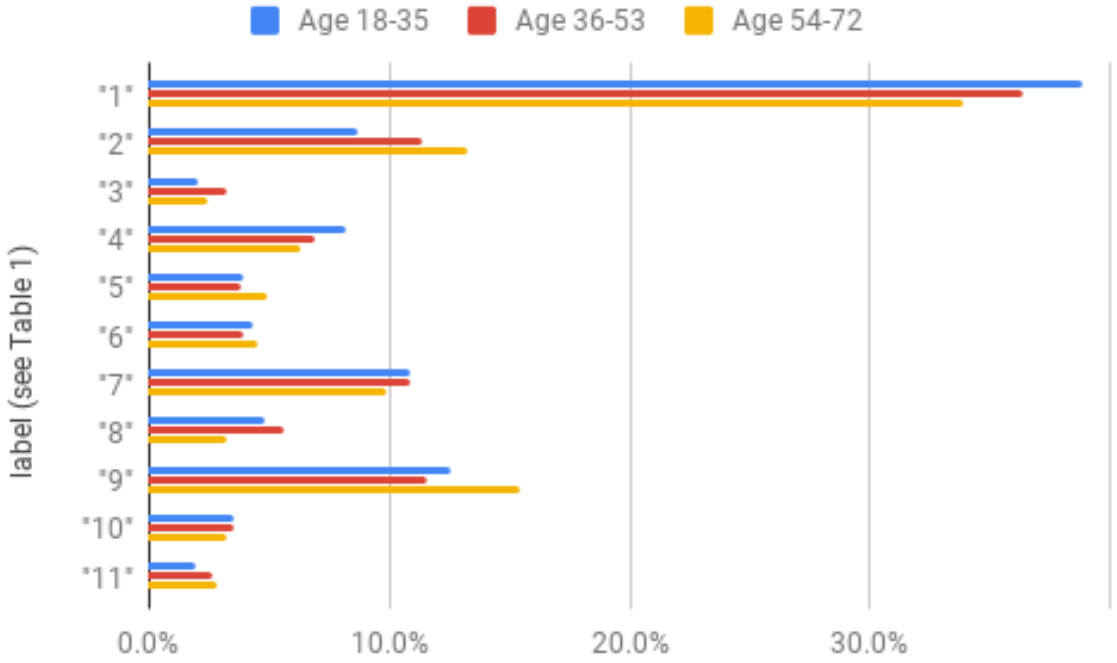


Figure 5.8: Age classification of textual observations into the themes of Table 5.1.

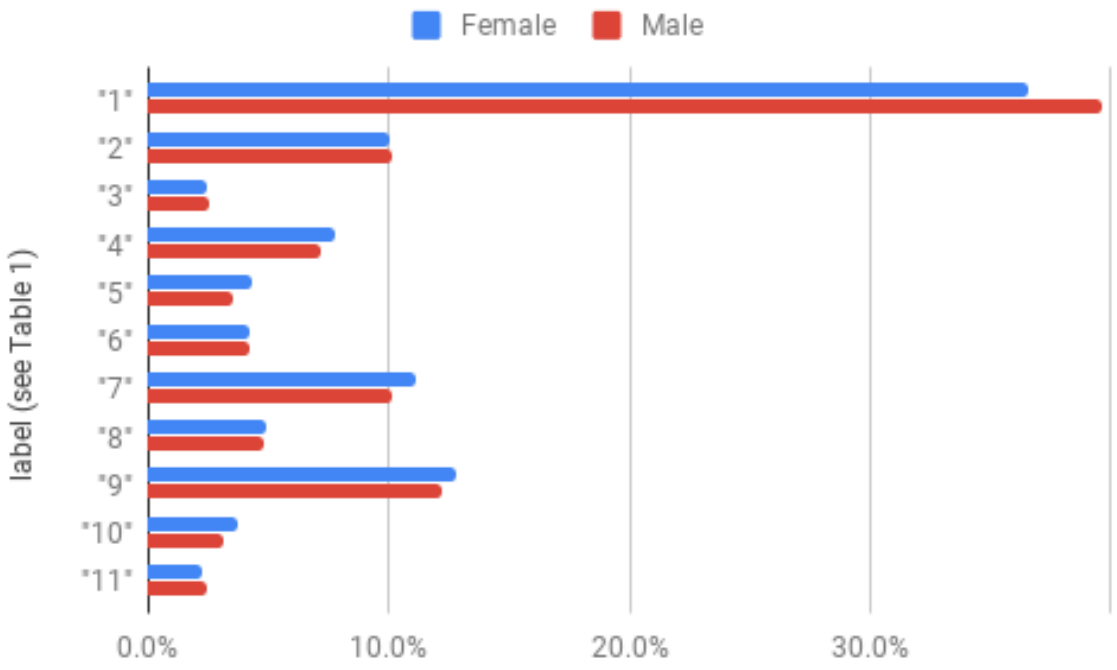


Figure 5.9: Gender classification of textual observations into the themes of Table 5.1.

5.5.3 How do image and text correspond?

By looking at the results from the text and image analysis, we can see that the most encountered label in both cases is *tree*. Furthermore, for the green users, 4 of the top 10 image labels have a direct correspondent in the identified clusters, namely *sky*, *tree*, *leaf* and *flower*. If we were to look at the other unique labels identified for the images outside the top 10, we would find other direct correspondents for some clusters, such as park, city, field etc. It is important to note that we can link image and text based on elements of nature that are rather static and do not involve movement, emotion, action, nor a specific time of day. This is due to the static nature of an image, which makes it hard for an algorithm to pick dynamism, or the emotion felt by a person. These aspects can be expressed more easily through text. As an experiment, we used the technique for the text classification described in Section 5.3.3 for labelling the set of labels for each image. In other words, each set of labels associated to one image was treated as a text observation. The result was overwhelmingly "label 1" (approximately 90%), which represents a specific part of nature. This is not surprising and just reinforces the idea mentioned above regarding the rather static nature of an image.

5.6 Time spent in green spaces

This section presents the results from the carried out time analysis described in Section 5.3.5. We report on the top green spaces, and the top active users, based on the average time spent in nature. Furthermore, we investigate park utilization for age and gender groups.

5.6.1 Top users and parks based on average time spent in green spaces

Tables 5.5 and 5.6 provide an overview of the top users and the most popular parks, sorted by the daily average time spent in a green area.

Table 5.5: Average time spent in parks, by the user.

User	Period of study	Tracked days	No. visits	Visits per day	No. parks	Total time	Avg. daily time	Avg. visit time
1	35 Days	30	106	4	10	5 days 20:40:01	04:41:20	01:19:37
2	70 Days	23	64	3	14	2 days 00:31:09	02:06:34	00:45:29
3	38 Days	20	47	2	7	1 days 07:22:15	01:34:07	00:40:03
4	8 Days	7	25	4	6	0 days 08:17:29	01:11:04	00:19:54
5	11 Days	11	48	4	11	0 days 12:54:22	01:10:24	00:16:08
6	43 Days	19	37	2	8	0 days 21:55:20	01:09:14	00:35:33
7	69 Days	24	37	2	7	1 days 03:19:41	01:08:19	00:44:19
8	114 Days	41	168	4	8	1 days 20:00:32	01:04:24	00:15:43
9	122 Days	74	154	2	26	3 days 04:23:51	01:01:57	00:29:46
10	24 Days	16	94	6	25	0 days 16:18:26	01:01:09	00:10:25

An important mention for the time analysis is that users could use the app for a longer time than the set study periods. Because of this, the total time spent by participants in nature cannot be directly compared. To this end, we consider the average time spent. In the case of Table 5.5, the column *Period of study* identifies the number of (consecutive) days that the users were part of the study, whereas the column *Tracked days* indicates the number of days in which the participants were actively using the app and had their location data recorded; this is to say that the database contained associated entries for the *Tracked days*.

When analysing Table 5.5, it is important to note that participants interact on average with a number of 7 parks. This highlights that, throughout the day, most people interact with a variety of green spaces. Therefore, it is important that citizens are provided with a large selection of green areas in terms of number, diversity, size, and location, rather than just a few large suburban parks.

To compute the average daily time that a user spends in a green area, we consider the *time spent* values from the days in which the participants interact with a park, namely the *Tracked days*. A user spent on average around 20 minutes a day interacting with green spaces. The top 10 participants shown in Table 5.5 spent a higher amount of time in nature. Most of them also interact with a larger than average (7) number of parks. A peculiar case is represented by the top user who spent on average more than 4 hours a day in green areas. Further analysis of the corresponding data indicates that user 1 spent almost all the time in a park. This could suggest that user 1 has a specific reason for interacting with the park, such as a job (examples could include park maintenance staff, dog-sitter, fitness instructor etc.).

Table 5.6: Average time spent inside green spaces, by the park.

No.	**Park**	**Tracked days**	**No. visits**	**No. users**	**Visits per day**	**Visits per device**	**Total time**	**Avg. daily time**	**Avg. visit time**
1	Endcliffe Park	124	358	68	2.89	5.26	10 days 19:28:10	02:05:33	00:43:29
2	Whiteley Woods	71	111	23	1.56	4.83	2 days 20:24:10	00:57:48	00:36:58
3	Weston Park	149	807	170	5.42	4.75	5 days 19:31:05	00:56:11	00:10:22
4	Botanical Gardens	97	191	39	1.97	4.90	2 days 22:46:01	00:43:46	00:22:14
5	Ponderosa Park	82	231	46	2.82	5.02	2 days 00:13:41	00:35:17	00:12:32
6	Hillsborough Park	52	165	29	3.17	5.69	1 days 02:24:19	00:30:28	00:09:36
7	Hallam Square	117	287	56	2.45	5.13	1 days 09:53:09	00:17:23	00:07:05
8	Crookes Valley Park	90	246	76	2.73	3.24	0 days 23:20:11	00:15:33	00:05:42
9	St. Georges Lecture Park	109	310	76	2.84	4.08	0 days 20:05:13	00:11:03	00:03:53
10	Peace Gardens	135	334	91	2.47	3.67	1 days 00:09:48	00:10:44	00:04:20

Table 5.6 offers an insight into the top parks from the point of view of the average time spent inside. Most parks on the list are fairly larger parks located towards the city outskirts. This explains the top average visit and daily times, as people tend to rather visit and spend

more time in the larger parks, as opposed to the smaller parks in the city centre which are often used for passing by or through on the way to work, a shop etc. To complement the information in the table, we use heat maps (also known as density maps) to illustrate the interaction that citizens have with the parks in terms of itinerary and the most commonly used paths.

Figure 5.10 depicts the participants' interaction with *Endcliffe Park*, the park ranked number 1 in Table 5.6. The heat map colours vary from green (low number of location points) to red (high number of location points). The red paths correspond to the actual built paths of the park, identifiable in the picture by the light coloured thin lines. The green location points and paths are observed mostly in the spaces without paths where users walk freely, such as on the grass, field etc. This type of visualization allows for the identification of the most used paths within a green space, as well as the less explored areas and ways. This information can be pivotal for the local authorities and public administration when deciding on new interventions, arrangements and upgrades for the park. Making use of this data allows for the impact of some changes to be maximized.

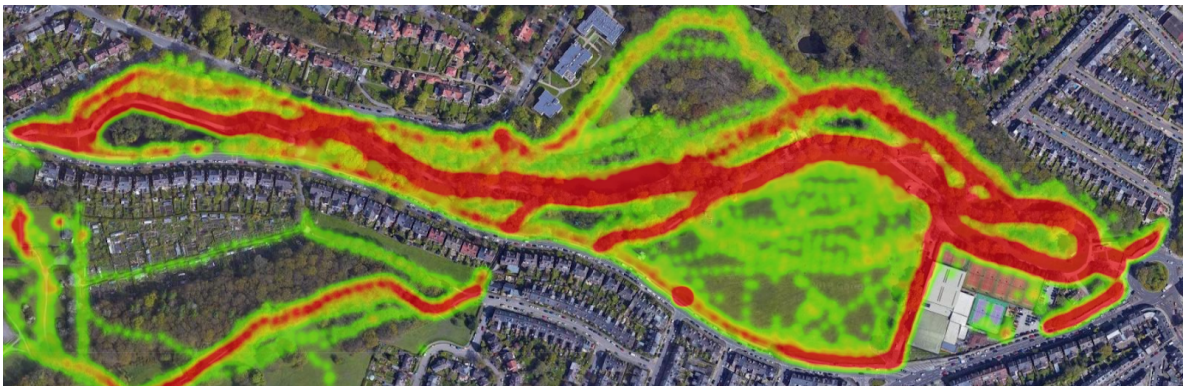
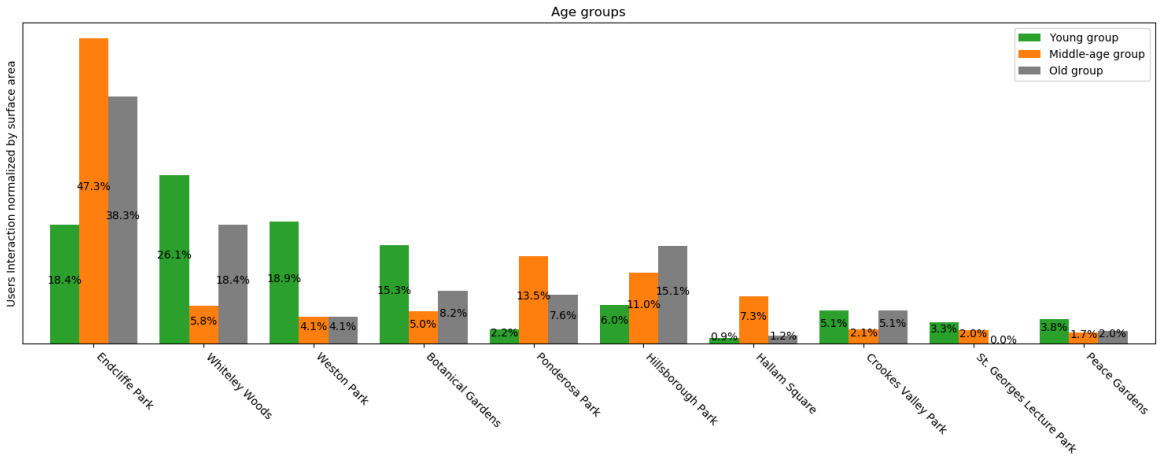


Figure 5.10: Endcliffe Park utilization based on the concentration of location points (green - low number, red - high number).

