



**HAL**  
open science

# Spatio-temporal data analytics in the context of environmental crowdsensing

Hafsa El Hafyani

► **To cite this version:**

Hafsa El Hafyani. Spatio-temporal data analytics in the context of environmental crowdsensing. Data Structures and Algorithms [cs.DS]. Université Paris-Saclay, 2022. English. NNT : 2022UPASG035 . tel-03938740

**HAL Id: tel-03938740**

**<https://theses.hal.science/tel-03938740>**

Submitted on 13 Jan 2023

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Analyse de données spatio-temporelles  
dans le contexte de la collecte  
participative de données  
environnementales  
*Spatio-temporal data analytics in the context of  
environmental crowdsensing*

**Thèse de doctorat de l'université Paris-Saclay**

École doctorale n° 580 sciences et technologies de l'information et de la  
communication (STIC)  
Spécialité de doctorat : Informatique  
Graduate School : Informatique et sciences du numérique  
Réfèrent : Université de Versailles Saint-Quentin-en-Yvelines

Thèse préparée dans l'unité de recherche **DAVID** (Université Paris-Saclay,  
UVSQ), sous la direction de **Karine ZEITOUNI**, Professeure, et le  
co-encadrement de **Yehia TAHER**, Maître de Conférence.

**Thèse soutenue à Versailles, le 05 mai 2022, par**

**Hafsa EL HAFYANI**

**Composition du jury**

<b>Ana-Maria Olteanu-Raimond</b> Directrice de recherche (HDR), LASTIG, Université Gustave Eiffel, IGN-ENSG	Présidente et rapporteur
<b>Sandro Bimonte</b> Directeur de recherche (HDR), INRAE	Rapporteur et examinateur
<b>Valérie Issarny</b> Directrice de recherche (HDR), INRIA	Examinatrice
<b>Cyril Ray</b> Maître de conférences, Institut de Recherche de l'École Navale	Examinateur
<b>Karine Zeitouni</b> Professeure, UVSQ - Université de Versailles Saint- Quentin-En-Yvelines	Directrice de thèse

**Titre:** Analyse de données spatio-temporelles dans le contexte de la collecte participative de données environnementales

**Mots clés:** Fouille de données, sciences de données, enrichissement, masses de données, internet des objets, données spatiotemporelles

**Résumé:** La qualité de l'air est l'un des principaux facteurs de risque pour la santé humaine. La collecte participative ou Mobile Crowd Sensing (MCS) en anglais, un nouveau paradigme basé sur la technologie émergente des micro-capteurs connectés, offre la possibilité de mesurer l'exposition individuelle à la pollution de l'air n'importe où et n'importe quand. Cela amène à générer en continu des séries de données géo-localisées, qui finissent par former une grande masses de données. Celle-ci constitue une mine d'information pour des analyses variées et une opportunité unique d'extraction de connaissances sur l'exposition à la pollution. Toutefois, cette analyse est loin d'être simple, car il y a un gap entre les séries de données brutes des

capteurs et les informations exploitables. En effet, les données brutes sont irrégulières, bruitées et incomplètes. Le défi majeur que cette thèse cherche à relever est de combler ce gap en proposant une approche holistique d'analyse et d'extraction de connaissance des données collectées dans le contexte du MCS. Nous mettons en œuvre un processus analytique complet comprenant le prétraitement des données, leur enrichissement avec des informations contextuelles, ainsi que la modélisation et le stockage de ces données. Nous l'avons implémenté en veillant à automatiser son déploiement. Les approches proposées sont appliquées sur des données réelles collectées au sein du projet Polluscope.

**Title:** Spatio-temporal Data Analytics in the Context of Environmental Crowdsensing

**Keywords:** Data mining, data science, enrichment, big data, internet of things, spatiotemporal data

**Abstract:** Air quality is one of the major risk factors in human health. Mobile Crowd Sensing (MCS), which is a new paradigm based on the emerging connected micro-sensor technology, offers the opportunity of the assessment of personal exposure to air pollution anywhere and anytime. This leads to the continuous generation of geolocated data series, which results in a big data volume. Such data is deemed to be a mine of information for various analysis, and a unique opportunity of knowledge discovery about pollution exposure. However, achieving this analysis is far from straightforward. In fact, there is a gap to

fill between the raw sensor data series and usable information: raw data is highly uneven, noisy, and incomplete. The major challenge addressed by this thesis is to fill this gap by providing a holistic approach for data analytics and mining in the context of MCS. We establish an end-to-end analytics pipeline, which encompasses data preprocessing, their enrichment with contextual information, as well as data modeling and storage. We implemented this pipeline while ensuring its automatized deployment. The proposed approaches have been applied to real-world datasets collected within the Polluscope project.

(وَقُلِ اعْمَلُوا فَسَيَرَى اللَّهُ عَمَلَكُمْ وَرَسُولُهُ وَالْمُؤْمِنُونَ)  
التوبة - ١٠٥

*"And say (O Muhammad SAW) "Do deeds! Allah will see your deeds, and (so will) His Messenger and the believers."*

*Quran, Chapter 9, Verse 105*



## *Dedicated to My Precious Family*

*My Mother (Nezha Mrani Alaoui) & my Father (Mohamed El Hafyani)*

*My Sisters (Mounia & Fatima)*

*My Brothers (Abdellatif, Oussama, Hamza & Amine)...*

*Hafsa @ Versailles, France*

*May 5th, 2022*



## Acknowledgments

The journey of this three-year PhD program at UVSQ-Université Paris-Saclay has proven to be a unique learning experience for me, both professionally and personally. My sincere thanks to the many people who have accompanied me during this challenging and exciting Ph.D.

First and foremost, I would like to express my deep gratitude to Prof. Karine Zeitouni for her motivation, enthusiasm and her immense knowledge. The three-year guidance from her was extremely beneficial, both scientifically and personally. She enlightens me about all aspects of doing research and being a scientist. Her determination, hard-work and her solid experience have always inspired me to keep going ahead. I am very lucky and happy to have such a tremendous mentor as Karine.

My genuine words of thanks should also go to my **co-advisor** Dr. Yehia Taher for always being so helpful and motivating. He always offered me guidance, support and friendship over the last three years. My greatest gratitude goes to Yehia for all of his kindness and supports from all perspectives.

I want to thank all of the jury members of my thesis defense, including Dr. Ana-Maria Olteanu-Raimond (HDR), Dr. Sandro Bimonte (HDR), Dr. Valérie Issarny (HDR), and Dr. Cyril Ray. I am appreciative of their time and efforts in reading and assessing this thesis work. My special words of thanks go to Dr. Ana-Maria Olteanu-Raimond (HDR) and Dr. Sandro Bimonte (HDR) for reviewing my dissertation.

I am infinitely indebted to my doctoral school STIC, and more broadly to Université Paris-Saclay, and the National Research Agency who has funded my work, through the Polluscope project under the grant agreement ANR-15-CE22-0018. I would like to thank all the members of the Polluscope consortia from LSCE, CEREMA and iPLESP who contributed in one way or another to this work, especially Boris Dessimond, Isabella Annesi-Maesano, Basile Chaix, Valerie Gros, Nicolas Bonnaire, Laura Bouillon, Salim Srairi and Jean-Marc Naude.

This thesis would not be able to reach this level without the contribution from many excellent colleagues in DAVID Lab. My deep appreciation goes to Dr. Laurent Yeh and Dr. Zoubida Kedad (HDR) for their invaluable feedback on my research and for always being so supportive of my work. I would like to thank my lab mates (Jingwei, Souheir, Redouane, Zoé, Mohammad Abboud, Mohammad Rihany, Alaa, Perla, Mariem, Lili, Ahmad, Baudouin, Robin, Julien, Ludovic, Julien, Alexandros, Riham, Pierre, Livia) for always being there and for supporting me. I am proud to say that my experience in the David lab was exciting and fun, and has energized me to continue in academic research.

I would like to take this opportunity to express my appreciation and thank to my friends Salima and Halima for their endless motivation and emotional support through the ups and downs.

Last but not least, even most importantly, my heartfelt gratitude goes to my precious family for their unconditional love, endless patience, and steady support, particularly to my mother Nezha Mrani Alaoui, my father Mohamed El Hafyani, My sisters Mounia and Fatima, and my brothers Abdellatif, Oussama, Hamza and Amine. This doctoral dissertation is dedicated to them.





# Contents

<b>List of Figures</b>	<b>11</b>
<b>List of Tables</b>	<b>13</b>
<b>1 Introduction</b>	<b>15</b>
1.1 Background . . . . .	16
1.2 Motivation . . . . .	17
1.3 Problem Statement & Research Questions . . . . .	19
1.4 Contributions . . . . .	22
1.5 Structure of the Dissertation . . . . .	24
1.6 Scientific Contributions . . . . .	25
<b>2 State of the Art</b>	<b>27</b>
2.1 Introduction . . . . .	28
2.2 Trajectory Data Preprocessing . . . . .	29
2.2.1 GPS Trajectory Data Noise Filtering . . . . .	29
2.2.2 Time Series Noise Filtering . . . . .	30
2.2.3 Discussion . . . . .	30
2.3 Trajectory Data Segmentation . . . . .	31
2.3.1 Stop & Move Detection . . . . .	31
2.3.2 Move Episodes Segmentation . . . . .	33
2.3.3 Time Series Segmentation . . . . .	33
2.3.4 Discussion . . . . .	35
2.4 Activity Recognition . . . . .	36
2.4.1 Activity Recognition from GPS Trajectories . . . . .	36
2.4.2 Activity Recognition from Wearable Sensors . . . . .	39
2.4.3 Discussion . . . . .	41
2.5 Generic Machine Learning Algorithms . . . . .	41
2.5.1 Multivariate Time Series Classification . . . . .	42
2.5.2 Multi-View Learning . . . . .	43
2.5.3 Discussion . . . . .	44
2.6 Moving Object Databases and Warehousing . . . . .	44
2.6.1 Moving Object Databases . . . . .	44
2.6.2 Trajectory Data Warehousing . . . . .	45
2.6.3 Other Works on DW . . . . .	48
2.6.4 Discussion . . . . .	49

<b>3</b>	<b>Trajectory Data Enrichment</b>	<b>51</b>
3.1	Introduction	52
3.2	Multidimensional time series segmentation	52
3.2.1	Change Point Detection: Summary of Related Work	53
3.2.2	Change Point Detection Model	54
3.2.3	Experiments and Results	59
3.2.4	Summary	61
3.3	Learning the Micro-environment	62
3.3.1	Activity Recognition: Summary of Related Work	64
3.3.2	Problem Formalization	65
3.3.3	Multi-view Learning Model	68
3.3.4	Micro-environment recognition model	69
3.3.5	Hybrid Multi-view Learning Model	74
3.3.6	Experiments and Results	79
3.3.7	Discussions & Perspectives	96
3.3.8	Summary	99
3.4	Conclusion	99
<b>4</b>	<b>Multidimensional Trajectory modeling</b>	<b>101</b>
4.1	Introduction	102
4.1.1	Motivation and Challenges	102
4.1.2	Problem Statement	104
4.1.3	Concepts of Semantic Trajectory Data Modeling	104
4.1.4	Contributions	106
4.2	Requirements of Multidimensional Data modeling in MCS	107
4.3	Background	108
4.4	Multidimensional Data Model	110
4.4.1	Overview	110
4.4.2	Spatial Discretization	111
4.4.3	Temporal Discretization	112
4.4.4	Spatial Indexing	113
4.4.5	Temporal Disaggregation	113
4.4.6	Spatial Disaggregation	114
4.4.7	MULTICS General Schema	115
4.4.8	Application Scenarios	116
4.5	Implementation and Experimentation	117
4.5.1	Experimental Design	117
4.5.2	Longitudinal Analysis	118
4.5.3	Spatial Analysis	118
4.5.4	Temporal Analysis	119
4.5.5	Temporal Disaggregation	121
4.5.6	Spatial Disaggregation	122
4.5.7	Computational Costs	123

4.6	Conclusion and Perspectives . . . . .	127
<b>5</b>	<b>Automating Data Analysis Pipelines</b>	<b>129</b>
5.1	Introduction . . . . .	130
5.2	Problem Statement . . . . .	131
5.3	Microservices Architectures: Related Work . . . . .	131
5.4	System Architecture . . . . .	132
5.4.1	Data Processing . . . . .	133
5.4.2	Visualisation . . . . .	134
5.5	Design of the Microservices . . . . .	136
5.6	Visualisation Demonstration . . . . .	139
5.6.1	COMIC Demonstration Scenario . . . . .	139
5.6.2	Grafana Demonstration . . . . .	141
5.7	Conclusion . . . . .	142
<b>6</b>	<b>Conclusions and Future Work</b>	<b>145</b>
6.1	Summary of Contributions . . . . .	146
6.2	Future Work . . . . .	149
6.2.1	Map-matching Based Enrichment . . . . .	149
6.2.2	Events Processing . . . . .	149
6.2.3	Exposure Profiles . . . . .	149
6.2.4	Privacy and Participants' Incentives . . . . .	149
<b>7</b>	<b>Bibliography</b>	<b>151</b>
<b>A</b>	<b>Appendix A</b>	<b>165</b>
A.1	Data Collection Campaigns . . . . .	165
A.2	Data Collection . . . . .	166
<b>B</b>	<b>Appendix B</b>	<b>169</b>
B.1	Longitudinal Analysis . . . . .	169
B.2	Spatial Analysis . . . . .	169
<b>C</b>	<b>Appendix C</b>	<b>171</b>
<b>D</b>	<b>Appendix D</b>	<b>173</b>



## List of Figures

1.1	An example of enriched trajectory collected in the context of MCS . . . . .	18
1.2	An illustration of the ETL data pipelines. . . . .	22
1.3	Dissertation structure organisation. . . . .	24
2.1	Structure of state-of-the-art chapter. . . . .	28
2.2	Stop and move segments in a trajectory . . . . .	32
2.3	A sample of time series with several change points . . . . .	34
2.4	From raw trajectory data to semantic trajectory. . . . .	37
2.5	General data flow for activity recognition from GPS trajectory data. . . . .	38
2.6	Generic data acquisition architecture for Human Activity Recognition. . . . .	39
2.7	HAR system architecture based on wearable sensors. . . . .	40
2.8	An example of a multidimensional model . . . . .	46
2.9	TDW framework structure . . . . .	47
3.1	Architecture of change point detection model. . . . .	54
3.2	An Overview of data collected in the context of MCS . . . . .	55
3.3	Performances of CPD during testing and validation phases . . . . .	61
3.4	Inter-sensor and micro-environment correlations. . . . .	63
3.5	Overview of the micro-environment recognition process. . . . .	70
3.6	Distribution of data over classes before class balancing. . . . .	72
3.7	Distribution of data over classes after class balancing. . . . .	72
3.8	Spatial Dimension Representation . . . . .	76
3.9	Sample trajectory . . . . .	76
3.10	Accuracy among different views. . . . .	81
3.11	Multi-view Approach Confusion Matrix . . . . .	84
3.12	Parameters' effect on Grid-Density Stop Detection (GDSD) . . . . .	85
3.13	MVP confusion matrix . . . . .	91
3.14	MLSTMP confusion matrix . . . . .	93
3.15	KNN-DTWP confusion matrix . . . . .	94
3.16	MVP + Office location Correction confusion matrix . . . . .	94
3.17	Predictions of VGP campaign for participant 9999988. . . . .	96
3.18	Predictions of VGP campaign for participant 9999944. . . . .	97
4.1	An example of rich trajectory collected in the context of MCS. . . . .	103
4.2	The conceptual architecture of the proposed solution for rich trajectories. . . . .	106
4.3	MULTICS conceptual Schema. . . . .	111
4.4	Spatial dimension representation . . . . .	112
4.5	Spatial hierarchy representation . . . . .	112
4.6	Longitudinal analysis per micro-environment. . . . .	119

4.7	Concentrations of NO <sub>2</sub> for all participants combined in Paris and Versailles regions. . . . .	120
4.8	10 minutes average concentrations of each day. . . . .	120
4.9	Maximum of hourly averages per micro-environment for all participants combined. . . . .	121
4.10	Airparif data versus the collected data for NO <sub>2</sub> in one trajectory. . . . .	121
4.11	Example of the disaggregation of PM <sub>10</sub> time series using NO <sub>2</sub> data as proxy with the Chow-Lin-Maxlog method . . . . .	122
4.12	Real versus estimated values of NO <sub>2</sub> after the spatial disaggregation. . . . .	123
4.13	Execution time varying the data volume. . . . .	126
5.1	Design of microservices for automating data analytics pipelines in MCS. . . . .	132
5.2	Visual analysis of a trajectory and its associated measurements . . . . .	136
5.3	The system implementation prototype . . . . .	137
5.4	COMIC visualisation GUI. . . . .	140
5.5	Classification and CPD Dashboard . . . . .	141
5.6	Comparison between single-view and multi-view models. . . . .	142
5.7	A zoom in on the collected data. . . . .	142
A.1	Illustration of the spatial distribution of participants in the two cohorts according to their place of residence. . . . .	166
B.1	Longitudinal analysis over time hierarchy. . . . .	169
B.2	Exposure at coarse levels of the spatial hierarchy. . . . .	170

## List of Tables

3.1	Overview of time spent in every micro-environment for four participants . . . . .	58
3.2	Cumulative Sum parameters optimization for each dimension . . . . .	60
3.3	An example of the new generated dataset $D'$ . . . . .	69
3.4	A concrete example of the new generated dataset $D'$ . . . . .	73
3.5	Average time spent per micro-environment. . . . .	80
3.6	Performance of Multi-view Learner ( with/out speed) . . . . .	82
3.7	Performance of MLSTM-FCN ( with/out speed) . . . . .	83
3.8	Performance of Multi-view Learner (2-step approach with/out speed) . . . . .	83
3.9	Comparison between Scikit-Mobility and grid density based model. . . . .	86
3.10	The description of various model variants . . . . .	88
3.11	Performance comparison of various privacy-friendly models . . . . .	89
3.12	Performance comparison of various privacy-invasive models . . . . .	89
3.13	Performance of Multi-view Learner on Participants' data (before/after) post-processing . . . . .	91
3.14	Performance of Multi-view Learner (2 steps classification) on Participants' data (before/after) post-processing . . . . .	92
3.15	Performance of MLSTM-FCN on Participants' data (before/after) post-processing . . . . .	92
3.16	Performance of KNN-DTW on Participants' data (before/after) post-processing . . . . .	93
3.17	Performance of Multi-view Learner with Location Correction and Post-processing on Participants' data . . . . .	95
3.18	Performance of MVB without NO2 and BC VS. MVB . . . . .	98
3.19	The accuracy results of TapNet on Polluscope data under different supervision ratios . . . . .	99
4.1	Existing work on mobility analysis in DW and OLAP systems. . . . .	105
A.1	General characteristics of the two campaigns VGP and RECORD. . . . .	165
C.1	Description of the used covariates for the spatial disaggregation of NO2 values. . . . .	171
C.2	A statistical summary of NO2 values. . . . .	171





# 1 - Introduction

## Contents

---

1.1	Background . . . . .	16
1.2	Motivation . . . . .	17
1.3	Problem Statement & Research Questions .	19
1.4	Contributions . . . . .	22
1.5	Structure of the Dissertation . . . . .	24
1.6	Scientific Contributions . . . . .	25

---

## 1.1 . Background

With the advancement of Internet of Things (IoT) and geolocation technology, an increasing number of connected objects are becoming location aware, including vehicles, vessels and biological moving entities such as humans and animals. Nowadays, smartphones with embedded GPS are no longer an emerging trend, but almost an essential feature. These kinds of mobile pervasive sensing technologies enhance the continuous generation of large volumes of geo data series, with no restriction on time and space, and promote the connectivity between physical objects and their surroundings.

Therefore, in order to monitor a certain spatial and temporal phenomena (e.g. air quality monitoring, traffic monitoring, etc), wireless sensing technologies are certainly used thanks to their ability to sense their surroundings and produce reliable measurements. However, traditional wireless sensing networks require a large number of deployed sensors to ensure the area coverage for large-scale and fine-grained sensing, which is economically unpractical and uninteresting. For instance, in order to monitor air quality in the area of the Paris region, Airparif<sup>1</sup>, which is responsible for this task, deploys around 50 fixed and permanent stations over a radius of 100 km around Paris. Around 30 stations are localised in Paris and its small crown, and the rest are scattered around its large crown. If we want to extend the air quality monitoring system to cover the whole area of Paris region at a finer grained level, certainly more than 50 fixed and permanent stations need to be deployed to ensure full area coverage. However, the expensive cost of fixed stations and their maintenance would make this system hard to implement.

Luckily, the new paradigm called Mobile Crowd Sensing (MCS) [51, 57] empowers volunteers to contribute data acquired by a multi-sensor box, and a mobile device to monitor large-scale phenomena that is not easy to monitor with a single sensor or does not offer full coverage with fixed sensors. In MCS scenarios, participants are equipped with various sensors plus a GPS embedded mobile device. They move freely within a monitoring region to take samples and report the collected data to monitoring center to cover the observed phenomena, which leads to a continuous generation of large volumes of geo data series. The particularity of this sensing paradigm is the combination of geo-location with observations and measurements of the observed phenomena over time. Several large-scale application scenarios are motivated by MCS paradigm such as noise monitoring [5], radioactivity monitoring [97], and air quality monitoring and individual exposure such as in our context of Polluscope project <sup>2</sup>.

The general objective of Polluscoe project is to monitor personal exposure to air pollution, since exposure to air pollution promotes the development of serious chronic pathologies, in particular cardiovascular and respiratory pathologies and

---

<sup>1</sup><https://www.airparif.asso.fr/>

<sup>2</sup><http://polluscope.uvsq.fr/>

cancers, which results in increased mortality, lower life expectancy and increased use of care. As a matter of fact, air pollution is responsible for nearly 1 in 10 deaths in the Paris region in 2019 [65]. In 2019, mortality related to air pollution in the Paris region is estimated at 7,920 premature deaths each year. Luckily, due to the COVID-19 pandemic and health restrictions, 2020 was an exceptional year in terms of air quality. The restrictive measures have led to a decrease in nitrogen dioxide (NO<sub>2</sub>) concentrations and particulate matters (PMs). This drop in nitrogen dioxide concentrations made it possible to avoid around 310 deaths, and the decrease of the concentrations of particulate matters pushed back the number of death by around 180, compared to 2019 [65].

Therefore, in order to monitor personal exposure to pollution, the Polluscope project recruited participants and equipped each one of them with a sensor kit and a mobile device to collect air quality measurement and GPS coordinates as geo-dated data series. The recruited participants, on a voluntary basis, collect air quality measurements such as Particulate Matters, NO<sub>2</sub>, Black Carbon, Temperature and Humidity by the multi-sensor box. The mobile device is used to collect GPS logs. In addition, a mobile application is provided to participants so that they can provide information on the context of the measurements. Thus, they are asked to indicate the type of place (called micro-environment) each time they change it. They also provide information on specific events that have an impact on the concentrations of pollutants and therefore on their exposure. This type of information is commonly referred to as self-reporting. These annotations are very important in MCS. They are used to interpret the observed measurements because they are largely dependent on the type of environment (indoor, outdoor or in transport). Without this information, the collected measurements cannot be interpreted correctly. In addition, they provide insight at a higher level of abstraction along participants trajectories.

With such increasing generation of large volume data, there is a growing need for putting forward a holistic approach for efficiently managing and analysing such huge amount of spatio-temporal data series produced by moving objects (i.e. participants) to fill the gap between data generation and data comprehension. While the literature proposes different solutions for handling moving objects such as moving objects data management in the database community, and several data mining techniques for analytical purposes, these approaches do not provide enriched trajectories mining, which leaves to application developers all the challenges to extract complex information from raw enriched trajectories.

## 1.2 . Motivation

One of the characteristics of the MCS paradigm is the combination of spatial location with continuous measurements and annotations which results in semantically enriched trajectories. Figure 1.1 describes the reconstruction of enriched

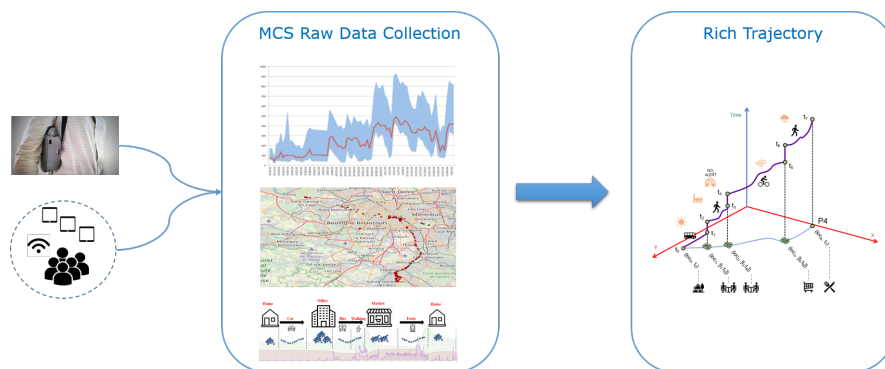


Figure 1.1: An example of enriched trajectory collected in the context of MCS

trajectory data in MCS. Basically, enriched trajectories are constructed from the combination of GPS tracks with ambient air measurements (e.g., atmospheric pollutants) plus annotation of contextual information such as the micro-environments of the participants (e.g., *home*, *office*, *restaurant*, etc.) and air pollution related events (e.g., smoking).

The construction of enriched trajectories from MCS has several advantages compared to the usage of traditional monitoring techniques (e.g. fixed stations). First, it promotes personalised exposure, i.e. each individual will be able to gain insights on his/her exposure. Second, it measures indoor and outdoor environments (e.g. home, work, transportation, streets, parks, etc.) and expands the spatial coverage, depending on human mobility. Finally, it enables insights at a higher resolution along the participants trajectories, thereby allowing to capture local variability and peaks of pollution.

Nevertheless, exploiting such complex data series for analytical purpose, such as exploratory analysis using data mining techniques, is far from straightforward, since raw sensor data are mostly noisy and acquired at irregular (and asynchronous) frequencies. Several research [140, 102, 142] has been proposed to shape the field of trajectory-based application and to provide a roadmap from the derivation of trajectory data, to trajectory preprocessing, to trajectory data management, and to a variety of mining tasks (such as trajectory pattern mining, outlier detection, and trajectory classification). Starting from this roadmap on trajectories, the motivation of this thesis is to further explore trajectory data mining and analytics and draw a protocol, not only for trajectory data from a geometric view, but for enriched trajectories with complex settings in MCS, due to the additional dimensions (i.e. temporal and semantic) and their low quality.

We aim at a generic computing model for extracting usable information from enriched trajectory data, which is qualified to provide information about exposure at the personal and collective levels. Therefore, we strive to integrate a semantic enrichment approach for trajectory data analysis, to better understand the personal

exposure and its relation to people's whereabouts (i.e micro-environment). Take the example where no information about participants whereabouts is available. It would be impossible to get insight about their exposure and therefore, the collected data would not be used properly. For this reason, there is a great interest in building an effective model to semantically enrich trajectories with participants micro-environments. That way, we can compare participants exposures to each others and build a medium exposure profile which will serve as a reference for an unobserved population.

Another objective of this thesis work is to compare the collected MCS data to traditional fixed stations network (i.e. Airparif data) while taking into consideration the micro-environment. In other words, we are interested in comparing both data sources at the same time and space; but enclosed micro-environment will not be object of this comparison. However, this comparison is not straightforward since the spatio-temporal coverage of both data sources are not aligned. For example, some fixed station models provide hourly pollutant measurements whilst MCS data measure air quality data every one minute. In addition, while fixed station network cover with their models the whole study area geographically at a certain spatial granularity (which can be a coarse granularity), MCS has low spatial coverage (depending on the number of volunteers) but with the finest spatial granularity. The objective here is to overcome the shortcomings of the integration of both data sources and provide a sound comparison.

As for the analysis of enriched trajectories analysis, a multidimensional analysis on such complex data is highly desirable, since it allows the exploration of data from several perspectives. Therefore, on one side, to take full advantage of these data, it should not be only analysed in isolation, but rather be matched with the context, and analyse it under multiple dimensionality and scale (e.g., spatial, user, micro-environment, time dimensions). On the other side, we need to design some solutions for spatial and temporal data imputation and /or interpolation to overcome the limitations of MCS (e.g. low coverage and missing data problems).

### 1.3 . Problem Statement & Research Questions

Data measured by mobile sensors can be represented by multivariate time series which are characterised by the presence of a spatial dimension forming trajectories. Equivalently, these data can be seen as spatio-temporal trajectories enriched by additional measurements throughout the collection period as show in Figure 1.1. While the human involvement in the MCS process is very important, it makes this paradigm a double-edged sword. On the one hand, (i) it is easier to deploy at lower cost compared to traditional stations, (ii) it enables insights at a finer-grained level of the observed phenomena including indoor and outdoor, and (iii) the network coverage may be further expanded if needed depending on the number of deployed volunteers. However, on the other hand, it brings a number of challenging

characteristics.

**Data Imperfection.** Data collected in the context of MCS is often imperfect compared to traditional stations, due to the limitations of accuracy and correctness of the sensors. The process needs to integrate a comparison step (called qualification campaign) between sensory data and traditional stations measurements, to ensure that the gap between the two sources of measurements is at its lowest level. In addition, after sensors deployment, the collected data may exhibit different problems such as noise, anomalies and sometimes data loss that requires cleaning and preprocessing. In fact, we could observe timestamps that are closely spaced or too sparse in different cases. Some sensors may be offline for hours or stay idle when the device is static then switch to a burst mode on the move. Some are configured to reduce data transmission when the variation is less than a predefined threshold. Such data imperfections should be taken into account, which affect both time series data and geolocation.

**Low-confidence in Self-reporting.** In real-life application of MCS, the annotation of participants contexts is by far the most difficult information to collect, since very few participants thoroughly annotate their micro-environment, plus the collected annotations are not guaranteed to be accurate. For example, some participants indicate that they are in their offices at 3am, or light a fireplace in a street, other participants completely exclude this annotation task. Therefore, there is a great interest in detecting automatically the participants context without burdening them (possibly from imperfect sensory data and participants annotation).

**Difficulty of merging sensory data and integrating external data.** In MCS, we deploy a large spectrum of sensors with different characteristics for sensitivity and sampling frequency. For example, taking two different sensors, one may generate measurements every minute while the other one may measure every 1 second. The data collected from all sensing objects should be merged, which could lead to measurements at irregular time intervals and missing data problems. Furthermore, if the spatial and temporal resolution of the observations is high, the coverage is very limited and very imbalanced depending on the visited places. In fact, a high density of data is concentrated in *home* and *office*, while very little data is located in punctual places such as *restaurant*, *station* and *store*, and even less data is located in crossed places (e.g., *street*, *bus*, etc.). Precisely, the spatial coverage is very irregular. Some places are characterised by a high spatial density, whilst zero information is available about other locations because nobody sets foot there. Therefore, it is really difficult to generalize and provide a densification coverage similar to the regulatory observation network. Plus, it is not easy to merge and compare with data of different spatio-temporal resolution and territorial coverage. For example, fixed station networks provide a full spatio-temporal coverage model for the study area. However, since the spatio-temporal resolution is different from sensory data, merging and comparing both sources of data is not easy.

To put our motivation of a "*holistic*" approach for enriched trajectory data

analysis into more accurate research statements, we break down the problem under consideration into a set of fundamental research questions that we will explore, discuss and answer during this thesis work. Each research question is either form a *scientific* or *technological* challenge :

- **R1.** What are the fundamental modelings for creating usable information from raw enriched trajectories ? What are the different facets and views of enriched trajectory data ? What is the personal exposure to pollution and how to quantify it ? What are the requirements for getting insights from enriched trajectories ? What is the gap between raw enriched trajectories and usable knowledge and how to bridge it ?
- **R2.** What are the fundamental preprocessing steps for spatio-temporal enriched trajectory data ? How to find spatio-temporal noise and outliers ? How to differentiate between an artifact peak and noise ? What is the gap between raw enriched trajectory data and clean enriched trajectories, and how to bridge such gap ? How to achieve purified enriched trajectories ?
- **R3.** Can we provide a more comprehensive and semantically enriched representation of enriched trajectory data ? Any intermediate models are necessary to achieve the semantically enriched representation aforementioned ? Can we contextualise the data and enrich it with the type of activity and movement (i.e. micro-environment) ? What are the spatio-temporal requirements to characterize the micro-environment and summarize their observed properties ? Can we combine different sensors (i.e., both GPS and AQ sensors) to automatically infer people's context ? Which types of algorithms and computational solutions need to be designed for this purpose ? Do data mining (e.g. feature representation) or statistical summary techniques have the ability to provide solutions for such recognition tasks ?
- **R4.** How to further enrich sensory data ? Does such semantic enrichment need additional external sources, such as the traditional network of fixed stations and models ? How to merge sensory data with external data ? How to align both sources of data with such different spatial and temporal scales and very low MCS coverage while taking into account micro-environments ? How to handle the problem of missing value provoked by merging two data sources with different spatial and temporal scales ? How to integrate and compare external data to sensory data ? Can we provide a generic model for external data integration and comparison with sensory data ?
- **R5.** Can we provide an interactive visualisation platform to explore every facet of the data, including GPS tracks and measurements ? Is it possible to visualise the difference between the detected and the declared micro-environments ? To what extent the computation model can affect the results



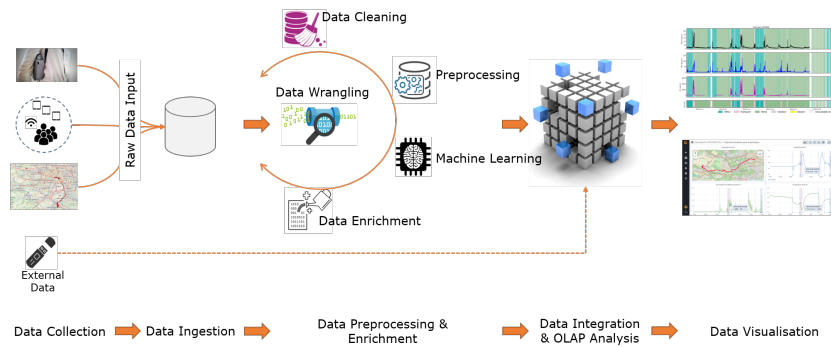


Figure 1.2: An illustration of the architecture pipelines.

of micro-environment's detection ? Can we visualise that effect ? Which sensory data contribute more in this inference ?