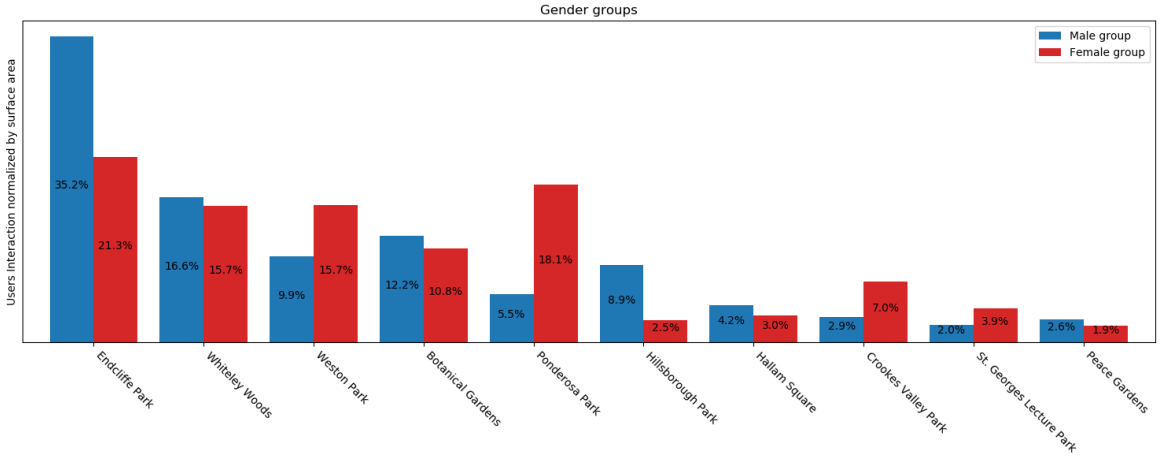
5.6.2 Age and gender distribution in park utilization

In this subsection, we offer an insight into the interaction of the users with the top 10 parks described in Table 5.6, while considering their age and gender. Figure 5.11a shows how different age groups, namely the young group (18 to 35 years old), the middle-aged group (36 to 53 years old), and the old group (54 to 72 years old) spent their time amongst the 10 parks. It is noticed that the middle group preferred to spend more time in the fairly large parks of Endcliffe Park, Ponderosa Park, and Hillsborough Park. The younger participants split their

time mainly amongst Endcliffe Park, Whitley Woods, Weston Park, and the Botanical Gardens. The older group favoured Endcliffe Park, Whitley Woods, and Hillborough Park.



(a) Age groups interactions with top 10 most visited green spaces.



(b) Gender groups interactions with top 10 most visited green spaces.

Figure 5.11: Age and gender groups interactions with top 10 most visited green spaces. The percentages are computed only on the samples in these top areas. The amount of interaction decreases from left to right.

Figure 5.11b shows how female and male participants in the study visited the top 10 parks. There are a few parks where there is a large difference in time spent in a specific green area between the two groups. Endcliffe Park and Hillsborough Park benefited from a higher presence of male users, while the female participants spent more time in Weston Park, Ponderosa Park, and Crookes Valley Park.

This basic analysis showcases how the data collection methodology of this study, and the

use of machine learning and data science techniques, can help to easier extract valuable data from social sciences studies. For this study case, local authorities and public administration can use the data to inform the design and provision of urban green spaces. More detailed analysis with specialized teams can delve into different park characteristics and their relationship with age and gender utilization, as well as the well-being of the participants.

5.7 Comparison between objective and subjective interaction

Initial analysis for this project included looking at the interaction between the participants and the green areas by considering the app utilization, namely the number of observations. Top parks were identified for both different age groups, and genders, similar to Section 5.6.2, but using number of observations instead of time spent. This introductory work was presented in [67]. In this section, we expand that initial work by comparing the subjective data based interaction (observations) to the objective one (location data). As previously, the data is subdivided in accordance with the demographic characteristics of the participants, i.e. age and gender. For the top parks based on the overall subjective interaction (density of user observations), we include for comparison the density of location points. The resulting graphics are presented in Figure 5.12 for the different age groups, and in Figure 5.13 for the genders. As opposed to the results in Section 5.6.2, in this case, the ranking is based on the density of observations, and not on the time spent in the park. The parks represented on the x-axis are ordered based on the overall density of the subjective data. Therefore, the first park on the left has the highest density, with the one on the very right having the lowest corresponding density.

It is interesting to note that, for many green areas, the subjective and objective data differ. In these cases, the participants who engaged more with a park from the point of view of time spent there, did not have the same amount of interaction with the app. For the green area of Peace Gardens, the objective data interaction exceeds the subjective one for all age and gender groups. Probably this is due to the fact that the park lies within the very city centre, surrounded by multiple office spaces and cafés. It is likely that many people regularly pass by or through the area as part of other activities, such as meeting with friends, going to/or leaving work etc. The position, accessibility and characteristics make Peace Gardens a green space of high objective interaction with the citizens. At the same time, it is likely that fewer people engage with the app to input observations because of other pressing matters and activities. A

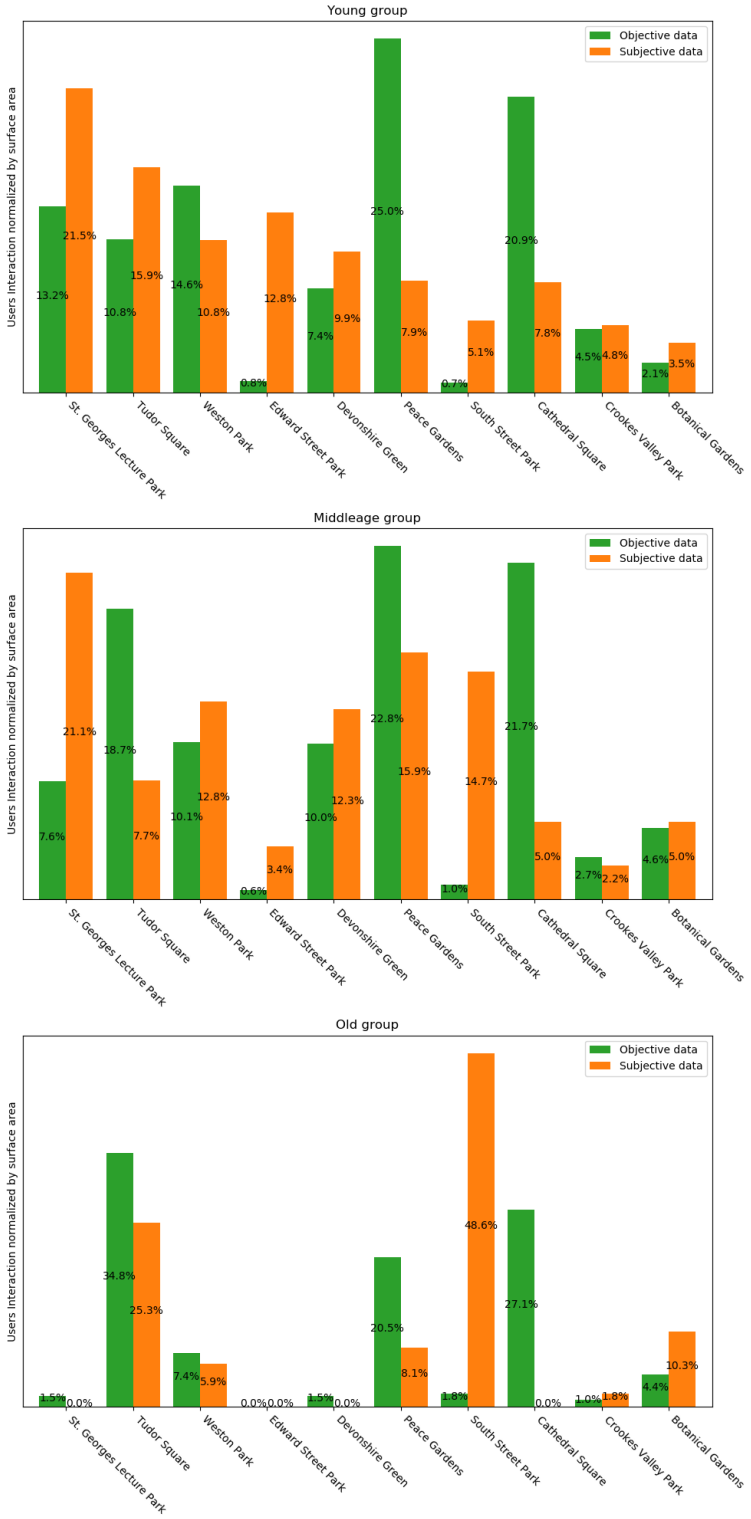


Figure 5.12: Comparison of objective (green) and subjective (orange) interaction for the top 10 most visited green spaces divided for the three age groups: young (18 to 35), middle-aged (36 to 53), and senior (54 to 72). The top 10 is according to the subjective data (density of observations). The percentages are computed only on the samples in these top areas.

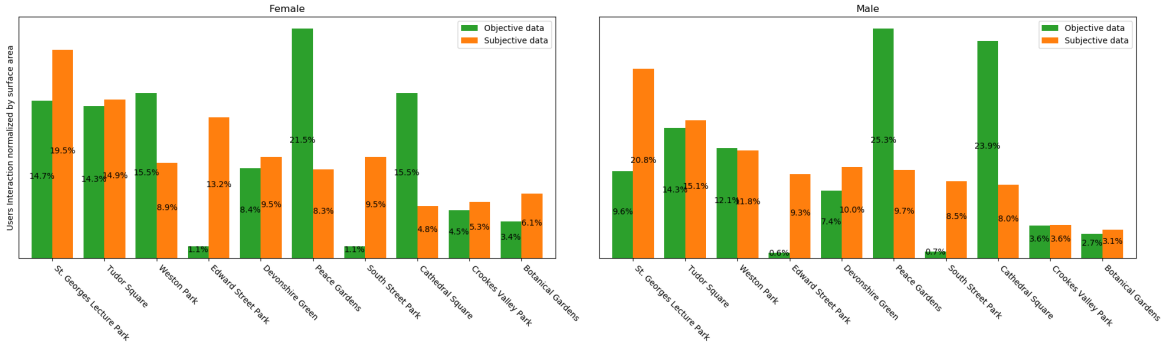


Figure 5.13: Comparison of objective (green) and subjective (orange) interaction for the top 10 most visited green spaces for the two gender groups. The top 10 is according to the subjective data (density of observations). The percentages are computed only on the samples in these top areas.

reversed situation is encountered in the South Street Park, where the amount of subjective interaction greatly exceeds the location data across all categories. In comparison to Peace Gardens, the South Street Park is a fairly large green space located outside the Sheffield central area, behind the train station of the city, neighbouring a residential area. Considering these characteristics, it is reasonable to assume that a subjective interaction is triggered while people return home, walking along the street by the park. The collection of location points is limited, as most people do not go through the park, but rather notice the green features while prompted by the app from the margin/distance.

In Figure 5.12, one can observe a difference between the different age groups in terms of overall interaction. St. George Lecture Park lacks both subjective and objective interaction data for the old group. However, there is high engagement for the young group, and medium engagement for the middle group. This can be explained by the fact that the park is part of the university site. Therefore, younger people are more likely to interact with the area.

5.8 Conclusion

The aim of this work was to present how data science and machine learning techniques can be used in social science studies in order to maximize insight gain. To this end, we made use of a pilot study, in which the problem at hand consists of understanding the interaction of citizens with green spaces. By making use of smartphones, data about the interaction is collected as it happens. This allows for monitoring of the exact moments in time. The data can be split into two main categories: subjective and objective. This allows for multiple levels of analysis and

comparison. Problems that occur are incomplete data, lack of data, or erroneous data, which can negatively impact statistical significance.