- **R6.** Can we automatise the aforementioned process of bridging the gap between data collection and data comprehension ? How to enable the pipelines to work properly and efficiently without human involvement ? Which technologies are best suitable for this purpose ?

## 1.4 . Contributions

Towards the motivation to establish an effective and efficient multidimensional data analytic framework in MCS, this dissertation focuses on providing an *end-to-end solution for creating usable information from raw enriched trajectories*. Specifically, this thesis formulates the following major six contributions to answer the aforementioned research questions:

- **C1: End-to-End solution for creating usable information from raw enriched trajectories** - The first solution provides a generic end-to-end solution for creating usable information from raw enriched trajectories from spatio-temporal semantically enriched data series. Specifically, we provide a roadmap from the derivation of enriched trajectories data, to spatio-temporal data series preprocessing, to semantic enrichment of trajectories, to enriched trajectories data management and a variety of mining tasks such as exposure profiles mining, to an interactive dashboard for real time data visualization, to the implementation of a micro-service based architecture for automating data analytics pipelines.
- **C2: Spatio-temporal enriched trajectory construction** - We design a practical computing platform for *constructing spatio-temporal enriched trajectories* from real-life MCS raw data. Instead of directly extract information

from raw enriched trajectories, we design an intermediate layer, i.e. pre-processing spatio-temporal data series, which can bridge the gap between raw trajectories and purified enriched trajectories. Specifically, we present a computing approach for reconstructing every view of the data, from real-life GPS tracks to every dimension of the multidimensional time series, in terms of cleaning multivariate time series and GPS data from outliers and noise, and interpolating missing values. As a result, a cleaner version of MCS data is achieved.

- **C3: Methods for trajectories semantic enrichment** - To further establish a sound semantic meaning to enriched trajectories, we add a contextualisation layer to the previous preprocessing platform. Semantic enrichment can further construct the enriched trajectories and add information about the context. In order to achieve this semantic enrichment, we develop a model for multidimensional data segmentation based on change point detection (CPD). This model divide the cleaned enriched trajectories into a set of coherent segments, where each segment represent a micro-environment. We contrast the proposed approach with a traditional CPD model and show the effectiveness and scalability of our approach. We further complete the semantic enrichment by designing a hybrid model for context recognition which can integrate geographic view and multivariate time series view to annotated enriched trajectories with the type of activity and movement. The geographic view adds semantic annotations to segments (i.e stop and move annotation; AKA *trajectory segmentation*) from GPS tracks only. The multivariate time series view detect the exact label of segments (e.g. *home, office, store, metro, park, etc.*).
- **C4: Adaptive and flexible system for data management and exploration in MCS coupled with external data** - We further propose a semantic enrichment for sensory data from additional external sources that provides networks data models of fixed stations. Since the two sources of data do not have the same scale, we first propose the transformation of the spatial and temporal dimensions into discrete values to facilitate their query, aggregation and comparison. The fundamental contribution here is the introduction of a multidimensional model for efficiently querying and computing different facets of data. In addition to the computation of several statistical measures, we introduce new operators of spatial and temporal disaggregation to extract, based on machine learning, finer grained data from coarse data and handle the problem of missing values and low MCS coverage.
- **C5: Interactive dashboards for real-time data visualisation** - We present a two-faces visualisation framework of the semantic enrichment and the enriched trajectories. The first dashboard allows users to customise the learning methods for detecting micro-environments and displays the detected

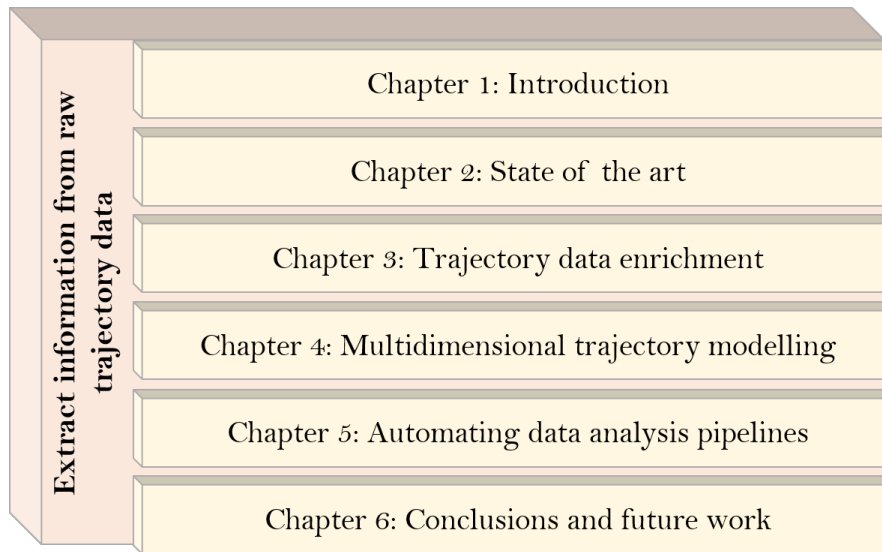


Figure 1.3: Dissertation structure organisation.

micro-environments vis-à-vis the declared micro-environments. The second interactive dashboard displays all the components of the enriched trajectory through time, including mobility, time series and context.

- **C6: Build a microservices based architecture for implementing and automating data analysis pipelines** - We deploy a scalable infrastructure based on micro-service for the whole model lifecycle. We propose an architecture that features the discussed components above (from C2 to C5) to build a scalable and reliable ecosystem for data ingestion, preprocessing, models predictions, storage and visualisation. Figure 1.2 depicts an illustration of the proposed architecture pipelines, which implements our contribution for automating data analysis pipelines based on micro-services, from raw data collection to knowledge extraction.

## 1.5 . Structure of the Dissertation

Figure 1.3 portrays the structure of the dissertation. Specifically, the rest of this thesis is organised as follows:

- Chapter 2 reviews existing works related to enriched trajectory data mining from several perspectives, including preprocessing, segmentation, activity recognition, management and warehousing.
- Chapter 3 focuses on trajectory data enrichment with contextual information. It investigates the time series segmentation approaches as well as activity recog-

tion algorithms which allows to add semantic enrichment to the trajectory data.

- Chapter 4 discusses enriched trajectory modeling requirements and issues. It investigates the mining and exploration of the enriched trajectory data from different perspectives (i.e. temporal, spatial, longitudinal, semantic).
- Chapter 5 presents a scalable infrastructure based on microservices for the implementation of the whole end-to-end data analysis system lifecycle.
- Chapter 5.7 provides general conclusions of this dissertation and highlights some future work directions.

## 1.6 . Scientific Contributions

During this thesis work, several contributions to the research field have been achieved. Twelve articles have been published or submitted in different venues, ranging between workshops, national conference, international conference, journals and book chapters.

1. H. El Hafyani, K. Zeitouni, Y. Taher, L. Yeh, and A. Ktaish, A Multidimensional Trajectory Model in the Context of Mobile Crowd Sensing. To appear in "Intelligent Distributed Computing for Trajectories: Metamodeling, Reactive Architecture for Analytics, and Smart Applications", which is to be published by CRC Press, Taylor & Francis Group, USA.
2. T. Nabil, K. Radja, P. Schembri, K. Zeitouni and H. EL Hafyani (2021). Variations spatio-temporelles de l'exposition individuelle aux polluants urbains mesurée par les micro-capteurs: quelles adaptations des comportements et des politiques urbaines? To appear in CyberGeo 2021.
3. H. El Hafyani, M. Abboud, J. Zuo, K. Zeitouni and Y. Taher. "Learning the Micro-environment from Rich Trajectories in the context of Mobile Crowd Sensing Application to Air Quality Monitoring". To appear in Geoinformatica.
4. H. El Hafyani, M. Abboud and Y. Taher. "A Microservices Based Architecture for Implementing and Automating ETL Data Pipelines for Mobile Crowdsensing Applications". In 2021 IEEE International Conference on Big Data (Big Data) (pp. 5909-5911). IEEE.
5. H. El Hafyani, M. Abboud, J. Zuo, K. Zeitouni and Y. Taher. "Tell Me What Air You Breathe, I Tell You Where You Are". The 17th International Symposium on Spatial and Temporal Databases (SSTD'21). Accepted also in 37ème Conférence sur la Gestion de Données – Principes, Technologies et Applications 2021 (BDA'21).

6. H. El Hafyani, K. Zeitouni, Y. Taher, L. Yeh and A. Ktaish. "Un Modèle de Trajectoire Multidimensionnel dans le Contexte de la Collecte Participative par Micro-Capteurs". 17ème journées Business Intelligence & Big Data (EDA'21).
7. M. Abboud, H. El Hafyani, J. Zuo, K. Zeitouni and Y. Taher. "Micro-environment Recognition in the context of Environmental Crowdsensing". In Big Mobility Data Analytics with EDBT 2021 (BMDA'21).
8. M. Brahem, H. EL Hafyani, et al., Data Perspective on Environmental Mobile Crowd Sensing, In Intelligent Data-Centric Systems, "Intelligent Environmental Data Monitoring for Pollution Management", Academic Press, 2021, Pages 269-288.
9. H. El Hafyani, K. Zeitouni, Y. Taher, M. Abboud. Leveraging change point detection for activity transition mining in the context of environmental crowdsensing. The 9th SIGKDD International Workshop on Urban Computing, San Diego, CA, 24/08/202. Accepted also as short research paper at 36ème conférence sur la Gestion de Données – Principes, Technologies et Applications 2020 (BDA'20).
10. H. El Hafyani. 2020. "Big Data Series Analytics in the Context of Environmental Crowd Sensing". The IEEE International Conference on Mobile Data Management (MDM'20), Versailles, France, 30/06-03/07/2020.
11. H. El Hafyani, K. Zeitouni and Y. Taher. "Micro-Environment Recognition in the Context of Mobile Sensing - A Holistic Approach", PhD symposium, 35ème Conférence sur la Gestion de Données – Principes, Technologies et Applications (BDA'19), Lyon, France, 2019.
12. M. Brahem, K. Zeitouni, L. Yeh and H. El Hafyani, "Prospective Data Model and Distributed Query Processing for Mobile Sensing Data Streams", Workshop On Multiple-Aspect Analysis Of Semantic Trajectories (MASTER 2019) In conjunction with ECML/PKDD Würzburg, Germany, 16-20/09/2019.
13. M. Brahem, M. Chachoua, H. El Hafyani, Z. Kedad, A. Ktaish, S. Mehanna, C. Ray, Y. Taher, R. Thibaud, L. Yeh and K. Zeitouni. "Polluscope – Vers un observatoire participatif de l'exposition individuelle à la pollution de l'air et de ses effets sanitaires", SAGEO 2019, Clermont-Ferrand 13-15/11/2019.

## 2 - State of the Art

### Contents

---

<b>2.1</b>	<b>Introduction</b>	<b>28</b>
<b>2.2</b>	<b>Trajectory Data Preprocessing</b>	<b>29</b>
2.2.1	GPS Trajectory Data Noise Filtering	29
2.2.2	Time Series Noise Filtering	30
2.2.3	Discussion	30
<b>2.3</b>	<b>Trajectory Data Segmentation</b>	<b>31</b>
2.3.1	Stop & Move Detection	31
2.3.2	Move Episodes Segmentation	33
2.3.3	Time Series Segmentation	33
2.3.4	Discussion	35
<b>2.4</b>	<b>Activity Recognition</b>	<b>36</b>
2.4.1	Activity Recognition from GPS Trajectories	36
2.4.2	Activity Recognition from Wearable Sensors	39
2.4.3	Discussion	41
<b>2.5</b>	<b>Generic Machine Learning Algorithms</b>	<b>41</b>
2.5.1	Multivariate Time Series Classification	42
2.5.2	Multi-View Learning	43
2.5.3	Discussion	44
<b>2.6</b>	<b>Moving Object Databases and Warehousing</b>	<b>44</b>
2.6.1	Moving Object Databases	44
2.6.2	Trajectory Data Warehousing	45
2.6.3	Other Works on DW	48
2.6.4	Discussion	49

---

## 2.1 . Introduction

Understanding user mobility from GPS and sensory data is a central theme in ubiquitous computing. Intensive and extensive researches have been done in the field of *sensory data mining* to extract usable knowledge from raw data. In this chapter, we review a large number of state-of-the-art studies related to the analysis and management of *enriched trajectory* data. All the figures presented in this chapter are constructed by the author.

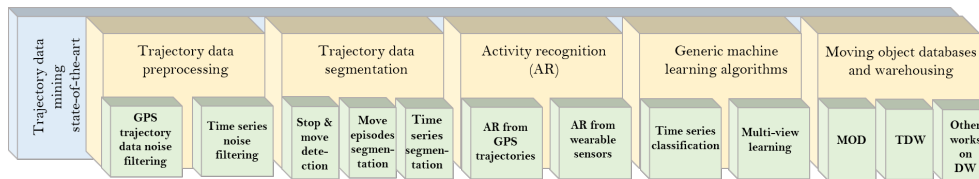


Figure 2.1: Structure of state-of-the-art chapter.

Primarily, a *trajectory* is defined as a multivariate time series in which one dimension represents the spatial positions. In contrast, *enriched trajectories* are trajectories enriched with other information such as the context or the whereabouts of the moving object. Thereby, *enriched trajectories* in the context of MCS are constructed from the combination of GPS tracks with temporal measurements plus other contextual information of the moving object. Therefore, we review the literature related to mining this type of enriched trajectory data according to the structure shown in Figure 2.1, which emphasises five components:

1. Trajectory data preprocessing.
2. Trajectory data segmentation.
3. Activity recognition.
4. Generic machine learning algorithms.
5. Moving object databases and warehousing.

The structure of enriched trajectory data mining shown in Figure 2.1 was inspired from the survey of Zheng [140], which provides a review of the existing techniques that allow to get insightful information from raw trajectory data. In his survey, Zheng explores the connections and differences between the existing methods in the literature and establishes a paradigm for trajectory data mining. The proposed paradigm allows to turn raw trajectory data into usable information by going through several steps, which have been classified in four groups: (i) the derivation of trajectory data, (ii) trajectory data preprocessing, (iii) trajectory data management, (iv) and a variety of mining tasks such as trajectory pattern mining and trajectory classification. While the author lists a variety of trajectory data

mining approach, an overall approach that combines temporal data besides GPS data is missing.

Therefore, in this chapter, we conduct a comprehensive review of related studies to *enriched trajectory* mining in term of preprocessing, segmentation, activity recognition, and management and warehousing as shown in Figure 2.1. We differentiate between trajectory data mining and time series mining since the combination of these two elements constitutes *enriched trajectories*.

## 2.2 . Trajectory Data Preprocessing

Zheng [141] describes the basic techniques that one needs to process trajectory data before starting the mining tasks. These techniques evolve around noise filtering, stay point detection, and trajectory segmentation. In this section, we will focus mainly on noise filtering, whilst trajectory segmentation and stay point detection will be the subject of Section 2.3. The objective of noise filtering is to remove noisy data that may bias the mining. Certainly, sensory data is never perfectly accurate. It is subject to several errors due to sensors noise and other factors such as poor receiving positional signals in urban areas. Filtering these imperfections aims to ensure the quality of the data before starting the mining tasks.

Elnahrawy and Nath [45] differentiate between the different sources of errors and broadly classify them into categories: systematic errors (bias) and random errors (noise). Systematic errors, on the one hand, arise due to changes in the operating conditions (e.g. temperature, humidity, etc.) or other factor such as aging of the sensor. They can be corrected by calibration as has been done in [77]. Random errors, on the other hand, comprise the unpredictable variation from one measurement to another, and they occur due - but not limited - to random hardware noise, noise from external sources, and environmental factors. We are particularly interested in random errors and how to reduce their effects on sensor readings since they may affect the readings precision. Particularly, the main objective is to overcome data quality issues within sensory data by employing some rule-based models or mathematical techniques. These techniques concern both trajectory data as well as time series data.

### 2.2.1 . GPS Trajectory Data Noise Filtering

Unfortunately, it is common to get noisy data in the form of inaccurate readings from GPS devices due to several reasons such as poor receiving signals. Although this problem is not completely solved, several studies take interest in filtering such noise. The most common approaches focus on heuristic methods to remove noise points directly from the trajectory, as has been done in GeoLife [144] and Scikit-mobility [101]. These heuristic approaches consist of first calculating the travel speed between consecutive points in the trajectory based on the distance and the time spent between a point and its successor. If the speed exceeds a certain



threshold - which is parameterised by the user - (e.g. 300 km/h), the segment of the two consecutive points is cut off.

However, Hendawi *et al.* [64] suggest that not all noisy GPS data should be discarded. Instead, They should be analysed since they may reveal some information about the place where the user is such as indoor spaces. For this purpose, Hendawi *et al.* focus on discovering the patterns of noisy GPS data and relate them to specific locations. For example, it is possible to “infer that a driver is passing by a tall building or through a forest based on the pattern of noise in the GPS readings”. Thereby, and by tracking the noise pattern, the authors come to the conclusion that it is possible to identify the road types, the height of surrounding buildings, the existence of urban constructions, and the existence of tunnels.

### 2.2.2 . Time Series Noise Filtering

The detection of noise and outliers in time series typically requires dedicated techniques of different types. Gupta *et al.* [58] and Blázquez-García *et al.* [18] provide an extensive and structured survey of latest techniques that have been widely used in the time series outlier detection domain. The authors discussed a numerous existing approaches ranging from statistical methods (such as Auto regressive (AR) models, and ARIMA models) to more complex and advanced methods (such as deep learning based approaches). The authors of [58] also provoked some studies in the literature that aim at detecting and filtering the noise from time series data such as the work of Cheng *et al.* [30].

In another line of work, Palshikar [100] proposes a method based on peaks detection (aka, spikes detection) to clean time series data from outliers. The peaks are identified as a sudden increase in the values of time series. While these spikes are easy to identify visually in a short time series, but it is not obvious to detect them automatically in any time series. Therefore, they introduce a formal notion of a peak in time series, and propose several algorithms for peak detection. They define the local peak as the point that is the local maximum within a window size, and it is isolated i.e. the value of this point is not similar to the other values in that window. The detection of peaks needs a user-defined threshold of the window average to differentiate between local outliers and real values.

### 2.2.3 . Discussion

In this section, we presented the difference between systematic errors and random errors, with a focus on the latter one. We discussed how to filter random noises from GPS data and time series based on heuristic methods and mathematical techniques. Heuristic methods based on a speed limit has shown remarkable results in filtering noise from GPS data as shown in [144] and [101]. However, setting the speed limit remains of a discussion and empirical tests need to be performed in order to find the optimal speed limit that allows to detect real noises. As for time series, whilst the work done by [100] is very useful, the human still needs to interact with the algorithm to set appropriate thresholds for the algorithms.

Automatizing the threshold choice based on the window mean value is very much needed.

### 2.3 . Trajectory Data Segmentation

In many real-world applications, we need to segment the trajectory data into coherent segments for further data analysis. The segmentation enables to mine rich knowledge from the trajectory data, such as the inference of stay locations. This is considered as the first step of semantic enrichment of data. The mobility literature is rich with approaches and methods proposed for trajectory data segmentation. Each proposal is built on the background of some specific application domain, such as deriving the stay area of a moving object, or detect transportation mode transitions within a trajectory, and aims at achieving the objectives of this specific application domain.

As a matter of fact, this modelling approach is very similar to the Time-Geography representation proposed by Hägerstrand [62], which is the first successful proposal to provide a conceptual approach for representing and analyzing human activity in space and time. Plus, Figure 1.1 is also inspired from the three-dimensional representation of a trajectory modelled according to the framework of Time-Geography. Whereas Time-Geography, which was featured before the era of smartphones, focused on modeling the constraints on human activity in space and time, the large collection of location data allows a more analytical approach of this paradigm. However, this has raised new challenges in terms of data pre-processing, among which trajectory reconstruction, stop and move detection, and activity labelling. Also, by instanciating the concept of Time-Geography to real life measurements, one has to define the scope (are we talking about a typical daily life mobility or cover a while period of interest?), and the detail level (should we distinguish between "stations" when the activity changes such as shopping and dining in the same mall? and to what extent a multi-modal journey would be detailed?) [90, 91]

In this section, we focus on and discuss three types of segmentation methods for trajectory data, namely: (i) stop & move detection based on location, (ii) move episodes segmentation, which includes trajectory segmentation based on the speed's time series, and (iii) general method of change-point-based segmentation for time series.

#### 2.3.1 . Stop & Move Detection

As mentioned above, trajectory segmentation is driven by the application domain. If the trajectory consists of several trips, such as going from start point A, passing by point B, then point C, to an end point D (cf Figure 2.2), a very intuitive way of segmenting this trajectory is to split it into segments where the moving object is stationary, and segments where the moving object is actually moving. The former segments are called stop segments while the latter are called move

segments. We refer to this approach as stop and move segments detection. With this presentation, between two stop segments, there is always a move segment.

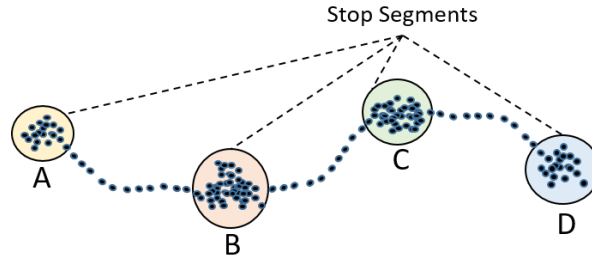


Figure 2.2: Stop and move segments in a trajectory.

The research challenge with this representation is the detection of stop and move segments. Parent *et al.* [102] discuss the latest approaches proposed for finding stops within a trajectory. The authors examine the existing criteria and assumption for the detection of stops episodes. Some of these assumptions are based on the spatial distance and temporal duration as investigated in [145]. In this work [145], Zheng *et al.* compute the stop episodes as sequences of consecutive GPS points, where their spatial distance is below a threshold and the temporal duration is higher than another threshold. In fact, the first work on stay point detection was proposed by Li *et al.* [81]. The proposed algorithm works as follows: first, it checks whether the distance between an anchor point (e.g.  $P_5$ ) in a trajectory and its successors is below a given threshold (e.g. 100 meters). Then, it measures the time span between the anchor point and the last successor (e.g.  $P_8$ ) within the distance threshold. If the time span is greater than a given threshold, a stop point (which comprises  $P_5, P_6, P_7,$  and  $P_8$ ) is detected; the algorithm starts from  $P_9$  to detect the next stop point. This work was further improved by Yuan *et al.* [137] based on the idea of density clustering. After finding the stay point (from  $P_5$  to  $P_8$  with  $P_5$  as an anchor point), their proposed algorithm further examines the successor points from  $P_6$ . For example, if the distance between  $P_6$  and  $P_9$  is within the threshold,  $P_9$  will be added to the stay point.

Other methods identify the stop segments by employing a combination of spatio-temporal indicators. For example, Yan *et al.* [133] detect the stop and move segments by taking into account several spatio-temporal criteria such as position density, velocity and direction. Other line of methods discover the stop segments by using only coordinates and time stamps from continuous GPS trajectory, without referring to the feature of velocity. For instance, Gong *et al.* [54] propose a two-step approach for stops detection based on a variant of DBSCAN [46] algorithm and support vector machine (SVM). The proposed variant, named DBSCAN-TE (which stands for density-based spatial clustering of applications with noise plus temporal and entropy constraints) identify the stops clusters, and feed the output to SVM in order to distinguish between activity stops (such as work, shopping,

etc.) and non activity stops (such as waiting for a green light, being stuck in a congestion, etc.).

### 2.3.2 . Move Episodes Segmentation

Trajectory data can be segmented with other specifications than stops and moves. For example, one of the popular criteria for segmenting trajectory data is transportation means. Zheng *et al.* [143] use speed and acceleration to segment trajectory data and identify change points. Based mainly on the common sense that people walk between two transportation modes, the authors conclude that the start point and end point of a *Walk Segment* could be a change point. Therefore, they first distinguish between walk point and non-walk points using a loose upper bound of velocity and that of acceleration. Then, they construct walk and non-walk segments by respectively consecutive points walk and non-walk points. As a result, the start point and end point of each walk-segment are potential change points to segment the trip. A similar approach was developed by Liao *et al.*. Using a Gaussian mixture model based on velocity, the authors classify the trajectory data into three speed ranges: walking, low speed, and high speed. Therefore, a new segment is created whenever there is a change in the speed range.

Beyond using any fixed spatial or temporal threshold, Bonavita *et al.* [19] provide a general methodology that inspects users mobility, and identifies segmentation thresholds that could match their mobility features. The proposed approach allows to avoid any input parameters and adapts the thresholds to user's trajectory data. Another work proposed by Yan *et al.* [133] segments the data based on semantic information such as road categories and public transport networks. For instance, if the user takes a bike path, we can determine that the transportation mode is either walking or bike, but certainly not bus, train or car. Also, the location of bus stops and train stations are indicating whether the the traveling mean is bus or train. We discuss further trajectory data annotation in Section 2.4.1 and how to extract the usable information from trajectory data.

### 2.3.3 . Time Series Segmentation

Time series segmentation refers to the problem of segmenting data into coherent segments. For instance, and depending on the application requirements, the segmentation can be used to improve the performance of people activity recognition by segmenting the data into the segments into non-overlapping segments, each segment represents an activity. In this thesis work, we mainly focus on change point detection which offers a valuable opportunity for univariate time series segmentation and the detection of activities breakpoints. While it is out of our focus, we list for reference some related work to time series segmentation that does not use change point detection, but may concern on multivariate time series segmentation.

Existing change point detection approaches in the literature can be classified in term of sensing technology, types of activities, segmentation methods, and online

or offline techniques. Offline time series change point detection techniques store the whole data set at once, and look for point locations where the changes have occurred based on a global view of the data. Online time series change point detection is an extension of the offline change point detection methods, where an offline change point detection is applied on each newly arrived sequence of data points. The survey of Aminikhanghahi *et al.* [6] enumerates, categorizes, and compares methods that have been proposed to detect change points in time series in both batch and online modes. The choice of the method depends on the desired outcome of the algorithm. The problem of activity segmentation constitutes one of the main interests of change point detection in IoT. Several advances based on wearable sensors [85], camera [2], or Smart Home [7] exist in the literature for collecting information, segmenting data as well as understanding and inferring human activities. As a matter of fact, the problem of activity recognition and activity segmentation are heavily inter-connected topics since the starting point of human activity recognition is to detect the transition points and label them with activities.

Figure 2.3 shows an example time series that contains several change points depicted by the green dashed lines. The time series illustrate the change of temperature as well as the change of activities over 9 hours. This plot highlights that the human changes its activity 9 times. The segment between two consecutive change points is referred to as an activity. Thereafter, the objective of change point detection is to discover these activity borders with change point detection.

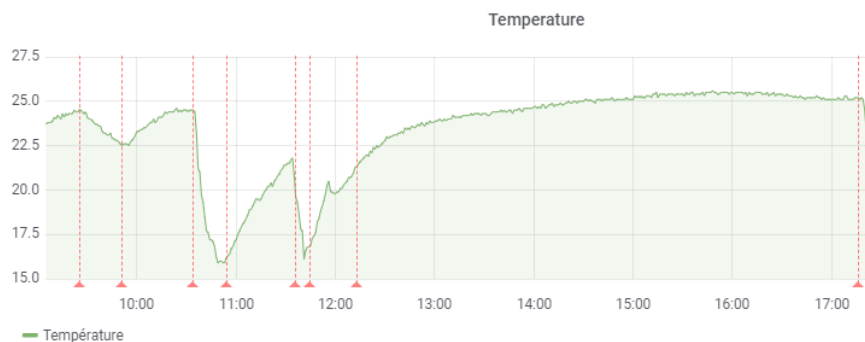


Figure 2.3: A sample of time series with several change points depicted by the green dashed lines.

In their survey paper, Aminikhanghahi *et al.* [6] provide an exhaustive review of the existing supervised (e.g. Decision Tree, *Support Vector Machine*, etc.) and unsupervised machine learning algorithms (e.g. *Cumulative Sum*, *Bayesian* model, etc.) that have been designed for change point detection. The supervised methods take a training set to learn a mapping to a target attribute from an input data. In supervised learning, data is already labeled by activity classes collected during data collection or provided by an expert. Unsupervised learning algorithms, on

the contrary, are used to discover change in pattern within unlabeled data. Since we are dealing with time series data, those approaches can be used to discover transitions based on statistical properties of time series without prior knowledge on class labels or a training set.

In another work, Aminikhanghahi *et al.* [7] used two different unsupervised methods to detect the transition in time series on unlabeled data. The first method is Relative Unconstrained Least-Squares Importance Fitting (RuLSIF), and the second one is Bayesian Change Point Detection (BCPD). The results show that the best performance for activity transition detection can be achieved with RuLSIF algorithm. However, setting appropriate values for RuLSIF parameters has a major influence on the change point detection results. The same authors proposed in another work a real-time non parametric change point detection, called SEP, which uses Separation distance as a divergence measure to detect change points in high-dimensional time series [9]. The goal of the proposal is to further advance the line of research in density ratio change point detection algorithms, and introduce a new unsupervised algorithm for change point detection in time-series data using the Separation distance metric. The results show that their algorithm exhibits similar behaviour as Kullback-Leibler importance estimation procedure (KLIEP) and uLSIF (Unconstrained Least-Squares Importance Fitting) estimation, which uses Pearson (PE) divergence as a dissimilarity measure.

Sadri *et al.* [109] propose an offline change point detection method based on Information Gain Theory. The authors proposal takes the number of change points and detects changes that affect the mean of the time series. The proposed method can also detect changes in the variance of the time series using a moving window. However, setting in advance the number of change points in the time series may affect the change point detection performance.

Liu *et al.* [85] propose two different ways for time series segmentation (not related to change point detection). The first methodology is an explicit segmentation whereby the data stream is segmented to a set of subsequences using certain window size and sliding length. However, the two parameters affect directly the transition detection precision. The second method is a sensor event-based segmentation, which divides the data stream into subsequences containing certain number of sensor events. This approach can dynamically adjust the window size to fit different activities during recognition.

Concerning multivariate time series segmentation, Gharghabi *et al.* [53] present a domain agnostic algorithm for multidimensional time series segmentation (which can handle data streaming at a high rate). The proposed approach called Fast Low-cost Unipotent Semantic Segmentation (FLUSS) exploits the Matrix Profile structure introduced in [135] and detects changes in the shape of the time series.

#### 2.3.4 . Discussion

To summarize, the studies of enriched trajectory data segmentation in the literature can be seen from two perspectives. The first one is based on GPS trajectory

data, and aims first at segmenting data into stop & move segments, then detect transportation means transitions within move segments by referring to the speed's time series. It is true that speed has shown good results in partitioning people's trajectories into several segments according to the change of transportation mode as shown in [142], however, speed is very sensitive to traffic conditions and weather. During a traffic jam, the average velocity of driving would be as slow as cycling, and then the change in participant activities may not be captured.

The second perspective is founded on time series segmentation. In this thesis work, we reviewed existing studies of change point detection methods for univariate time series segmentation. Whilst the proposal of [7] showed the best performance with RuLSIF algorithm for activity transition detection, setting appropriate values for RuLSIF parameters has a major influence on the change point detection results. Similarly, in the proposal of [109], which can detect changes in the variance of the time series using a moving window, setting the number of change points in the whole of the time series affect the change point detection results. Besides, with the study of [85] where the authors use a certain window size and sliding length to segment the data, two different activities may appear in the same window. Therefore, detecting the exact change point timestamp may not be possible.

Furthermore, the above discussed methods concern only mono-variate time series. While the work of [53] can be generalized to address multidimensional time series, it assumes the existence of prior knowledge about the subset of time series dimensions that are relevant for detection of each change point. In the context of environmental crowdsensing, the multivariate time series' dimensions do not contribute evenly in the detection of activity transition, and no prior knowledge about the weight of each dimension is known.