In this work, we looked to better understand the behaviour of the inhabitants of Sheffield, UK. We cleaned and pre-processed the initial dataset and proceeded towards a more in depth analysis. The main types of data we used include text observations, images taken by the users, as well as participants' location points in the geo-fenced green spaces.

Machine learning techniques allowed us to automatically extract the main topics of interest in the text, as well as to categorize the comments in 11 classes as described in previous research using traditional methodologies, thus, showing how content analysis can be automated while allowing for similar results. Furthermore, these methods enabled us to identify features noticed by the users based on the pictures they uploaded. The output from the text and the images were compared, and some similarities could be pinpointed regarding what the citizens notice as the good things in their green surroundings. The location points provided us with the time spent in various green spaces and allowed us to identify the most active users, as well as the most popular parks. In order to broaden the analysis, we compared the interaction based on inputted observations with the interaction based on automatically recorded location points when in green spaces. The challenge in this type of study comes from fusing the information and finding the relationships between different variables.

The lessons learned from undertaking this work allowed for a better understanding of how to carry out large-scale social studies and which techniques can be employed in order to target results from both objective and subjective data. In order to increase the value that this type of research project brings, I believe that a partnership between public institutions and researchers from both social science and computer science fields should exist. Some limitations we encountered throughout the project included a lack of understanding/insight into certain human behaviour, missing data, not being able to influence the original study design, and the lack of impactful change. By having joint teams, social study cases could be designed while also taking into account different technical details useful for further data analysis. Furthermore, with the involvement of public institutions, more data could be collected, and impactful changes could be made to the life of citizens based on the result analysis done by the researchers.

Chapter 6

Conclusions and Future Work Directions

This chapter provides a summary of the main contributions of this thesis, draws conclusions, and discusses possible avenues for future work.

6.1 Conclusions and thesis contributions

The Internet of Things (IoT) continues to expand both in terms of the number of connected devices, and the amount of data it generates. Furthermore, it encompasses more and more application domains, including smart cities, smart industry, and smart healthcare. The challenge that remains is to manage and interpret all the generated data and obtain valuable insight.

The goal of this thesis was to carry out a variety of data analytics within the context of IoT in the Edge Cloud setting. The focus was on smart anomaly detection, and data imputation within the edge environment, as well as a real social case study. The novelty and originality that this thesis brings to the research field are focused on a synthesis of the state of the art and novel applications of existing methods and knowledge.

Chapter 3 introduces the reader to the topic of smart anomaly detection in the context of IoT and sensor systems. Anomaly detection is an important problem within the IoT and not only, due to its wide range of application domains. We review state-of-the-art methods pertaining to both the conventional approach (statistical methods, time-series analysis, signal processing, etc.) and the data-driven realm (supervised learning, reinforcement learning, deep learning, etc.), which can be used to detect anomalies in sensor systems. This particular context poses numerous challenges, as one needs to identify the optimal computational-energy-accuracy trade-offs. Furthermore, we consider and discuss the impact and influence of different architectures (Cloud, Fog, Edge) on anomaly detection within the sensor systems scenario.

Moreover, we identify an interesting set of open issues and challenges in the field that do not limit themselves to the development of suitable algorithms but explore the intersection between computing (learning models), communications (efficiency), and engineering (constraints). This piece of work highlights the major role that machine learning has in the problem of anomaly detection in sensor systems, taxonomizes techniques ranging from conventional to data-driven techniques while taking into account also the main architectural models impacting the area.

Chapter 4 is concerned with using machine learning techniques at the edge for the task of data imputation. This is part of the movement of pushing intelligence towards the edge of the network. We consider data imputation to be an essential data-cleaning operation that can be performed at the edge in order to avoid the transmission of corrupted data to the next processing layers of the system. To this extent, we evaluate two machine learning techniques (kNN and missForest) against two statistical techniques (mean and MICE) for the task of real-time embedded data imputation, while considering the following metrics: RMSE, density distribution, execution time, RAM, and CPU utilization. We start from a publicly available pollution dataset, which we impair with both random missing data (non-bursty case), and bursts of missing data (bursty case). We carry out our experimental work on board of both a constrained device (Raspberry Pi 4B with 4GB of RAM) and a laptop. We find out that kNN and missForest outperform the other two techniques, being able to cope with impairment rates of up to 40% for the non-bursty case, as well as being able to recover blocks of up to 100 missing data samples for the bursty case before dropping to the performance level of mean and MICE. Execution times do show that for our considered scenarios, real-time data imputation on board of constrained devices can be achieved. The work in this chapter encourages taking advantage of the edge, optimizing existing processing pipelines in IoT, and building on top of existing smart sensor systems.

Chapter 5 presents a real social study case, representative of the smart city scenario, IoT analytics in an Edge Cloud Setting, as well as how technology can improve day to day life. The goal of this work is to showcase how data science and machine learning techniques can be used in social science studies to maximize insight gain. The IWUN (Improve Well-being through Urban Nature) project is a pilot study that attempts to understand the impact that interaction with nature, in particular urban green spaces, has on citizens. A field experiment was carried out in Sheffield, UK, engaging 1870 participants for two different time periods (7 and 30 days). With the help of a specially developed smartphone app (Shmapped), both objective (sensor information) and subjective data (direct input provided by the users) were collected. Each time a participant was entering designated green spaces, tracking was active

and the corresponding location data was saved. This data was complemented by textual and photographic information provided by the user, either spontaneously or when prompted by the app. With the help of both data science and machine learning techniques, we extract from the collected data the main features noticed by the participants in their interaction with nature, as well as dwelling time and top interaction areas. We identified the main topics of interest in the comments, along a categorization of those in 11 classes from a psychological study which uses traditional methodology. These results were compared to the main features identified in the images, and some similarities could be pinpointed. Furthermore, making use of the dwelling time, top active users and the popular green spaces were discovered. Moreover, we compared the interaction based on subjective data with the one based on objective data. The challenge that arises is fusing the available data and finding the relations between different variables. However, this work showcases the possibilities of integrating technology into large scale social studies.

The work within this thesis helped us provide a multi perspective view of the state of the art, along a novel taxonomy for smart anomaly detection in sensor systems, while considering both conventional and data-driven techniques, as well as different architectural environments; evaluate machine learning techniques (kNN and missForest) against statistical techniques (mean and MICE) for real-time embedded data imputation (on a Raspberry Pi), while considering both bursty and non-bursty missing data in an environmental IoT scenario; showcase how applying data science and machine learning techniques to the data from a real social field experiment can complement the traditional analysis and provide new insight.

6.2 Future work

When discussing future work, we consider a more general direction of taking advantage of the edge by pushing computation to the level of IoT devices, as well as specific research avenues pertaining to the three research topics tackled in this thesis.

In relation to Chapter 3, promising research directions include those within the identified open issues and challenges, namely miniaturization and acceleration of algorithms, improving energy efficiency and security, tackling data heterogeneity, building on different architectural models, or using sensors softwarization. In addition to these general lines of expanding smart anomaly detection in sensor systems, it would also be interesting to carry out an experimental comparative analysis of techniques from both the statistical and the machine learning world for the task of anomaly detection on board of constrained environments, such as the Raspberry

Pi.

In connection to Chapter 4, expansions could be carried out along the following lines: technique optimization for edge-based data imputation, multivariate data imputation, evaluation of new techniques along a variety of application scenarios, as well as computing platforms (other edge devices besides the Raspberry Pi 4), development of new data imputation methods, and the design of a reinforcement learning based system that can choose the best data imputation method for a specific set of circumstances.

The first research direction looks at optimizing the techniques for the conditions of the environmental scenario at hand, as well as making use of time windows for the data imputation step, so that we can minimize execution needs. In our work, all chosen data imputation methods have not been optimized, or tuned in any way, in order to offer a ground truth and bare-bones evaluation. However, we believe that an analysis considering different time windows could help to further bring down the execution time, as well as lead to further increases in accuracy.

Moreover, it would be worth investigating the performance of the edge data imputation methods in the scenario of data missing from multiple data columns (multivariate imputation). In our work, for simplicity, we considered the case where the environmental collected data had missing values only in the ozone dimension. However, in real situations, data would be missing from any of the sensors of the systems. We expect that in this situation, further optimization can be done, so as to take advantage of the possible correlation between different neighbouring sensors. More importantly, we believe that future work should focus on different methods and scenarios in which one can optimize existing processing pipelines by pushing part of the data processing to the edge of the network, as there are many benefits, such as important savings on energy, storage space, and transmission costs.

Furthermore, an interesting avenue for expanding this work would be looking at different types of data collected by sensors, such as health data. The use of different datasets would allow for the expansion and evaluation of the suggested approach to other application scenarios. Moreover, besides different datasets, different computationally powerful IoT devices could be evaluated along the Raspberry Pi 4B, as other application domains could already have promising IoT devices as part of their overall infrastructure.

Finally, a comprehensive comparative evaluation of the most promising machine learning based techniques from past years, for the task of edge data imputation, would prove to be valuable for the research field. Our work could benefit from comparing our chosen algorithms against a multitude of other techniques to discuss advantages and disadvantages, and make recommendations for the appropriate techniques to use, depending on the application scenarios

and technical requirements. Furthermore, this would support the design of novel, ad-hoc algorithms that improve the state-of-the-art methods by looking at the strong and weak points of the compared methods, as well as the design and development of a reinforcement learning based system that can automatically choose the most appropriate methods for the data imputation task based on the application scenario and dataset characteristics.

For the work presented in Chapter 5, interesting research avenues could include the development of smart online chatbots which can dynamically adapt their user prompts based on different research goals or requirements. Also, more processing could be done on the edge devices (smartphones), an example including federated learning. For the former research direction, the first step would be designing a new app that takes advantage of artificial intelligence. In this context, the chatbot that prompts the user for information would be smart, meaning that based on the information it already has, it will decide which questions it should ask in order to maximize the knowledge gain. This is different from the static approach where each user always gets asked the same questions. The aim is to manage the asking of questions in order to build statistical significance and minimize intrusion. This also implies that we are moving from an offline approach to an online one. The data is to be analysed as it comes and depending on the present results and the current statistical significance, the system decides which questions should be addressed to which user. This approach can also be applied to the *green prescription* element of the app, with real-time text and image analysis used to vary the prompts towards those known to be associated with improvements in well-being. Thus, in the future the app may actively stimulate the improvement of well-being based on known causes of well-being variation; work in this direction is only preliminary at the moment. This kind of app fits into the framework of a smart city and can be used for both social studies, and city planning and to improve the quality of life for citizens. It represents a scenario where technology, IoT and artificial intelligence can be used in order to improve current conditions in cities and to implement and monitor large-scale studies.

Along these specific research avenues, we would like to actually encourage future work along the lines of pushing intelligence and computation to the edge, and developing edge-friendly algorithms and computational infrastructure. There are many application domains within IoT which could benefit from this approach, such as smart healthcare, smart cities, Industry 4.0, etc.

Bibliography

[1] M. M. N. Aboelwafa, K. G. Seddik, M. H. Eldefrawy, Y. Gadallah, and M. Gidlund. "A Machine-Learning-Based Technique for False Data Injection Attacks Detection in Industrial IoT". In: *IEEE Internet of Things Journal* 7.9 (2020), pp. 8462–8471. DOI: 10.1109/JIOT.2020.2991693.

[2] E. Adi, A. Anwar, Z. Baig, and S. Zeadally. "Machine learning and data analytics for the IoT". In: *Neural Computing and Applications* 32.20 (2020), pp. 16205–16233. DOI: 10.1007/s00521-020-04874-y.

[3] C. C. Aggarwal and C. Zhai. *Mining text data*. Springer Science & Business Media, 2012. ISBN: 9781461432227.

[4] C. C. Aggarwal. *Outlier Analysis*. Springer Publishing Company, Incorporated, 2013. ISBN: 1461463955.

[5] S. Ahmad, A. Lavin, S. Purdy, and Z. Agha. "Unsupervised real-time anomaly detection for streaming data". In: *Neurocomputing*. Online Real-Time Learning Strategies for Data Streams 262 (2017), pp. 134–147. DOI: 10.1016/j.neucom.2017.04.070.

[6] E. Ahmed, I. Yaqoob, A. Gani, M. Imran, and M. Guizani. "Internet-of-things-based smart environments: state of the art, taxonomy, and open research challenges". In: *IEEE Wireless Communications* 23 (2016), pp. 10–16. DOI: 10.1109/MWC.2016.7721736.

[7] J. Åkerberg, M. Gidlund, and M. Björkman. "Future research challenges in wireless sensor and actuator networks targeting industrial automation". In: *2011 9th IEEE International Conference on Industrial Informatics*. 2011, pp. 410–415. DOI: 10.1109/INDIN.2011.6034912.

[8] H. N. Akouemo and R. J. Povinelli. "Data Improving in Time Series Using ARX and ANN Models". In: *IEEE Transactions on Power Systems* 32.5 (2017), pp. 3352–3359. DOI: 10.1109/TPWRS.2017.2656939.

[9] M. G. R. Alam, M. M. Hassan, M. Z. Uddin, A. Almogren, and G. Fortino. "Autonomic computation offloading in mobile edge for IoT applications". In: *Future Generation Computer Systems* 90 (2019), pp. 149–157. DOI: 10.1016/j.future.2018.07.050.

[10] M. I. Ali, F. Gao, and A. Mileo. "CityBench: A Configurable Benchmark to Evaluate RSP Engines Using Smart City Datasets". In: *The Semantic Web - ISWC 2015*. Springer International Publishing, 2015, pp. 374–389. ISBN: 978-3-319-25010-6.

[11] M. Z. Alom, V. Bontupalli, and T. M. Taha. "Intrusion detection using deep belief networks". In: *2015 National Aerospace and Electronics Conference (NAECON)*. 2015, pp. 339–344. DOI: 10.1109/NAECON.2015.7443094.

[12] "An overview of next-generation architectures for machine learning: Roadmap, opportunities and challenges in the IoT era". In: 2018, pp. 827–832. DOI: 10.23919/DATE.2018.8342120.

[13] P. Anantharam, P. Barnaghi, K. Thirunarayan, and A. Sheth. "Extracting City Traffic Events from Social Streams". In: *ACM Trans. Intell. Syst. Technol.* 6.4 (2015). DOI: 10.1145/2717317.

[14] S. Ando. "Clustering Needles in a Haystack: An Information Theoretic Analysis of Minority and Outlier Detection". In: *Seventh IEEE International Conference on Data Mining (ICDM 2007)*. 2007, pp. 13–22. DOI: 10.1109/ICDM.2007.53.

[15] M. Antonini, M. Vecchio, F. Antonelli, P. Ducange, and C. Perera. "Smart Audio Sensors in the Internet of Things Edge for Anomaly Detection". In: *IEEE Access* 6 (2018), pp. 67594–67610. DOI: 10.1109/ACCESS.2018.2877523.

[16] M. S. Aslanpour, S. S. Gill, and A. N. Toosi. "Performance evaluation metrics for cloud, fog and edge computing: A review, taxonomy, benchmarks and standards for future research". In: *Internet of Things* 12 (2020), p. 100273. DOI: 10.1016/j.iot.2020.100273.

[17] L. Atzori, A. Iera, and G. Morabito. "The Internet of Things: A survey". In: *Computer Networks* 54.15 (2010), pp. 2787–2805. DOI: 10.1016/j.comnet.2010.05.010.

[18] K. S. Awaisi, A. Abbas, M. Zareei, H. A. Khattak, M. U. Shahid Khan, M. Ali, I. Ud Din, and S. Shah. "Towards a Fog Enabled Efficient Car Parking Architecture". In: *IEEE Access* 7 (2019), pp. 159100–159111. DOI: 10.1109/ACCESS.2019.2950950.

[19] M. J. Azur, E. A. Stuart, C. Frangakis, and P. J. Leaf. "Multiple imputation by chained equations: what is it and how does it work?" In: *International Journal of Methods in Psychiatric Research* 20.1 (2011), pp. 40–49. DOI: 10.1002/mpr.329.

[20] T. Bakıcı, E. Almirall, and J. Wareham. "A Smart City Initiative: the Case of Barcelona". In: *Journal of the Knowledge Economy* 4.2 (2012), pp. 135–148. DOI: 10.1007/s13132-012-0084-9.

[21] I. Bakolis, R. Hammoud, M. Smythe, J. Gibbons, N. Davidson, S. Tognin, and A. Mechelli. "Urban Mind: Using Smartphone Technologies to Investigate the Impact of Nature on Mental Well-Being in Real Time". In: *BioScience* 68.2 (2018), pp. 134–145. DOI: 10.1093/biosci/bix149.

[22] P. Baldi. "Autoencoders, Unsupervised Learning, and Deep Architectures". In: *Proceedings of ICML Workshop on Unsupervised and Transfer Learning*. Vol. 27. Proceedings of Machine Learning Research. PMLR, 2012, pp. 37–49.

[23] H. Bangui, S. Rakrak, S. Raghay, and B. Buhnova. "Moving to the Edge-Cloud-of-Things: Recent Advances and Future Research Directions". In: *Electronics* 7.11 (2018), p. 309. DOI: 10.3390/electronics7110309.

[24] M. Behniafar, A. Nowroozi, and H. Shahriari. "A Survey of Anomaly Detection Approaches in Internet of Things". In: *The ISC International Journal of Information Security* 10.2 (2018), pp. 79–92. DOI: 10.22042/isecure.2018.116976.408.

[25] K. Bonawitz, H. Eichner, W. Grieskamp, D. Huba, A. Ingerman, V. Ivanov, C. Kiddon, J. Konečný, S. Mazzocchi, B. McMahan, T. Van Overveldt, D. Petrou, D. Ramage, and J. Roselander. "Towards Federated Learning at Scale: System Design". In: *Proceedings of Machine Learning and Systems*. Vol. 1. 2019, pp. 374–388.

[26] M. Borova, M. Prauzek, J. Konecny, and K. Gaiova. "Environmental WSN Edge Computing Concept by Wavelet Transform Data Compression in a Sensor Node". In: *IFAC-PapersOnLine* 52.27 (2019), pp. 246–251. DOI: 10.1016/j.ifacol.2019.12.646.

[27] H. Bosman. "Anomaly detection in networked embedded sensor systems". PhD thesis. Technische Universiteit Eindhoven, Department of Electrical Engineering, 2016. ISBN: 978-90-386-4125-6.

[28] H. H. W. J. Bosman, G. Iacca, A. Tejada, H. J. Wörtche, and A. Liotta. "Ensembles of incremental learners to detect anomalies in ad hoc sensor networks". In: *Ad Hoc Networks* 35 (2015), pp. 14–36. DOI: 10.1016/j.adhoc.2015.07.013.

[29] H. H. Bosman, G. Iacca, A. Tejada, H. J. Wörtche, and A. Liotta. "Spatial anomaly detection in sensor networks using neighborhood information". In: *Information Fusion* 33 (2017), pp. 41–56. DOI: 10.1016/j.inffus.2016.04.007.

[30] A. Botta, W. de Donato, V. Persico, and A. Pescapé. "Integration of Cloud computing and Internet of Things: A survey". In: *Future Generation Computer Systems* 56 (2016), pp. 684–700. DOI: 10.1016/j.future.2015.09.021.

[31] A. Bouguettaya, Q. Z. Sheng, B. Benatallah, A. G. Neiat, S. Mistry, A. Ghose, S. Nepal, and L. Yao. "An internet of things service roadmap". In: *Communications of the ACM* 64.9 (2021), pp. 86–95. DOI: 10.1145/3464960.

[32] D. Brauckhoff, K. Salamatian, and M. May. "A Signal Processing View on Packet Sampling and Anomaly Detection". In: *2010 Proceedings IEEE INFOCOM*. 2010, pp. 1–9. DOI: 10.1109/INFCOM.2010.5462154.

[33] S. V. Buuren and K. Groothuis-Oudshoorn. "MICE: Multivariate Imputation by Chained Equations in R". In: *Journal of Statistical Software* 45.1 (2011), pp. 1–67. DOI: 10.18637/jss.v045.i03.

[34] F. Calabrese, M. Colonna, P. Lovisolo, D. Parata, and C. Ratti. "Real-Time Urban Monitoring Using Cell Phones: A Case Study in Rome". In: *IEEE Transactions on Intelligent Transportation Systems* 12.1 (2011), pp. 141–151. DOI: 10.1109/TITS.2010.2074196.

[35] G. O. Campos, A. Zimek, J. Sander, R. J. G. B. Campello, B. Micenková, E. Schubert, I. Assent, and M. E. Houle. "On the evaluation of unsupervised outlier detection: measures, datasets, and an empirical study". In: *Data Mining and Knowledge Discovery* 30.4 (2016), pp. 891–927. DOI: 10.1007/s10618-015-0444-8.

[36] R. Casadei, G. Fortino, D. Pianini, W. Russo, C. Savaglio, and M. Viroli. "Modelling and simulation of Opportunistic IoT Services with Aggregate Computing". In: *Future Gener. Comput. Syst.* 91 (2019), pp. 252–262. DOI: 10.1016/j.future.2018.09.005.

[37] F. Cauteruccio, L. Cinelli, E. Corradini, G. Terracina, D. Ursino, L. Virgili, C. Savaglio, A. Liotta, and G. Fortino. "A framework for anomaly detection and classification in Multiple IoT scenarios". In: *Future Generation Computer Systems* 114 (2021), pp. 322–335. DOI: 10.1016/j.future.2020.08.010.

[38] F. Cauteruccio, G. Fortino, A. Guerrieri, A. Liotta, D. C. Mocanu, C. Perra, G. Terracina, and M. T. Vega. "Short-long term anomaly detection in wireless sensor networks based on machine learning and multi-parameterized edit distance". In: *Information Fusion* 52 (2019), pp. 13–30. DOI: 10.1016/j.inffus.2018.11.010.

[39] R. Chalapathy and S. Chawla. *Deep Learning for Anomaly Detection: A Survey*. 2019. arXiv: 1901.03407 [cs.LG].

[40] V. Chandola, A. Banerjee, and V. Kumar. "Anomaly Detection for Discrete Sequences: A Survey". In: *IEEE Transactions on Knowledge and Data Engineering* 24.5 (2012), pp. 823–839. DOI: 10.1109/TKDE.2010.235.

[41] V. Chandola, A. Banerjee, and V. Kumar. "Anomaly detection: A survey". In: *ACM computing surveys* 41.3 (2009), pp. 1–58. DOI: 10.1145/1541880.1541882.

[42] A. Chattopadhyay and U. Mitra. "Security Against False Data-Injection Attack in Cyber-Physical Systems". In: *IEEE Transactions on Control of Network Systems* 7.2 (2020), pp. 1015–1027. DOI: 10.1109/TCNS.2019.2927594.

[43] S. Chauhan and L. Vig. "Anomaly detection in ECG time signals via deep long short-term memory networks". In: *2015 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*. IEEE. 2015, pp. 1–7. DOI: 10.1109/DSAA.2015.7344872.

[44] S. Chen, Y. Zheng, W. Lu, V. Varadarajan, and K. Wang. "Energy-Optimal Dynamic Computation Offloading for Industrial IoT in Fog Computing". In: *IEEE Transactions on Green Communications and Networking* 4.2 (2020), pp. 566–576. DOI: 10.1109/TGCN.2019.2960767.

[45] Y. Chen, S. Su, and H. Yang. "Convolutional Neural Network Analysis of Recurrence Plots for Anomaly Detection". In: *International Journal of Bifurcation and Chaos in Applied Sciences and Engineering* 30.1 (2020). DOI: 10.1142/S0218127420500029.

[46] M. Chincoli and A. Liotta. "Self-Learning Power Control in Wireless Sensor Networks". In: *Sensors* 18.2 (2018), p. 375. DOI: 10.3390/s18020375.

[47] M. Chincoli and A. Liotta. "Transmission Power Control in WSNs: From Deterministic to Cognitive Methods". In: *Integration, Interconnection, and Interoperability of IoT Systems*. Springer International Publishing, 2018, pp. 39–57. DOI: 10.1007/978-3-319-61300-0_3.

[48] Y. S. Chong and Y. H. Tay. *Abnormal Event Detection in Videos using Spatiotemporal Autoencoder*. 2017. arXiv: 1701.01546 [cs.CV].

[49] Cisco. *Cisco Global Cloud Index: Forecast and Methodology, 2016-2021 White Paper*. 2019. URL: https://www.cisco.com/c/en/us/solutions/collateral/service-provider/global-cloud-index-gci/white-paper-c11-738085.html (visited on 02/10/2020).

[50] A. Deng and B. Hooi. "Graph Neural Network-Based Anomaly Detection in Multivariate Time Series". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 35. 5. 2021, pp. 4027–4035.

[51] H. Deng, Z. Guo, R. Lin, and H. Zou. "Fog Computing Architecture-Based Data Reduction Scheme for WSN". In: *2019 1st International Conference on Industrial Artificial Intelligence (IAI)*. 2019, pp. 1–6. DOI: 10.1109/ICIAI.2019.8850817.

[52] S. Deng, H. Zhao, W. Fang, J. Yin, S. Dustdar, and A. Y. Zomaya. "Edge Intelligence: The Confluence of Edge Computing and Artificial Intelligence". In: *IEEE Internet of Things Journal* 7.8 (2020), pp. 7457–7469. DOI: 10.1109/JIOT.2020.2984887.

[53] W. Ding, X. Jing, Z. Yan, and L. T. Yang. "A survey on data fusion in internet of things: Towards secure and privacy-preserving fusion". In: *Information Fusion* 51 (2019), pp. 129–144. DOI: 10.1016/j.inffus.2018.12.001.

[54] P. Domingos. "MetaCost: A General Method for Making Classifiers Cost-Sensitive". In: *Proceedings of the Fifth ACM SIGKDD International Conference*. 1999, pp. 155–164. DOI: 10.1145/312129.312220.

[55] D. Dua and C. Graff. *UCI Machine Learning Repository*. 2017. URL: http://archive.ics.uci.edu/ml (visited on 07/15/2020).