## 2.4 . Activity Recognition

Human activity recognition involves a wide range of applications from smart homes activities [8] to daily human activities [138][85][31], to human mobility [37][142] to cite a few. It represents a typical scenario of machine learning, and some public datasets are widely used in the benchmarks. In this section, we review some activity recognition studies from mobile sensing data. We mainly focus on inferring activities from GPS trajectories and wearable sensors.

### 2.4.1 . Activity Recognition from GPS Trajectories

In the last decades, several studies start to focus on activity recognition using GPS-based trajectory data. This type of problems aims to tag raw trajectory data (or its segments) with semantic labels that enables to understand the trajectory data on a semantic level and the mobility of the moving object as shown in Figure 2.4. The semantic information can be used in several applications, such as identifying the most visited places by the moving object or offering trip recommendation.

In his survey, Zheng [141] differentiate between application requirements for

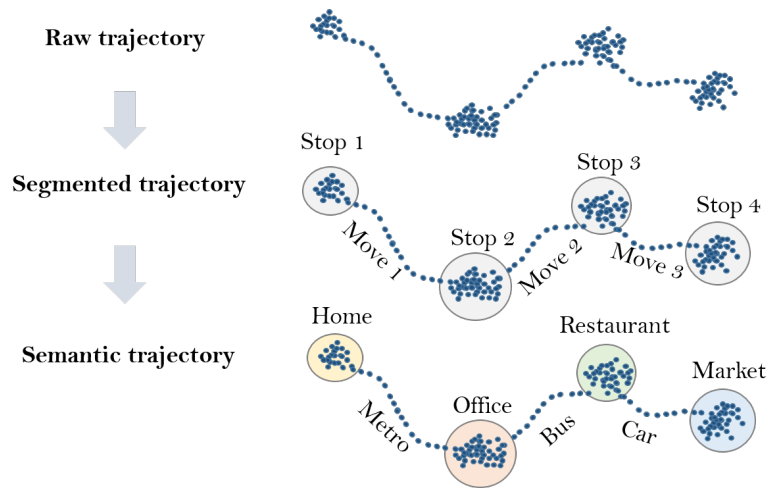


Figure 2.4: From raw trajectory data to semantic trajectory.

tagging raw trajectory data (or its segments) with a semantic label such as transportation modes inference and human activities detection. Following his proposed diagram, the activity recognition from GPS trajectory abides by 3 steps: (1) Segment trajectory data using a segmentation methods. For instance, segment trajectory data into stop & move segments, then segment the move segments according the change of transportation means. (2) Extract features for each trajectory segment (or point). (3) Build a model that allows the classification of segments (or points). Figure 2.5 depicts the general data flow for activity recognition from GPS trajectory data. Since trajectory is primarily a sequence, sequence inference models such as Dynamic Bayesian Network (DBN), Hidden Markov Model (HMM) and Conditional Random Field (CRF), can be used to integrate information from trajectory local points (or segments) and the sequential patterns between adjacent points (or segments). In a more recent survey, Mazimpaka and Timpf [88] review some generic methods for trajectory data mining and the relationships between them. The authors comply with Zheng’s approach and state that most trajectory classification algorithms follow a traditional two-step approach: first extracting a set of discriminative features and then using the extracted features to train an existing standard classification model.

Rehrl *et al.* [105] propose and evaluate a three-step trajectory data mining approach based on machine learning techniques. The authors focus on the detection and classification problems of stops points in vehicle trajectories. The proposed approach describes three mining steps that comprise stop detection, feature extraction, and stop segments classification. The authors first segment the trajectory into stay points clusters (refer to Section 2.3.1 for stay points detection), then after extracting 14 characteristics of each stop, they classify the detected stops into two categories: traffic-relevant and non-traffic-relevant stops.

As for the field of transportation mode detection, the research community has



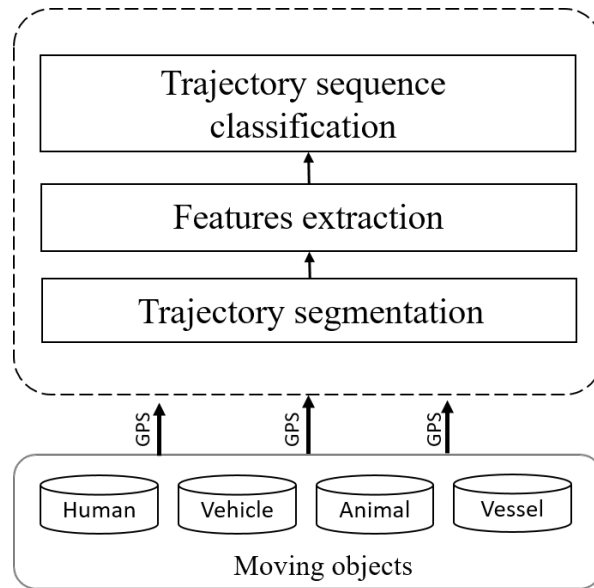


Figure 2.5: General data flow for activity recognition from GPS trajectory data.

provided several extensive works based on machine learning techniques. Etemad *et al.* [47] provide a framework for the prediction of transportation mode based on GPS data only. The key contribution of the authors work is to propose trajectory point features generation and trajectory segments feature extraction which comprise bearing rate, the change rate of the bearing rate and the global and local trajectory features. Thereafter, with the help of several machine learning algorithms such as *SVM*, *Decision Tree*, and *XGBoost*, the authors attempt to classify the moving object's trajectory segments by transportation modes, which comprise walking, train, bus, bike, driving, etc.

Instead of using hand-crafted features with traditional machine learning algorithms, Dabiri and Heaslip [34] propose a travel mode inference model based on convolutional neural network (CNN) schemes which are able to automatically drive high-level features from raw input. The authors predict travel mode labels, which include walk, bike, bus, driving, and train, from raw GPS trajectory data. The proposed approach based on CNN architectures attains state-of-the-art accuracy on GPS data from GeoLife dataset [143].

In the work of Zheng *et al.* [142], and based on GPS logs, the authors propose a supervised learning-based approach to infer people's transportation modes, including driving, walking, bus, and bicycle. The authors start by defining a set of features that are more robust to traffic conditions, such as heading change rate, stop rate, and velocity change rate, etc. Thereafter, and by using change point-based detection to segment the trajectory data (refer to Section 2.3.1), the authors extract features from each segment and employ them to train a supervised-learning

model from transportation mode inference.

In another line of work related to vessels trajectory mining, Kontopoulos *et al.* [74] attempt to classify vessel trajectories from real time stream into three activities, which include trawling, longlining, and under way. The proposed method first splits vessel trajectories into temporal segments of 1, 2, 4, 8, 12, and 24 hours, and generate a set of features that are representative of the activities in question, such as the average speed and its standard deviation, the average drift and its standard deviation, and number of turns. The authors then train a *Random Forest* classifier and compare it against three other classifiers: *Gradient Boosted Tree*, *Linear Discriminant Analysis*, and *Logistic Regression* to learn the fishing activities of the vessels.

### 2.4.2 . Activity Recognition from Wearable Sensors

With the recent exceptional development of mobile devices and high-computational, small-sized, and low-cost sensors, an active area of research has emerged with the main goal of extracting information from data collected by pervasive sensors. In particular, human activity recognition from wearable sensors constitutes a high interest task for researchers within the field, and covers a wide range of applications, including, but not limited to, monitoring diabetes or heart disease patients with their daily routines, human computer interaction, and following athletic activities. Therefore, recognizing activities such as running, walking, standing up, and raising hand is fundamental to provide feedback about the application scenario. In this section, we review the existing work related to human activity recognition (HAR) from wearable sensors.

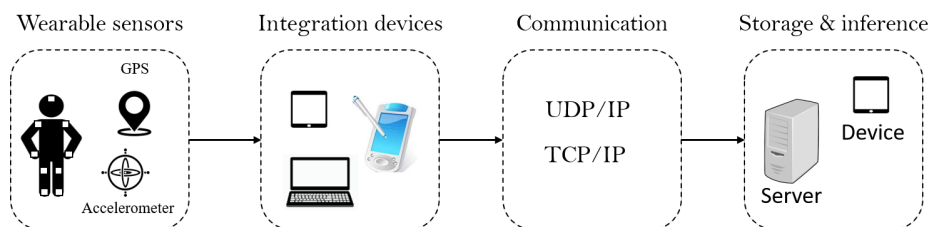


Figure 2.6: Generic data acquisition architecture for Human Activity Recognition.

In their survey, Lara and Labrado [78] examine the state of the art in HAR based on wearable sensors, and propose a generic data acquisition architecture for HAR system based on wearable sensors. Figure 2.6 illustrates the data acquisition structure from wearable sensors to the storage in a local device or a remote server. First, sensors that are attached to the human's body measure information of interest about a certain phenomena such as motion [67], location [32], temperature [104], ECG [69], etc. These wearable sensors communicate with an integration

device such as cellphone, PDA, laptop or a customized embedded system. The received data from the sensors are then sent to an application server for visualization, analysis or real time monitoring. UDP/IP or TCP/IP can be used as communication protocol, according to the desired level of reliability. It is worth mentioning that these components are not necessarily all implemented in every HAR system, and their deployment depends on the application scenario.

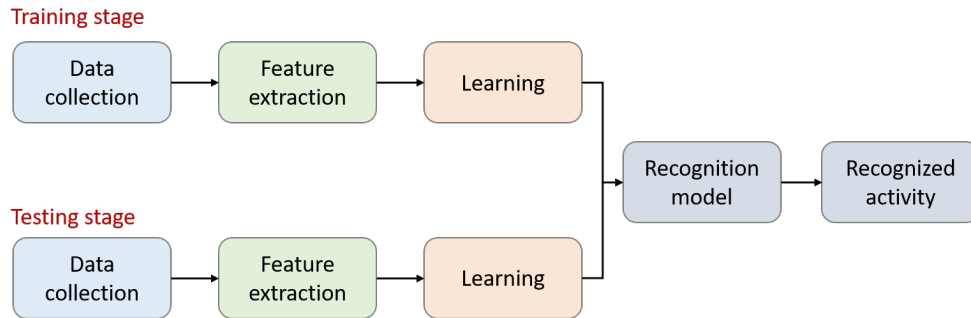


Figure 2.7: HAR system architecture based on wearable sensors.

Furthermore, the authors present a general architecture of any HAR system. They argue that activity recognition from wearable sensors, as any other machine learning application, goes under two stages: the training phase and the testing (also called evaluation) phase. Figure 2.7 shows the common phases involved in these two stages. During the training phase, the time series signals are split into time windows on which features extraction is applied and relevant information is filtered. Thereafter, machine learning methods are used to generate a HAR model from the extracted features dataset. Similarly, during the testing phase, data segmented according to a time window. The segmented data are used to extract the same features and evaluate the priorly trained model.

Following the same HAR system architecture, Parkka *et al.* [104] used several data signal acquired by wearable sensors, such as accelerometer, microphone, and air pressure, to classify everyday activities such as walking, running, and cycling. In their approach, the authors segmented the signal into 1-s segments (72 272 segments were used), and extract six features, such as peak frequency of up-down chest acceleration, median of up-down chest acceleration, peak power of up-down chest acceleration, from these segments. For classifying the segments into daily activities, three different classifiers, namely custom decision tree, an automatically generated decision, and an artificial neural network (ANN) were used.

In another interesting work of the healthcare assessment domain, Zhang and Sawchuk [138] present a framework based on Bag-of-Features (BoF) to build activity recognition models using motion primitive symbols. The authors validate the effectiveness of their BoF-based approach for the recognition of nine activity classes, which are walking forward, walk left, walk right, go up stairs, go down stairs, run forward, jump up, sit on a chair and stand. In contrast, Liu *et al.* [85]

build a dictionary of time series patterns (called *shapelets*) to address the problem of complex activity recognition, such as gestures or actions, from multiple sensors. The authors extend the concept of shapelet to represent complex activities by re-defining the shapelet as a representation of the activity.

Alternately, and instead of using handcrafted features from sensors signals, the study of Jiang *et al.* [71] proposes to recognize human physical activities from accelerometers and gyroscopes signals by learning automatically the optimal features. The proposed approach transforms sensors signal sequences into images and use Deep Convolutional Neural Networks (DCNN) to learn the best discriminative features for recognize activities such as walking, standing, and walking downstairs.

Remaining in deep learning-based proposals, Ordóñez *et al.* [98] a framework for activity recognition based on convolutional and LSTM recurrent units. The main key of their proposal is the automatic design of features that does not require expert knowledge. The authors demonstrate the suitability of their framework for activity recognition from wearable sensors to infer activities of locomotion, postures and gestures. Besides, Wang *et al.* [126] provide a summary of the recent advances of sensor-based deep learning approaches for activity recognition.

### 2.4.3 . Discussion

In the last decades, several studies start to focus on activity recognition. We reviewed recent approaches for activity detection and classification from wearable sensors signals and GPS. We covered a breadth of activities, including daily human activities, such as standing, walking, and going up stair, etc., and transportation means, to provide a complete and comprehensive review. In addition to the type of activity, we covered two taxonomy systems for activity recognition based on GPS data and wearable sensors.

In addition, we provided a large set of research proposals that employ feature extraction and classification to infer the type of activity, and whether the processing is based on GPS data or wearable sensors signals. We highlighted that several types of wearable sensors are used in the HAR research proposals, such as accelerometers, microphone, and air pressure.

However, in the context of MCS, we do not use accelerometer or sound since this information is privacy invasive. Furthermore, the previous disused proposals are either based on the geographical or temporal information, but an overall methodological approach for combining these different aspects on real-world enriched trajectory data is missing. This combination may lead to a more robust inference model rather than the usage of a single attribute, and it needs to be investigated, which we will discussed in Chapter 3 of this dissertation.

## 2.5 . Generic Machine Learning Algorithms

The previous activity recognition works are based on data collected from GPS and wearable sensors. As a matter of fact, the human activity recognition problem

is not confined to only these two approaches, but can also falls into other application scenarios where multivariate time series are generated by heterogeneous sources of data [52]. In this section, we review some generic research designs that can be employed to infer the human activity from multivariate time series. We focus on multivariate time series classification (MTSC) and multi-view learning as the main generic machine learning methods.

### 2.5.1 . Multivariate Time Series Classification

Human activity recognition falls in the problem of labelling data segments with the type of activity which leads to a multivariate time series classification (MTSC) problem based on data collected by multiple sources. There is a wide range of time series classification approaches that can be classified into four categories: distance-based methods [12], feature-based methods [104], ensemble methods [52], and deep learning models [48][29][126]. The one-nearest neighbor (1-NN) classifier with different distance measures, such as euclidean distance (ED) or dynamic time wrapping (DTW) [12], is always considered as the benchmark to give a preliminary evaluation in the MTSC problem. Feature-based methods is based on a variety of features learned from TS data, through which we can distinguish the differences between data and classify them. The disadvantages of these methods lie in the complexity and weak generality of building features, which obviously limits their versatility. This type of methods follow exactly the approach discussed above and depicted in Figure 2.7.

Besides hand-engineered features, some methods use deep neural network (DNN) to extract the features of time series for classification. In their survey, Fawaz *et al.* [48] review the current studies of deep learning algorithms for time series classification (TSC), and present an empirical study of the most recent DNN architectures for TSC, including convolutional neural network (CNN), recurrent neural network (RNN), echo state network (ESN), and multi layer perceptron (MLP). Besides univariate time series, the authors tested the approaches on 12 multivariate time series datasets, and give an overview of the most successful deep learning applications.

Considering the real-life scenarios, where it is difficult or expensive to obtain a large amount of labeled data for training, some studies used both labeled and unlabeled data to learn the human activity, that is semi-supervised learning (SSL) [127] on MTSC. The pioneering work by [127] propose a semi-supervised technique for time series classification. The authors demonstrated that semi-supervised learning requires less human effort and generally achieves higher accuracy than training on limited labels. The semi-supervised model [127] is based on the self-learning concept with the one-nearest-neighbor (1-NN) classifier. First, the labeled set denoted by  $P$  (as positively labeled) is applied to train the 1-NN classifier  $C$ . Then, the unlabeled samples  $U$  are given the pseudo labels progressively based on their distance to the samples in  $P$ . Thereafter, the enriched labeled set  $P$  allows iteratively repeating the previous step and improving the classifier. More recently, the

deep learning-based models on MTSC show promising performance under weak supervision. For instance, Zhang et al. [139] propose a novel semi-supervised MTSC model named time series attentional prototype network (TapNet) to explore the valuable information in the unlabeled samples. TapNet projects the raw MTS data into a low-dimensional representation space. The unlabeled samples approach themselves to the class prototype in the representation space, where pseudo labels are generated by the distance-based probability allowing training the model progressively. Moreover, the hybrid convolutional neural network (CNN) and long short-term memory (LSTM) structure adopted in TapNet allows modeling, respectively, the variable interactions and the temporal features of MTS.

### 2.5.2 . Multi-View Learning

Another line of studies propose multi-view learning to classify time series data originated from multiple sensors to recognize user activities. Garcia-Ceja et al. [52] propose a method based on multi-view learning and stacked generalization for fusing audio and accelerometer sensor data for human activity recognition using wearable devices. Each sensor's data is seen as a different "view", and they are combined using stacked generalization [131]. The approach trains a specific classification model over each view and an extra meta-learner using the view models as input. The general idea of the authors is to combine data from heterogeneous types of sensors to complement each other and thus, increase recognition accuracy.

Wang et al. [125] propose a framework based on deep learning to learn features from different aspects of the data based on features of sequence and visualization. In order to imitate the human brain, which can classify data based on visualization, the authors transform the time series into an area graph. Area graph here is used to model time series as images in order to apply Convolutional Neural Network (CNN) on top of it. They use well-trained Long short-term memory with an attention mechanism (LSTM-A) neural networks and CNN with attention (CNN-A) to extract the features of time series data. LSTM-A extracts sequence features, while CNN-A extracts visual features from the time series. Then, based on the fusion of features, the authors carry out the time series classification task. Although the approach gained promising results, further performance gain was achieved by recent deep learning methods such as InceptionTime [49].

Li et al. [82] propose a multi-view discriminative bilinear projections (MDBP) for multi-view MTSC. The proposed approach is a multi-view dimensionality reduction method for time series classification, which aims to extract discriminative features from multi-view MTS data. MDBP mainly projects multi-view data to a shared subspace through view-specific bilinear projections that preserve the temporal structure of MTS, and learns discriminative features by incorporating a novel supervised regularization.

### 2.5.3 . Discussion

To summarize, the HAR problem falls into the problem of multivariate time series classification, with sensors readings as input and the label of the activity as output. However, MCS data is characterised by its heterogeneous property, designating that data is originated from different sensors readings. In fact, some sensors may be offline and do not transfer any data for hours, which may lead to missing data problems. Therefore, the usage of MTSC is not straightforward. Naturally, it is necessary to design a model that combines data from heterogeneous sensors, and has the ability to classify it efficiently even if one (or more) dimension is missing.

Furthermore, multi-view learning approach seems like a candidate solution for the heterogeneity characteristic of the used sensors. As stated in [52], Each sensor's data is seen as a different "view", and the combination of the different views may be achieved by ensemble learning [146]. If some dimensions are missing, the model would have the ability to infer the final label with the available dimensions.

## 2.6 . Moving Object Databases and Warehousing

Over the last decades, the interest in mobility data modeling and warehousing has substantially grown with the larger availability of mobility data generated by moving objects. Certainly, exploiting and managing moving objects trajectories are necessary to discover knowledge about the moving object behaviours in several applications domains which can include traffic monitoring, wildlife migrations and movements, vessels trajectories, visitors behaviour in a museum, and many location-based scenarios. A large number of database researches have been established to enhance the data management field such as spatio-temporal database [80], moving object database [60], and trajectory data warehouse (TDW) [122]. Similar to traditional databases researches, the ultimate purpose of trajectory databases is to establish an ad-hoc data representation and storage for the trajectories of the moving objects. In this section, we summarize database research proposals that deal with trajectory data in terms of moving objects databases (MOD) and trajectory data warehouses (TDW).

### 2.6.1 . Moving Object Databases

Similar to traditional databases researches, the study of moving objects databases starts by defining specific representations of the moving objects spatio-temporal trajectories. These specific representations enable trajectory data querying and processing of moving objects. However, in traditional databases, data is assumed to be constant, unless it is explicitly modified, which can not be applied on moving objects.

Solutions such as the proposal of Wolfson *et al.* [129] offer a representative data model to deals with the continuous variation of the moving object location. Their model called Moving Objects Spatio-Temporal (MOST) was proposed for modeling dynamic attributes whose value changes continuously with time. DOMINO

(Databases fOrMovINg Objects tracking) which is a corresponding moving object database prototype, was produced by Wolfson *et al.* [130]. Built on top of existing object-relational databases (e.g., Oracle), the system provides temporal capabilities, uncertainty management, and location prediction.

Another major research prototype for MOD is built by Güting *et al.* [61] and called SECONDO. SECONDO is an extensible database system which supports new kinds of data models, especially spatial and spatio-temporal database models, and consists of three major components namely the kernel, the optimizer, and the GUI. The kernel is implemented on top of BerkeleyDB and written in C++. It includes an extensible list of algebras to support query processing. The optimizer is implemented in PROLOG, and provides SQL-like syntax with conjunctive query optimization. The GUI is written in Java and offers an interface for visualizing the moving objects.

In addition, SECONDO provides two algebras called Parallel SECONDO [87] and Distributed SECONDO [95] for big data management. Parallel SECONDO, developed by Lu and Güting [87], is a hybrid parallel processing system built upon Hadoop and a set of single-computer SECONDO databases to achieve an efficient parallel system for the mobility data processing. Following the same concept, Distributed SECONDO, developed by Nidzwetzki and Güting [95], uses Apache Cassandra as data storage and the extensible DBMS Secondo as a query processing engine.

Besides SECONDO, HERMES is a similar prototype database engine for moving objects developed by Pelekis *et al.* [103]. The system provides a SQL-like query language on top of OGC-compliant ORDBMS, namely Oracle and PostgreSQL. HERMES exploits the spatial data types available in Oracle and extends them through a new data cartridge with the MO types.

On top of the above mentioned trajectory database systems, a new research prototype of trajectory database called MobilityDB was proposed by Zimányi *et al.* [147]. Based on PostgreSQL and PostGIS, MobilityDB extends their type system with abstract data types for representing moving object data. For example, the system defines the `tgeompoint` type to represent moving geometry point objects. This integration allows a seamless reuse of the powerful data management features of the platform. Furthermore, MobilityDB is built on top of the existing operations, indexing, aggregation, and optimization framework.

### 2.6.2 . Trajectory Data Warehousing

It is certain that MOD uses optimized queries and indexing techniques. However, according to Leonardi *et al.* [80], these queries are costly in terms of processing and fetching the required data because of the usage of many JOIN operators. Therefore, an alternative solution for handling trajectory data is to use trajectory data warehouses technologies.

Data warehouse (DW), which uses the concept of multidimensional models, provides tools to integrate heterogeneous sources of data into one common storage



repository for further analysis. The concept of multidimensional models organises data into a set of dimensions and fact tables. A fact table is composed of different dimensions and measures, whereas dimensions provide information for the measures included in the fact table. Figure 2.8 shows an example of the multidimensional model for a company data warehouse that stores information about sales. DW provides different analytical tools such as Online Analytical Processing (OLAP) and data mining techniques to extract usable knowledge. Likewise Trajectory Data Warehouse (TDW) can take advantage of DW utilities, and exploit moving object data using OLAP and data mining techniques. In this section, we review the existing studies in the literature on trajectory data warehousing.

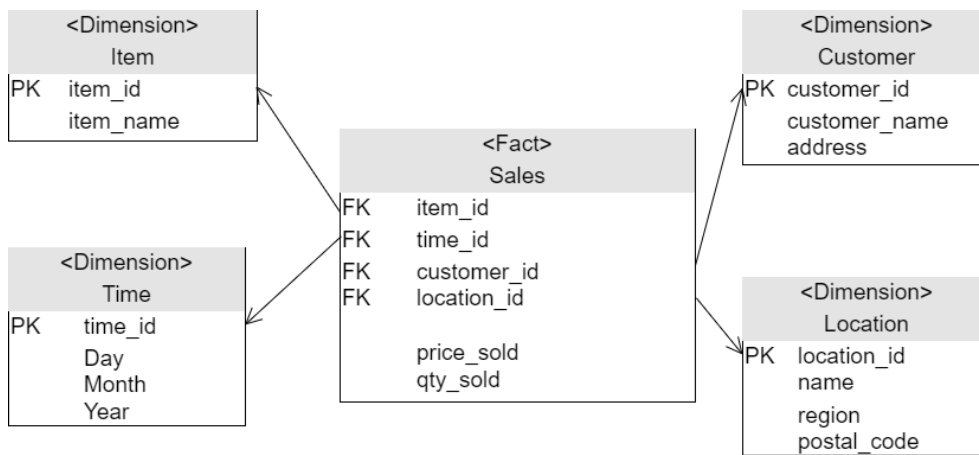


Figure 2.8: An example of a multidimensional model.

In their survey, Alshafi *et al.* [4] provide a review on existing studies of the management, the storage, and the analysis of trajectory data using data warehouse technologies. In addition, the authors propose a framework that summarizes the requirements to build the TDW. Figure 2.9 describes the structure of the proposed framework. First, trajectory data is collected from different data sources. Then, the extraction and integration of data from the different data sources is performed, followed by a step of transformation of the data to the desired format. Thereafter, after loading the data to the TDW, OLAP, data mining and visualization techniques can be performed.

Recent studies follow the same structure as in Figure 2.9 to design a framework for TDW. For instance, Leonardi *et al.* [80] create two TDW prototype systems; the first one is for vessels trajectories and the other for cars trajectories moving along a road network. The trajectory data cube of the vessels scenario consists of two fact tables and six dimensions including the spatial and temporal dimensions, as well as non-spatial dimensions. The spatial domain granularity is structured by setting a grid of rectangles of size  $330 \text{ m} \times 440 \text{ m}$ . The spatial hierarchy is then drawn by a collection of regular grids of increasing size. As for the temporal dimension, the base granularity of the hierarchy is a one day interval. For the road traffic

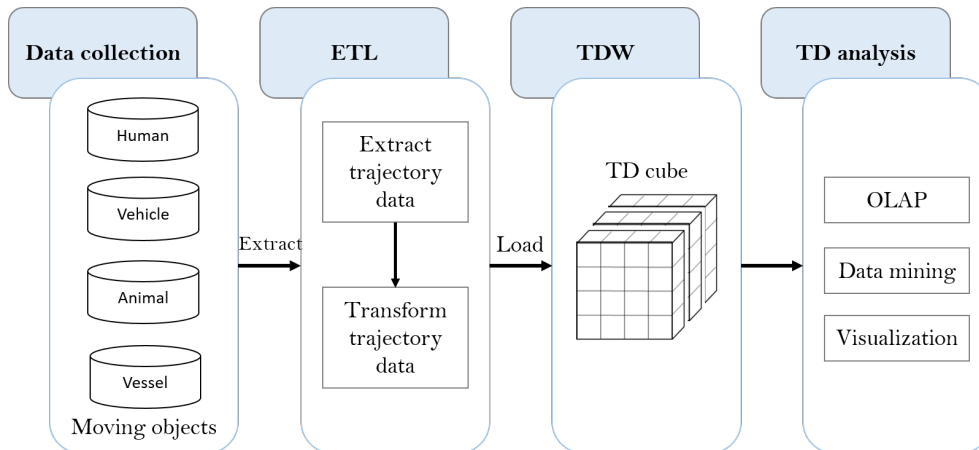


Figure 2.9: TDW framework structure.

scenario, the base granular of the spatial domain hierarchy is set to the segment of the road network. Similar to the study in [80], Leonardi *et al.* [79] use the multidimensional modeling to design a TDW called T-WAREHOUSE. The proposed framework is implemented to cover all the required steps for TDW, from trajectory reconstruction to OLAP analysis on mobility data. Furthermore, the study of Campora *et al.* [24] designs and implements a TDW that support spatio-temporal concepts on top of relational DBMS. The propose framework demonstrates that it supports OLAP, SOLAP, and STOLAP queries using traffic data.

Wanger *et al.* [122] tackle the problem of semantic trajectories data enriched with domain knowledge such as transportation means, and propose a TDW model for mobility called *Mob-Warehouse*. The raised questions of Who, Where, When, What, Why and How (or 5W1H) address the trajectory's features and analyse the different aspects of the "mobility story". The authors implement a prototype of the model using a large dataset of car trajectories. According to their proposed TDW conceptual model, the authors define one fact table and six dimensions, which address the six question of 5W1H. The two dimension that represent space and time address the question of, respectively, where and when.

In the context of OLAP framework, studies such as Spatial OLAP (SOLAP) [106] offer OLAP functionalities coupled with GIS functionalities for the analysis of geo-located data. They have the ability to perform multidimensional exploring of the data which can be presented in both detailed and aggregated forms [16]. The Open Geospatial Consortium (OGC) seeks to take inspiration from OLAP cubes to apply them on a multi-dimensional ("n-D") array of values. Data Cubes for geospatial information provide a way to integrate observations and geospatial data for efficient data analytics, using the geospatial coverages (e.g. rasters and imagery) data structure. Unlike conventional datacubes in OLAP, the dimensions refer to metrics and not categorical or semantic data, and cover entirely some spatial region.

In addition, sequential OLAP was proposed by Lo *et al.* [86] to support OLAP operations for sequences (S-OLAP). An event in an S-OLAP system consists of a number of **dimensions** and **measures** and each dimension may be associated with a **concept hierarchy**. If there is a logical ordering among a set of events, the events can form a sequence. A logical ordering can be based on another attribute (e.g. *time* attribute). Built upon Sequential OLAP, Interval OLAP or I-OLAP [73] was proposed to analyse and process efficiently data organized as intervals in an OLAP way. Koncilia *et al.* (2014) define an interval as the time between two consecutive events.

Last but not least, based on their proposed MOD called MobilityDB, Vaisman and Zimányi [120] extend existing proposal on TDW and integrate relational warehouse data with moving object data to realize the notion of spatio-temporal queries as defined in [119]. Therefore, the authors implement Mobility DW which gives the possibility to define moving objects as measures in a DW fact table. Similar to MobilityDB, Mobility DW is implemented on top of PostgreSQL, but the authors claim that it can be extended to big data hadoop-based environments.

### 2.6.3 . Other Works on DW

The previous studies address the problem of defining a DW in a unified-granular framework. However, the studies of Bimonte *et al.* [17] and Iftikhar and Pedersen [66] tackle the problem of multi-granularity in a DW. The study Bimonte *et al.* [17] propose a linear framework to handle the missing values in a multi-granular data warehouse. The authors propose a generic approach called adjustment to impute missing values from the detailed and aggregated facts. The aggregated facts are used to estimate detailed missing facts in vertical manner. The detailed facts, on the other side, are used to estimated missing facts based on facts with the same dimensions levels in an horizontal manner. Motivated by the lacking ability of current models to store data at different granularity in DW, Iftikhar and Pedersen [66] present alternative schema designs to structure both the detail and the aggregated data at different levels of granularity. The proposed alternatives define the time dimension granularity as a single hierarchy. Therefore, data in the fact table is associated with a time dimension in a particular granularity. Both proposal [17] [66] deal with DW in general and do not concern moving objects.

In the context of moving objects, a tailor-made data model has been proposed in [124] and [123] where the concepts of continuous dimension and continuous fact make it possible to capture the spatio-temporal fact of mobility in a pre-defined network. An adapted indexing method makes it possible to respond effectively to spatio-temporal aggregate queries. The advantage of this model is to allow spatial and temporal queries on the fly, without being limited to a prior division of space or time. The downside is that this model is more difficult to implement. Defining a granularity and a spatial and temporal frame of reference for the dimensions is a solution often adopted in practice. This is the case in the model proposed for the analysis of spatio-temporal activities in [111]. These works are the most similar

to our context of enriched trajectory data, but they were limited to trajectories without associated measures.

#### 2.6.4 . Discussion

To summarize, MODs such as MOST [129], SECONDO [61] and HERMES [103] were proposed to support spatio-temporal data. While these MODs have the ability to handle MO data, they are oriented toward addressing the problem of trajectory DWs.

Solutions such as Spatial OLAP [106] have the ability to perform multidimensional exploring of the data which can be presented in both detailed and aggregated forms. However, in a SOLAP model, the spatial attribute is represented as a cartographic object (i.e. points, lines and polygons) [16]. This opens the issue of drawing the spatial dimension's hierarchy in SOLAP model. Typically, spatial hierarchy is depicted by the topological relations (i.e. inclusion, overlap) between members of the same and/or different spatial levels. This affects the accuracy of the aggregation process [16]. In OGC and unlike conventional data cubes in OLAP, the dimensions refer to metrics and not categorical or semantic data, and cover entirely some spatial region. In MCS, only visited locations are materialized.