[56] H. E. Egilmez and A. Ortega. "Spectral anomaly detection using graph-based filtering for wireless sensor networks". In: *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2014, pp. 1085–1089. DOI: 10.1109/ICASSP.2014.6853764.

[57] H. Elazhary. "Internet of Things (IoT), mobile cloud, cloudlet, mobile IoT, IoT cloud, fog, mobile edge, and edge emerging computing paradigms: Disambiguation and research directions". In: *Journal of Network and Computer Applications* 128 (2019), pp. 105–140. DOI: 10.1016/j.jnca.2018.10.021.

[58] *ELKI outlier dataset*. URL: https://elki-project.github.io/datasets/outlier (visited on 07/15/2020).

[59] L. Erhan, M. Ndubuaku, M. Di Mauro, W. Song, M. Chen, G. Fortino, O. Bagdasar, and A. Liotta. "Smart anomaly detection in sensor systems: A multi-perspective review". In: *Information Fusion* 67 (2021), pp. 64–79. DOI: 10.1016/j.inffus.2020.10.001.

[60] L. Erhan. *GitHub repository: data_imputation_rpi_comparison*. https://github.com/lauraerhan/data_imputation_rpi_comparison. 2022.

[61] L. Erhan, M. Di Mauro, A. Anjum, O. Bagdasar, W. Song, and A. Liotta. "Embedded Data Imputation for Environmental Intelligent Sensing: A Case Study". In: *Sensors* 21.23 (2021), p. 7774. DOI: 10.3390/s21237774.

[62] L. Erhan, M. Di Mauro, O. Bagdasar, and A. Liotta. "Critical comparison of data imputation techniques at IoT Edge". In: *Intelligent Distributed Computing XIV*. 2022.

[63] L. Erhan, M. Ndubuaku, E. Ferrara, M. Richardson, D. Sheffield, F. J. Ferguson, P. Brindley, and A. Liotta. "Analyzing Objective and Subjective Data in Social Sciences: Implications for Smart Cities". In: *IEEE Access* 7 (2019), pp. 19890–19906. DOI: 10.1109/ACCESS.2019.2897217.

[64] T. Erl, R. Puttini, and Z. Mahmood. *Cloud Computing: Concepts, Technology & Architecture*. 1st. USA: Prentice Hall Press, 2013. ISBN: 0133387526.

[65] B. Fekade, T. Maksymyuk, M. Kyryk, and M. Jo. "Probabilistic Recovery of Incomplete Sensed Data in IoT". In: *IEEE Internet of Things Journal* 5.4 (2018), pp. 2282–2292. DOI: 10.1109/JIOT.2017.2730360.

[66] N. Fernando, S. W. Loke, and W. Rahayu. "Mobile cloud computing: A survey". In: *Future Generation Computer Systems* 29.1 (2013), pp. 84–106. DOI: 10.1016/j.future.2012.05.023.

[67] E. Ferrara, A. Liotta, L. Erhan, M. Ndubuaku, D. Giusto, M. Richardson, D. Sheffield, and K. McEwan. "A Pilot Study Mapping Citizens' Interaction with Urban Nature". In: *2018 IEEE 16th Intl Conf on Dependable, Autonomic and Secure Computing, 16th Intl Conf on Pervasive Intelligence and Computing, 4th Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress (DASC/PiCom/DataCom/CyberSciTech)*. 2018, pp. 836–841. DOI: 10.1109/DASC/PiCom/DataCom/CyberSciTec.2018.00-21.

[68] E. Ferrara, A. Liotta, M. Ndubuaku, L. Erhan, D. Giusto, M. Richardson, D. Sheffield, and K. McEwan. "A Demographic Analysis of Urban Nature Utilization". In: *2018 10th Computer Science and Electronic Engineering (CEEC)*. 2018, pp. 136–141. DOI: 10.1109/CEEC.2018.8674206.

[69] P. Ferrari, S. Rinaldi, E. Sisinni, F. Colombo, F. Ghelfi, D. Maffei, and M. Malara. "Performance evaluation of full-cloud and edge-cloud architectures for Industrial IoT anomaly detection based on deep learning". In: *2019 II Workshop on Metrology for Industry 4.0 and IoT (MetroInd4.0 IoT)*. 2019, pp. 420–425. DOI: 10.1109/METROI4.2019.8792860.

[70] U. Fiore, F. Palmieri, A. Castiglione, and A. De Santis. "Network anomaly detection with the restricted Boltzmann machine". In: *Neurocomputing* 122 (2013), pp. 13–23. DOI: 10.1016/j.neucom.2012.11.050.

[71] A. Fischer and C. Igel. "An introduction to restricted Boltzmann machines". In: *Iberoamerican congress on pattern recognition*. Springer. 2012, pp. 14–36. DOI: 10.1007/978-3-642-33275-3_2.

[72] E. Fitzgerald, M. Pióro, and A. Tomaszwski. "Energy-Optimal Data Aggregation and Dissemination for the Internet of Things". In: *IEEE Internet of Things Journal* 5.2 (2018), pp. 955–969. DOI: 10.1109/JIOT.2018.2803792.

[73] G. Fortino and P. Trunfio. *Internet of Things Based on Smart Objects*. Springer International Publishing, 2014. ISBN: 978-3-319-00491-4.

[74] G. Fortino, W. Russo, C. Savaglio, W. Shen, and M. Zhou. "Agent-Oriented Cooperative Smart Objects: From IoT System Design to Implementation". In: *IEEE Trans. Syst. Man Cybern. Syst.* 48.11 (2018), pp. 1939–1956. DOI: 10.1109/TSMC.2017.2780618.

[75] P. Fountas and K. Kolomvatsos. "A Continuous Data Imputation Mechanism based on Streams Correlation". In: *2020 IEEE Symposium on Computers and Communications (ISCC)*. 2020, pp. 1–6. DOI: 10.1109/ISCC50000.2020.9219548.

[76] P. Fountas and K. Kolomvatsos. "Ensemble based Data Imputation at the Edge". In: *2020 IEEE 32nd International Conference on Tools with Artificial Intelligence (ICTAI)*. 2020, pp. 961–968. DOI: 10.1109/ICTAI50040.2020.00150.

[77] S. Fu, J. Liu, and H. Pannu. "A hybrid anomaly detection framework in cloud computing using one-class and two-class support vector machines". In: *International Conference on Advanced Data Mining and Applications*. Springer. 2012, pp. 726–738.

[78] X. Fu, G. Fortino, P. Pace, G. Aloi, and W. Li. "Environment-fusion multipath routing protocol for wireless sensor networks". In: *Information Fusion* 53 (2020), pp. 4–19. DOI: 10.1016/j.inffus.2019.06.001.

[79] Y. Fu, F. R. Yu, C. Li, T. H. Luan, and Y. Zhang. "Vehicular Blockchain-Based Collective Learning for Connected and Autonomous Vehicles". In: *IEEE Wireless Communications* 27.2 (2020), pp. 197–203. DOI: 10.1109/MNET.001.1900310.

[80] Y. Fujiki, K. Kazakos, C. Puri, P. Buddharaju, I. Pavlidis, and J. Levine. "NEAT-o-Games: Blending Physical Activity and Fun in the Daily Routine". In: *Comput. Entertain.* 6.2 (2008). DOI: 10.1145/1371216.1371224.

[81] Z. D. G. Wang and K. Choi. "Tackling Missing Data in Community Health Studies Using Additive LS-SVM Classifier". In: *IEEE Journal of Biomedical and Health Informatics* 22.2 (2018), pp. 579–587. DOI: 10.1109/JBHI.2016.2634587.

[82] V. Garcia-Font, C. Garrigues, and H. Rifà-Pous. "A comparative study of anomaly detection techniques for smart city wireless sensor networks". In: *Sensors* 16.6 (2016), p. 868. DOI: 10.3390/s16060868.

[83] P. Garcia-Laencina, J. Sancho-Gomez, and A. Figueiras-Vidal. "Pattern classification with missing data: a review". In: *Neural Computing and Applications* 19.2 (2010), pp. 263–282. DOI: 10.1007/s00521-009-0295-6.

[84] S. Garg, K. Kaur, N. Kumar, and J. J. P. C. Rodrigues. "Hybrid Deep-Learning-Based Anomaly Detection Scheme for Suspicious Flow Detection in SDN: A Social Multimedia Perspective". In: *IEEE Transactions on Multimedia* 21.3 (2019), pp. 566–578. DOI: 10.1109/TMM.2019.2893549.

[85] S. Garg, K. Kaur, S. Batra, G. S. Aujla, G. Morgan, N. Kumar, A. Y. Zomaya, and R. Ranjan. "En-ABC: An ensemble artificial bee colony based anomaly detection scheme for cloud environment". In: *Journal of Parallel and Distributed Computing* 135 (2020), pp. 219–233. DOI: 10.1016/j.jpdc.2019.09.013.

[86] S. Garg, K. Kaur, S. Batra, G. Kaddoum, N. Kumar, and A. Boukerche. "A multi-stage anomaly detection scheme for augmenting the security in IoT-enabled applications". In: *Future Generation Computer Systems* 104 (2020), pp. 105–118. DOI: 10.1016/j.future.2019.09.038.

[87] M. Ge, H. Bangui, and B. Buhnova. "Big Data for Internet of Things: A Survey". In: *Future Generation Computer Systems* 87 (2018), pp. 601–614. DOI: 10.1016/j.future.2018.04.053.

[88] X. Ge, R. Zhou, and Q. Li. "5G NFV-Based Tactile Internet for Mission-Critical IoT Services". In: *IEEE Internet of Things Journal* 7.7 (2020), pp. 6150–6163. DOI: 10.1109/JIOT.2019.2958063.

[89] J. Goh, S. Adepu, M. Tan, and Z. S. Lee. "Anomaly detection in cyber physical systems using recurrent neural networks". In: *2017 IEEE 18th International Symposium on High Assurance Systems Engineering (HASE)*. IEEE. 2017, pp. 140–145. DOI: 10.1109/HASE.2017.36.

[90] A. González-Vidal, P. Rathore, A. S. Rao, J. Mendoza-Bernal, M. Palaniswami, and A. F. Skarmeta-Gómez. "Missing Data Imputation With Bayesian Maximum Entropy for Internet of Things Applications". In: *IEEE Internet of Things Journal* 8.21 (2021), pp. 16108–16120. DOI: 10.1109/JIOT.2020.2987979.

[91] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT Press, 2016. ISBN: 9780262035613.

[92] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. "Generative Adversarial Nets". In: *Advances in Neural Information Processing Systems 27*. Curran Associates, Inc., 2014, pp. 2672–2680.

[93] Google Cloud. *Google Cloud Vision API*. URL: https://cloud.google.com/vision/ (visited on 01/12/2019).

[94] A. K. Gopalakrishna, T. Ozcelebi, J. J. Lukkien, and A. Liotta. "Evaluating machine learning algorithms for applications with humans in the loop". In: *2017 IEEE 14th International Conference on Networking, Sensing and Control (ICNSC)*. 2017, pp. 459–464. DOI: 10.1109/ICNSC.2017.8000136.

[95] B. Guo, Z. Yu, X. Zhou, and D. Zhang. "Opportunistic IoT: Exploring the social side of the internet of things". In: *Proceedings of the 2012 IEEE 16th International Conference on Computer Supported Cooperative Work in Design (CSCWD)*. 2012, pp. 925–929. DOI: 10.1109/CSCWD.2012.6221932.

[96] B. Guo, D. Zhang, Z. Wang, Z. Yu, and X. Zhou. "Opportunistic IoT: Exploring the harmonious interaction between human and the internet of things". In: *Journal of Network and Computer Applications* 36.6 (2013), pp. 1531–1539. DOI: 10.1016/j.jnca.2012.12.028.

[97] J. Han, M. Kamber, and J. Pei. "13 - Data Mining Trends and Research Frontiers". In: *Data Mining (Third Edition)*. Morgan Kaufmann, 2012, pp. 585–631. DOI: 10.1016/B978-0-12-381479-1.00013-7.

[98] D. Hawkins. *Identification of Outliers*. Monographs on Statistics and Applied Probability. Springer Netherlands, 1980. ISBN: 978-94-015-3996-8.

[99] V. J. Hodge and J. Austin. "A Survey of Outlier Detection Methodologies". In: *Artificial Intelligence Review* 22.2 (2004), pp. 85–126. DOI: 10.1007/s10462-004-4304-y.

[100] S. Hoghooghi and R. Javidan. "Proposing a new method for improving RPL to support mobility in the Internet of Things". In: *IET Networks* 9.2 (2020), pp. 48–55. DOI: 10.1049/iet-net.2019.0152.

[101] E. R. Hruschka and M. do Carmo Nicoletti. "Roles Played by Bayesian Networks in Machine Learning: An Empirical Investigation". In: *Emerging Paradigms in Machine Learning*. Springer Berlin Heidelberg, 2013, pp. 75–116. DOI: 10.1007/978-3-642-28699-5_5.

[102] K. Hsieh, A. Harlap, N. Vijaykumar, D. Konomis, G. R. Ganger, P. B. Gibbons, and O. Mutlu. "Gaia: Geo-Distributed Machine Learning Approaching LAN Speeds". In: *14th USENIX Symposium on Networked Systems Design and Implementation*. 2017, pp. 629–647.