Other solutions propose conceptual framework to support semantic trajectories such as Mob-warehouse [122], provide analysis of the mobility story of the moving object. Indeed, the spatial dimension is an important one in the analysis, yet it is not the only focus of the analysis related to enriched trajectories. One of the important facets is to address the continuous measurements facet of the enriched trajectory combined with any dimensions, and not only on the spatial one, which is not possible with *Mob-Warehouse*. We are interested in the analysis of the desired phenomena during specific time periods with discard to the spatial dimension. In contrast, Sequential OLAP [86] and Interval OLAP [73] provide analysis for the temporal perspectives. However, in Sequential OLAP, it is necessary to have a logical ordering among a set of events. Yet, in MCS context, events are not as dense and regular as measurements and do not necessarily indicate a logical ordering. Plus, we are interested in the spatial dimension in the data analysis and exploration, unlike Sequential OLAP and Interval OLAP.



## 3 - Trajectory Data Enrichment

### Contents

---

<b>3.1</b>	<b>Introduction . . . . .</b>	<b>52</b>
<b>3.2</b>	<b>Multidimensional time series segmentation .</b>	<b>52</b>
3.2.1	Change Point Detection: Summary of Related Work . . . . .	53
3.2.2	Change Point Detection Model . . . . .	54
3.2.3	Experiments and Results . . . . .	59
3.2.4	Summary . . . . .	61
<b>3.3</b>	<b>Learning the Micro-environment . . . . .</b>	<b>62</b>
3.3.1	Activity Recognition: Summary of Related Work . . . . .	64
3.3.2	Problem Formalization . . . . .	65
3.3.3	Multi-view Learning Model . . . . .	68
3.3.4	Micro-environment recognition model . . . . .	69
3.3.5	Hybrid Multi-view Learning Model . . . . .	74
3.3.6	Experiments and Results . . . . .	79
3.3.7	Discussions & Perspectives . . . . .	96
3.3.8	Summary . . . . .	99
<b>3.4</b>	<b>Conclusion . . . . .</b>	<b>99</b>

---

### 3.1 . Introduction

In this chapter, we present the first contributions of this thesis, i.e. trajectory data construction and semantic enrichment. Trajectory data construction consists of pre-processing the raw enriched trajectory data, which paves the way for purified and sound trajectory data series, by cleaning the trajectory data from outliers and noise, and interpolating missing values. Therefore, a sound version of the trajectory data is achieved. Furthermore, the major contributions of this thesis work is the design of a comprehensive annotation framework to enrich trajectory data with contextual information. Precisely, we build a model for multidimensional data segmentation based on change point detection (CPD) to detect participants activities boundaries. Thereafter, we design a hybrid model for context recognition which can integrate geographic information as well as multivariate time series data to annotate trajectory data with the type of activity and movement.

This chapter is organised as follows: Section 3.2 presents the segmentation framework based on change point detection for multidimensional time series. Section 3.3 is the fundamental part of trajectory data enrichment, which annotate MCS data with the type of activity and movement.

### 3.2 . Multidimensional time series segmentation

Air quality and exposure to pollution are a central concern for people living in urban areas. As the harmful effects of air pollutants on their health is alarming. The key concern to reduce the risk of these pollutants on individual's health is by understanding the totality of exposure. Air pollution monitoring is getting more interest nowadays, due to the rapid advances of the Internet of things (IoT) along with the emergence of the Mobile Crowd Sensing (MCS) paradigm.

The mentioned technologies coupled with the widespread use of GPS, allows volunteers to contribute their collected data in order to get personalized insights about their exposures to pollution. Polluscope <sup>1</sup> is a French project deployed in Île-de-France (i.e., Paris region), and is a typical use case study based on MCS. In Polluscope, participants are equipped with a sensor kit which can measure different pollutants such as Nitrogen dioxide (NO<sub>2</sub>), Particulate Matters (PMS), Black Carbon (BC), Temperature, etc., independently from their environment either indoor or outdoor.

Air quality strongly depends on the context<sup>2</sup> of the participant, thus in order to understand and identify participants' exposure to pollution, it is essential to identify the context of the participants. To avoid miss-classification of the exposure wrt the context (micro-environment), the participants need to fill a time-use diary, but

---

<sup>1</sup><http://polluscope.uvsq.fr>

<sup>2</sup>In this thesis, the terms "context" and "micro-environment" are used interchangeably.

in real-life, they rarely thoroughly do this self-reporting task. Therefore, there is a need to identify the context automatically from the collected raw data. However, since the context changes with the participants' activity and whereabouts, this also means that we need to detect the changes and segment the geo-data series into non overlapping segments according to participants micro-environments (i.e. home, work, transport, streets, park, etc.). Segmented data is a prerequisite for activity recognition mining task, which assigns to each segment a labeled with a single activity.

As a matter of fact, segmentation in our process means the detection of changes in micro-environment and events, and not only the change from a stop segment to a move segment. Segmentation can occur in the same stop segment such as moving from a shopping gallery to a restaurant in the same shopping mall. It can also occur with the change of transportation means from one to another in the same move segments, or because of the events.

In this section, we propose a multi-dimensional time series segmentation to discover activities and events boundaries in the context of mobile crowd sensing. The main contribution is precisely the combination of different dimensions in the change point detection, when not all dimensions may cause or contribute evenly in discovering the change in participant's activities or events. Our approach combines data pre-preparation, change point detection on individual dimensions, and a post-processing phase to fuse the detection from multiple dimensions. This last phase is based on a supervised learning approach. We implement and test our framework in a real-application setting. The rest of this section is organised as follows. We present a summary of works related to change point detection in Section 3.2.1. The formal presentation of our change point detection model is explained in Section 3.2.2. Section 3.2.3 presents the experimental results and evaluation of the change point detection model on real-world data. Section 3.2.4 summarises the main contributions regarding multi-dimensional data segmentation model based on change point detection.

### 3.2.1 . Change Point Detection: Summary of Related Work

During the investigation of existing approaches in the literature in Section 2.3.3, there are a lot of studies on time series segmentation and more precisely change point detection in time series. Such research operates on limited types of time series. Some algorithms, such as kernel-based method or Guassian Process, require the time series to be i.i.d or stationary, and others, such as uLSIF and KLIEP, offer a parametric approach for non-stationary time series [6]. Another issue raises with multi-dimensional data, where not all dimensions may cause the change. These constraints are not coherent with the nature of mobile crowd sensing data, where time series segmentation should not be subject to any constraints. Additionally, the inclusion of weak or irrelevant dimensions should not degrade the performance of the change point detection.



However, Aminikhangahi *et al.*[9] state that, compared with other change-point detection methods, density ratio based algorithms offer several advantages for real world problems. One of the direct density ratio change point detection methods that has no limitations on time series data distribution and does not require any condition on data stationarity, is the cumulative sum (CUSUM) algorithm [6]. The contribution of this work is formulated as a proposed change point detection combination of multi-dimensional time series obtained from real world geo-location and sensors data, where time series dimensions do not contribute evenly in the activity transition detection. To the best of our knowledge, no prior work focuses on the segmentation of human whereabouts using climatic sensors, environmental sensors and geo-location.

### 3.2.2 . Change Point Detection Model

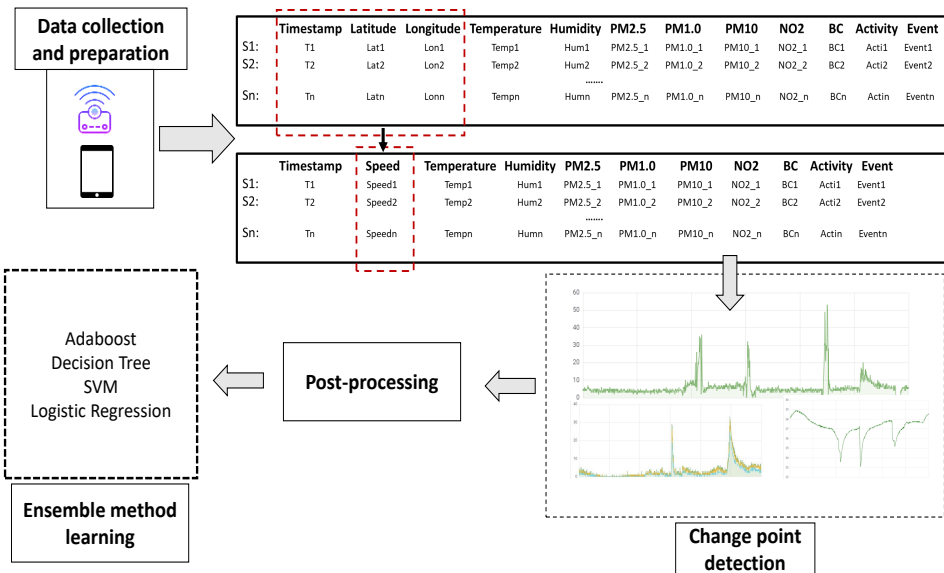


Figure 3.1: Architecture of change point detection model.

In this section, we introduce our change point detection approach based on CUSUM algorithm for multi-dimensional time series in environmental crowd sensing. Both the theory and the implementation aspects of the approach will be discussed. The holistic schema of our proposed approach is shown in Figure 3.1. The implementation of our proposed process includes four parts: data collection and preparation, change point detection, post-processing and ensemble method learning.

## Data Collection and Preparation

Real-life data are collected within the scope of Polluscope project. Three cohorts of volunteers are recruited to collect sensory spatio-temporal data series. Participants are equipped with air pollution sensors and tablets empowered with GPS chipsets. The sensors collect time annotated measurements of Particulate Matter (PM1.0, PM10, PM2.5), NO<sub>2</sub>, Black Carbon, Temperature and Humidity, and the tablet records participants' geo-locations and allows to annotate data with activities and pollution related events. Activities are depicted by micro-environments of participants (e.g., Home, Office, Park, Restaurant, etc.). Events are temporary actions for a brief period (e.g., Start cooking, Open a window, Close a window, Smoking, Turn on a chimney, etc.). Activity and event recognition allow to enrich semantically the collected data with the context. More than 86 volunteers participated in the data collection phase. Each participant carries a kit composed of an aethalometer for measuring Black Carbon, a gas sensor which measures the Nitrogen dioxide (NO<sub>2</sub>), and a sensor for various size particulate matters (PM) measurements (the sensors have been selected after performing evaluation in [76]) - all bundled with a tablet for seven days with no restrictions on space and time. For more information on data collection's protocol, refer to Appendix A. As such the MCS scheme is opportunistic, aiming at reporting the participants' exposure in accordance with their habits and their daily life. A sequence of these data contains kit ID, timestamp, and values from different sensors. Figure 3.2 shows an example of sequences collected by tablets and different sensors. This sequence contains a series of logs which include timestamp, kit ID, Latitude, Longitude, ambient air data (i.e., Temperature, Humidity, PM2.5, PM10, PM1.0, NO<sub>2</sub> and Black Carbon), activity (Car) and events (Close a window). The air quality (AQ) data plus GPS data have been undergone a pre-processing phase, which consists of cleaning data from noise and outliers, as well as interpolating missing values. This phase is discussed further in Section 3.3.4.

Participants geo-locations are collected as GPS logs. As shown in the upper part of Figure 3.1 by dotted rectangle, each GPS log is a sequence of GPS points that contain latitude, longitude and timestamp. We derive the velocity time series from GPS coordinates.

time	kit_id	lat	lon	Temperature	Humidity	PM2.5	PM10	PM1.0	NO2	BC	activity	event
2019-10-20 12:52:57	57	48.766292	2.032283	26.2	51.5	2.0	2.0	1.0	10.0	1027.0	Voiture	Fermeture De Fenêtre
2019-10-20 12:53:28	57	48.766292	2.032283	26.5	50.7	2.0	2.0	2.0	0.0	921.0	Voiture	Fermeture De Fenêtre
2019-10-20 12:53:59	57	48.766292	2.032283	26.5	50.7	2.0	2.0	2.0	0.0	921.0	Voiture	Fermeture De Fenêtre
2019-10-20 12:54:30	57	48.764743	2.034910	26.6	50.2	2.0	3.0	1.0	0.0	888.0	Voiture	Fermeture De Fenêtre
2019-10-20 12:55:01	57	48.762178	2.038895	26.8	49.7	3.0	4.0	1.0	3.0	885.0	Voiture	Fermeture De Fenêtre
2019-10-20 12:55:32	57	48.762240	2.044100	26.8	49.7	3.0	4.0	1.0	3.0	885.0	Voiture	Fermeture De Fenêtre

Figure 3.2: An Overview of data collected in the context of MCS

It is important to emphasize that not all data are thoroughly annotated with

participants context. Also most sensor data are noisy, and require a preprocessing phase to clean them from irrelevant measurements. We have observed this especially in the GPS data, BC, and some PM data. The sensors for climatic parameters do not show such defect. Therefore, the data preparation is twofold. On the one hand, a de-noising process is applied to clean the data. On the other hand, the highest quality sample of annotated data is selected as a baseline to validate the process of data segmentation. The idea is to generalize the change point detection to all participants data, by using the model derived from a good-quality dataset.

### Change Point Detection

The change point detection problem is the process of detecting abrupt changes in time series data. The change points are detected when the probability distribution of time series changes abruptly between two consecutive intervals. The overall question is: how to combine all these different aspects of the data (geo-location, sensors, partially annotated activities and events) to segment and discover the context of the user, and to discriminate the observations in different micro-environment? This is called a holistic approach of activity recognition [40].

The segmentation phase consists mainly in splitting spatio-temporal data into coherent segments. Each segment represents a micro-environment. One way to do this segmentation is to detect the changes either in the ambient time series, or in the geo-location. The former corresponds to the problem of change point detection (CPD) in time series. Many solutions exist in the literature when it comes to mono-variate time series. As for the the GPS data, it is related to the so-called stop & move detection in trajectories. In this thesis work, we use the change point detection in time series for both problems, simply by adding the velocity dimension, which is easily derived from geo-location data.

The CUMulative SUM (CUSUM) is one of the main CPD techniques in mono-dimensional time series. It has no limitation on time series data distribution and does not require any condition on data stationarity [6]. First introduced by [99], CUSUM algorithm is a sequential analysis technique, and it is the most familiar change point detection algorithm. The CUSUM proposed by [99] uses the Sequential probability ratio test (SPRT) to detect change points. The algorithm performs by comparing probability distributions of two time series intervals. As the two intervals are moving, the test issues an alarm of a change point when the probability distributions of the two intervals are significantly different.

One form of implementing the cumulative sum test is given by the Expression 3.1, which detects changes in the positive and negative direction ( $g_j^+$  and  $g_j^-$ ) in the data ( $x$ ) [59]. The decision rule is: if  $g_j^+$  and  $g_j^-$  exceeded a user defined **threshold** ( $h$ ), then an alarm is given ( $t_{alarm}$ ), a change point has been detected, and the test statistic is reset ( $g_t^+ = 0$  and  $g_t^- = 0$ ). This algorithm depends also on another parameter called **drift** ( $\nu$ ) for drift correction to avoid false alarm or slow drift. The CUMulative SUM (CUSUM) test is given by:

$$\begin{cases} s_t = x_t - x_{t-1} \\ g_t^+ = \max(g_{t-1}^+ + s_t - \nu, 0) \\ g_t^- = \max(g_{t-1}^- - s_t - \nu, 0) \end{cases} \quad (3.1)$$

$$\text{if } g_t^+ > h \quad \text{or} \quad g_t^- > h : \begin{cases} t_{alarm} = t \\ g_t^+ = 0 \\ g_t^- = 0 \end{cases}$$

The CUSUM test accuracy depends on tuning the parameters  $h$  and  $\nu$ . Both parameters present a trade-off between faster detection of true alarms and allowing more false alarms. High values of  $\nu$  allow false alarms at the cost of obtaining a delayed detection [59]. As soon as the CUSUM test exceeds a threshold  $h$ , the change is detected. The accuracy of this algorithm is often computed by indicators such as false positive rate.

### Post-Processing

In multi-dimensional time series, some dimensions may contribute more in the explanation of the change, while others may be considered as irrelevant. The participant context is very highly correlated with ambient air temperature and humidity more than, for example, speed. Because, first, the temperature and humidity indoor are different than outdoor. When participant changes their micro-environment, temperature and humidity change abruptly. Second, speed is very sensitive to traffic conditions. During a traffic jam, using a transportation mode such as Car or Bus, participant's speed drops frequently without indicating a change in micro-environment.

The application of the CUSUM algorithm on different dimensions of time series may generate numerous false alarms. Post-processing the CUSUM algorithm results will improve the change point detection accuracy by merging the detected change points into one change point if a certain condition is verified. The condition, in this work, is the time difference between two consecutive detected change points should be less than 5 minutes. If the time difference between two consecutive detected change points is less than 5 minutes, then the detected change points will be merged into one detected change point. As shown in Table 3.1, participants do not stay in the same micro-environment for a short period of time. On the average, the least time spent in one micro-environment are found to be 8 minutes.

Table 3.1: Overview of time spent in every micro-environment for four participants

Micro-Environment	Mean	SD	Min	Median	Max
Office	03:58:36	02:32:04	00:02:00	03:58:30	09:19:00
Bus	00:08:02	00:05:09	00:03:00	00:06:00	00:18:00
Cinema	02:00:00	0	02:00:00	02:00:00	02:00:00
Home	04:30:26	05:48:49	00:00:00	01:10:00	23:16:00
Store	00:19:25	00:26:53	00:01:00	00:10:00	01:39:00
Metro	00:25:07	00:09:10	00:15:00	00:23:30	00:40:00
Park	01:20:00	0	01:20:00	01:20:00	01:20:00
Restaurant	00:43:52	00:27:12	00:24:00	00:37:00	01:50:00
Street	00:07:53	00:09:05	00:00:00	00:05:00	00:55:00
Train	00:22:42	00:13:20	00:01:00	00:20:30	00:48:00
Car	00:26:12	00:37:32	00:01:00	00:08:30	02:52:00
Bike	01:08:00	00:44:27	00:14:00	01:12:00	02:22:00

### Ensemble Method Learning

One of the contribution of this work is the combination of multi-dimensional sensory time series data and geo-located data (i.e. GPS data) to detect the changes boundaries of participants micro-environments when some dimensions may be considered irrelevant to the change detection or not all dimensions cause the change. In order to enhance the accuracy of the change point detection, many ensemble methods [146] have been proposed to further enhance the algorithms accuracy by combining learners rather than trying to find the best single learner.

There are different types of combination methods, among which, the most popular are: **Averaging**, **Voting** and **Combining by Learning** [146]. *Averaging* is the most popular and fundamental combination method for numeric outputs. Regression is an explicit example of how *Averaging* works. *Voting*, on the other hand, is the most popular and fundamental combination method for nominal output. Classification is an explicit example of how *Voting* works. There are four types of voting [146]: (1) *Majority Voting* is the most popular voting method. Here, every classifier votes for one class label, and the final output class label is the one that receives more than half of the votes. (2) *Plurality Voting* takes the class label which receives the largest number of votes as the final winner. (3) If the individual classifiers are with unequal performance, it is intuitive to give more power to the stronger classifiers in voting and this is realized by *Weighted Voting*. These three voting methods are suitable for classifiers that use *crisp class labels*. However, if individual classifier produce class probability outputs (such as Naive Bayes, Logistic Regression), *Soft Voting* is the choice.

However, CUSUM algorithm produces the timestamp when the change has occurred. Some dimensions (such as Temperature and Humidity) are more important

and correlated to the change in participants activities than others. The most suitable for that type of multi-dimensional time series would be the *Weighted Voting*. The key here is to assign weights in proportion to the performance of individual learners. Assigning inadequate weights has a major effect on the learning accuracy.

To assign automatic weights to single learner, *Combining by Learning* method is a procedure where individual learners are trained for the *first level learners* and combined by a learner for the *second-level learner*. **Stacking** introduced by [132] [113] [22], can be viewed as a generalization of many *Combining by Learning* methods.

In this work, we propose a model that integrates the CUSUM change point detection algorithm with multi-dimensional time series to achieve a strong combination abilities. The model works as follows: (1) the change point detection algorithm is applied on each time series dimension separately; (2) each dimension generates a set of detected change points, with a certain accuracy to the ground truth; (3) the weights of every dimension are then learned from the gold set data annotated by activities and events of participants. The model used in this experiment include: *AdaBoost* with Decision Tree, *Decision Tree* (DT), *SVM* and *Logistic Regression*. The proposed model allows to understand which dimension is affected by the changes in participants micro-environments and pollution related events.

### 3.2.3 . Experiments and Results

The above-mentioned segmentation model was implemented in Python using scikit-learn (0.22.2) for *Combining by Learning* method, and tested on real environmental crowd sensing data to detect boundaries transition in activity and pollution related events.

## Experiments

We evaluate our change point detection model to segment participants' daily activities and pollution related events. We use for the experiment phase environmental crowd sensing data collected over 7 days by two participants. Each participant is equipped with three sensors that record ambient air data (i.e. Temperature, Humidity, Particulate Matter: PM2.5, PM10, PM1.0, NO2 and Black Carbon) and a tablet for geo-location and data annotation.

From geo-location data, speed of participant every minute is calculated. Then, our multi-dimensional time series has six dimensions: *Speed*, *Black Carbon*, *NO2*, *Temperature*, *Humidity*, and *Particulate Matter PM2.5*.

## Parameters optimization

Setting *Cumulative Sum* algorithm parameters, **threshold**  $h$  and **drift**  $\nu$ , depend on each dimension. Several parameters combinations have been tested to choose

Table 3.2: Cumulative Sum parameters optimization for each dimension

Dimension	Threshold	Drift	Precision	Recall
Temperature	0.6	0.05	0.72	0.78
Humidity	4	0.05	0.70	0.82
PM 2.5	25	0.03	0.87	0.30
NO2	15	15	0.66	0.26
Black Carbon	900	500	0.22	0.65
Speed	1	0.1	0.12	0.52
Post-processing	-	-	0.45	1

the one that yields the highest performance. The parameters that have been used in this experiment are given in Table 3.2. To evaluate the overall performance of the algorithm, we compute the precision to measure the ratio of true positive change points to total points classified as change points. The recall (true positive rate) is also computed to measure the portion of change points that was correctly detected.

To evaluate the performance of the change point detection, we consider as a true positive every detected change point that belongs to a buffered interval of 5 minutes before and after the actual change. Overall, **Temperature** and **Humidity** have the highest *precision* and *recall* at the same time. However, **Speed** generates the highest number of false positives. Thereby, *Temperature* and *Humidity* should be assigned with more power than the rest of dimensions. After post-processing the results, the recall, the true positive rate, records a score of 100%, which means that all the change points have been detected successfully. However, the precision indicates that many false alarms are still being detected due to some weak dimensions

### Ensemble learning performance

For the ensemble learning step, a second data set is generated. This data set contains the output of the CUSUM algorithm results. In other words, the output of the first-level learners is considered as input for the second-level learner. The response vector is the vector of the actual change points. This vector is binary re-coded and takes 1 if there is a change point, and 0 if not.

We study the performance of different classifier models as a predictor of the response vector. We divided the first participant data into two sets. 70% of the data is for training the models, and the rest 30% for testing the models. In order to validate our approach and generalize it on other un-annotated data, we use the second participant data of one day and compare it with our ground truth. The

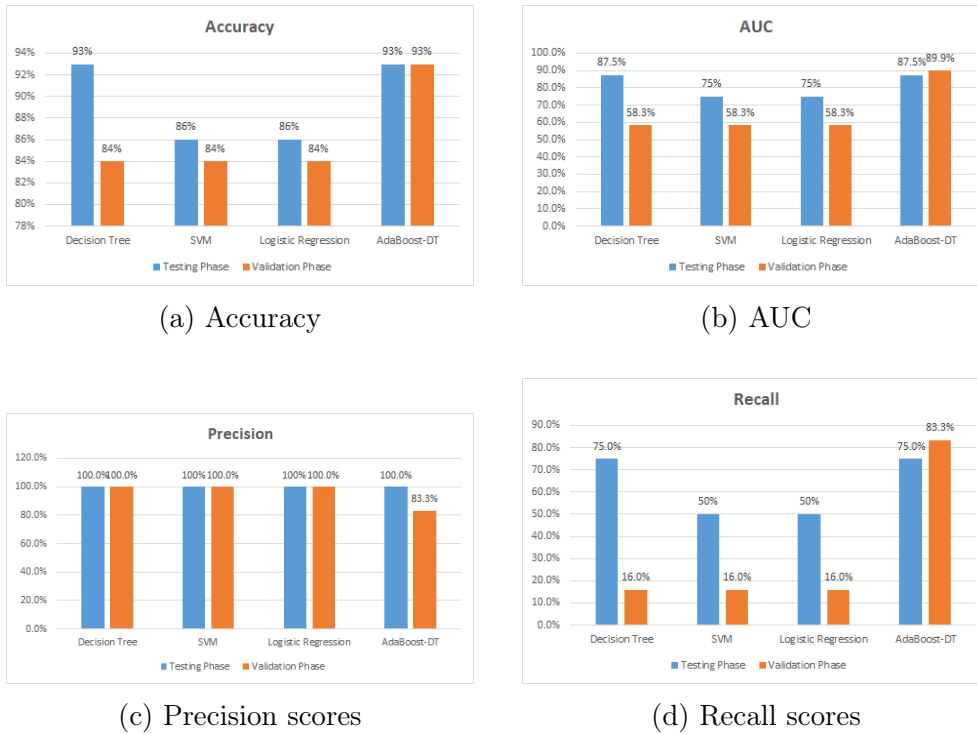


Figure 3.3: Performances of CPD during testing and validation phases.

results of the change point detection (CPD) experiment are summarized in Figures 3.3a, 3.3b, 3.3c and 3.3d.

The experience conducted on real world data shows that our approach outperforms the traditional cumulative sum algorithm. During the testing phase, when comparing the combining learning algorithms' precisions to the overall precision of the traditional CUSUM algorithm in Figure 3.3c, *Decision Tree*, *SVM*, *Logistic Regression* and *AdaBoost* with *Decision Tree* base learner outperform the CUSUM algorithm. Considering the recall of the CUSUM after post-processing shown in Figure 3.3d, *Decision Tree* and *AdaBoost* show good recall scores with 75% each.

When comparing the performance of the combining learners algorithms, considering the accuracy in Figure 3.3a, all the algorithms perform well during the testing phase, and *Decision Tree* and *AdaBoost* outperform the other algorithms with 93%. When considering the validation phase, *AdaBoost* outperforms all the other algorithms with an accuracy of 93%. Considering the *Area Under the ROC Curve* (AUC), during the testing phase, *Decision Tree* and *AdaBoost* outperform the other algorithms with 87.5%. However, during the validation phase, *AdaBoost* outperforms the other algorithms with 89%.

#### 3.2.4 . Summary

Change point detection segmentation can provide insights about human behaviour's transition. Participants' whereabouts can be learned after segmenting



the collected multi-dimensional time series, and discover insights about individual exposure to pollution.

In this section, we presented a change point detection approach based on the *Cumulative Sum* algorithm to discover transition points in multi-dimensional time series using real world data collected in the context of environmental crowd sensing. The experiment conducted in multi-dimensional time series, where not all dimensions may cause the change, shows that our approach outperforms the traditional CUSUM algorithm, using *AdaBoost* as a combining learner algorithm.

### 3.3 . Learning the Micro-environment from Rich Trajectories in the context of Mobile Crowd Sensing

As mentioned in Section 3.2, air quality strongly depends on the context, and so is the individual exposure to pollution. For this reason, there is a great interest in making exposure analysis context-aware. Beyond that, ignoring the context would make the data collection useless, precisely because of the influence of the micro-environment. However, the context annotation is by far the most difficult information to collect in a real-life application setting since very few participants thoroughly annotate their micro-environment. Therefore, there is a great interest in unburdening the participants by automatically detecting the context.

The problem of automatically annotating MCS data can be seen as a problem of activity recognition from enriched trajectory data collected by heterogeneous sensors. There is a broad variability of research studies on the subject of activity recognition. The survey by Yu Zheng [141] proposes a systematic review of the major research in trajectory mining. Whilst the author provides a variety of trajectory data mining methods, an overall approach that combines several sensors data besides GPS data is missing. In contrast, combining several sensory data suggests the usage of multivariate time series classification (MTSC) for activity recognition. Although this solution showed excellent performance in some application domains [107], its success is not guaranteed with heterogeneous sensors such as environmental data. First, the usage of heterogeneous data may induce some missing data problems when some sensors stop working. Therefore, there is a need for a model that characterises the micro-environments even with missing dimensions. Second, it is not known to what extent environmental data can be determining of micro-environments, which needs to be investigated.

As a matter of fact, when visually exploring the data, we noticed that micro-environments preserve a certain pattern. Moreover, we observe the existence of an inter-sensor correlation and with the context. Figure 3.4 shows the evolution of three dimensions (i.e. Black Carbon (BC), NO<sub>2</sub> and Particulate Matters (PM)) with micro-environments identification. As shown in Figure 3.4, BC and NO<sub>2</sub> preserve the same shapes and statistical characteristics in the micro-environment “car”. Specifically, BC maintains the same fluctuations pattern in the micro-environment

“car” and conserves approximately the same average value in these segments. Likewise, NO<sub>2</sub> fluctuates promptly and preserves roughly the same average value in both segments of the micro-environment “car” as well as approximate maximum values. We also note that NO<sub>2</sub> values keep roughly the same pattern in the micro-environment “indoor”. Moreover, we can observe the existence of a correlation between the three dimensions during the whole timeline, meaning that when one of the dimensions fluctuates, the other two follow.

The idea we promote in this section is to utilize a wisely chosen annotated dataset in order to train a model on the acquired enriched trajectories (composed of both environmental and mobility dimensions) as predictors of the micro-environment. We hypothesize that the multivariate time series collected by the MCS campaigns not only depends on the micro-environment but could be a proxy of it.

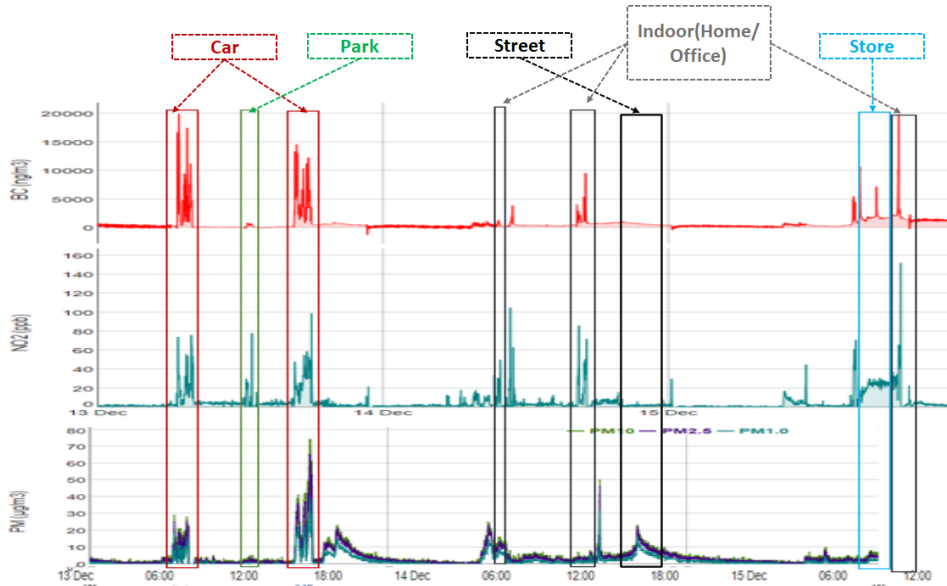


Figure 3.4: Inter-sensor and micro-environment correlations.

The question that arises now is how to combine all these different heterogeneous aspects of the data (geo-location, sensors) to identify the user’s context automatically, and how much a model can discriminate the observations in different micro-environments. In this section, we evaluate different approaches and provide a framework dedicated to the preparation, the application, and the comparison of different machine learning algorithms. Precisely, we make the following contributions in this thesis work:

- first and foremost, we identify the problem of micro-environment recognition in the MCS context.
- we demonstrate that AQ determines the type of micro-environment.

- thereafter, we propose a ML approach based on multi-view learning for the recognition of micro-environment.
- afterwards, we extend the proposed micro-environment recognition approach to include another layer which consists of the detection of stay locations (i.e. stop detection) and transportation modes from GPS data; also known as trajectory segmentation. We refer to this extension as the hybrid approach.
- we optimise the proposed approaches either by analysing the exact geolocation, or simply applying some a priori rules. We emphasize that the first optimisation is privacy invasive whilst the other one is privacy friendly.
- we conduct extensive experiments in a real scenario setting and compare with baselines, which shows the effectiveness of our proposed approaches.

In the current work, Polluscope data is considered as a running application example based on MCS. However, the proposed approach can be generalised on other MCS applications besides the AQ scenarios. In fact, the same process of activity recognition of moving objects holds with other sensory data such as sound for noise sensing or temperature for heat comfort assessment. For instance, the collaborative AIRLESS project between Cambridge and Beijing, which is a typical use case study based on MCS, aims to understand the impact of air pollution on human health in the world's largest country. One important element of AIRLESS is to automatically detect and classify major exposure-related micro-environments (home, work, other static, in-transit) using GPS coordinates, accelerometry, and noise. The classification of micro-environment can remarkably improve exposure metrics since pollutant inhalation rates vary significantly by location and micro-environment [27].

The rest of this section is organised as follows. We summarise the works related to activity recognition in Section 3.3.1. Section 3.3.2 gives a thorough problem description. Section 3.3.3 describes the multi-view learning approach. The presentation of our micro-environment recognition model is discussed in Section 3.3.4 and 3.3.5. Section 3.3.6 presents the experimental results and evaluation of the micro-environment recognition model in the context of environmental crowd sensing. Section 3.3.7 gives an extensive discussion of the perspectives of this work. In section 3.3.8, we summarize our conclusions and provide directions for future work.

### 3.3.1 . Activity Recognition: Summary of Related Work

In the last recent years, Human activity recognition (HAR) has gained a great interest from the research community. Its application domains encompasses all the activities performed within daily human activities and human mobility. Micro-environment recognition falls in the same problem of HAR.

As discussed in Section 2.4, a wide range of research proposals exist in the literature. Such studies are either based on multivariate time series classification

without any concern or attention to trajectory data to classify the label of the micro-environment, or related to stop and move detection in trajectory data where the label of the micro-environment is reduced to stop/move (where stops are indoor and moves are outdoor, in general). Transportation mode detection approaches can reveal the label of the move segments but not indoor segments.

However, with the advent of MCS, the focus has shifted to considering both aspects of the two sources of data, and bring them in the activity recognition loop. Driven from multivariate time series classification and stop & move detection within trajectories studies, this thesis investigates semantic enrichment of MCS data coming from multi-sensors and GPS tracks in order to add contextual information to the data. To the best of our knowledge, no previous approaches that combine environmental time series and trajectory data in the micro-environment recognition problem have been proposed so far, which is the focus of the current work.

### 3.3.2 . Problem Formalization

Before detailing the methodology of the proposed approach, we start with a thorough description of the problem.

#### What are rich trajectories ?

In the context of MCS, the collected type of data are typically continuous sensors measures along with the participant's spatial location (e.g., GPS coordinates). They represent a specific type of trajectories that we call rich trajectories. We define hereafter this concept, starting from the definition of time series.

**Definition 3.3.1.** (Univariate Time Series). A univariate time series is a sequence  $U = [(t_1, v_1), \dots, (t_l, v_l)]$  where  $l$  is the length of  $U$  and for  $i = 1 \dots l$ ,  $t_i \in T$  is a timestamp from a time domain  $T$  and  $v_i \in D$  is a scalar value of a domain  $D$ .

**Example 3.3.1.** Environmental sensor measurements such as temperature constitute a univariate time series.

**Definition 3.3.2.** (Multivariate Time Series). A multivariate time series  $MV$  is defined as  $MV = (U_1, U_2, \dots, U_i, \dots, U_n)$  where  $U_i$  is a univariate time series for dimension  $D_i$ , and  $i = 1, \dots, n$ .

**Example 3.3.2.** Environmental sensor measurements such as temperature, humidity and NO2 constitute a 3-Dimensional time series.

**Definition 3.3.3.** (Trajectory). A trajectory  $T$  is defined as a multivariate time series with two or three dimensions for the spatial position.

**Example 3.3.3.** A multivariate time series with latitude and longitude as dimensions represent a trajectory.

**Definition 3.3.4.** (Rich Trajectory). A rich trajectory  $RT$  is defined as a multivariate time series where a subset of the dimensions  $D_i$  where  $i \in [1, \dots, n]$  constitutes a spatial position, plus additional non-spatial informational.

**Example 3.3.4.** A GPS trajectory data of a moving object associated with environmental sensor measures such as temperature, humidity and NO2 is a typical example of a rich trajectory.

### What is micro-environment recognition ?

First, we define a trajectory segmentation, then we introduce the annotated version of rich trajectories before defining the target problem of micro-environment recognition learning.

**Definition 3.3.5.** (Rich Trajectory Segment). A rich trajectory segment  $RTS$  is defined as a sub-sequence of contiguous vectors of  $RT$  between  $j$  and  $k$  ( $1 \leq j \leq k \leq l$ ). So,  $RTS = RT(j, k) = (U'_1, U'_2, \dots, U'_i, \dots, U'_n)$  where  $U'_i = [(t_{ij}, v_{ij}), \dots, (t_{lk}, v_{lk})]$ , and  $\forall 1 \leq i \leq n$ .

**Example 3.3.5.** A one hour trajectory constitutes a rich trajectory segment of a one week rich trajectory data.

**Definition 3.3.6.** (Trajectory Segmentation). Given a trajectory or a rich trajectory as input, trajectory segmentation is a process that splits it into non overlapping trajectory segments.

**Example 3.3.6.** Splitting trajectory data of a moving object into hourly segments represent a one form of trajectory segmentation.

An annotated rich trajectory is defined as a sequence of trajectory segments along with annotations that belong to a predefined list of categories. Formally:

**Definition 3.3.7.** (Annotated Rich Trajectory). An annotated rich trajectory  $ART$  is defined as a sequence of couples  $ART = [(RT(1, i_1), a_1), (RT(i_1, i_2), a_2), \dots, (RT(i_j, i_{j+1}), a_2), \dots, (RT(i_p, l), a_{p+1})]$ , where  $RT(i_j, i_{j+1})$  are rich trajectory segments  $RTS$  between  $j$  and  $j + 1$ ,  $a_k \in A$ , and  $A$  is a discrete domain.

**Example 3.3.7.** Rich trajectory segments enriched with contextual information such as the whereabouts of a moving object represent an annotated rich trajectory.

In this work, annotations describe the micro-environment of the participant. In this work, micro-environments can either be an indoor space (e.g. home, office, restaurant, etc.), outdoor space (e.g. street, park, etc.) or a transportation mode (e.g. metro, bus, car, etc.). The micro-environment recognition question relates to the problem of segmenting data and assigning a label to each segment by combining every available data.

**Definition 3.3.8.** (Micro-environment Recognition). Given a rich trajectory  $RT$  as input, micro-environment recognition is a process that outputs the corresponding annotated rich trajectory  $ART$ .

**Definition 3.3.9.** (Micro-environment Recognition Learning). Given a set of annotated rich trajectories, train a model where the rich trajectory segments are the predictors, and the annotations constitute the class labels.

Using a trained model on a wisely chosen annotated dataset, we aim at predicting the annotation on a completely unseen data by the model.

*Why is this information of micro-environment important ?* Personal exposure to pollution is directly correlated to people's habits and where they spend most of their time. For instance, if a person is highly exposed in his/her home during cooking time without much room ventilation, it would be time for them to revisit his/her habits and start ventilating the room when cooking. Therefore, the information of micro-environment is necessary to interpret correctly the collected AQ data, get insight on the individual exposure, and for a participant, adapt his/her behavior to reduce his/her exposure.

While there are several works related to activity recognition from spatio-temporal trajectory (e.g. Sardianos *et al.* [110] and Toch *et al.* [118]), this work investigates the capability of environmental data in characterizing and inferring automatically the activity of the moving object. Therefore, we envision to combine every available information (i.e., AQ data, mobility data, declared annotations) to detect efficiently the micro-environment of the moving object.

### How can micro-environments be recognised ?

Micro-environments can mainly be characterised by the temporal attributes (i.e. AQ measures) as well as the spatial one. There are several works for activity recognition that are either based on the geographical or temporal information. However, an overall methodological approach for combining these different aspects on real-world complex trajectory data is missing. This combination may lead to a more robust detection model rather than the usage of a single attribute, and it needs to be investigated.

To employ every available facet of the rich trajectories, the design of the micro-environment recognition model needs to integrate two layers: a geographic layer and a multivariate time series layer. The geographic layer may infer the stop and move segments (aka *trajectory segmentation*) from GPS tracks only. This layer can go further and discover the location of home and work. The second layer of the multivariate time series may detect the exact label of segments (e.g. *home, office, store, metro, park*, etc.). Usually, this problematic leads to a multivariate time series (MTS) classification with AQ data as input and the detected micro-environments as output. However, MCS data is characterised by its heterogeneous

property, designating that data is originated from different sensors readings. In fact, some sensors may be offline and do not transfer any data for hours, which may lead to missing data problems. Therefore, the usage of MTS classification is not straightforward. Naturally, it is necessary to design a model that combines data from heterogeneous sensors, and has the ability to classify it efficiently even if one (or more) dimension is missing.

Furthermore, in real-world settings, problems such as imbalanced data occur. For instance, we observe that the predominant labels are home and work since people spend most of their time there. Thereby, most segments are naturally mistaken by the model as home or work. Therefore, the proposed model should take into consideration all these aspects of the data and be efficient and robust enough to overcome these challenges. This model is explained further in Section 3.3.4.

### 3.3.3 . Multi-view Learning Model

In this section, we present the approach of multi-view learning with stacked generalisation. We followed the proposal of Garcia-Ceja *et al.* [52], and adapted it to best fit for solving our problem.

It is not unusual to have applications in which heterogeneous types of sensors (e.g. accelerometers, gyroscopes) are involved for activity recognition. One way to deal with this problem is to extract features from each sensor and aggregate them to build the final classification model. However, this approach is not optimal since each sensor has its own statistical properties. Hence the idea of multi-view stacking to fuse data from heterogeneous sensors.

The multi-view paradigm consists of learning a model based on the different views of the data. The key idea is to consider each source of data independently and fuse them with *stacked generalization* (also called *stacking*), which is a type of ensemble methods [146] for combining multiple learners.

The overall process is described as follows:

1. The first step consists of defining the set of first-level learner and *meta learner*.
2. Train the first-level learner on each view of the original data.
3. Predict the labels of each view using the first-level learner. Each view will produce a prediction vector with associated prediction probabilities.
4. Form a new matrix by column binding the prediction vectors and the true labels. This matrix forms the new training data  $D'$  for the meta-learner.
5. Train the meta-learner with  $D'$ .
6. Generate the final multi-view stacking model.

Table 3.3: An example of the new generated dataset  $D'$ .

First-Level Learners						Associated Prediction Probabilities					True Label	
$l_1$	$l_2$	...	$l_i$	...	$l_n$	$p_1$	$p_2$	...	$p_i$	...	$p_n$	$y$

From an abstract view, assuming that  $Y_{it}$  is a dimension of the  $n$ -dimensional time series  $Y_t = (Y_{1t}, Y_{2t}, \dots, Y_{it}, \dots, Y_{nt})$ , each view  $V_i$ , where  $V = (V_1, V_2, \dots, V_i, \dots, V_n)$  is the set of views, represents a dimension  $Y_{it}$  of the multivariate time series  $Y_t$ . Thus, we have as many views as dimensions.

The first-level learner takes as input the time series values coming from each view. Then, each view will generate its own predicted labels with associated prediction probabilities with the form  $[l_i, p_1, p_2, \dots, p_j, \dots, p_k, y]$ , where  $l_i$  is the predicted label of the first-level learner  $i$ ,  $p_j$  is the associated prediction probability for each class  $j$  of the  $k$  possible classes, and  $y$  is the true label.

A new dataset  $D'$  is then created by column binding the output of each view and the true labels. We remind that these outputs consist of the predicted labels and the associated prediction probabilities for each of the  $k$  possible classes. Thus  $D'$  has the form shown in Table 3.3, where  $l_i$  is the predicted label of the first-level learner  $i$ ,  $p_i$  is the probability of this prediction, and  $y$  is the true label.

After generating a new dataset  $D'$ , a second-level classifier, or *meta-learner*, is trained over  $D'$  through ensemble learning [146]. This approach allows to preserve the statistical properties of each view and learn the classes of the MTS instances with a significant improvement in the classification accuracy.

Many ensemble methods [146] have been proposed to further enhance the algorithm accuracy by combining learners rather than trying to find the best single learner. Due to their versatility and flexibility, ensemble methods attract many researchers and can be applied in different domains including, but not limited to, time series classification [52] and time series segmentation [43]. In a previous work [43], we used a multi-view approach for segmenting MCS data where we employed an unsupervised learning for change detection on each view.

### 3.3.4 . Micro-environment recognition model

In this section, we provide an overview of our proposed framework for micro-environment recognition in the context of MCS [1]. Figure 3.5 provides a panorama of the steps followed to achieve the micro-environment recognition objective. It shows a roadmap from the derivation of air quality and trajectory data (i.e. step 1), to data preparation (i.e. step 2) which produce data ready to be consumed by a univariate time series classification model (e.g. kNN-DTW, LSTM, random forest, decision tree, etc.) (i.e. step 3). The outputs of the univariate time series classification constitute a new data set (i.e. step 4) which serves as an input for a meta-learner (i.e. step 5). The meta-learner produces the final classification results. In the following sections, we discuss each step separately. It is necessary to mention that the red dashed lines represent the hybrid approach, which we



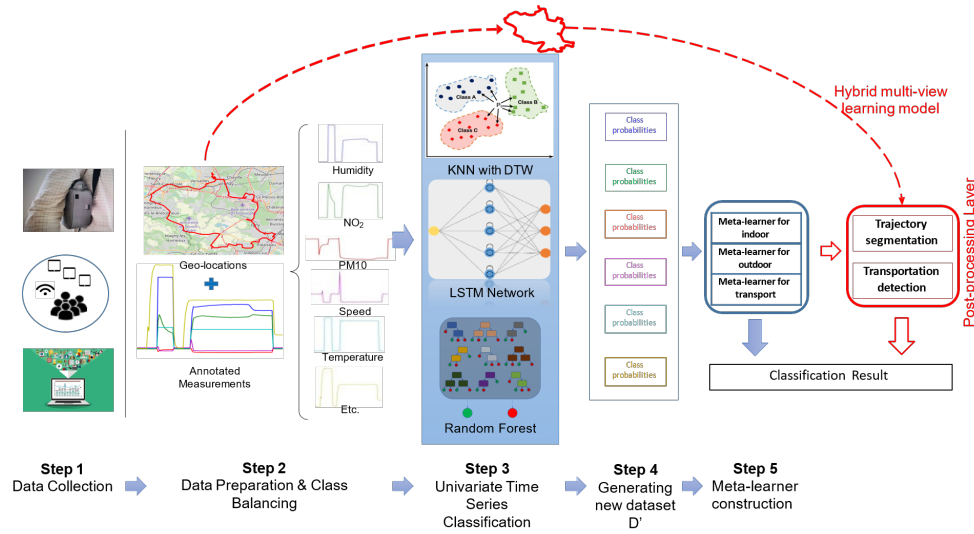


Figure 3.5: Overview of the micro-environment recognition process.

thoroughly discuss in Section 3.3.5..

## Data Collection

The first step of the micro-environment recognition process is the data collection, and it has been fully addressed in Section 3.2.2.

## Data Preparation

The second step is the data pre-processing which includes data de-noising, data imputation, data segmentation, and class balancing.

First, most sensor data are noisy, with irrelevant measurements from the actual condition. Even though the sensor data quality is a permanent preoccupation of the project, we observed the noisy data in both GPS (due to signal loss) and air quality data after careful evaluation before data selection and periodic qualification during the campaign [77]. The sensors for climatic parameters do not show such defects. Therefore, a de-noising process is applied on both GPS and air quality data. Precisely, we distinguish between peaks and artefacts by referring to the expert's judgment. The peaks that are judged to be real are then conserved. The same goes for GPS data which has been cleaned based on a minimum threshold of velocity (here 130km/h as it is the speed limit in France). Beyond this threshold, GPS points that are in charge of producing the velocity will be removed.

Second, the collected sensory data are usually incomplete due to device error or communication issues, with missing values at some time stamps. Therefore, we set a threshold of ten consecutive missing steps to conduct the imputation process. In other words, we perform data imputation on missing intervals that do

not exceed 10 minutes (i.e., 10 steps). Precisely, new values are inferred with the linear interpolation approach on the non-missing temporal neighbors; that says, new values are interpolated by a linear function of the two temporal ends of the missing values. Globally, the highest quality sample of annotated data is selected as a baseline to validate the process of micro-environment recognition. The idea is to generalize the micro-environment recognition to all participants' data by using the model derived from a good-quality dataset. We describe in Section 3.3.6 how the high quality sample of annotated data is selected.

Third, data is segmented into samples of fixed length (here 5 minutes). The choice of the fixed length value is discussed in Section 3.3.6. Each segment will be assigned a unique label. Essentially, the proposed model will take the observed measurements of the segment as input and produce a unique label by virtue of the multi-view learning.

Last but not least, micro-environment recognition is also subject to class imbalance problem. Usually, individuals spend most of their time indoors, either at home or at the office. A dataset is imbalanced if the classification categories are not equally represented, which is the case in our study. Therefore, because of this problem (home is the majority class, followed by office), the likelihood of having a good accuracy value of the classification is very high. The classifier will practically attribute the majority class to almost every data segment and fail to detect the minority classes, which leads to an overall good accuracy but does not necessarily reflect the actual performance of the classifier. Hence the solution of re-sampling and data augmentation, which are the commonly used techniques to solve this problem.

For data re-sampling, random oversampling of the minority classes and random under-sampling of majority classes are the most popular approaches. However, the random oversampling approach usually introduces duplicates to stabilize the training process, which does not thoroughly explore the valuable information from the data. Therefore, some work considers synthesizing new samples from the minority class. For instance, synthetic minority oversampling technique (SMOTE) [28] under samples the majority class and over samples the minority one based on the K-nearest neighbors. SMOTE selects samples that are close in the feature space, then generates a synthetic sample nearby. This procedure can be used to create as many synthetic examples for the minority class as required.

For data augmentation, Generative Adversarial Network (GAN) [56] has shown promising performance among various types of data, which uses existing data more effectively than re-sampling techniques. In Time series domain, the Time series Generative Adversarial Networks (TimeGAN or TGAN) [136] was proposed recently to generate realistic time series data considering the temporal dependency. However, in practice, it is generally hard to converge the adversarial training process with very limited samples [10], which is the case in our context.

Therefore, we combine both approaches of data re-sampling and data aug-

mentation. First, we adopt SMOTE to under-sample the majority classes and over-sample slightly the minority classes. Then we apply the TimeGAN network to generate new samples over the minority classes. Figure 3.6 and 3.7 show the data distributions before and after class balancing respectively.

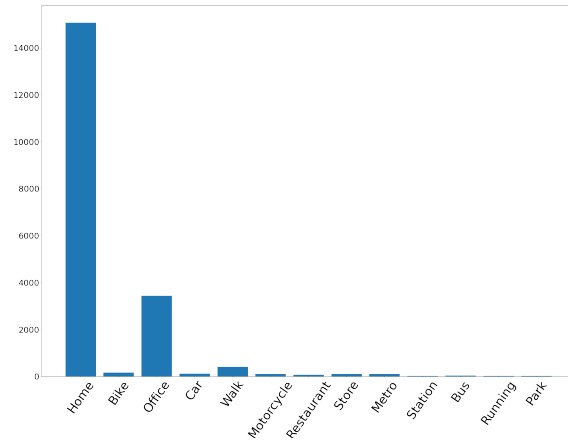


Figure 3.6: Distribution of data over classes before class balancing.

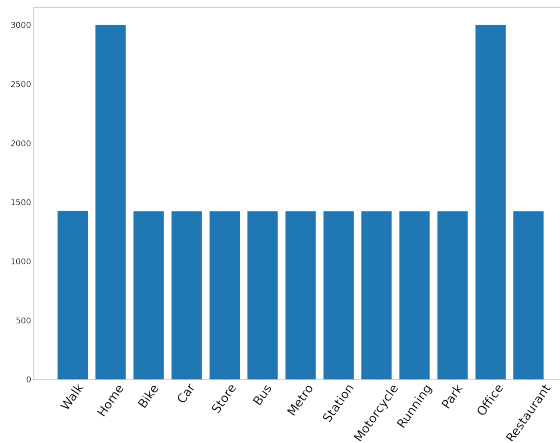


Figure 3.7: Distribution of data over classes after class balancing.

### Multi-View Learning Model Application

We propose to learn the micro-environment of participants from multivariate time series (MTS) through a two-stage model based on multi-view learning. The classification model consists of training a first-level learner on each view (i.e. step 3

in Figure 3.5), and then train a meta-learner (i.e. step 5 in Figure 3.5) to combine the output of each view and enhance the global accuracy of the classification. As stated before, we have as many views as dimensions. For instance, given a multivariate time series with four dimensions: temperature, humidity, speed, and NO2, each dimension will be considered as a view. Therefore, four different views will be considered in the multi-view learning model. The spatial dimension as GPS tracks is not considered as a view because of the low spatial coverage. Also the target class is the type of micro-environment. The spatial dimension has less impact than the temporal pattern (2 locations could be spatially close but have different patterns in terms of exposure if one is indoor and the other outdoor).

In step 3, the first-level learner (e.g.  $k$ -NN, LSTM, random forest, decision tree, etc.) takes as input the values of the time series data coming from each view, and outputs, for each view, a vector in the form  $[l_i, p_1, p_2, \dots, p_j, \dots, p_k, y]$ , where  $l_i$  is the predicted label of the first-level learner  $i$ ,  $p_j$  is the associated prediction probability for each class  $j$  of the  $k$  possible classes, and  $y$  is the true label. Let us take the example above of the multivariate time series with four dimensions which are temperature, humidity, speed, and NO2, and examine the output of the first level learners. Let us say that our objective is to classify the MTS into three classes that are indoor, outdoor, and transport, while supposing that the true label is indoor. The temperature view will generate its own predicted label - let us say - indoor, and the associated predictions probabilities in this form  $[l_{temperature} = indoor, p_{indoor} = 0.6, p_{outdoor} = 0.2, p_{transport} = 0.2, y = indoor]$ . In the same way, the three remaining dimensions shall generate their own predicted labels with corresponding probabilities in this structure:

$[l_{humidity} = indoor, p_{indoor} = 0.7, p_{outdoor} = 0.1, p_{transport} = 0.2, y = indoor]$ ,  
 $[l_{speed} = outdoor, p_{indoor} = 0.4, p_{outdoor} = 0.5, p_{transport} = 0.1, y = indoor]$ ,  
 $[l_{NO2} = transport, p_{indoor} = 0.2, p_{outdoor} = 0.2, p_{transport} = 0.6, y = indoor]$ .

In step 4, we aimed at giving a weight for each learner. Therefore, a new dataset  $D'$  is generated by column binding the output of first-level learner and the true label as shown in Table 3.3, where  $l_i$  is the predicted label of the first-level learner  $i$ ,  $p_i$  is the probability of this prediction, and  $y$  is the true label. Continuing with the same example of the four dimensional MTS above, the feature structure of the generated dataset would be in the structure shown in Table 3.4.

Table 3.4: A concrete example of the new generated dataset  $D'$ .

First-Level Learners				Associated Prediction Probabilities				True Label
temperature	humidity	speed	NO2	temperature	humidity	speed	NO2	
indoor	indoor	outdoor	transport	0.6	0.7	0.5	0.6	indoor

In step 5, and after generating a new dataset  $D'$ , a *meta-learner* is trained over  $D'$ . That said, by referring to the example above, the second level-learner (e.g.

*Random Forest*) will take the generated features (i.e. every view detected label plus its corresponding probability) as input and produce the final detected label. For instance and from  $D'$  shown in Table 3.4, the meta-learner takes as input the label produced by the view "temperature" (i.e. indoor) and its associated prediction probability (i.e. 0.6), plus the labels and their corresponding probabilities from the other three views (i.e. humidity, speed and NO2). Therefore, the meta-learner's input has the following structure [indoor, indoor, outdoor, transport, 0.6, 0.7, 0.5, 0.6] and produces the final label (e.g. indoor) from the combination of labels and their associated prediction probabilities.

One of the advantages of multi-view learning is its versatility in first and second level learners' choices. One can flexibly substitute classifier choices as wished between kNN, LSTM, random forest decision tree, or any other classifier [44]. In this work, we opt for *Random Forest* classifier for the first as well as meta-learners since it has shown high performance when applied in the human activity recognition domain [52].

### 3.3.5 . Hybrid Multi-view Learning Model

The multi-view learning model records some limitations, especially when it comes to discriminating between some indoor micro-environments that share similar characteristics such as "home" and "office", or between some transportation modes. Besides, the time of presence are often characteristic of some micro-environments (e.g. night time is likely to indicate home, and working time usually denotes the office). Identifying precisely some stay locations from GPS data is possible, and may improve discriminating the corresponding micro-environments. Thus, the need for an improvement in the model seems necessary.

We introduce new optimisations based on these observations. Specifically, the new optimisations include two approaches. The first one is *privacy invasive*. It includes the exact locations of home and office reported by participants in the post-processing layer. Certainly, it requires to have this private information ahead. The second approach is *privacy friendly*. The latter approach (i.e. privacy friendly) will be the subject of discussion in this section.

Figure 3.5 shows the new privacy friendly optimisations presented by the red dashed lines, which consist mainly of adding trajectory data as another layer while post-processing the results of the multi-view learning model, along with the disambiguation between home and office based on the location.

This post-processing layer consists of splitting trajectory data into stop and move segments (i.e. the trajectory segmentation box in Figure 3.5). We propose a stop detection algorithm based on grid density that we will present and discuss in the following section. We tag every stop with a unique and specific number to distinguish between them. Plus, we infer the location of home and office based on a priori rules according to the time of presence in the stop and the density of the stop. We further discuss these rules in Section 3.3.5. Furthermore, and after distinguishing between stop and move segments, the move segments are labeled

by transportation means (e.g., metro, bus, car, etc.), which is represented by the transportation detection box in Figure 3.5. We take advantage of the work of Etemad *et al.* [47], which we have already discussed in Section 3.2.1, to detect the transportation mode and include the results in the post-processing layer. In the following section, we present our proposed algorithm for Grid Density-Based Stop Detection (GDSD).

### Stop Detection

In this section, we present a novel and robust algorithm for stop detection based on GPS data only, namely Grid Density-Based Stop Detection (GDSD). This approach can be used either as a separate view in the multi-view learning model to infer the stay places from GPS data (i.e. mobility view), or as a post-processing layer to correct the ambiguity between home, work, and other stop places. In the current work, the mobility data is used as a post-processing layer.

GDSD approach takes GPS points as input, and outputs segments of the same fixed length as the multi-view model's segments, to ensure their comparability. Each segment is labeled with the number of the stops or the label "home" or "office". Let us take an example of four segments  $s_1$ ,  $s_2$ ,  $s_3$ , and  $s_4$ . Each segment is 5 minutes long. The multi-view model results assign to segments  $s_1$  and  $s_2$  the home label, and to  $s_3$  and  $s_4$  the work label. However, according to the results of stop detection model, all the four segments belong to the same cluster, which implies that the four segments are all together either at home, work, or any other indoor micro-environment. But, since these four segments ( $s_1$ ,  $s_2$ ,  $s_3$ , and  $s_4$ ) are labeled "home" by the GDSD approach, the final class shall be "home". That is precisely the objective of this stop detection extension: to eliminate the ambiguity between stops micro-environments and improve the performance of the multi-view model.

Therefore, in this approach, the GPS tracks (i.e. latitude and longitude) are transformed into discrete values referencing a pixel of a rectangular grid with a spatial resolution (here of  $50 \text{ m}^3$ ). Then, in order to organize the cells in a way that allows to maintain the locality of spatially close GPS points, we adopt spatial indexing using 2D Hilbert Space-Filling Curves (SFC), which provides a grouping feature per proximity [93]. In other words, neighboring cells are likely to be assigned to close Hilbert indices. Figure 3.8 shows the rasterization of the spatial dimension using Hilbert SFC. The spatial extent is defined to cover the study area (i.e. Paris region). It is worth mentioning that we only maintain cells corresponding to the locations with GPS data, and discard cells with no GPS data within.

The adopted rasterization approach allows us to derive the stay areas (often indoors) of participants. We can discover the places where a participant spends

---

<sup>3</sup>This value has been chosen in accordance with the resolution adopted by Airparif (the agency in charge of AQ monitoring in Paris Region, also part of Polluscope consortium) in their simulation models.

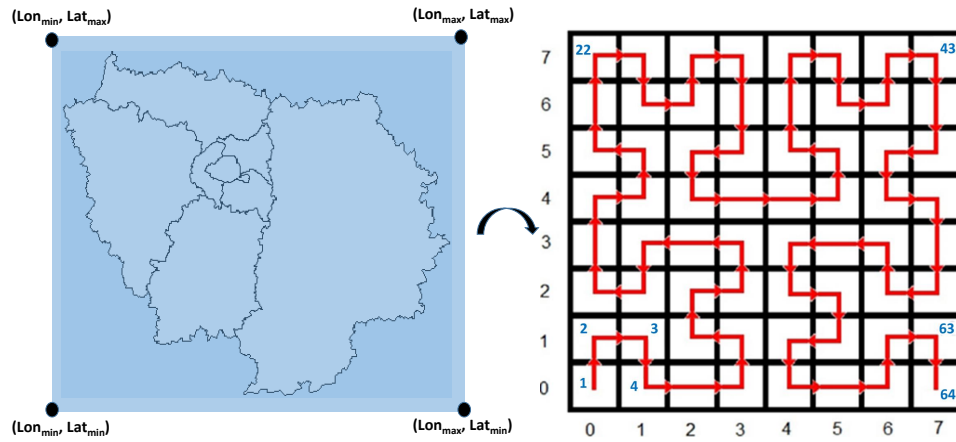


Figure 3.8: Spatial Dimension Representation

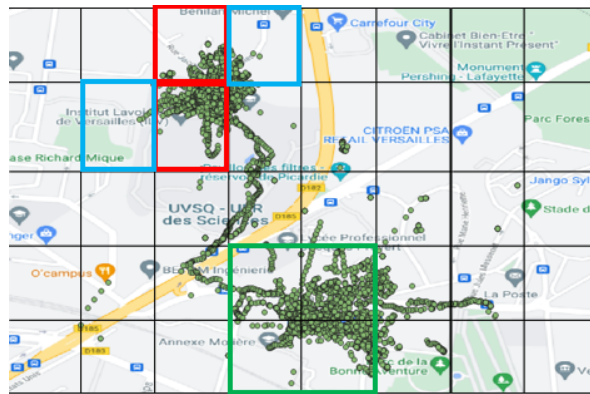


Figure 3.9: Sample trajectory

time the most based on cells densities.

When looking at the sample of GPS trajectory points in Figure 3.9, we can easily and unambiguously detect the existence of two stops plus some noise points that may or may not belong to any of the stops. The reason why the human mind detects or recognizes the stops is because the density of points inside the stops is higher than outside them. We mimic the same human brain rational reasoning to detect the stops based on the cell density.

We formalize our intuitive notion of deriving stop places from a spatial dataset  $D$  in a 2D euclidean space. The key idea is to set a minimum density threshold  $MinDens$  in a cell in order to be detected as a stop cell. The adjustment of  $MinDens$  is utterly empirical and depends on the dataset size and GPS sampling frequency. For instance, the  $MinDens$  in a dataset collected over one week with

a high sampling frequency (e.g. every one second) is naturally different than a *MinDens* chosen from a dataset collected over one day with the same sampling frequency. We explain further in Section 3.3.6 how we set the *MinDens* threshold.

Furthermore, because of the limitations of GPS readings, some stop clusters may be shared by more than one cell (e.g. green cells in Figure 3.9). For this reason, there is a need to include in the detected stop the neighbouring cells that may belong to the same stop. Yet, as Hilbert SFC are by definition hierarchical, one has only to divide the Hilbert index by  $2^{2n}$ , where  $n \in \mathbb{N}$  to access the neighbouring cells.

Thereafter, the stop detection algorithm based on grid density takes as input the minimum density threshold *MinDens* and  $n$  (as in  $2^{2n}$ ). The algorithm then joins the neighbouring cells to the detected stop even though these neighbouring cells do not verify the density condition, forming a new cluster composed of several cells and at least one cell that verifies the density condition.

Algorithm 1 presents a basic version of the stop detection algorithm without focusing on GPS data pre-processing and cleaning. Primarily, the algorithm rasterizes the data and creates a Hilbert index that is assigned to each cell. The algorithm then selects a set of cells indices whose density is higher than the desired density threshold *MinDens*. All the selected cells are stops candidates. The algorithm moves systematically to higher level of hierarchy by dividing the acquired Hilbert index by  $2^{2n}$  (i.e., a grouping of 4 cells, 16 cells, 64 cells, etc.) and considers the whole grouping cells around a stop candidate as a stop. The remaining cells will be considered as move segments. For instance, in Figure 3.9, let us suppose that only one of the four green cell verifies the density condition (the upper right green cell), then this cell forms a stop candidate. After dividing its Hilbert index by  $2^{2*1}$ , we systematically go to a higher level of hierarchy and the whole grouping of the four green cells will be considered as a stop.

However, some outlier points may slip out of the grouping cells (e.g. blue cells in Figure 3.9). The algorithm has another step which consists of post-processing the trajectory segments based on a temporal threshold *MinDur*. In other words, if a move (resp. stop) segment is jammed between two stop (resp. move) segments and the duration of this segment is less than *MinDur*, then this segment is to be merged with the previous segments.