[103] C. Huang, Y. Wu, Y. Zuo, K. Pei, and G. Min. "Towards Experienced Anomaly Detector Through Reinforcement Learning". In: *Proceedings of the AAAI Conference on Artificial Intelligence* 32.1 (2018).

[104] L. Huang, X. Nguyen, M. Garofalakis, M. I. Jordan, A. Joseph, and N. Taft. "In-Network PCA and Anomaly Detection". In: *Proceedings of the 19th International Conference on Neural Information Processing Systems*. NIPS'06. Canada: MIT Press, 2006, pp. 617–624.

[105] IBM. *What is edge computing?* 2021. URL: https://www.ibm.com/cloud/what-is-edge-computing (visited on 11/16/2021).

[106] *Improving Wellbeing through Urban Nature (IWUN) | Shmapped*. URL: http://iwun.uk/shmapped/ (visited on 07/15/2018).

[107] INFSO D.4 Networked Enterprise, RFID INFSO G.2 Micro, and Nanosystems in co-operation with the Working Group RFID of the ETP EPoSS. "Internet of Things in 2020. Roadmap for the future. Version 1.1". In: *European Commission: Information Society and Media* (2008).

[108] IoT Analytics. *State of IoT 2021: Number of connected IoT devices growing 9% to 12.3 billion globally, cellular IoT now surpassing 2 billion*. URL: https://iot-analytics.com/number-connected-iot-devices/ (visited on 07/25/2021).

[109] N. A. S. Al-Jamali and H. S. Al-Raweshidy. "Intelligent Traffic Management and Load Balance Based on Spike ISDN-IoT". In: *IEEE Systems Journal* 15.2 (2021), pp. 1640–1651. DOI: 10.1109/JSYST.2020.2996185.

[110] O. Janssens, V. Slavkovikj, B. Vervisch, K. Stockman, M. Loccufier, S. Verstockt, R. Van de Walle, and S. Van Hoecke. "Convolutional neural network based fault detection for rotating machinery". In: *Journal of Sound and Vibration* 377 (2016), pp. 331–345. DOI: 10.1016/j.jsv.2016.05.027.

[111] A. Javaid, Q. Niyaz, W. Sun, and M. Alam. "A deep learning approach for network intrusion detection system". In: *Proceedings of the 9th EAI International Conference on Bio-inspired Information and Communications Technologies (formerly BIONETICS)*. 2016, pp. 21–26.

[112] J. Jin, J. Gubbi, S. Marusic, and M. Palaniswami. "An Information Framework for Creating a Smart City Through Internet of Things". In: *IEEE Internet of Things Journal* 1.2 (2014), pp. 112–121. DOI: 10.1109/JIOT.2013.2296516.

[113] M. V. Joshi, R. C. Agarwal, and V. Kumar. "Mining Needle in a Haystack: Classifying Rare Classes via Two-Phase Rule Induction". In: *SIGMOD Rec.* 30.2 (2001), pp. 91–102. DOI: 10.1145/376284.375673.

[114] A. Joulin, E. Grave, P. Bojanowski, and T. Mikolov. *Bag of Tricks for Efficient Text Classification*. 2016. arXiv: 1607.01759 [cs.CL].

[115] Y. T. K. Arima N. Okada and K. Kiguchi. "Evaluations of a multiple SOMs method for estimating missing values". In: *Proc. IEEE SICE*. 2014, pp. 796–801.

[116] K. Kalyanarangan. *Text-Clustering-API*. 2017. URL: https://github.com/vivekkalyanarangan30/Text-Clustering-API (visited on 12/15/2018).

[117] Y. Kalyani and R. Collier. "A Systematic Survey on the Role of Cloud, Fog, and Edge Computing Combination in Smart Agriculture". In: *Sensors* 21.17 (2021), p. 5922. DOI: 10.3390/s21175922.

[118] M.-J. Kang and J.-W. Kang. "Intrusion Detection System Using Deep Neural Network for In-Vehicle Network Security". In: *PLOS ONE* 11.6 (2016), pp. 1–17. DOI: 10.1371/journal.pone.0155781.

[119] Z. Khan, A. Anjum, K. Soomro, and M. A. Tahir. "Towards cloud based big data analytics for smart future cities". In: *Journal of Cloud Computing* 4.1 (2015). DOI: 10.1186/s13677-015-0026-8.

[120] R. Kitchin. "Big Data, new epistemologies and paradigm shifts". In: *Big Data & Society* 1.1 (2014). DOI: 10.1177/2053951714528481.

[121] C. Kolias, G. Kambourakis, A. Stavrou, and J. Voas. "DDoS in the IoT: Mirai and Other Botnets". In: *Computer* 50.7 (2017), pp. 80–84. DOI: 10.1109/MC.2017.201.

[122] K. Kolomvatsos, P. Papadopoulou, C. Anagnostopoulos, and S. Hadjiefthymiades. "A Spatio-Temporal Data Imputation Model for Supporting Analytics at the Edge". In: *Lecture Notes in Computer Science, Springer, Cham.* Vol. 11701. 2019, pp. 138–150.

[123] L. Kong, M. Xia, X.-Y. Liu, M.-Y. Wu, and X. Liu. "Data loss and reconstruction in sensor networks". In: *2013 Proceedings IEEE INFOCOM*. 2013, pp. 1654–1662. DOI: 10.1109/INFCOM.2013.6566962.

[124] N. Koroniotis, N. Moustafa, E. Sitnikova, and B. Turnbull. "Towards the development of realistic botnet dataset in the Internet of Things for network forensic analytics: BoT-IoT dataset." In: *Future Generation Computer Systems* 100 (2019), pp. 779–796. DOI: 10.1016/j.future.2019.05.041.

[125] R. Kotian, G. Exarchakos, S. Stavros, and A. Liotta. "Impact of Transmission Power Control in multi-hop networks". In: *Future Generation Computer Systems* 75 (2017), pp. 94–107. DOI: 10.1016/j.future.2016.10.010.

[126] T. Kraska, A. Talwalkar, J. C. Duchi, R. Griffith, M. J. Franklin, and M. I. Jordan. "MLbase: A Distributed Machine-learning System". In: *CIDR*. 2013.

[127] K. Krippendorff. *Content Analysis: An Introduction to its Methodology*. Beverly Hills: SAGE Publications, Inc, 1980.

[128] S. Kumar, P. Tiwari, and M. Zymbler. "Internet of Things is a revolutionary approach for future technology enhancement: a review". In: *Journal of big data* 6.1 (2019). DOI: 10.1186/s40537-019-0268-2.

[129] N. Laptev, S. Amizadeh, and I. Flint. "Generic and Scalable Framework for Automated Time-series Anomaly Detection". In: *Proceedings of the 21st ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. Association for Computing Machinery, 2015, pp. 1939–1947. DOI: 10.1145/2783258.2788611.

[130] M. Lavassani, S. Forsström, U. Jennehag, and T. Zhang. "Combining Fog Computing with Sensor Mote Machine Learning for Industrial IoT". In: *Sensors* 18.5 (2018), p. 1532. DOI: 10.3390/s18051532.

[131] A. Lavin and S. Ahmad. "Evaluating Real-time Anomaly Detection Algorithms - the Numenta Anomaly Benchmark". In: (2015). DOI: 10.1109/ICMLA.2015.141.

[132] P. Lea. *IoT and Edge Computing for Architects: Implementing edge and IoT systems from sensors to clouds with communication systems, analytics, and security, 2nd Edition*. 2nd ed. Packt Publishing, 2020. ISBN: 9781839214806.

[133] J. H. Lee, M. G. Hancock, and M.-C. Hu. "Towards an effective framework for building smart cities: Lessons from Seoul and San Francisco". In: *Technological Forecasting and Social Change* 89 (2014), pp. 80–99. DOI: 10.1016/j.techfore.2013.08.033.

[134] W. Lee, S. J. Stolfo, P. K. Chan, E. Eskin, W. Fan, M. Miller, S. Hershkop, and J. Zhang. "Real time data mining-based intrusion detection". In: *Proceedings DARPA Information Survivability Conference and Exposition II. DISCEX'01*. Vol. 1. IEEE. 2001, pp. 89–100. DOI: 10.1109/DISCEX.2001.932195.

[135] W. Lee and D. Xiang. "Information-theoretic measures for anomaly detection". In: *Proceedings 2001 IEEE Symposium on Security and Privacy. S P 2001*. 2001, pp. 130–143. DOI: 10.1109/SECPRI.2001.924294.

[136] C. Li, Y. Xue, J. Wang, W. Zhang, and T. Li. "Edge-oriented computing paradigms: A survey on architecture design and system management". In: *ACM Computing Surveys (CSUR)* 51.2 (2018), pp. 1–34. DOI: 10.1145/3154815.

[137] D. Li, D. Chen, B. Jin, L. Shi, J. Goh, and S.-K. Ng. "MAD-GAN: Multivariate anomaly detection for time series data with generative adversarial networks". In: *International Conference on Artificial Neural Networks*. Springer. 2019, pp. 703–716.

[138] F. Li, X. Zhang, C. Du, and L. Huang. "A hybrid NRS- CART algorithm and its application on coal mine floor water-inrush prediction". In: *TENCON 2015 - 2015 IEEE Region 10 Conference*. 2015, pp. 1–4. DOI: 10.1109/TENCON.2015.7372795.

[139] L. Li, G. Xu, L. Jiao, X. Li, H. Wang, J. Hu, H. Xian, W. Lian, and H. Gao. "A Secure Random Key Distribution Scheme Against Node Replication Attacks in Industrial Wireless Sensor Systems". In: *IEEE Transactions on Industrial Informatics* 16.3 (2020), pp. 2091–2101. DOI: 10.1109/TII.2019.2927296.

[140] Z. C. Lipton, J. Berkowitz, and C. Elkan. *A Critical Review of Recurrent Neural Networks for Sequence Learning*. 2015. arXiv: 1506.00019 [cs.LG].

[141] S. Liu, D. C. Mocanu, A. R. R. Matavalam, Y. Pei, and M. Pechenizkiy. *Sparse evolutionary Deep Learning with over one million artificial neurons on commodity hardware*. 2021. arXiv: 1901.09181 [cs.NE].

[142] Y. Liu, T. Dillon, W. Yu, W. Rahayu, and F. Mostafa. "Missing Value Imputation for Industrial IoT Sensor Data With Large Gaps". In: *IEEE Internet of Things Journal* 7.8 (2020), pp. 6855–6867. DOI: 10.1109/JIOT.2020.2970467.

[143] T. Luo and S. G. Nagarajan. "Distributed anomaly detection using autoencoder neural networks in WSN for IoT". In: *2018 IEEE International Conference on Communications (ICC)*. IEEE. 2018, pp. 1–6. DOI: 10.1109/ICC.2018.8422402.

[144] J. Maas, R. A. Verheij, P. P. Groenewegen, S. de Vries, and P. Spreeuwenberg. "Green space, urbanity, and health: how strong is the relation?" In: *Journal of Epidemiology & Community Health* 60.7 (2006), pp. 587–592. DOI: 10.1136/jech.2005.043125.

[145] G. MacKerron and S. Mourato. "Happiness is greater in natural environments". In: *Global Environmental Change* 23.5 (2013), pp. 992–1000. DOI: 10.1016/j.gloenvcha.2013.03.010.

[146] R. Mahmud, R. Kotagiri, and R. Buyya. "Fog Computing: A Taxonomy, Survey and Future Directions". In: *Internet of Everything: Algorithms, Methodologies, Technologies and Perspectives*. Springer Singapore, 2018, pp. 103–130. DOI: 10.1007/978-981-10-5861-5_5.

[147] P. Malhotra, A. Ramakrishnan, G. Anand, L. Vig, P. Agarwal, and G. Shroff. *LSTM-based Encoder-Decoder for Multi-sensor Anomaly Detection*. 2016. arXiv: 1607.00148 [cs.AI].

[148] I. Mani and I. Zhang. "KNN Approach to Unbalanced Data Distributions: A Case Study Involving Information Extraction". In: *Proceedings of the ICML'2003 Workshop on Learning from Imbalanced Datasets*. 2003.

[149] M. Markou and S. Singh. "Novelty detection: a review-part 1: statistical approaches". In: *Signal Processing* 83.12 (2003), pp. 2481–2497. DOI: 10.1016/j.sigpro.2003.07.018.

[150] M. Markou and S. Singh. "Novelty detection: a review-part 2: neural network based approaches". In: *Signal Processing* 83.12 (2003), pp. 2499–2521. DOI: 10.1016/j.sigpro.2003.07.019.

[151] B. M. Marlin. "Missing Data Problems in Machine Learning". PhD thesis. Canada: University of Toronto, 2008. ISBN: 9780494578988.

[152] I. P. S. Mary and L. Arockiam. "Imputing the missing data in IoT based on the spatial and temporal correlation". In: *2017 IEEE International Conference on Current Trends in Advanced Computing (ICCTAC)*. 2017, pp. 1–4. DOI: 10.1109/ICCTAC.2017.8249990.