A comparison between Grid Density-based Stop Detection (GDSD) and state-of-the-art approaches is discussed further in Section 3.3.6.

Next comes the step of inferring the location of home and work based on some a priori rules. The straightforward way is to draw the location of home according to the time of presence in the stop. If the participant is static between 2am and 4am every day at the same location, it is very likely that the location in question is home. Another criterion for inferring the location of home and work is the density of the stops. Usually and based on common sense, people spend the most of their time in their home followed by their work. Thereby, the densest stop is likely to be



---

**Algorithm 1** Grid Density-Based Stop Detection (GDSD)
 

---

```

1: procedure GDSD(MinDens, n, MinDur)
2:   SetOfCells = {}
3:   create HilbertIndex from (Lon, Lat) ▷ Create the Hilbert index
   from latitude and longitude.
4:   SetOfIndex = {HilbertIndex} ▷ Create a set of all the possible
   Hilbert indices.
5:   density ← GroupBy(HilbertIndex) and count
6:   stops ← HilbertIndex where density ≥ MinDens
7:   k = 1
8:   for Hindex in stops do
9:     x ← Floor(Hindex/ $2^{2n}$ )
10:    SetOfCells := SET( $i/2^{2n} = x$ )  $\forall i \in$  SetOfIndex
    label(SetOfCells) ← k
11:    k := k + 1
12:  end for
13:  for Hindex in SetOfIndex \ stops do
14:    label(Hindex) ← -1
15:  end for
16:  segments := Set(segments) ▷ segments is the set of all the stop
   segments
17:  j := 1
18:  while j < segments.size do
19:    if Duration(segments[j]) < MinDur
20:    & segments[j].label == segments[j + 1].label then
21:      segments[j - 1] ← concat(segments[j - 1], segments[j])
22:      del segments[j]
23:    end if
24:  end while
25:  return segments
26: end procedure

```

---

home and the second densest stop is work, *ceteris paribus*. These assumptions are confirmed by the participants declared annotations of their micro-environments - if they exist.

### 3.3.6 . Experiments and Results

The experiments are carried out in different environments. The multi-view learning model was implemented in Python 3.6 using scikit-learn 0.23.2 and tslearn [116]. The deep-learning models (MLSTM-FCN [72], TapNet [139]) were trained on a single Tesla V100 GPU of 32 Go memory with CUDA 10.2, using respectively Keras 2.2.4 and PyTorch 1.2.0.

## Experimental Settings

We evaluate the proposed models in these experiments using real-life data collected in the scope of the Polluscope project. In Polluscope, three data collection campaigns have been conducted, covering the whole study area (i.e., Paris region). Each campaign was spread over 12 weeks, with a collection generally carried out every other week (in order to check and re-qualify the sensors). 103 volunteers participated in the data collection phase, which lasted one week for each participant. These participants are equipped with a kit that contains air pollution sensors and tablets empowered with GPS chipsets. The sensors collect, every one minute, time annotated concentrations of Particulate Matters (PM1.0, PM10, PM2.5), Nitrogen dioxide NO<sub>2</sub>, Black Carbon (BC), temperature, and relative humidity. The tablet serves to geolocate the participants and to fill in their time activity via an Android app developed for this purpose. The speed dimension was derived from the geo-locational data.

In total, 13 activities (i.e., micro-environment to recognize) are considered in this study, which can be organized into three categories:

- Indoor environment: *home, office, restaurant, store, station*
- Outdoor environment: *park, walk, run, bike*
- Transport environment: *metro, car, bus, motorcycle*

Previously (i.e. in [1]), we related to the annotation of the given tool (i.e., an Android app installed in the tablet). In this work, data have been enriched both based on a tool (called TripBuilder Web) [26], and a thorough human control of participants' annotations within the third campaign called RECORD [25]. Thereby, this data is more reliable than our previously used data collected during the second campaign, called VGP. We select the data of 13 participants with the best annotation activities in the RECORD campaign. Overall, the dataset contains 8 dimensions, more than 1 million rows (per dimension), with an average of 82071 rows per participant. The collected data are split into two thirds for training and one third for testing, with care taken to keep the data of each participant grouped

Table 3.5: Average time spent per micro-environment.

Activity	Stay duration (in minutes)
Office	446
Bus	13
Home	899
Station	4
Store	24
Motorcycle	20
Metro	17
Park	76
Restaurant	46
Running	76
Car	29
Bike	50
Walk	12

either in training or testing set. We use the cross-validation score with “repeated stratified k-fold” to re-split the training set into training and validation sets, while we evaluate the overall model performance on the testing set.

Considering the temporal feature of the data, we segment the collected data into samples of 5 minutes’ length at maximum. Usually, people spend most of their time indoors. We should thus consider outdoor activities with a short period compared to indoor activities. For example, the average time spent in “station” is around 4 minutes as shown in Table 3.5 which depicts the average time spent per micro-environment. Participants tend to spend more time in some micro-environment (typically “home” and “office”) than others (e.g. “walk”, “metro”, “store”, etc.). Globally, as shown in Figure 3.6, the distribution of data samples is highly imbalanced over the different classes, leading to poor classification performance, especially for the minority classes. More precisely, the model tends to optimize the global loss error which is biased towards the majority classes while ignoring the minority ones. In consequence, the obtained accuracy is not reliable to evaluate the actual model performance. To cope with this problem, we re-balanced the classes via data re-sampling and data augmentation as mentioned in section 3.3.4 when pre-processing the data. Figure 3.6 and figure 3.7 show the data distributions before and after class balancing, respectively.

### Experimental Design

First, we evaluate our basic multi-view learning model without integrating the post-processing layer. Considering the mobility information in our data, we carry out our experiments on the datasets with or without integrating the *speed* variable.

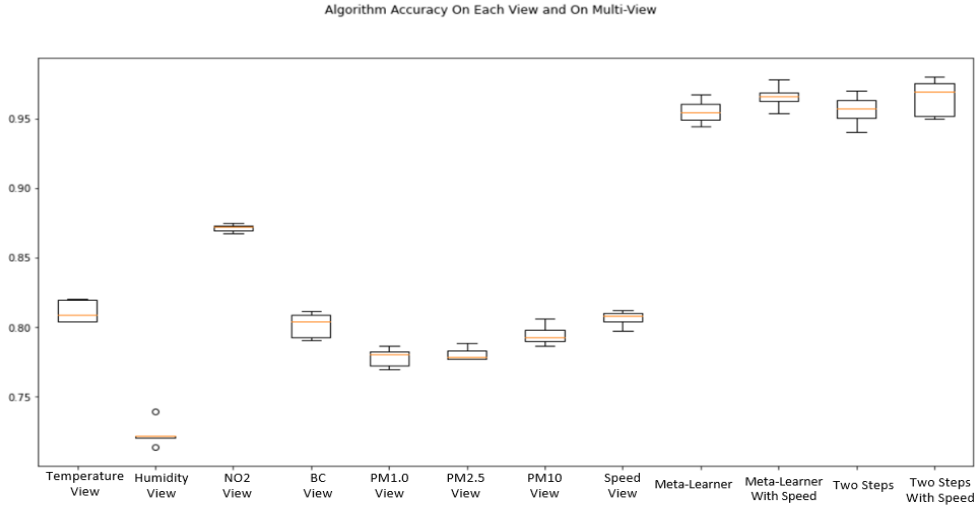


Figure 3.10: Accuracy among different views.

Furthermore, to thoroughly evaluate the importance of the mobility information, we introduce and evaluate a two-step approach by first discriminating between *indoor*, *outdoor*, and *transport* micro-environments, followed by a refinement step to learn a more specific class.

Then, we evaluate our proposed algorithm for stop detection, which is a key component in our post-processing layer. We compare it with the state-of-the-art models implemented in Scikit-Mobility [101].

Finally, we conduct an extensive experiment considering various optimization techniques proposed in the post-processing step. We evaluate the effect of the post-processing layer not only on our multi-view learning model but also on other classic MTSC models. We optimize the proposed approaches by either analyzing the exact geolocation of the participant (privacy-invasive method) or using a priori rules (privacy-friendly method).

### Model Performance without Post-processing

In this section, we detail the experimental results of the multi-view learning model without integrating the post-processing layer. First, we evaluate the first-level learners on each single view and the multi-view learner on the global view. We evaluate as well the multi-view learner when applying the two-step approach which learns firstly the coarse-grained classes (i.e., *indoor*, *outdoor* and *transport*) then refine them into more specific classes (e.g., *home*, *park*, *metro*, etc.). Then, we compare the multi-view learner with MLSTM-FCN [72], the state-of-the-art on Multivariate Time Series Classification.

As mentioned in Section 3.3.4, the micro-environment recognition can be formulated as a Multivariate Time Series Classification (MTSC) problem, and the

Table 3.6: Performance of Multi-view Learner ( with/without speed)

class	Without Speed			With Speed		
	Precision	Recall	F1 Score	Precision	Recall	F1 Score
Walk	0.96	0.86	<b>0.91</b>	0.95	0.87	<b>0.91</b>
Bus	0.99	0.96	<b>0.98</b>	0.98	0.97	<b>0.98</b>
Office	0.96	0.88	0.92	0.97	0.92	<b>0.95</b>
Restaurant	0.97	0.97	0.97	0.99	0.97	<b>0.98</b>
Home	0.87	0.97	0.92	0.90	0.99	<b>0.94</b>
Bike	0.92	0.97	0.94	0.96	0.99	<b>0.97</b>
Car	0.99	0.98	0.98	0.99	0.99	<b>0.99</b>
Store	0.94	0.93	0.94	0.96	0.96	<b>0.96</b>
Metro	0.96	0.93	0.94	0.98	0.95	<b>0.96</b>
Station	0.98	0.96	0.97	0.99	0.97	<b>0.98</b>
Motorcycle	0.99	0.99	<b>0.99</b>	0.99	0.99	<b>0.99</b>
Running	0.99	0.99	<b>0.99</b>	0.99	0.99	<b>0.99</b>
Park	0.99	0.98	0.98	0.99	0.98	<b>0.99</b>

multi-view learner combines the predictions of each independent view (i.e., dimension) from the first-level learners to get the final classification results. In Figure 3.10, we report the accuracy of the first-level learners over the different views, as well as the multi-view learner and the two-step approach with and without considering the mobility (i.e., speed) dimension. Globally, the results suggest that the multi-view learner shows comparable performance, with or without adopting the two-step approach. Integrating the *speed* dimension helps slightly improve the performance of the multi-view learner. We observe that the first-level learners usually have low accuracy performance, which is not surprising as the incomplete local information is not enough to train a reliable model. By combining the local information from different views, the multi-view learner can improve the model accuracy significantly.

To know how our multi-view learner performs compared to the state-of-the-art work, we select MLSTM-FCN [72], a powerful deep learning model for Multivariate Time Series Classification. We show as well the detailed evaluation results when applying the two-step approach. Since MLSTM-FCN requires enormous computational resources for parameter optimization, we train the model on GPU. In contrast, our multi-view-based approaches are trained on a normal CPU with less requirement on computational resources. For each of the models, we study the impact of using or not the mobility data and report the performance in terms of *precision*, *recall*, and *F1 score*.

The detailed results are grouped in Table 3.6, 3.7, and 3.8. Globally, the three models have comparable results before and after adding mobility. While MLSTM shows slightly better performance than the two-step model, the latter outperforms the multi-view model. Looking at the F1-score, the out-performance of MLSTM,

Table 3.7: Performance of MLSTM-FCN ( with/out speed)

class	Without Speed			With Speed		
	Precision	Recall	F1 Score	Precision	Recall	F1 Score
Walk	0.98	0.95	<b>0.96</b>	0.94	0.97	<b>0.96</b>
Bus	1.0	1.0	<b>1.0</b>	1.0	1.0	<b>1.0</b>
Office	0.97	0.95	<b>0.96</b>	0.96	0.94	0.95
Restaurant	1.0	1.0	<b>1.0</b>	1.0	1.0	<b>1.0</b>
Home	0.97	0.97	<b>0.97</b>	0.98	0.97	<b>0.97</b>
Bike	0.98	1.0	<b>0.99</b>	0.98	1.0	<b>0.99</b>
Car	0.99	1.0	<b>1.0</b>	0.98	1.0	0.99
Store	0.99	1.0	<b>0.99</b>	0.99	1.0	<b>0.99</b>
Metro	0.99	1.0	<b>0.99</b>	1.0	0.97	<b>0.99</b>
Station	0.99	1.0	<b>1.0</b>	1.0	1.0	<b>1.0</b>
Motorcycle	1.0	1.0	<b>1.0</b>	1.0	1.0	<b>1.0</b>
Running	1.0	1.0	<b>1.0</b>	0.99	1.0	0.99
Park	1.0	1.0	<b>1.0</b>	1.0	1.0	<b>1.0</b>

Table 3.8: Performance of Multi-view Learner (2-step approach with/out speed)

class	Without Speed			With Speed		
	Precision	Recall	F1 Score	Precision	Recall	F1 Score
Walk	0.93	0.97	<b>0.95</b>	0.95	0.96	<b>0.95</b>
Bus	0.99	0.99	<b>0.99</b>	0.99	0.99	<b>0.99</b>
Office	0.97	0.92	<b>0.94</b>	0.97	0.91	<b>0.94</b>
Restaurant	0.99	0.98	<b>0.98</b>	0.99	0.98	<b>0.98</b>
Home	0.93	0.97	<b>0.95</b>	0.93	0.98	<b>0.95</b>
Bike	0.97	0.96	0.96	0.97	0.97	<b>0.97</b>
Car	0.98	0.99	<b>0.99</b>	0.99	0.99	<b>0.99</b>
Store	0.98	0.97	<b>0.97</b>	0.98	0.96	<b>0.97</b>
Metro	0.98	0.97	<b>0.98</b>	0.98	0.98	<b>0.98</b>
Station	1.0	1.0	<b>1.0</b>	0.99	1.0	0.99
Motorcycle	0.99	0.98	0.98	0.99	0.99	<b>0.99</b>
Running	0.98	0.98	<b>0.98</b>	0.98	0.98	<b>0.98</b>
Park	0.99	0.96	0.97	0.99	0.97	<b>0.98</b>

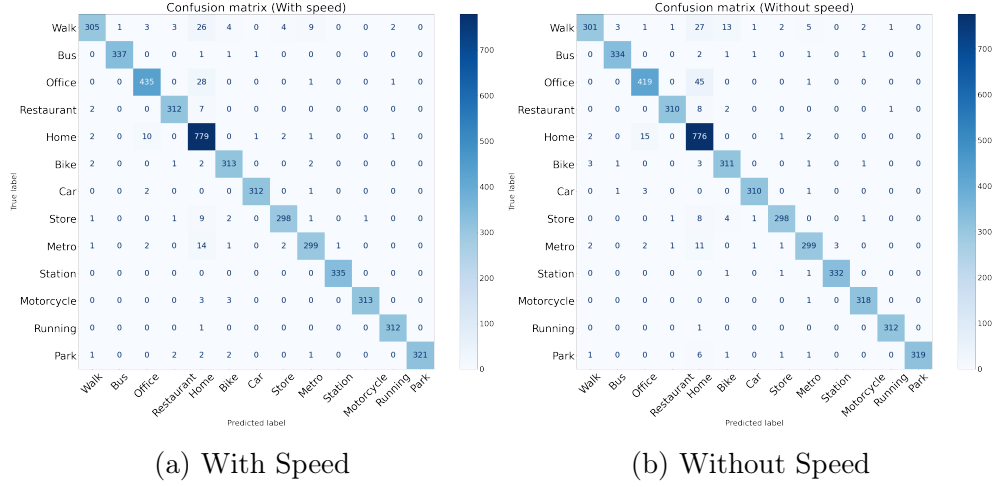


Figure 3.11: Multi-view Approach Confusion Matrix

compared to the two-step model, does not go beyond 3 point (e.g. 0.96 and 0.99 for the class bike) before adding mobility, and 2 points (e.g. 0.97 and 0.99 for the class store) after adding mobility, whereas the difference between the two-steps model and the multi-view model does not exceed 4 points (e.g. 0.91 and 0.95 for the class walk) before and after adding mobility.

As for our multi-view learner, when integrating the speed dimension for model training, we observe an improvement in the model's performance, particularly the F1-score, while the performance of MLSTM-FCN does not improve or even deteriorates. Figure 3.11 shows the confusion matrix of multi-view approach. Figure 3.11a reports the confusion matrix with the presence of the mobility dimension (i.e. speed), while figure 3.11b corresponds to the confusion matrix of the model with the absence of mobility dimension. We notice that the model can easily discriminate between the "indoor", "outdoor" and "transport" activities, but it cannot perfectly distinguish between the micro-environments inside each category. For example, even though some of the samples in the "home" micro-environment are falsely predicted as "restaurant" or "office", the three micro-environments, "home", "office" and "restaurant" can be classified as indoor. Thereby, we introduced a grouping step before recognizing the micro-environment. In this step we classify the sample into either an "indoor", "outdoor", or "transport" environment. Based on the classification result, a model will be specialized for each indoor, outdoor or transport micro-environments. Table 3.8 shows the results of the added step.

### Stop Detection Performance

In this section, we evaluate our proposed algorithm, Grid Density-based Stop Detection (GDSD). First, we study the parameter effects on GDSD's performance and select the best ones when applying the algorithm. Then, we compare GDSD

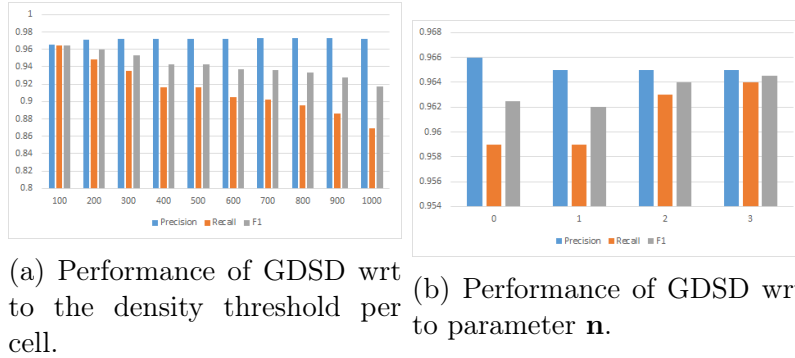


Figure 3.12: Parameters' effect on Grid-Density Stop Detection (GDSD)

with Scikit-Mobility [101], the state-of-the-art approach designed for stop detection. Here, we adopt only the GPS data of the 13 participants in the Polluscope RECORD campaign.

Before applying the proposed approach, the critical question is to set the minimum density per cell (i.e., density threshold) and the number of grouping cells within a stop. Generally, these parameters are set empirically. Without prior knowledge of the data, it is necessary to conduct various tests to discover the best parameters. To this end, we tune one parameter while blocking another one to observe how the model performance evolves. Precisely, we set the values between 100 and 1000 points, with a step of 100, for the density threshold. The number of grouping cells  $n$  varies between 0, 1, 2, and 3, meaning a grouping of, respectively, 1, 4, 16, and 64 cells. Firstly, we start by searching for the optimal value of the number of grouping cells. We iterate over the values of the density threshold (i.e. between 100 and 1000 with a step of 100). On each round, we compute the performance of the model while iterating the values of  $n$ . We report the results with the optimal iterations. Figure 3.12 shows the parameters' effects on the model's performance. From Figure 3.12a (with  $n=3$ ), we observe that the precision improves slightly, whereas the recall drops with the increase of density threshold, indicating a trade-off between precision and recall when the density threshold equals 100. This observation is supported by the F1-score, which values is optimum at 100. As for the parameter' effect of  $n$ , Figure 3.12b (with density threshold equal to 100) shows that when  $n$  increases, the precision drops slightly, whereas the recall score increases a little, depicting an adequate trade-off between precision and recall when  $n$  equals 3 (i.e. a grouping of 64 cells). We observe as well an optimal F1-score under this setting. In the rest of the chapter, we set grid density threshold to 100 and set  $n$  to 3.

To validate the performance of our proposed stop detection algorithm GDSD, we compare it with the state-of-the-art models implemented in Scikit-Mobility [101]. We conduct the experiments of the stop detection for each campaign participant separately. Table 3.9 depicts the results of this comparison. On the one



Table 3.9: Comparison between Scikit-Mobility and grid density based model.

Row ID	Participant ID	Scikit-Mobility			Grid Density Based		
		Precision	Recall	F1	Precision	Recall	F1
1	988088403	0.988	0.891	0.937	0.985	0.993	<b>0.989</b>
2	988231648	0.988	0.961	0.974	0.981	0.995	<b>0.988</b>
3	982228564	0.936	0.849	0.89	<b>0.953</b>	0.945	<b>0.949</b>
4	986002161	1.0	0.82	0.901	1.0	0.969	<b>0.984</b>
5	986939872	0.813	0.78	0.796	0.826	0.895	<b>0.859</b>
6	988335737	<b>0.908</b>	0.299	0.45	0.858	0.658	<b>0.745</b>
7	986174566	0.96	0.854	0.904	<b>0.991</b>	0.971	<b>0.981</b>
8	986884172	0.92	0.66	0.769	<b>0.985</b>	0.956	<b>0.97</b>
9	986938604	0.995	0.684	0.811	0.995	0.99	<b>0.992</b>
10	985935431	0.812	0.325	0.464	<b>1.0</b>	0.6	<b>0.75</b>
11	987014104	0.936	0.92	0.928	<b>0.993</b>	0.995	<b>0.994</b>
12	82119412	0.84	0.559	0.671	<b>0.953</b>	0.9	<b>0.926</b>
13	983602168	0.934	0.963	0.948	<b>0.966</b>	0.973	<b>0.969</b>

hand, when looking at the precision of the two models, the two approaches are comparable for some participants (e.g., rows ID 1, 2, 4, 5, and 9), our approach outperforms the baseline for others except for one participant (i.e., row ID 6). On the other hand, the recall score and the F1-score show that the grid-density-based approach outperforms the baseline on all participants. Overall, the proposed approach always has better performance than the baseline.

### Experiment Extension

In this section, we extend the experiments by applying the proposed post-processing techniques which are designed to enhance the micro-environment detection model. We apply four basic MTSC models:

- MVB: our proposed Basic Multi-View learning model.
- MV-2steps: our proposed Basic Multi-View learning model with two-step classification as shown in Section 3.3.6.
- MLSTM-FCN [72]: a powerful deep learning model for Multivariate Time Series Classification.
- KNN-DTW [12]: the most popular benchmark for Time Series Classification which adopts K-nearest neighbor (K-NN) classifier with dynamic time wrapping (DTW) distance.

As the models are trained on different hardware environments (e.g., MLSTM-FCN is training on a GPU, which is ten times faster than running on CPU), it is

unfair to compare them in terms of efficiency. However, according to the recent study [108], the deep learning-based models usually require more computational resources than classic data mining approaches; the lazy classifiers (e.g., KNN-DTW) are much slower than the tree-based classifier (e.g., Random Forest) due to the costly distance computations (e.g., DTW). As the first-level learner and meta-learner in our multi-view learning model are based on Random Forest, thus the model training and prediction are quite efficient compared to other models.

The model variants after applying the post-processing techniques are detailed in Table 3.10. We go through extensive experiments and test various model variants to select the best model combinations. We organize the model variants into two categories: **privacy-friendly** and **privacy-invasive** models. For privacy-friendly models, post-processing is performed using stop detection and transportation mode detection techniques, while for privacy-invasive models, additional private information is adopted such as *Location of Home (LH)* and *Location of Office (LO)*.

### Global accuracy comparison on the model variants

In this section, we used the trained model to predict the context of our real MCS data and we adopted the post-processing techniques on the results. Here, we show the global accuracy comparison between the models within each privacy category. Tables 3.11 and 3.12 report the accuracy of various privacy-friendly and privacy-invasive models, respectively. The *NaN* in the results of MLSTM and KNN models indicates that no complete data is collected, thus, the models are not applicable. More precisely, some variables are missing during the data collection process. However, the multi-view-based models succeed all to detect the micro-environment even some dimensions are missing.

For the privacy-friendly models, all the proposed multi-view-based models show higher accuracy compared to baselines (i.e., KNN-based and MLSTM-based models). On the one hand, there is a big performance difference between multi-view-based models and the baseline models, especially for the participants who did not collect the complete variable data on which the baseline models are not applicable (i.e., *NaN* value). On the other hand, the post-processing does show its generalizability which improves the performance of both multi-view-based and baseline models. Among all the privacy-friendly models, the MVP (Multi-view with Post-processing) model shows the best performance, which validates our reliability of our proposed model.

For the privacy-invasive models, we adopt the additional private information: Location of Home (LH) and Location of Office (LO), to check their impact on the models. However, since the baselines showed poor performance in the privacy friendly models, they will not be considered in this comparison. It is already known that, even with the exact locations of home and office, they will fail with missing dimensions. By considering the office location correction, the MVP+LO model demonstrates the best performance among the privacy-invasive models. More

Table 3.10: The description of various model variants

<b>Model</b>	<b>Description</b>
MVB	Basic multi-view model
MV-2steps	Multi-view model having 2 steps, first discriminate between indoor/outdoor/transport and then classify the micro-environment.
MV-2stepsP	Multi-view model having 2 steps, first discriminate between indoor/outdoor/transport and then classify the micro-environment, with a pre-processing step based on stop detection transportation mode detection models.
PMV	Multi-view model with a pre-processing step based on stop detection transportation mode detection models.
MVP	Multi-view model with a post-processing layer based on stop detection and transportation mode detection models.
MLSTMB	Basic MLSTM-FCN model.
MLSTMP	MLSTM-FCN with a post-processing layer based on stop detection and transportation mode detection models.
KNN-DTWB	Basic KNN-DTW model.
KNN-DTWP	KNN-DTW with a post-processing layer based on stop detection and transportation mode detection models.
MVB+LO	Basic multi-view model with a post-processing step based on the location of office.
MVB+LH	Basic multi-view model with a post-processing step based on the location of home.
MVP+LO	Multi-view model with a post-processing layer based on stop detection and transportation mode detection models as well as the location of office.
MVP+LH	Multi-view model with a post-processing layer based on stop detection and transportation mode detection models as well as the location of home.

Table 3.11: Performance comparison of various **privacy-friendly** models

Participant ID	MVB	MV-2steps	PMV	MVP	MV-2stepsP	MLSTMB	MLSTMP	KNN-DTWP	KNN-DTWP
988088403	88.2	89.2	89.6	<b>95.9</b>	95.8	63.5	66.6	85.8	86.5
988231648	90.9	90.7	90.9	93.5	<b>93.6</b>	NaN	NaN	NaN	NaN
982228564	94.3	93.8	91.7	94.8	<b>94.9</b>	23.6	24.0	74.8	76.7
986002161	<b>91.0</b>	89.5	89.5	<b>91.0</b>	89.7	NaN	NaN	NaN	NaN
986939872	<b>83.3</b>	82.2	80.0	88.1	86.4	34.4	34.4	72.0	75.2
988335737	60.9	59.2	59.2	<b>61.2</b>	59.5	NaN	NaN	NaN	NaN
986174566	92.0	92.5	92.1	92.2	<b>93.5</b>	10.8	11.3	76.7	78.3
986884172	85.9	85.7	86.5	<b>88.6</b>	88.3	NaN	NaN	NaN	NaN
986938604	98.6	98.4	98.3	<b>98.8</b>	98.7	37.6	37.3	91.4	91.9
985935431	90.7	90.8	<b>91.1</b>	90.9	<b>91.1</b>	NaN	NaN	NaN	NaN
987014104	98.1	97.6	97.6	<b>99.1</b>	98.6	38.5	39.6	89.8	92.2
82119412	96.8	95.9	96.0	<b>97.2</b>	96.2	10.0	9.8	66.0	65.5
983602168	89.8	89.9	89.1	92.4	<b>92.5</b>	NaN	NaN	NaN	NaN
Overall Accuracy	91.33	91.0	90.71	<b>93.43</b>	93.10	31.14	31.70	83.48	85.06

Table 3.12: Performance comparison of various **privacy-invasive** models

Participant ID	MVB +LO	MVB +LH	MVP +LO	MVP +LH
988088403	92.3	92.6	94.9	<b>96.0</b>
988231648	95.4	94.8	<b>96.0</b>	95.6
982228564	95.8	94.4	<b>95.9</b>	94.3
986002161	<b>95.7</b>	<b>95.7</b>	<b>95.7</b>	<b>95.7</b>
986939872	87.2	87.3	<b>89.2</b>	<b>89.2</b>
988335737	<b>67.2</b>	<b>67.2</b>	<b>67.2</b>	<b>67.2</b>
986174566	93.2	93.2	<b>93.3</b>	<b>93.3</b>
986884172	<b>94.4</b>	94.2	94.3	93.6
986938604	99.7	99.1	<b>99.8</b>	99.2
985935431	94.9	94.9	<b>95.0</b>	<b>95.0</b>
987014104	92.2	97.7	<b>99.4</b>	97.9
82119412	NaN	98.0	NaN	<b>98.3</b>
983602168	94.4	92.7	<b>94.5</b>	93.3
Overall Accuracy	94.70	94.20	<b>95.27</b>	94.87

importantly, MVP+LO shows as well the highest overall accuracy among both privacy-friendly and privacy-invasive models. The *NaN* in the results of MVB+LO and MVP+LO models indicates that the location of office is unknown, and thereafter, the post-processing task is not applicable. Globally, the privacy-invasive models show better performance than the privacy-friendly models, indicating that the private information does help improve the models. However, in practice, the private information is not always available. Therefore, a trade-off between model performance and user privacy should be considered in practice.

### Post-processing effects on various basic MTSC models

In this section, we report the detailed results of various model variants applied to new data (not seen before by the model) to show the effects of the post-processing techniques. First, we show the performance of the privacy-friendly models before and after adopting the post-processing layer (i.e., stop-mode and transportation-mode detection). Then, for privacy-invasive models, we briefly compare the effects between various location-correction techniques (i.e., LO and LH) on our multi-view learner after post-processing (i.e., MVP).

For privacy-friendly models, we show the results on four basic MTSC models (i.e., multi-view learner, two-step multi-view learner, MLSTM-FCN, and KNN-DTW) with or without pre-processing. Tables 3.13, 3.14, 3.15 and 3.16 show the metric comparison (i.e., *precision*, *recall*, *F1-Score*) of the four models, respectively. Globally, the multi-view models (i.e., MVB and MVP) show better performance than both MLSTM-FCN and the two-step multi-view models. Furthermore, the post-processing allows improving all the basic models, especially for the F1-score, in which we can observe a noticeable improvement. To have a more detailed understanding of the results, we show in Figures 3.13, 3.14, and 3.15 the related confusion matrices of the models where we report the percentage of true predictions in each class. From the results, we observe that the post-processing improved largely the recognition of the **outdoor** (e.g., *walk*, *bike*, *park*) and **transport** (e.g., *car*, *bus*, *metro*) micro-environments, whereas the performance on **indoor** micro-environments (e.g. *station*, *restaurant*, and *store*) recognition is only slightly improved, which is mainly due to the limited sample numbers in the testing set.