[153] V. Matta, M. Di Mauro, and M. Longo. "Botnet identification in randomized DDoS attacks". In: *2016 24th European Signal Processing Conference (EUSIPCO)*. 2016, pp. 2260–2264. DOI: 10.1109/EUSIPCO.2016.7760651.

[154] V. Matta, M. Di Mauro, M. Longo, and A. Farina. "Cyber-Threat Mitigation Exploiting the Birth–Death–Immigration Model". In: *IEEE Transactions on Information Forensics and Security* 13.12 (2018), pp. 3137–3152. DOI: 10.1109/TIFS.2018.2838084.

[155] V. Matta, M. Di Mauro, M. Longo, and A. Farina. "Multiple Cyber-Threats Containment Via Kendall's Birth-Death-Immigration Model". In: *2018 26th European Signal Processing Conference (EUSIPCO)*. 2018, pp. 2554–2558. DOI: 10.23919/EUSIPCO.2018.8553618.

[156] B. McMahan and D. Ramage. *Federated Learning: Collaborative Machine Learning without Centralized Training Data*. https://ai.googleblog.com/2017/04/federated-learning-collaborative.html. 2017. (Visited on 11/15/2021).

[157] P. Mell and T. Grance. *The NIST definition of cloud computing*. 2011. URL: https://csrc.nist.gov/publications/detail/sp/800-145/final (visited on 10/10/2021).

[158] T. Meng, X. Jing, Z. Yan, and W. Pedrycz. "A survey on machine learning for data fusion". In: *Information Fusion* 57 (2020), pp. 115–129. DOI: 10.1016/j.inffus.2019.12.001.

[159] P. d. Meo, E. Ferrara, F. Abel, L. Aroyo, and G.-J. Houben. "Analyzing User Behavior across Social Sharing Environments". In: *ACM Trans. Intell. Syst. Technol.* 5.1 (2014). DOI: 10.1145/2535526.

[160] R. van der Meulen. *What Edge Computing Means For Infrastructure And Operations Leaders*. 2018. URL: https://www.gartner.com/smarterwithgartner/what-edge-computing-means-for-infrastructure-and-operations-leaders (visited on 11/16/2021).

[161] G. Miller. "The Smartphone Psychology Manifesto". In: *Perspectives on Psychological Science* 7.3 (2012), pp. 221–237. DOI: 10.1177/1745691612441215.

[162] Missing Link Electronics. *Accelerating Machine-Learning*. URL: https://www.missinglinkelectronics.com/index.php/menu-products/menu-machine-learning (visited on 05/15/2020).

[163] D. C. Mocanu, E. Mocanu, P. Stone, P. H. Nguyen, M. Gibescu, and A. Liotta. "Scalable training of artificial neural networks with adaptive sparse connectivity inspired by network science". In: *Nature Communications* 9.1 (2018). DOI: 10.1038/s41467-018-04316-3.

[164] N. Mohamudally and M. Peermamode-Mohaboob. "Building An Anomaly Detection Engine (ADE) For IoT Smart Applications". In: *Procedia Computer Science*. The 15th International Conference on Mobile Systems and Pervasive Computing (MobiSPC 2018) / The 13th International Conference on Future Networks and Communications (FNC-2018) / Affiliated Workshops 134 (2018), pp. 10–17. DOI: 10.1016/j.procs.2018.07.138.

[165] F. Molaei, E. Rahimi, H. Siavoshi, S. Ghaychi Afrouz, and V. Tenorio. "A Comprehensive Review on Internet of Things (IoT) and its Implications in the Mining Industry". In: *American Journal of Engineering and Applied Sciences* 13.3 (2020), pp. 499–515. DOI: 10.3844/ajeassp.2020.499.515.

[166] A. Morshed, P. P. Jayaraman, T. Sellis, D. Georgakopoulos, M. Villari, and R. Ranjan. "Deep Osmosis: Holistic Distributed Deep Learning in Osmotic Computing". In: *IEEE Cloud Computing* 4.6 (2017), pp. 22–32. DOI: 10.1109/MCC.2018.1081070.

[167] A. Munawar, P. Vinayavekhin, and G. De Magistris. "Spatio-Temporal Anomaly Detection for Industrial Robots through Prediction in Unsupervised Feature Space". In: *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*. 2017, pp. 1017–1025. DOI: 10.1109/WACV.2017.118.

[168] M. Munir, S. A. Siddiqui, M. A. Chattha, A. Dengel, and S. Ahmed. "FuseAD: Unsupervised Anomaly Detection in Streaming Sensors Data by Fusing Statistical and Deep Learning Models". In: *Sensors* 19.11 (2019), p. 2451. DOI: 10.3390/s19112451.

[169] K. P. Murphy. *Machine Learning: A Probabilistic Perspective*. The MIT Press, 2012. ISBN: 9780262018029.

[170] K. Ni, N. Ramanathan, M. N. H. Chehade, L. Balzano, S. Nair, S. Zahedi, E. Kohler, G. Pottie, M. Hansen, and M. Srivastava. "Sensor Network Data Fault Types". In: *ACM Transactions on Sensor Networks* 5.3 (2009), 25:1–25:29. DOI: 10.1145/1525856.1525863.

[171] J. H. Nord, A. Koohang, and J. Paliszkiewicz. "The Internet of Things: Review and theoretical framework". In: *Expert Systems with Applications* 133 (2019), pp. 97–108. DOI: 10.1016/j.eswa.2019.05.014.

[172] M.-h. Oh and G. Iyengar. "Sequential Anomaly Detection using Inverse Reinforcement Learning". In: *Proceedings of the 25th ACM SIGKDD - KDD '19*. ACM Press, 2019, pp. 1480–1490. DOI: 10.1145/3292500.3330932.

[173] P. Pace, G. Aloi, R. Gravina, G. Caliciuri, G. Fortino, and A. Liotta. "An Edge-Based Architecture to Support Efficient Applications for Healthcare Industry 4.0". In: *IEEE Transactions on Industrial Informatics* 15.1 (2019), pp. 481–489. DOI: 10.1109/TII.2018.2843169.

[174] P. Pace, G. Fortino, Y. Zhang, and A. Liotta. "Intelligence at the Edge of Complex Networks: The Case of Cognitive Transmission Power Control". In: *IEEE Wireless Communications* 26.3 (2019), pp. 97–103. DOI: 10.1109/MWC.2019.1800354.

[175] L. Pan and J. Li. "K-Nearest Neighbor Based Missing Data Estimation Algorithm in Wireless Sensor Networks". In: *Wireless Sensor Network* 2 (2010), pp. 115–122. DOI: 10.4236/wsn.2010.22016.

[176] N. Pandeeswari and G. Kumar. "Anomaly detection system in cloud environment using fuzzy clustering based ANN". In: *Mobile Networks and Applications* 21.3 (2016), pp. 494–505. DOI: 10.1007/s11036-015-0644-x.

[177] N. Patel, A. N. Saridena, A. Choromanska, P. Krishnamurthy, and F. Khorrami. "Adversarial Learning-Based On-Line Anomaly Monitoring for Assured Autonomy". In: *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2018, pp. 6149–6154. DOI: 10.1109/IROS.2018.8593375.

[178] M. L. M. Peixoto, I. Souza, M. Barbosa, G. Lecomte, B. G. Batista, B. T. Kuehne, and D. M. L. Filho. "Data Missing Problem in Smart Surveillance Environment". In: *2018 International Conference on High Performance Computing Simulation (HPCS)*. 2018, pp. 962–969. DOI: 10.1109/HPCS.2018.00152.

[179] M. A. F. Pimentel, D. A. Clifton, L. Clifton, and L. Tarassenko. "A review of novelty detection". In: *Signal Processing* 99 (2014), pp. 215–249. DOI: 10.1016/j.sigpro.2013.12.026.

[180] K. M. Prasad, A. R. M. Reddy, and K. V. Rao. "BARTD: Bio-inspired anomaly based real time detection of under rated App-DDoS attack on web". In: *Journal of King Saud University - Computer and Information Sciences* 32.1 (2020), pp. 73–87. DOI: 10.1016/j.jksuci.2017.07.004.

[181] A. Al-Qamash, I. Soliman, R. Abulibdeh, and M. Saleh. "Cloud, Fog, and Edge Computing: A Software Engineering Perspective". In: *2018 International Conference on Computer and Applications (ICCA)*. 2018, pp. 276–284. DOI: 10.1109/COMAPP.2018.8460443.

[182] Y. Qin, Q. Z. Sheng, N. J. Falkner, S. Dustdar, H. Wang, and A. V. Vasilakos. "When things matter: A survey on data-centric internet of things". In: *Journal of Network and Computer Applications* 64 (2016), pp. 137–153. DOI: 10.1016/j.jnca.2015.12.016.

[183] J. Qiu, Q. Wu, G. Ding, Y. Xu, and S. Feng. "A survey of machine learning for big data processing". In: *EURASIP Journal on Advances in Signal Processing* 2016.1 (2016), p. 67. DOI: 10.1186/s13634-016-0355-x.

[184] T. Raghunathan, J. Lepkowksi, J. Van Hoewyk, and P. Solenbeger. "A multivariate technique for multiply imputing missing values using a sequence of regression models". In: *Survey Methodology* 27 (2001), pp. 85–95.

[185] V. Rajagopalan and A. Ray. "Symbolic time series analysis via wavelet-based partitioning". In: *Signal Processing* 86.11 (2006), pp. 3309–3320. DOI: 10.1016/j.sigpro.2006.01.014.

[186] S. Rajasegarar, C. Leckie, M. Palaniswami, and J. C. Bezdek. "Distributed anomaly detection in wireless sensor networks". In: *2006 10th IEEE Singapore International Conference on Communication Systems*. IEEE. 2006, pp. 1–5. DOI: 10.1109/ICCS.2006.301508.

[187] D. Reinsel, J. Gantz, and J. Rydning. *The Digitization of the World: From Edge to Core. An IDC White Paper*. 2018. URL: https://www.seagate.com/files/www-content/our-story/trends/files/idc-seagate-dataage-whitepaper.pdf (visited on 04/10/2020).

[188] M. Richardson, J. Hallam, and R. Lumber. "One Thousand Good Things in Nature: Aspects of Nearby Nature Associated with Improved Connection to Nature". In: *Environmental Values* 24.5 (2015), pp. 603–619. DOI: 10.3197/096327115x14384223590131.

[189] M. Richardson and D. Sheffield. "Three good things in nature: noticing nearby nature brings sustained increases in connection with nature". In: *PsyEcology* 8.1 (2017), pp. 1–32. DOI: 10.1080/21711976.2016.1267136.

[190] C. Roy, S. Misra, and S. Pal. "Blockchain-Enabled Safety-as-a-Service for Industrial IoT Applications". In: *IEEE Internet of Things Magazine* 3.2 (2020), pp. 19–23. DOI: 10.1109/IOTM.0001.1900080.

[191] S. Ruiz-Correa, D. Santani, and D. Gatica-Perez. "The Young and the City: Crowdsourcing Urban Awareness in a Developing Country". In: *Proceedings of the First International Conference on IoT in Urban Space*. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2014, pp. 74–79. DOI: 10.4108/icst.urb-iot.2014.257453.

[192] M. Salehi and L. Rashidi. "A Survey on Anomaly detection in Evolving Data: [with Application to Forest Fire Risk Prediction]". In: *ACM SIGKDD Explorations Newsletter* 20.1 (2018), pp. 13–23. DOI: 10.1145/3229329.3229332.

[193] K. Samuelsson, M. Giusti, G. D. Peterson, A. Legeby, S. A. Brandt, and S. Barthel. "Impact of environment on people's everyday experiences in Stockholm". In: *Landscape and Urban Planning* 171 (2018), pp. 7–17. DOI: 10.1016/j.landurbplan.2017.11.009.

[194] A. Sari. "A Review of Anomaly Detection Systems in Cloud Networks and Survey of Cloud Security Measures in Cloud Storage Applications". In: *Journal of Information Security* 6 (2015), pp. 142–154. DOI: 10.4236/jis.2015.62015.

[195] I. Sarrigiannis, K. Ramantas, E. Kartsakli, P. Mekikis, A. Antonopoulos, and C. Verikoukis. "Online VNF Lifecycle Management in an MEC-Enabled 5G IoT Architecture". In: *IEEE Internet of Things Journal* 7.5 (2020), pp. 4183–4194. DOI: 10.1109/JIOT.2019.2944695.

[196] C. Savaglio and G. Fortino. "A Simulation-Driven Methodology for IoT Data Mining Based on Edge Computing". In: *ACM Trans. Internet Technol.* 21.2 (2021), pp. 1–22. DOI: 10.1145/3402444.

[197] J. Schneible and A. Lu. "Anomaly detection on the edge". In: *2017 IEEE Military Communications Conference (MILCOM)*. IEEE. 2017, pp. 678–682. DOI: 10.1109/MILCOM.2017.8170817.

[198] F. Serdio, E. Lughofer, K. Pichler, T. Buchegger, M. Pichler, and H. Efendic. "Fault detection in multi-sensor networks based on multivariate time-series models and orthogonal transformations". In: *Information Fusion* 20 (2014), pp. 272–291. DOI: 10.1016/j.inffus.2014.03.006.