However, as shown in Table 3.15 and 3.16, MLSTM-FCN-based and KNN-DTW-based models show bad performances even after post-processing. For instance, in Table 3.15, the MLSTM-FCN-based models fail to detect most of the micro-environments, even they perform relatively better on detecting *home*, the performance is still much worse than multi-view-based models. Therefore, we draw a similar conclusion as mentioned in Section 3.3.6: the baseline models are not applicable on such complex scenarios where some dimensions are missing during data collection process; In other words, some sensors may be inoperative due to a technique issue for a long time, thus, some samples contain less dimensions than others. On this aspect, our multi-view-based models show a key advantage

Table 3.13: Performance of Multi-view Learner on Participants' data (before/after) post-processing

class	MVB			MVP		
	Precision	Recall	F1 Score	Precision	Recall	F1 Score
Walk	0.82	0.79	0.80	0.85	0.83	<b>0.84</b>
Bus	0.85	0.64	0.73	0.96	0.69	<b>0.79</b>
Office	0.86	0.85	0.85	0.92	0.90	<b>0.90</b>
Restaurant	0.42	0.60	<b>0.50</b>	0.44	0.60	<b>0.50</b>
Home	0.95	0.95	0.95	0.96	0.96	<b>0.96</b>
Bike	0.57	0.61	0.59	0.61	0.61	<b>0.61</b>
Car	0.51	0.18	0.27	0.78	0.25	<b>0.38</b>
Store	0.61	0.61	0.61	0.64	0.68	<b>0.64</b>
Metro	0.62	0.70	0.66	0.71	0.70	<b>0.71</b>
Station	0.16	0.17	0.16	0.25	0.16	<b>0.20</b>
Motorcycle	0.33	0.08	0.12	0.65	0.30	<b>0.41</b>
Running	0.30	0.61	0.40	0.38	0.61	<b>0.48</b>
Park	0.32	0.86	0.47	0.36	0.85	<b>0.50</b>

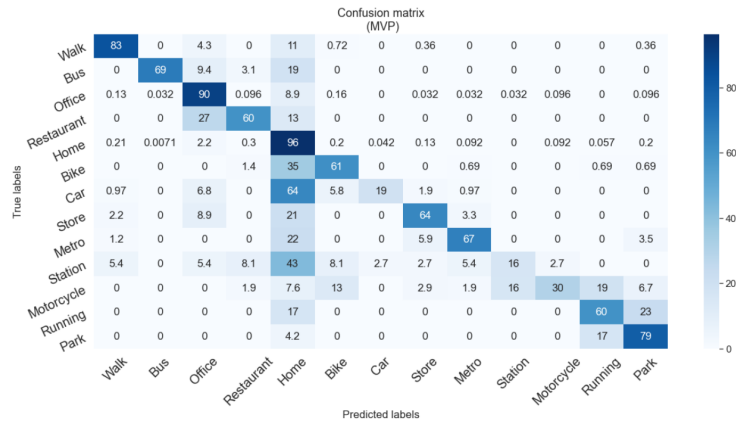


Figure 3.13: MVP confusion matrix

Table 3.14: Performance of Multi-view Learner (2 steps classification) on Participants' data (before/after) post-processing

class	MV-2steps			MV-2stepsP		
	Precision	Recall	F1 Score	Precision	Recall	F1 Score
Walk	0.75	0.79	0.77	0.78	0.83	<b>0.80</b>
Bus	0.73	0.64	0.68	0.81	0.69	<b>0.74</b>
Office	0.85	0.87	0.86	0.92	0.96	<b>0.94</b>
Restaurant	0.43	0.60	0.50	0.48	0.60	<b>0.54</b>
Home	0.95	0.95	0.95	0.97	0.97	<b>0.97</b>
Bike	0.43	0.38	0.41	0.52	0.41	<b>0.44</b>
Car	0.50	0.14	0.22	0.88	0.23	<b>0.36</b>
Store	0.45	0.62	0.53	0.60	0.68	<b>0.60</b>
Metro	0.57	0.50	<b>0.53</b>	0.76	0.45	<b>0.53</b>
Station	0.46	0.15	<b>0.23</b>	0.80	0.14	0.22
Motorcycle	0.27	0.04	0.07	0.76	0.28	<b>0.40</b>
Running	0.24	0.61	0.35	0.38	0.60	<b>0.46</b>
Park	0.41	0.93	<b>0.57</b>	0.38	0.96	0.54

Table 3.15: Performance of MLSTM-FCN on Participants' data (before/after) post-processing

class	MLSTMB			MLSTMP		
	Precision	Recall	F1 Score	Precision	Recall	F1 Score
Walk	0.13	0.02	0.03	0.13	0.09	0.01
Bus	0.0	0.0	0.0	0.0	0.0	0.0
Office	0.13	0.60	0.22	0.13	0.58	0.21
Restaurant	0.0	0.0	0.0	0.0	0.0	0.0
Home	0.73	0.27	0.40	0.74	0.28	0.41
Bike	0.0	0.0	0.0	0.67	0.06	0.11
Car	0.0	0.0	0.0	0.0	0.0	0.0
Store	0.0	0.0	0.0	0.0	0.0	0.0
Metro	0.0	0.0	0.0	1.0	0.08	0.16
Station	0.0	0.0	0.0	0.0	0.0	0.0
Motorcycle	0.0	0.0	0.0	0.55	0.33	0.42
Running	0.0	0.0	0.0	0.0	0.0	0.0
Park	0.0	0.0	0.0	0.0	0.0	0.0

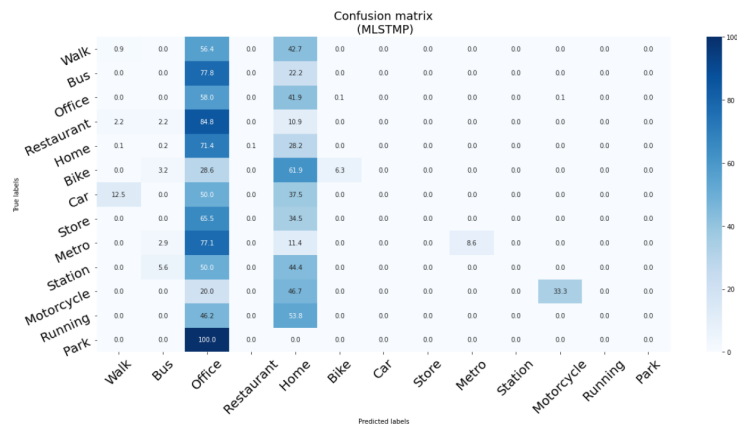


Figure 3.14: MLSTMP confusion matrix

Table 3.16: Performance of KNN-DTW on Participants' data (before/after) post-processing

class	KNN-DTWB			KNN-DTWP		
	Precision	Recall	F1 Score	Precision	Recall	F1 Score
Walk	0.13	0.52	0.20	0.26	0.62	0.36
Bus	0.06	0.55	0.11	0.1	0.83	0.17
Office	0.59	0.64	0.62	0.66	0.82	0.74
Restaurant	0.04	0.24	0.06	0.05	0.33	0.09
Home	0.92	0.74	0.82	0.97	0.87	0.92
Bike	0.22	0.33	0.26	0.55	0.40	0.46
Car	0.0	0.0	0.0	0.0	0.0	0.0
Store	0.11	0.41	0.18	0.0	0.0	0.0
Metro	0.09	0.40	0.16	0.37	0.55	0.44
Station	0.09	0.11	0.10	0.33	0.20	0.25
Motorcycle	0.05	0.08	0.06	0.44	1.0	0.62
Running	0.17	0.71	0.27	0.11	0.27	0.15
Park	0.04	0.50	0.08	0.33	0.50	0.40



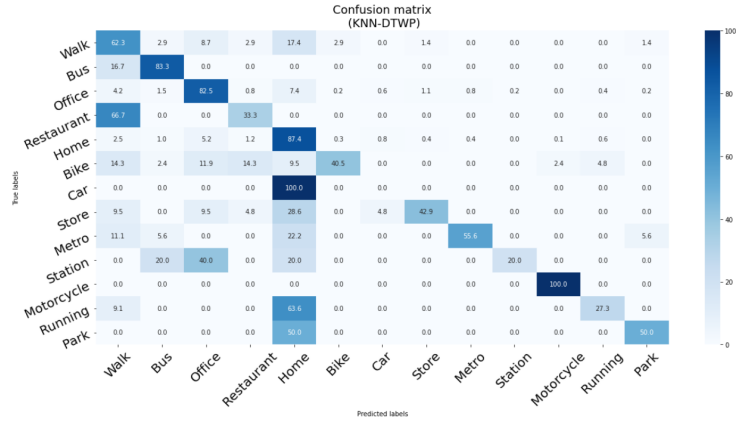


Figure 3.15: KNN-DTWP confusion matrix

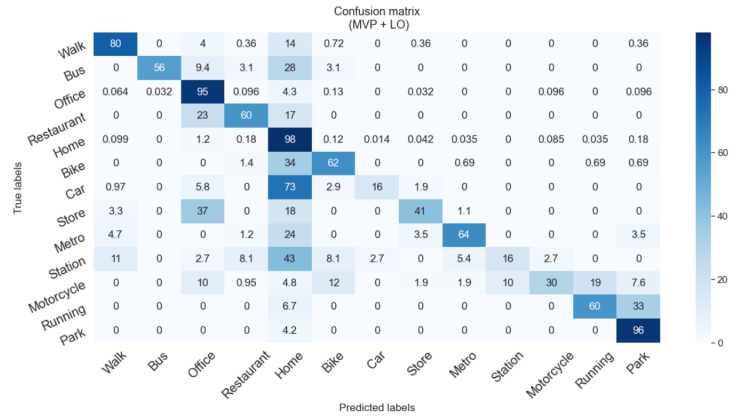


Figure 3.16: MVP + Office location Correction confusion matrix

compared to the baseline models, which can be explained by the fact that the meta-learner allows weighting the predictions of the first-level learners, thus eliminating the effects of the missed dimensions. To this end, considering as well the high computation cost of the MLSTM-FCN and KNN-DTW models, we judge that the the baseline approaches are not qualified as appropriate models for predicting the micro-environments.

For privacy-invasive modes, we show in Table 3.17 and Figure 3.16 the performance of the MVP model when adopting the location-correction techniques (i.e., LO and LH). Compared to the MVP model's performance reported in Table 3.13, we observe that adding the location corrections of home or office not only leads to better predictions on the target classes (i.e., home, office) but also improves the general model performance. Moreover, the location of office (LO) helps the MVP model achieve better predictions in most classes than the location of home (LH), which is coherent with our conclusion in Section 3.3.6 where the MVP+LO model shows the best accuracy for most campaign participants. However, even though

Table 3.17: Performance of Multi-view Learner with Location Correction and Post-processing on Participants' data

class	MVP + LH			MVP + LO		
	Precision	Recall	F1 Score	Precision	Recall	F1 Score
Walk	0.89	0.78	0.83	0.88	0.80	0.84
Bus	1.0	0.53	0.69	0.95	0.56	0.71
Office	0.94	0.93	0.93	0.92	0.95	0.94
Restaurant	0.51	0.60	0.55	0.49	0.60	0.54
Home	0.96	0.98	0.97	0.97	0.98	0.98
Bike	0.67	0.54	0.59	0.68	0.62	0.65
Car	0.84	0.15	0.26	0.84	0.15	0.26
Store	0.78	0.54	0.64	0.71	0.41	0.52
Metro	0.78	0.49	0.60	0.83	0.64	0.72
Station	0.42	0.14	0.20	0.35	0.16	0.22
Motorcycle	0.74	0.35	0.47	0.67	0.31	0.42
Running	0.24	0.30	0.27	0.41	0.60	0.49
Park	0.32	0.96	0.48	0.31	0.96	0.47

the location information allows to greatly improve the model's performance, the privacy stays as a crucial issue during both the data collection and data application process. In practice, a trade-off between the privacy and model performance should be considered.

In conclusion, MVP with location correction and MVP classifiers have comparable results. Although location corrections (i.e., LH and LO) can improve the model's performance, those location data are not always available due to privacy issues.

### Model Generalization

In practice, we should consider the model generalization on unseen data, which allows evaluating the model in more complex scenarios. We have used the multi-view model (which have been trained over RECORD campaign data) to classify data that have never been seen by it before. We opt for the VGP campaign data, which was collected during a different time period from RECORD, to prove the generalization ability of the proposed model. For VGP campaign, we don't have the ground truth for the data, so we have plotted the predictions versus the declared activities (which is not guaranteed to be accurate). Figure 3.17 shows the plot of declared versus predicted micro-environments. For this participant (i.e. participant 9999988), we trust his/her annotations, so we can notice that the model have performed well. While for figure 3.18, as we don't have the real ground truth, we can see that the model's predictions are more reliable than the annotations. For instance, the participant in the plot has declared three times staying outdoors in the

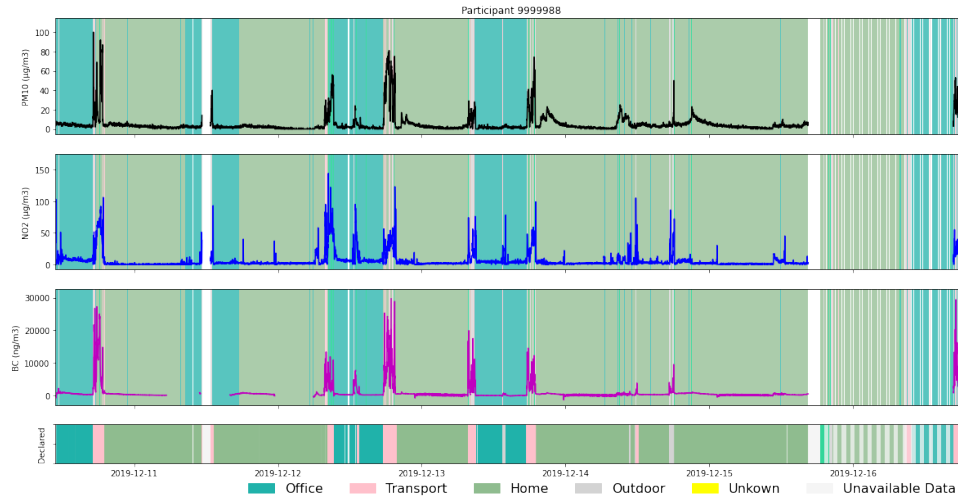


Figure 3.17: Predictions of VGP campaign for participant 9999988.

middle of the night (i.e. 24, 25 and 26th of October 2019), which is very unlikely to be true. Some participants may completely forget to annotate the change of micro-environment, so the declared annotations are indeed imperfect.

### 3.3.7 . Discussions & Perspectives

In this section, we discuss the perspectives for improving our multi-view learning model and the possibility for tackling the practical label issue in the context of Polluscope.

## Multi-view Learner

The multi-view learner adopted in this work is composed of the base learner (i.e., Random Forest) and the meta-learner (i.e., Random Forest), which has greatly improved the performance compared to the single kNN-DTW classifier. The objective of this work is not to propose the best classifier for MTS classification, but to provide an insight that the multi-view learner is capable of coordinating effectively the information from different variables and achieving more reliable performance than a single base learner. Moreover, the results of the grouping approach which is based on the multi-view approach confirms that there is a clear signature for each micro-environment, thus we can have an effective prediction with this approach. Moreover, multi-view approach offers the reusability of the first-level learners, and allows using different classifiers and combinations for the first-level learners. Multi-view model doesn't require a special hardware such as GPU for training Neural Networks (i.e. MLSTM-FCN). In addition, it doesn't require a long time for classification as other classifiers such as KNN-DTW do. Besides, using multi-view approach allows the prediction of micro-environments in the absence

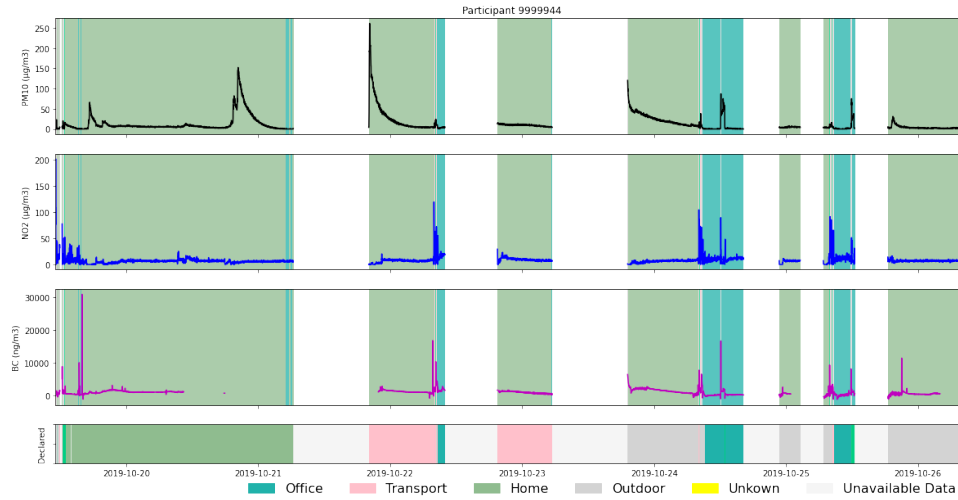


Figure 3.18: Predictions of VGP campaign for participant 9999944.

of some dimensions in the data. Another advantage of using multi-view approach is that its meta-learner is trained on out-of-fold predictions thus the model will not over fit.

Nevertheless, the kNN-DTW is considered as the baseline for MTS classification and is widely outpaced by the advanced approaches such as Shapelets [134, 149, 148] or the frequent patterns [94]. Essentially, the kNN-DTW captures the global feature based on the distance measure between the entire sequences, while the local features (e.g., the frequent patterns [94], the interval features [35], Shapelets [134], etc.) are more appropriate when a specific pattern characterizes a class. More specifically, a combination of features extracted from different domains may dramatically improve the performance of the base learner [83]. Therefore, one of the perspectives consists of the **optimization** of the base learner and the exploration of the **explainability** of the multi-view learner on both the feature interpretation and the variable importance for building the classifier. For this reason, we have removed the NO2 and BC dimensions to show their importance for some classes. Table 3.18 shows the precision, recall, and F1 score for MVB while removing some dimensions (NO2 and BC) compared to the MVB model containing all dimensions. The comparison shows that the F1-score of the MVB model for all classes is greater than that model without NO2 and BC. Except for *Running* class which is only one point difference. This comparison shows the importance and role of those dimensions (NO2 and BC) in context prediction. We have chosen to remove NO2 and BC because depending on figure 3.10, these 2 dimensions have the highest accuracy compared to other dimensions. The visual representation of Shapelets make them good candidates for such improvement.

Table 3.18: Performance of MVB without NO2 and BC VS. MVB

class	MVB without NO2 and BC			MVB		
	Precision	Recall	F1 Score	Precision	Recall	F1 Score
Walk	0.74	0.76	0.75	0.82	0.79	<b>0.80</b>
Bus	0.58	0.54	0.56	0.85	0.64	<b>0.73</b>
Office	0.78	0.74	0.76	0.86	0.85	<b>0.85</b>
Restaurant	0.43	0.60	<b>0.50</b>	0.42	0.60	<b>0.50</b>
Home	0.92	0.93	0.93	0.95	0.95	<b>0.95</b>
Bike	0.49	0.47	0.48	0.57	0.61	<b>0.59</b>
Car	0.34	0.15	0.20	0.51	0.18	<b>0.27</b>
Store	0.54	0.57	0.55	0.61	0.61	<b>0.61</b>
Metro	0.52	0.60	0.55	0.62	0.70	<b>0.66</b>
Station	0.10	0.12	0.11	0.16	0.17	<b>0.16</b>
Motorcycle	0.25	0.07	0.11	0.33	0.08	<b>0.12</b>
Running	0.32	0.61	<b>0.42</b>	0.30	0.61	0.40
Park	0.26	0.89	0.41	0.32	0.86	<b>0.47</b>

### Label Shortage Issue

The label shortage is a practical issue when building the learning model. In the context of Polluscope particularly, post-labelling for time series sensor data is much more costly than classic data (e.g., image, text, etc.) due to the low interpretability over the real-valued sequence. Therefore, the data need to be annotated during the data collection process. However, certain practical factors limit the availability of labels. For instance, the participants are not always conscious in annotating their micro-environment. Therefore, for certain time periods, no annotations were marked.

In order to give an insight about the consistency between the labeled and unlabeled data, and to see if the unlabeled data are valuable for improving the classifier’s performance in our context, we conduct a preliminary test on the Polluscope data with the newly proposed semi-supervised MTSC model TapNet [139].

TapNet [139] is a deep learning based approach designed for multivariate time series classification. By adopting the prototypical network [114], TapNet allows learning a low-dimensional embeddings for the input MTS where the unlabelled samples help adjust the class prototype (i.e., class centroid), which leads to a better classifier than using only the labelled samples. Table 3.19 shows the semi-supervised learning results on Polluscope data considering or not the *speed* variable. We evaluate the performance of TapNet under different supervision ratios in the training set. The results show that the unlabeled samples and the *speed* variable do improve the performance of the classifier. Besides, the accuracy didn’t drop a lot when eliminating the annotations in training set (from ratio=1 for fully labelled to 0.5, and even for 0.2 when only 20% data in labelled), indicating that the collected data within each class is not sparsely distributed. Thus learning under

weak supervision is reliable with the aid of the unlabeled samples.

Table 3.19: The accuracy results of TapNet on Polluscope data under different supervision ratios

Condition	Sup_ratio=1	Sup_ratio=0.5	Sup_ratio=0.2
Speed	0.746	0.725	0.717
No speed	0.713	0.703	0.695

Giving the promising results on the data distribution consistency, another avenue worth exploring is to consider and integrate a semi-supervised model into our multi-view learner. Various semi-supervised frameworks are applicable to our model, such as applying self-learning [127] to produce the pseudo labels on the multi-view learner, or adopting the label propagation and manifold regularization techniques [121] on the base learner.

### 3.3.8 . Summary

Activity recognition has gained the interest of many researchers nowadays, due to the widespread use of mobility sensors. Micro-environment recognition is essential in MCS projects such as Polluscope, in order to analyse the individual's exposure to air pollution and to relate it to her context. The major finding of our study is to show to some extent that the environmental observations can characterize the micro-environment. Moreover, the accuracy of the model is high enough to consider an automatic detection of the micro-environment without burdening the participants with self-reporting. By using the mobility feature as a time series, the accuracy improves slightly though the gain is moderate. Therefore, we can keep characterizing the micro-environment even in the absence of the speed dimension.

## 3.4 . Conclusion

In this work, we covered time series segmentation based on change point detection to demonstrate the change points in participants' contexts detected automatically by a multi-dimensional CPD model. The experiment conducted using real-world data, showed the effectiveness of our proposed approach for multidimensional time series segmentation, where not all dimensions may cause or detect the change.

Additionally, we promoted the idea of multi-view learning with stacking to detect the user context from environmental data collected from several sensors plus mobility. We employed different approaches and learners, and conducted a thorough experimental study, which shows the efficiency of the multi-view approach for time series classification, even some dimensions are missing. We have also compared the results with the MLSTM-FCN and kNN-DTW classifier which were considered as the baselines. During training phase, MLSTM-FCN, on the one hand,

showed promising results that could not be confirmed during the phase of applying the model on new data due to over-fitting. kNN-DTW, on the other hand, was not comparable, plus it is not suitable because of its time consumption. Furthermore, training on a previous data set was biased by the quality of the annotation. But this limitation was overcome by using a more reliable data that we did not have before.

Furthermore, we extend the proposed approach to include the detection of stay locations based on trajectory segmentation into stop and move segments. The move segments were labeled by the type of transportation mode. We combine time series plus trajectory data as a pre-processing and a post-processing layers to bring the best of them.

In addition, we present two optimisation methods which are either privacy friendly or privacy invasive. The later approach adds the private location of home and office in post-processing, whilst the first approach uses a priori rules. According to our experiments, we highly recommend using the privacy friendly models due to their ability to respect the private lives of the participants.

## 4 - Multidimensional Trajectory modeling

### Contents

---

<b>4.1</b>	<b>Introduction</b>	<b>102</b>
4.1.1	Motivation and Challenges	102
4.1.2	Problem Statement	104
4.1.3	Concepts of Semantic Trajectory Data Modeling	104
4.1.4	Contributions	106
<b>4.2</b>	<b>Requirements of Multidimensional Data modeling in MCS</b>	<b>107</b>
<b>4.3</b>	<b>Background</b>	<b>108</b>
<b>4.4</b>	<b>Multidimensional Data Model</b>	<b>110</b>
4.4.1	Overview	110
4.4.2	Spatial Discretization	111
4.4.3	Temporal Discretization	112
4.4.4	Spatial Indexing	113
4.4.5	Temporal Disaggregation	113
4.4.6	Spatial Disaggregation	114
4.4.7	MULTICS General Schema	115
4.4.8	Application Scenarios	116
<b>4.5</b>	<b>Implementation and Experimentation</b>	<b>117</b>
4.5.1	Experimental Design	117
4.5.2	Longitudinal Analysis	118
4.5.3	Spatial Analysis	118
4.5.4	Temporal Analysis	119
4.5.5	Temporal Disaggregation	121
4.5.6	Spatial Disaggregation	122
4.5.7	Computational Costs	123
<b>4.6</b>	<b>Conclusion and Perspectives</b>	<b>127</b>

---



## 4.1 . Introduction

In this chapter, we address the fourth contribution of this thesis, i.e. trajectory data management and warehousing. Resulting from trajectory data pre-processing and enrichment in Chapter 3, we have already achieved a sound trajectory data enriched with contextual information. To analyse these semantically enriched trajectory data and better understand the exposure to pollution per participant and for all participants combined, this chapter focuses on designing a multidimensional trajectory data model in the context of mobile crowd sensing called **MULTICS**, which allows trajectory data mining and exploration from different perspectives (i.e. temporal, spatial, longitudinal).

This chapter is organised as follows: The following section presents motivation for a multidimensional data model and the encountered challenges. Section 4.2 presents the main challenging characteristics for multidimensional rich trajectories modeling. Section 4.3 introduces MULTICS conceptual model. Section 4.4 introduces the proposed data model. Section 4.5 presents the implementation of MULTICS. Experiments are conducted on real environmental data from MCS campaigns. Finally, the last section summarizes our main contributions and draws our perspectives.

### 4.1.1 . Motivation and Challenges

As has been addressed in Chapter 1, combining spatial location with continuous measurements, and time activity diary result in rich trajectories. Figure 4.1 shows an example of a typical rich trajectory evolving along space and time. In addition to ambient air measurements such as temperature and air pollutants, this trajectory is enriched with contextual information such as participants' micro-environments (e.g. home, office, restaurant, etc.) and air pollution related events (e.g. smoking). Therefore, rich trajectory data does confirm the representation mentioned above, which can be described by a vector (user id, timestamp, latitude, longitude, measurements, semantic). The challenge then is to perform analytical processing of rich trajectory data. For instance, to answer a query like *Which category of participant is the most exposed to NO<sub>2</sub> and in which micro-environment?* requires aggregating measures (here NO<sub>2</sub>) at some space and time granularity while considering other categorical dimensions. Other queries may need not only trajectory data and contextual information, but also the ability to integrate external data and perform complex analysis. For instance, answering this query: *What is the difference between the collected AQ measurements and fixed stations AQ data for the same localization and at the same time?* This query involves the integration of AQ data from fixed stations and the alignment of the spatial and temporal granularities of the two data sources, in order to match and compare them. The combination and the aggregation of rich trajectories data calls for the concept of Trajectory Data Warehouse (**TDW**).

However, one particularity of MCS is its heterogeneous and imperfection prop-

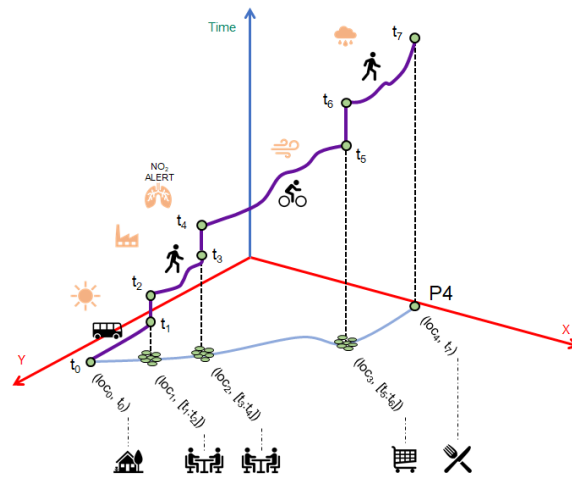


Figure 4.1: An example of rich trajectory collected in the context of MCS.

erties, designating that rich trajectory data is originated from different sensors that may exhibit some issues. In fact, some sensors may be offline and do not transfer any data for hours, which may lead to missing data problems. Furthermore, participants may not thoroughly annotate their micro-environments or may completely forget to fill in this information, resulting in trajectory data with low-confidence in self-reporting, or in worst cases, trajectory data with no semantic contextual information. With this missing data issue, analysing rich trajectory data and extracting the *complete* exposure story of participants is not straightforward. The complete story of a subject has been addressed in [122] to analyse the different aspects of the “mobility story” of a moving object. In our case, we intend to use this approach for extracting the *complete* information from rich trajectories, including interpolating missing values, inferring the context of the participants, and analysing the exposure story of the participants.

Information of the exposure story benefits participants in triple ways: (i) Discover periods and/or location with high pollution phenomena along the participant’s trajectory, so they can change their mobility habits - if possible. (ii) Have a view on their exposure over time and detect the micro-environment with the highest/lowest level of pollution. Users can then take actions to improve their AQ (e.g., open/close window). (iii) Provide participants with complete information on their exposure even if they did not thoroughly annotate their data or the data were not acquired. Hence providing information from such enriched trajectory data is a fundamental issue in real-world applications.

Consequently, the multidimensional feature of rich trajectories motivates a multidimensional analysis since it allows the exploration of these data from several perspectives (i.e., longitudinal, spatial and temporal perspectives) and multi-scale, which allows exploring the MCS data at different granularity levels.

### 4.1.2 . Problem Statement

Despite extensive research efforts on modeling Trajectory Data Warehouse (TDW) and OLAP systems [106, 86, 122], none of them are applicable on our MCS data enriched with measurements and semantics since it is non-trivial to adapt generic Data Warehouse (DW) for spatio-temporal trajectories.

One of the strengths of MCS is the usage of different sensors designed by different manufacturers. The used sensors may differ in their sampling frequencies, which could lead to measurements at irregular, potentially asynchronous time intervals, and missing values. In other words, we can not get a complete measured state due to various times of sensor data acquisitions. For instance, GPS tracks are collected every second, while AQ data are collected every minute. The user needs to recognize the periods and location with a high level of exposure. Combining GPS tracks and AQ time series will create a missing value issue. Plus, the time and spatial dimensions are not finite and discrete and cannot be used for aggregation, unless they are previously discretized according to a given granularity. This raises the question of *how to manage the diversity of the granularity of these data ? How to ensure the usage of grouping and aggregating of measurements whilst some dimensions such as time and space do not present finite and discrete domains?* Since the data captured by the sensors are provided with a given accuracy, for comparable measurements, aggregation or grouping these data may not be possible. For example, the coordinates of two participant trajectories walking together may not be numerically identical even though they are acquired at the same time. Thus, a query that requires the grouping of these similar trajectories cannot be computed. In addition to this, another question that arises itself is *how to associate a concept hierarchy to these two continuous dimensions?* And lastly, *how to handle the missing values in a multi-granular data warehouse?*

Furthermore, the semantic of events reporting the time activity (i.e. context) is also different from the sensor updates, because it is categorical and relates to large intervals or sporadic events while sensor updates are numerical and reported continuously. For instance, micro-environments depict participants' contexts for a period of time (possibly a large interval), whilst air pollution related events report temporary and sporadic activities for a brief period. Therefore, a natural question arises *how to model events reporting with respect to their semantics?*

Additionally, in the context of MCS, sensors continuously collect huge amounts of rich trajectory data. Thereby, we need an efficient implementation of the data model to handle the volume and the velocity of this large-scale data.

### 4.1.3 . Concepts of Semantic Trajectory Data Modeling

Following the huge generation of spatio-temporal data, it became commonly known that non-spatial data warehouses are not sufficient to fully exploit the spatial dimension of geo-located data [106] [70]. A re-thinking of the traditional solutions was needed. In this section, we provide a summary of the main research on rich

Solutions	Semantic Trajectories	Spatial perspective	Temporal perspective	Missing Data Solution
Spatial OLAP [106]	No	Yes	No	No
Iftikhar and Pederson [66]	No	No	Yes	Yes
Sequential OLAP [86]	Yes	No	Yes	No
Interval OLAP [73]	Yes	No	Yes	No
OGC [96]	No	Yes	No	No
STOLAP [124]	No	Yes	Yes	No
Mob-Warehouse [122]	Yes	Yes	No	No
Leonardi <i>et al.</i> [80]	No	Yes	Yes	No
Mobility DW [120]	Yes	Yes	No	No
MULTICS	Yes	Yes	Yes	Yes

Table 4.1: Existing work on mobility analysis in DW and OLAP systems.

trajectories modeling. Depending on whether the model requires spatio-temporal data enriched with measurements and events, we discuss their compliance to MCS context. Table 4.1 presents some existing works related to mobility DW and OLAP systems.

Based on our deep study and analysis of the literature related to moving object management (see Section 2.6), we have identified two families of databases research that have been successfully tackled in the literature; i.e. moving objects databases and trajectory data warehousing. More specifically, we have considered trajectory-based models and enriched trajectory models from trajectory data warehouses' family, which are the case studies of this thesis work.

A thorough comparative analysis between state-of-the-art proposals reflects that each work involves various facets of the trajectory data. Notably, the Mob-warehouse [122] proposal is an interesting study introduced as a key work to investigate all the aspect of the "mobility story". Whilst the authors interrogate the data from different perspectives and answer questions in the form of where, when, who, whom, what why and How, one important facet of the data is still missing and it concerns the continuous measurement. Indeed, the continuous measurements are as important as any other dimensions in MCS data. Other initiatives such as Mobility DW [120] provide a data warehouse for moving objects. It is true that the semantic information about the moving object is not excluded in the proposal, However, continuous measurements are not included, and it remains an important facet in MCS data mining and analysis. Sequential OLAP [86] and Interval OLAP also [73] deal with semantically enriched trajectory data. However, they do not introduce the spatial dimension in the data analysis and exploration.

On the other hand, trajectory-based models encompass studies such as Spatial OLAP [106], the work of Iftikhar and Pederson [66], OGC [96], STOLAP [124] and the work of Leonardi *et al.* [80], which do not consider the semantic facet of the trajectory data. While these approaches are key guidance for trajectory data modelling, they present some limitations concerning the spatial perspective or the temporal one as described in Table 4.1.