[199] A. Servin and D. Kudenko. "Multi-agent Reinforcement Learning for Intrusion Detection". In: *Adaptive Agents and Multi-Agent Systems III. Adaptation and Multi-Agent Learning*. Springer Berlin Heidelberg, 2008, pp. 211–223. DOI: 10.1007/978-3-540-77949-0_15.

[200] H. Shahrokni, B. V. der Heijde, D. Lazarevic, and N. Brandt. "Big Data GIS Analytics Towards Efficient Waste Management in Stockholm". In: *Proceedings of the 2014 conference ICT for Sustainability*. Atlantis Press, 2014. DOI: 10.2991/ict4s-14.2014.17.

[201] A. B. Sharma, L. Golubchik, and R. Govindan. "Sensor faults: Detection methods and prevalence in real-world datasets". In: *ACM Transactions on Sensor Networks (TOSN)* 6.3 (2010), pp. 1–39. DOI: 10.1145/1754414.1754419.

[202] R. Shebuti. *ODDS Library*. 2016. URL: http://odds.cs.stonybrook.edu (visited on 07/15/2020).

[203] A. Sheth. "Citizen Sensing, Social Signals, and Enriching Human Experience". In: *IEEE Internet Computing* 13.4 (2009), pp. 87–92. DOI: 10.1109/MIC.2009.77.

[204] D. J. Stekhoven and P. Bühlmann. "MissForest—non-parametric missing value imputation for mixed-type data". In: *Bioinformatics* 28.1 (2012), pp. 112–118. DOI: 10.1093/bioinformatics/btr597.

[205] D. Strom. *Big Data Makes Things Better*. URL: https://insights.dice.com/2012/08/03/big-data-makes-things-better/ (visited on 02/22/2019).

[206] P. Sun and S. Chawla. "On local spatial outliers". In: *Fourth IEEE International Conference on Data Mining (ICDM'04)*. IEEE. 2004, pp. 209–216. DOI: 10.1109/ICDM.2004.10097.

[207] Z. Sun, L. Wei, C. Xu, T. Wang, Y. Nie, X. Xing, and J. Lu. "An Energy-Efficient Cross-Layer-Sensing Clustering Method Based on Intelligent Fog Computing in WSNs". In: *IEEE Access* 7 (2019), pp. 144165–144177. DOI: 10.1109/ACCESS.2019.2944858.

[208] R. Sutton and A. Barto. *Reinforcement Learning: An Introduction*. Adaptive Computation and Machine Learning series. MIT Press, 2018. ISBN: 978-0-262-03924-6.

[209] D. W. J. T. Rockel and U. Bankhofer. "Decision Trees for the Imputation of Categorical Data". In: *Archives of Data Science* 2.1 (2017), pp. 1–15. DOI: 10.5445/KSP/1000058749/14.

[210] Y. K. Tan. *Energy harvesting autonomous sensor systems: design, analysis, and practical implementation*. CRC Press, 2013. ISBN: 9781138074095.

[211] U. U. Tariq, H. Ali, L. Liu, J. Hardy, M. Kazim, and W. Ahmed. "Energy-Aware Scheduling of Streaming Applications on Edge-Devices in IoT-Based Healthcare". In: *IEEE Transactions on Green Communications and Networking* 5.2 (2021), pp. 803–815. DOI: 10.1109/TGCN.2021.3056479.

[212] The Raspberry Pi Foundation. *Raspberry Pi 4 Model B*. https://www.raspberrypi.org/products/raspberry-pi-4-model-b/specifications/. 2020. (Visited on 05/10/2021).

[213] K. M. Ting. "An instance-weighting method to induce cost-sensitive trees". In: *IEEE Transactions on Knowledge & Data Engineering* 3 (2002), pp. 659–665. DOI: 10.1109/TKDE.2002.1000348.

[214] *TinyOS Project*. URL: http://www.tinyos.net/ (visited on 05/15/2020).

[215] R. Tkachenko, I. Izonin, N. Kryvinska, I. Dronyuk, and K. Zub. "An Approach towards Increasing Prediction Accuracy for the Recovery of Missing IoT Data based on the GRNN-SGTM Ensemble". In: *Sensors* 20.2625 (2020). DOI: 10.3390/s20092625.

[216] O. Troyanskaya, M. Cantor, G. Sherlock, P. Brown, T. Hastie, R. Tibshirani, D. Botstein, and R. B. Altman. "Missing value estimation methods for DNA microarrays". In: *Bioinformatics* 17.6 (2001), pp. 520–525. DOI: 10.1093/bioinformatics/17.6.520.

[217] F. Turchini, L. Seidenari, T. Uricchio, and A. Del Bimbo. "Deep Learning Based Surveillance System for Open Critical Areas". In: *Inventions* 3.4 (2018), p. 69. DOI: 10.3390/inventions3040069.

[218] *UCSD Anomaly Detection Dataset*. URL: http://www.svcl.ucsd.edu/projects/anomaly/dataset.htm (visited on 07/15/2020).

[219] X. R. Wang, J. T. Lizier, O. Obst, M. Prokopenko, and P. Wang. "Spatiotemporal anomaly detection in gas monitoring sensor networks". In: *European Conference on Wireless Sensor Networks*. Springer. 2008, pp. 90–105. DOI: 10.1007/978-3-540-77690-1_6.

[220] Y. Wang, N. Masoud, and A. Khojandi. "Real-Time Sensor Anomaly Detection and Recovery in Connected Automated Vehicle Sensors". In: *IEEE Transactions on Intelligent Transportation Systems* (2020), pp. 1–11. DOI: 10.1109/TITS.2020.2970295.

[221] R. Weber. *Basic Content Analysis*. SAGE Publications, Inc, 1990. DOI: 10.4135/9781412983488.

[222] X. Wei and L. Wu. "A New Proposed Sensor Cloud Architecture Based on Fog Computing for Internet of Things". In: *2019 International Conference on Internet of Things and IEEE Green Computing and Communications and IEEE Cyber, Physical and Social Computing and IEEE Smart Data*. 2019, pp. 615–620. DOI: 10.1109/iThings/GreenCom/CPSCom/SmartData.2019.00120.

[223] G. M. Weiss and F. Provost. "Learning when training data are costly: The effect of class distribution on tree induction". In: *Journal of artificial intelligence research* 19 (2003), pp. 315–354.

[224] M. Wess, P. D. S. Manoj, and A. Jantsch. "Neural network based ECG anomaly detection on FPGA and trade-off analysis". In: *2017 IEEE International Symposium on Circuits and Systems (ISCAS)*. 2017, pp. 1–4. DOI: 10.1109/ISCAS.2017.8050805.

[225] T. Winter, P. Thubert, A. Brandt, J. W. Hui, R. Kelsey, P. Levis, K. Pister, R. Struik, J.-P. Vasseur, and R. K. Alexander. "RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks." In: *IETF RFC* 6550 (2012), pp. 1–157.

[226] Y. Wu, H.-N. Dai, and H. Tang. "Graph Neural Networks for Anomaly Detection in Industrial Internet of Things". In: *IEEE Internet of Things Journal* (2021), pp. 1–1. DOI: 10.1109/JIOT.2021.3094295.

[227] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and S. Y. Philip. "A Comprehensive Survey on Graph Neural Networks". In: *IEEE Transactions on Neural Networks and Learning Systems* 32.1 (2021), pp. 4–24. DOI: 10.1109/TNNLS.2020.2978386.

[228] D. Wulsin, J. Blanco, R. Mani, and B. Litt. "Semi-supervised anomaly detection for EEG waveforms using deep belief nets". In: *2010 Ninth International Conference on Machine Learning and Applications*. IEEE. 2010, pp. 436–441. DOI: 10.1109/ICMLA.2010.71.

[229] X. Xiang, J. Gui, and N. N. Xiong. "An Integral Data Gathering Framework for Supervisory Control and Data Acquisition Systems in Green IoT". In: *IEEE Transactions on Green Communications and Networking* 5.2 (2021), pp. 714–726. DOI: 10.1109/TGCN.2021.3068257.

[230] M. Xie, S. Han, B. Tian, and S. Parvin. "Anomaly detection in wireless sensor networks: A survey". In: *Journal of Network and Computer Applications*. Advanced Topics in Cloud Computing 34.4 (2011), pp. 1302–1325. DOI: 10.1016/j.jnca.2011.03.004.

[231] Z. Xu, K. Kersting, and L. von Ritter. "Stochastic Online Anomaly Analysis for Streaming Time Series". In: *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence*. International Joint Conferences on Artificial Intelligence Organization, 2017, pp. 3189–3195. DOI: 10.24963/ijcai.2017/445.

[232] H. Xue, B. Huang, M. Qin, H. Zhou, and H. Yang. "Edge Computing for Internet of Things: A Survey". In: *2020 International Conferences on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData) and IEEE Congress on Cybermatics (Cybermatics)*. 2020, pp. 755–760. DOI: 10.1109/iThings-GreenCom-CPSCom-SmartData-Cybermatics50389.2020.00130.

[233] X. Yan, W. Xiong, L. Hu, F. Wang, and K. Zhao. "Missing Value Imputation Based on Gaussian Mixture Model for the Internet of Things". In: *Mathematical Problems in Engineering* 15 (2015). DOI: 10.1155/2015/548605.

[234] Y. Yao, A. Sharma, L. Golubchik, and R. Govindan. "Online Anomaly Detection for Sensor Systems: A Simple and Efficient Approach". In: *Performance Evaluation* 67.11 (2010), pp. 1059–1075. DOI: 10.1016/j.peva.2010.08.018.

[235] T. Yéiémou, H. Tall, and D. A. Rollande Sanou. "BAIWL: Blacklisting Approach to Improve Wireless Sensor Network Lifetime". In: *2020 IEEE International Conf on Natural and Engineering Sciences for Sahel's Sustainable Development - Impact of Big Data Application on Society and Environment (IBASE-BF)*. 2020, pp. 1–5. DOI: 10.1109/IBASE-BF48578.2020.9069588.

[236] W. Yu, F. Liang, X. He, W. G. Hatcher, C. Lu, J. Lin, and X. Yang. "A Survey on the Edge Computing for the Internet of Things". In: *IEEE Access* 6 (2018), pp. 6900–6919. DOI: 10.1109/ACCESS.2017.2778504.

[237] D. Zaldivar, L. A. Tawalbeh, and F. Muheidat. "Investigating the Security Threats on Networked Medical Devices". In: *2020 10th Annual Computing and Communication Workshop and Conference (CCWC)*. 2020, pp. 0488–0493. DOI: 10.1109/CCWC47524.2020.9031212.

[238] A. Zanella, N. Bui, A. Castellani, L. Vangelista, and M. Zorzi. "Internet of Things for Smart Cities". In: *IEEE Internet of Things Journal* 1.1 (2014), pp. 22–32. DOI: 10.1109/JIOT.2014.2306328.

[239] A. M. Zarca, J. B. Bernabe, A. Skarmeta, and J. M. Alcaraz Calero. "Virtual IoT HoneyNets to Mitigate Cyberattacks in SDN/NFV-Enabled IoT Networks". In: *IEEE Journal on Selected Areas in Communications* 38.6 (2020), pp. 1262–1277. DOI: 10.1109/JSAC.2020.2986621.

[240] C. Zhang, P. Patras, and H. Haddadi. "Deep Learning in Mobile and Wireless Networking: A Survey". In: *IEEE Communications Surveys and Tutorials* 21.3 (2019), pp. 2224–2287. DOI: 10.1109/COMST.2019.2904897.

[241] P. Zhang, M. Zhou, and G. Fortino. "Security and trust issues in Fog computing: A survey". In: *Future Generation Computer Systems* 88 (2018), pp. 16–27. DOI: 10.1016/j.future.2018.05.008.

[242] N. Zhong, J. Ma, R. Huang, J. Liu, Y. Yao, Y. Zhang, and J. Chen. "Research Challenges and Perspectives on Wisdom Web of Things (W2T)". In: *Wisdom Web of Things*. Springer International Publishing, 2016, pp. 3–26. DOI: 10.1007/978-3-319-44198-6_1.

[243] C. Zhou and R. C. Paffenroth. "Anomaly detection with robust deep autoencoders". In: *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM. 2017, pp. 665–674. DOI: 10.1145/3097983.3098052.

[244] X. G. Zhou and L. Q. Zhang. "Abnormal event detection using recurrent neural network". In: *2015 International Conference on Computer Science and Applications (CSA)*. IEEE. 2015, pp. 222–226. DOI: 10.1109/CSA.2015.64.

[245] W. Zhuang, Q. Ye, F. Lyu, N. Cheng, and J. Ren. "SDN/NFV-Empowered Future IoV With Enhanced Communication, Computing, and Caching". In: *Proceedings of the IEEE* 108.2 (2020), pp. 274–291. DOI: 10.1109/JPROC.2019.2951169.

[246] A. Zimek and P. Filzmoser. "There and back again: Outlier detection between statistical reasoning and data mining algorithms". In: *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 8.6 (2018). DOI: 10.1002/widm.1280.