Motivated by these limitations, in this work, we propose a multidimensional data model for rich trajectories data modeling and analysis which takes into consideration the particularity of MCS data, as well as the specific semantic and nature

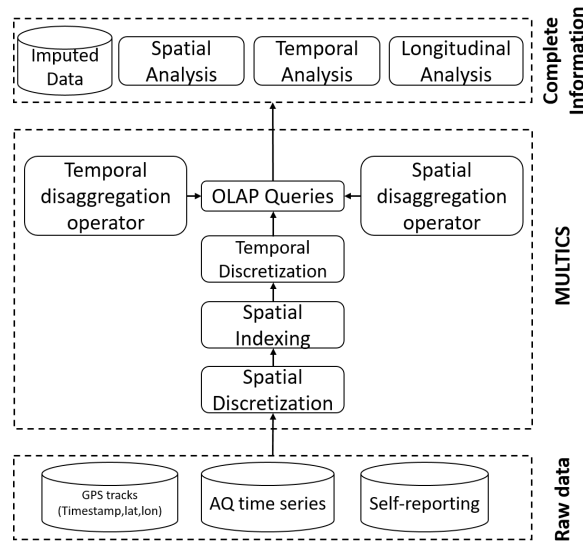


Figure 4.2: The conceptual architecture of the proposed solution for rich trajectories.

of self-reporting (i.e. micro-environments and events declared by participants) to capture every facet of the data.

#### 4.1.4 . Contributions

To the best of our knowledge, this is the first contribution that adopts a multi-dimensional model to meet the requirements of the complete exposure and mobility stories. We investigate the capability of online analytical processing (OLAP) systems in handling MCS data and identify their limitations. Precisely, we adopt the discretization method of the spatial and temporal dimensions so they can be used in an OLAP system. Based on machine learning, we propose new operators for spatial and temporal disaggregation to deal with missing values in a multi-granular DW. Figure 4.2 illustrates the conceptual architecture of the proposed solution. Specifically, the main contributions of this work are :

- We propose *MULTICS*, a **MULTI**-dimensional model for **Crowd Sensing** to deal with the multidimensional feature of MCS data and capture every facet.
- We adopt the discretization method of the spatial and temporal dimensions by setting a minimal granularity.
- Time is also discretized (in our application, the granularity of the minute was chosen).
- For the spatial dimension, the study area is divided into pixels of predefined

## 4.2. REQUIREMENTS OF MULTIDIMENSIONAL DATA MODELING IN MCS107

size<sup>1</sup>. Trajectories are converted to pixel references combined with time units (i.e., members of spatial and time dimensions), which adapts to the exploration at different scales in a hierarchical manner. We also adopt a spatial index to speed-up the spatial queries.

- Besides the embedded operators, we introduce two operators of spatial and temporal disaggregation based on machine learning to handle the problem of missing values by producing finer resolution data from coarse data.
- Along with the collected data, the proposed model has the ability to store external facts and/or dimensions, as well as data derived by, e.g., a machine learning process.
- Finally, extensive experiments on real-world data collected by participants demonstrate the usefulness of our proposed model in solving real applications scenarios.

### 4.2 . Requirements of Multidimensional Data modeling in MCS

This section introduces the main characteristics of rich trajectories data collected in MCS, which are subject to many limitations. Indeed, data measured by mobile sensors can be represented by multivariate time series which are characterised by the presence of a spatial dimension forming trajectories. Equivalently, we can use these data as spatio-temporal trajectories enriched by additional measurements throughout the collection period. Such type of data exhibits a number of challenging characteristics.

**Spatial and Temporal Autocorrelation.** From the modeling view, a distinctive aspect of such data series is the spatial autocorrelation [89]. The same holds for consecutive observations, as the variation of physical phenomena is usually smooth. This means that collected data are not independent, and so, the spatial and the temporal dimensions should be organized and indexed accordingly.

**Multi-Granularity.** Another fundamental characteristic of mobile sensor data is the diversity of their granularity, both under the temporal and spatial dimensions. The temporal domain is typically represented at different time granularities. The spatial entity can be represented at different scales within a hierarchy of regions or cells. Combining multiple datasets with several granularities or changing the granularity of a dataset are important analysis tasks that we intend to deal with. Thus, we need to define a framework that takes into account spatial and temporal granularities, and allows the shifting from one granularity to another. The passage from a finer resolution to a higher resolution is motivated by temporal / spatial

---

<sup>1</sup>The chosen division follows the same division as the Air Quality Monitoring Association in the Paris region AirParif (<https://www.airparif.asso.fr/>). It is set to 12.5x12.5 m<sup>2</sup> in Paris, 25x25m<sup>2</sup> in the inner suburbs and 50x50m<sup>2</sup> elsewhere in the rest of the region.

aggregation. While spatial / temporal disaggregation is advocated when the lower granularity data is missing.

**Data Volume.** Huge amounts of data are being collected continuously in MCS campaigns (as many as the number of equipped holders) in different geographical areas. Big data processing techniques are necessary to allow an efficient interactive data analysis.

### 4.3 . Background

In this section, we define the main concepts for multidimensional data models.

The notion of granularities has been deeply studied in the literature, Bettini et al [13, 14, 15] define the temporal granularity as a partition of the time domain.

**Definition 4.3.1.** (Temporal Granularity). Formally, a temporal granularity  $g_T$  is a function from an ordered set  $I_T$  to the power set of the temporal domain  $T$  such that:

$$\forall i, j, k \in I_T, (i < k < j \wedge g_T(i) \neq \emptyset \wedge g_T(j) \neq \emptyset \implies g_T(k) \neq \emptyset)$$

$$\forall i, j \in I_T, (i < j \implies \forall x \in g_T(i) \forall y \in g_T(j) x < y)$$

Typical examples of temporal granularities are days, weeks, months.  $g_T(i)$  are called temporal granules of the granularity  $g_T$ . The first condition states that the subset of the set that maps to non-empty subsets of the time domain is contiguous. The second condition states that granules do not overlap and that their order is the same as their time domain order. Besides, Camossi et al. [23] define the spatial granularity as a mapping from an index set to subsets of the spatial domain (i.e. a set of 2-dimensional points)

**Definition 4.3.2.** (Spatial Granularity) Formally, a spatial granularity  $g_S$  is a function from an ordered set  $I_S$  to the power set of the spatial domain  $S$  such that:

$$\forall i, j \in I_S,$$

$$(i \neq j \wedge g_S(i) \neq \emptyset \wedge g_S(j) \neq \emptyset \implies intersects(g_S(i), g_S(j)) \neq \emptyset)$$

Typical examples of spatial granularities are pixels of different sizes, or a spatial hierarchy such as administrative subdivisions of a country.  $g_S(i)$  are called spatial granules of the granularity  $g_S$ .

**Definition 4.3.3.** (Time Series). We define a time series as an infinite sequence of values where a value is a couple  $(t, v)$  where  $t \in T$  is a timestamp (at a given granularity) from a time domain  $T$  with discrete time units in increasing order and  $v$  is a vector  $(v_1, \dots, v_n)$  where each value is a measurement or scalar value,  $v$  is an n-uplet of a fixed size.

**Definition 4.3.4.** (Spatio-Temporal Data Series). We define a Spatio-Temporal Data Series (STDS) as a time series where the location (e.g., latitude and longitude) belongs to the vector  $v$ .

**Definition 4.3.5.** (Hierarchy) A hierarchy is a set of binary relationships between the members of dimension. It is a systematic way of organizing the dimension into a logical tree structure, which defines parent-child aggregation relationships. The aggregate member, called parent, corresponds to the consolidation of another member called child.

Typical example of a simple hierarchy in a multidimensional Data Warehouse (DW) is the date (day, month, quarter, year).

**Definition 4.3.6.** (Dimension) Dimension ( $D_i$ ) is an object that contains attributes that belong to the same category in the user's perception of data.

**Example:** Minute, Hour, Day, Month and Year may make up a Time dimension.

**Definition 4.3.7.** (Fact) A fact is represented by a set of values which are related to a set of dimensions. The values of a Fact are usually measurements that can be numeric or alphanumeric.

**Example:** measurement in Figure 4.3 constitutes a fact table containing five dimensions (i.e. `user_id`, `campaign_id`, `time_id`, `location_id` and `measurement_value_id`) and relating to the measure `measurement_value` which is a numeric value.

**Definition 4.3.8.** (Temporal Disaggregation) Temporal disaggregation, also known as Temporal Distribution, is the process of converting a low granularity time series (e.g. annual time series) to a higher granularity time series (e.g. monthly time series). Denoting  $m$  the disaggregation function and  $S$  (optional) one or more external time series used to perform the disaggregation, temporal disaggregation is defined by the following expression:

$$TD\text{Agg}_{g_T, m, [T\text{Proj}_f(S)]}(R) = \{(v, s) \mid s \in G_T(R)\}$$

The set  $G_T$  (e.g.,  $G_T = \text{March, April, June, July}$ ) ranges over the granules of a granularity  $g_T$  (e.g., months).

**Example:** For each air pollutant measure that is initially sampled every hour, estimate its value every minute. This might be based on mathematical criteria or time series models such as ARIMA, or on other consistent temporal data used as a proxy. For instance, if we know either the sum, the average, the first or the last value of Particulate Matter (for which the sampling frequency is lower), we can use a consistent time series such as NO<sub>2</sub> (which is sampled every minute)



to disaggregate or interpolate Particulate Matter from low frequency to higher frequency time series with Chow-Lin-Maxlog method (See section 4.5).

$$TDAgg_{PM10,Chow-Lin-Maxlog[TProj(NO_2)]}(PM10)$$

**Definition 4.3.9.** (Spatial Disaggregation) Spatial disaggregation refers to the process of producing high resolution estimates of data distribution (e.g. 25m square) from coarse geospatial data (e.g. 1km square). The generation of fine gridded data can be done based on techniques that are often in conjunction with ancillary data, or statistical modeling methods. Denoting  $m$  the spatial disaggregation method and  $S$  the external geospatial data (optional), spatial disaggregation is defined as :

$$SDAgg_{(g_S, m, S)}(R) = \{(v, s) \mid s \in G_S(R)\}$$

$G_S$  is the spatial counterpart of  $G_T$ .

**Example:** Disaggregating the values of NO<sub>2</sub> from a coarse resolution (e.g. 300  $m^2$ ) to a finer resolution (e.g., 50  $m^2$  using the distribution of roads for example as ancillary data with a *machine learning algorithm* such as *Random Forest* (RF) (See section 4.5).

$$SDAgg_{(g_S, RF, Roads)}(NO_2)$$

## 4.4 . Multidimensional Data Model

As discussed in Section 4.1, the key insight of this work is to adopt a multidimensional model to mine rich trajectory data and extract users' complete story of exposure to pollution so that they possess the ability to take appropriate actions. In this section, we define the proper steps taken to achieve the ultimate goal. First, we introduce an overview of MCS data used for modeling rich trajectories. Afterwards, we *discretize* the spatial and temporal dimensions along with introducing two operators: *spatial and temporal disaggregation* (cf. Figure 4.2). Thereafter, we demonstrate the proposed model for rich trajectories, MULTICS, including the overall framework and the details of each component.

### 4.4.1 . Overview

Data collected in the context of MCS combines geolocation with observations and measurements over time, resulting in "rich trajectories". As a running application example, we consider a database obtained from the Polluscope project. A cohort of volunteers have been equipped with individual sensors collecting several pollution measurements along with GPS data. In Polluscope, three data collection campaigns have been conducted. Each campaign is characterised by a start date, an end date and a person in charge (i.e. responsible). Each campaign was

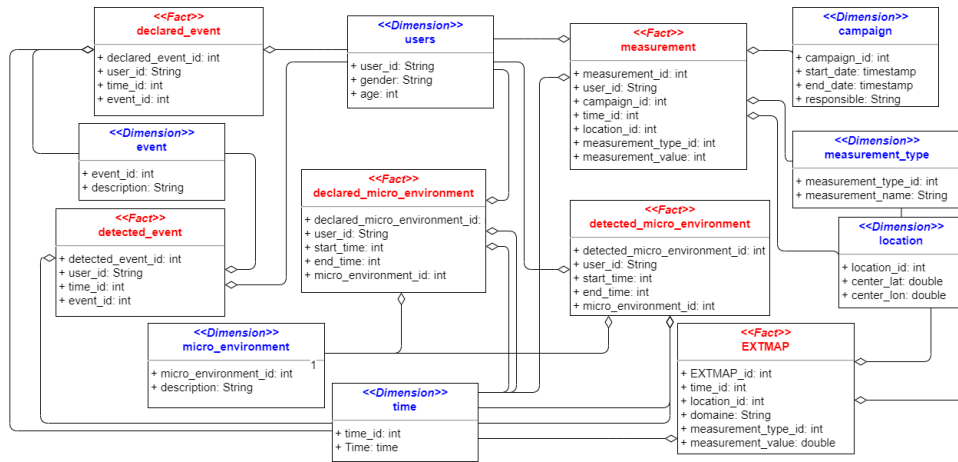


Figure 4.3: MULTICS conceptual Schema.

spread over 12 weeks with a collection generally carried out every other week (in order to check and re-qualify the sensors). 103 volunteers participated in the campaigns. We remind that these participants are equipped with a kit which contains air pollution sensors and tablets empowered with GPS chipsets. The sensors collect, every one minute, time annotated measurements of Particulate Matters (PM1.0, PM10, PM2.5), Nitrogen dioxide NO<sub>2</sub>, Black Carbon (BC), temperature and relative humidity. The tablet was used to geolocate the participants (GPS tracks are collected between 1 to 30 seconds) and to fill in their time activity via a mobile app developed for this purpose. Activities last a certain time and represent micro-environments which can be indoor environments (e.g. home, office, restaurant, etc.), outdoor environments (e.g., park, street, etc.) or even transportation modes (e.g., car, bus, etc.). In addition, the participant fills in the events related to air pollution, designating temporary actions over a short period (e.g., opening a window, cooking, smoking, lighting the fireplace, etc.). For more information on data collection's protocol, refer to Appendix A

It becomes obvious that the collected data show properties of auto-correlation and multi-granularity. The proposed solution should maintain the locality of the spatially close data, and to take into account the diversity of granularities. We introduce *spatial discretization and indexing* in order to keep spatially close data together and guarantee a hierarchical spatial representation.

#### 4.4.2 . Spatial Discretization

In a multidimensional model and OLAP systems, dimensions have finite and generally known values in advance, so as to be used for grouping and aggregating the measures reported in the fact table. However, spatial and temporal data represent a continuous domain. They can not be used for aggregation as they are represented. Subsequently and in order to allow the representation of the spatial dimension, we transform the reported positions (i.e. latitude and longitude) into

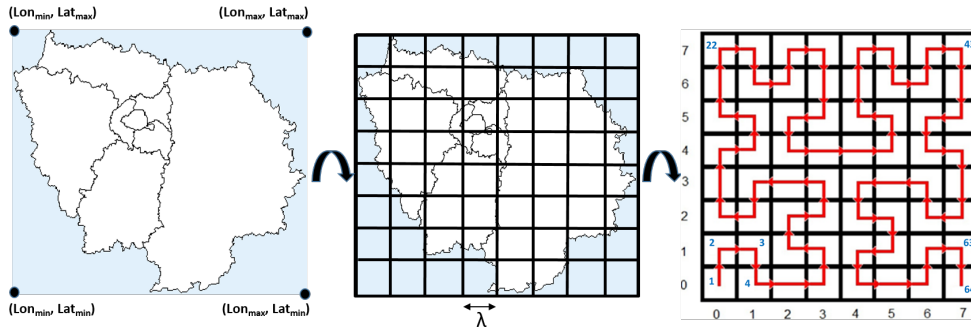


Figure 4.4: Spatial dimension representation

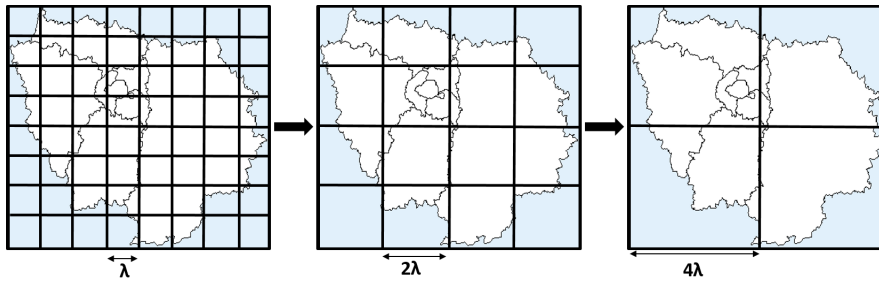


Figure 4.5: Spatial hierarchy representation

discrete values referencing a pixel of a rectangular grid with a spatial resolution (here of 50 m). The center subplot of Figure 4.4 illustrates the partition of the Paris region into a rectangular grid along the longitude and latitude dimensions.

Assuming the minimum latitude and longitude of the region are respectively  $Lat_{min}$  and  $Lon_{min}$ , and the maximum values are  $Lat_{max}$  and  $Lon_{max}$ . We split the region into  $cw * ch$  cells along the 2D axes with a grid side length of  $\lambda$  (here  $\lambda = 50m$ ), where  $cw$  and  $ch$  are the number of vertical and horizontal splits of the grid. The finer granularity of the spatial dimension is a cell of 50m (the choice of this value is discussed in Section 4.1.4). The spatial hierarchy can then be represented by grouping cells with a grid side length of  $k * \lambda$ , where  $2^k$  is the number of cells in the grouping (cf Figure 4.5).

#### 4.4.3 . Temporal Discretization

Likewise, the temporal data are brought back to a minimum threshold. A careful choice of the lowest level of granularity in time is needed in order to provide a good trade-off between precision and storage costs. In fact, while AQ measurements are acquired every one minute, GPS data are collected every 1 to 30 seconds. However, participants do not change their location very frequently within one minute inside a cell of 50 m spatial resolution. Hence, the temporal minimum threshold of one minute was chosen here. In this way, spatial and temporal di-

mensions can be supported by OLAP systems, unlike the original representation of infinite space and time.

#### 4.4.4 . Spatial Indexing

There remains the question of maintaining locality in the organization of the spatial dimension. For this, we adopt the spatial indexing using 2D Hilbert Space-Filling Curves (SFC), which provides a grouping feature per proximity [93]. In other words, neighboring cells are likely to be assigned to a close Hilbert index. Moreover, Hilbert SFC shows a fractal property that eases the exploration at different levels of spatial hierarchy, leaving the way for the "Roll-up" and "Drill-down" within the spatial dimension like zooming in images. By dividing the Hilbert index, we systematically move to a higher level of hierarchy with a grouping of  $2^{2n}$  cells.

Figure 4.4 shows our spatial data indexation and representation using Hilbert space-filling curves. The spatial extent is defined to cover the study area (i.e. Paris region). It is worth noting that we only maintain cells corresponding to the locations with GPS data, which is much more compact than solutions like geo-cubes [96].

The adopted rasterization approach is gainful in different ways. It allows to derive the stay areas (often indoors) of participants. For example, we can discover the places where a participant spends time the most based on cell densities, which can be calculated by counting the number of measurements associated with the cell (the pixel) per participant in the period. This approach also makes it possible to detect spatial outliers and spatial noise. Furthermore, it provides an equal-area pixelization that makes it easy to share the spatial dimension with external sources provided in raster formats (tiff, geoTIFF, NetCDF,...) such as Air Quality Maps<sup>2</sup>.

#### 4.4.5 . Temporal Disaggregation

One of the main contributions of our MULTICS model is the introduction of new operators, namely, spatial and temporal disaggregation, to derive finer grained data from coarse data.

Temporal disaggregation methods aim at deriving the data from low frequency time series to higher frequency time series. These methods can be categorised as models that:

- do not rely on any indicator series. These models purely rely on mathematical criteria or time series models such as ARIMA to obtain data at a higher level.
- use one or more indicators series observed at higher frequency as proxy to derive the finer level time series.

The former approach deals with the case where the only available data are the aggregated time series. It includes mainly mathematical methods proposed by [20]

---

<sup>2</sup><https://www.breezometer.com/>

and [68] and more theoretically founded model-based methods [128] relying on the ARIMA representation of the series to be disaggregated.

On the other hand, when one or more logically correlated high frequency indicators are available, ideal approach for temporal disaggregation are those belonging to the family of regression models, among which the Chow-Lin method proposed by [33], further developed by [21], [50] and [84]. The regression model of Chow-Lin estimates the coefficients at the low frequency level, then uses them to estimate the target time series at a higher frequency using as input the original high frequency indicators.

In this chapter, we offer an illustrative example of temporal disaggregation using R package “tempdisagg” [112] to perform the optimal procedures of Chow-Lin-Maxlog [33]. This example illustrates how we can descend from a low granularity time series to a higher granularity time series by applying one of the temporal disaggregation methods with the help of ancillary time series data. The temporal disaggregation allows then to unify data on the same level of granularity as ancillary data, and consequently solve the multi-granularity problem. (See Section 4.5).

#### 4.4.6 . Spatial Disaggregation

Spatial disaggregation, on the other hand, refers to the process of converting low resolution spatial data to higher resolution data. The most basic approach for spatial disaggregation is mass-preserving areal weighting [55] whereby we assume a homogeneous distribution of data. The process is based on a discretized grid, where each cell in the grid is assigned a value based on the proportion of the source zone (i.e. the polygon over which data is aggregated) contained in each cell (e.g. disaggregating the count of population based on the proportion of polygon that overlaps with the cell). However, mass-preserving areal weighting is based on the assumption that the observed phenomena is evenly distributed on the polygon which is often incorrect. Pycnophylactic interpolation [117] is an extension of mass-preserving areal weighting, which starts by applying mass-preserving areal weighting on the grid. Then, a smoothing function is applied which replaces the cell values with the average of their four nearest neighbours. To avoid the incorrect assumption that the data is evenly distributed, mask area weighting schemes [39] are based on creating a binary dasymetric mapping within the target zone where the source data should be allocated. Source units are divided into binary sub-regions (i.e., populated and unpopulated) and the source information is then allocated only to the populated areas. Dasymetric disaggregation [39] is an improvement of mask areal weighting that uses ancillary spatial data to augment the interpolation process.

In recent research, methods exploring machine learning techniques for combining pycnophylactic interpolation and dasymetric weighting were proposed. Monteiro et al. [92] present a hybrid disaggregation procedure to historical data (e.g., estimate population in one census year within the units of another year). Stevens et al. [115] combine widely available, remotely-sensed and geospatial data (also

referred to as covariates), such as presence of hospitals, road networks, land cover, etc., that contribute to the modelled dasymetric weights to disaggregate census counts at a country level. Each covariate is projected on a grid pixel of 100 m spatial resolution, and then aggregated by census units or villages. Their key contribution is to extract a training set at villages level to learn a Random Forest (RF) regression model. The RF model is then able to predict the population density at a finer level (100m spatial resolution). An illustrative example of spatial disaggregation is discussed in Section 4.5 in which we combine different geospatial data such as road networks and the presence of parks to disaggregate NO2 values.

#### 4.4.7 . MULTICS General Schema

With the transformation of the spatial dimension, the use of OLAP functionalities is obviously possible via our presentation. In this subsection, the schema of MULTICS is described in detail. It adopts a snowflakes multi-dimensional model.

The multi-dimensional model in MULTICS is generic and can be used for many MCS applications. We use the context of Polluscope in this subsection for illustration purpose.

The general schema is presented in Figure 4.3, which illustrates how we define the dimensions and fact tables. It contains six fact tables. The first table `measurement` stores the sensors' readings. The fact table `measurement` relates air pollutants measurements values depicted by the attribute `measurement_value`, to five dimensions:

1. `users` assigns participants their demographic data.
2. `campaign` assigns to each campaign a specific `campaign_id` and gives information about the `start_date`, the `end_date` and the person in charge, i.e. the responsible.
3. `location` is the spatial dimensions which gives information about the Hilbert SFC indices assigned to the grid cells.
4. `time` is the time dimension.
5. `measurement_type` depicts a dictionary of the type of measurements (e.g., PM2.5, NO2). This could be any observation or measurement, as for instance, noise, ozone, pollen, etc. The schema is therefore generic and applicable to any MCS application context.

Thereafter, MULTICS defines five additional fact tables: `declared_micro_environment_record`, `detected_micro_environment_record`, `declared_event_record`, `detected_event_record` and `EXTMAP`. The first table links the participants information to their declared self-reporting. It describes the micro-environment with a start and end time of presence. The dimension `micro_environment` contains descriptions about indoor and outdoor micro-environments as well as the

exhaustive list of transportation means. Identically, the fact table `declared_event_record` is similar to the `declared_micro_environment_record`, except the events are characterized by a temporary timestamp because they are brief. The dimension `event` depicts the information about the exhaustive dictionary of air pollution related events.

However, not all the participants thoroughly annotate their micro-environment. Therefore, there is a great interest in automatically detecting the context of the participants without burdening them, which have been addressed in Section 3.3.

We emphasise that MULTICS tackles the particularity of rich trajectories collected in the context of MCS as location-based data series. But, in addition to this, MULTICS can integrate external information as dimensions or fact tables. We extend the traditional data warehouse to support external geographic and temporal information from other sources. For instance, we can enrich the data warehouse with external geographic sources such as cartographic layers (e.g. roads, city boundaries) and Points of Interest (Pols). External temporal sources can be, for instance, temporary events related to the observed phenomena, such as a fire that emits pollutants, or a confinement leading to a significant drop in traffic which is a source of pollution. `EXTMAP` fact table represents spatio-temporal external sources, such as air quality maps that can be compared with the collected MCS data.

#### 4.4.8 . Application Scenarios

The multidimensional model thus proposed allows analysing data at different scales and hierarchies. Besides, it enables data to be viewed and modelled in different views, more generally from different facets of dimensions, and more precisely at different locations and periods of time. We emphasise its usefulness in analysing and exploring rich annotated trajectories in the context of MCS by introducing some use cases:

- **Longitudinal Analysis** which refers to the analysis and assessment of individual exposure over time. It allows to follow the evolution of individual exposure to pollution while detecting periods of high and low levels of pollution. Data can be aggregated over time periods, such as rush hours, weekend, weekdays, etc. The analysis can also be broken down into periods spent by micro-environment, which is valuable for understanding the exposure contexts and the difference between them.
- **Spatial Analysis** consists of detecting locations with high level pollution phenomena. It allows to emphasize pollution phenomena in different locations. For one participant, the spatial analysis permits to follow the level of pollution throughout the participant's trajectory, and therefore discover locations with high levels of pollution. Likewise, for all participants combined, we can identify on the map the locations with high level pollution phenomena. Spatial analysis can be generalized to types of micro-environments

as reported by participants, which opens the way to the traceability of the micro-environment with the highest and lowest exposure for each participant or for all participants combined.

- **Temporal Analysis** which refers to the analysis of measurements over time. In addition to the aforementioned longitudinal analysis which focuses on the individual dimension, we are interested in other temporal analysis which combine data from several participants. One example is to analyze the set of measurements reported for different time periods (e.g., peak hours, day of the week, weekend, month, season, year). Another is to focus on a specific micro-environment or area to assess the impact of certain policies on the level of pollution over time.
- **Temporal Disaggregation**, also known as Temporal Distribution, is the process of converting a low frequency time series (e.g. annual time series) to a higher frequency time series (e.g. monthly time series). This might be based on mathematical criteria or time series models such as ARIMA, or on other consistent temporal data used as a proxy. For instance, if we know either the sum, the average, the first or the last value of Particulate Matter (for which the sampling frequency is lower), we can use a consistent time series such as  $NO_2$  (which is sampled every minute) to disaggregate or interpolate *Particulate Matters* from low frequency to higher frequency time series.
- **Spatial Disaggregation** refers to the process of producing high resolution estimates of data distribution (e.g.  $25m^2$ ) from coarse geospatial data (e.g.  $1km^2$ ). The generation of fine gridded data can be done by utilising techniques that are often in conjunction with ancillary data, or statistical modeling methods. For instance, disaggregating  $NO_2$  values from a coarse resolution, such as  $200m^2$ , to a finer resolution, such as  $50m^2$ , using the distribution of buildings and roads for instance as ancillary data in a machine learning model.

## 4.5 . Implementation and Experimentation

### 4.5.1 . Experimental Design

MULTICS is implemented under *Spark*<sup>3</sup> 3.1.2, *Hadoop*<sup>4</sup> 3.2.2 and *Python* 3.9.2. We take advantage of *Spark SQL* OLAP-Like querying capabilities. *Spark SQL* employs Hadoop HDFS for data distribution and answer to simple queries, plus it provides optimized OLAP operators.

---

<sup>3</sup><https://spark.apache.org>

<sup>4</sup><https://hadoop.apache.org>



As mentioned above, more than 103 volunteers have participated in the data collection phase which lasts one week for each participant. GPS data is collected at a frequency of 1 to 30 seconds, whilst pollutants' measurements are collected every minute, thus, resulting in approximately 10 millions rows of time annotated measurements, plus few annotations of data by the type of micro-environment and pollution related events.

The spatial dimension is defined by the pixels (finite set and easiest to compare) rather than the exact position and is organized according to the Hilbert index order. This spatial organisation is gainful for dimensional analysis. Indeed, by referring to longitudinal analysis, one individual can discover the pixels with high pollution and the time spent within, and thus generate heat maps of their exposure. Likewise, this comparison can be performed for different participants combined at the same location (at the fine pixel level or at any level of the spatial hierarchy), in order to identify the participants with the highest exposure. As we can see, there are many different facets for exploring the data. All the possible combinations need to be reachable. In the next section, we introduce and discuss some possible combinations of the dimensions, i.e. location, time, pollutant types, etc.

#### 4.5.2 . Longitudinal Analysis

Longitudinal analysis consists of analysing participant exposure over time. It captures the individual exposure view. In our context, we intend to compare participants individual exposure to a fictitious medium exposure profile. That says, we need to calculate different aggregates of pollutants per participant and for all participants combined to constitute the fictitious medium exposure profile.

We take advantage of the ROLLUP operator to navigate through the dimension hierarchy and explore all possible facets. This operator is used to respond to queries such as the query in Example B.1.1 (see Appendix B).

The analysis can go beyond and illustrate the perspective of the individual analysis along their daily activities (i.e., for each time interval that the individual has spent in each micro-environment). Figure 4.6 depicts the evolution of the collected pollutants throughout the whole period of the campaign as well as the correlation between the concentrations of pollutants and the detected micro-environments.

#### 4.5.3 . Spatial Analysis

Spatial analysis addresses the problem of detecting locations with high level pollution phenomena. It consists of expressing this phenomena spread in the geographical localisation and selecting the most visited locations by all participants with high or low level of pollution.

As Hilbert SFC is by definition hierarchical, moving to a coarse level of hierarchy only needs to divide the Hilbert index by  $2^{2n}$  where  $n$  is a given hierarchy level as shown in Example B.2.1 (see Appendix B). Moreover, this harmonized presentation of the spatial dimension allows to maintain proximate objects close. In Figure 4.7, the upper map indicates the aggregated trajectories of all our participants combined

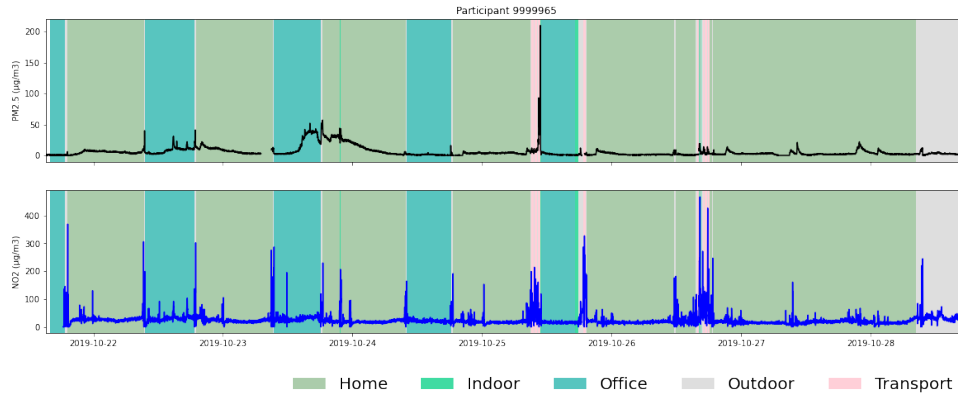


Figure 4.6: Longitudinal analysis per micro-environment.

in Paris region, the lower left map consists of a zoom of the upper map on Versailles region, while the lower right map depicts a parcel of one participant's trajectory over the whole period of the campaign. On the one hand, this spatial presentation allows us to discover the stay areas of participants based on cell density. The circles on the maps depict the places where participants spend most of their time; the radius of the circle is proportional to the time spent in the cell. On the other hand, by setting a threshold, this approach allows the detection of spatial outliers and spatial noise that are close to stay areas; below this threshold, the points will be considered as outliers. Moreover, because neighboring cells are likely to be assigned a close hilbert index, finding the cells that are close to the stay areas is not problematic.

#### 4.5.4 . Temporal Analysis

Temporal analysis consists of analysing the exposure to pollution over time. It permits to get insight about the phenomena during specific periods of time while navigating through the temporal hierarchy. The temporal hierarchy can be defined in many ways. For instance, it can be illustrated by minute  $\rightarrow$  hour  $\rightarrow$  weekdays/weekend  $\rightarrow$  month  $\rightarrow$  year.

Besides obtaining each pollutant time series, one of the applications of our MULTICS model is to get aggregates of pollutants over a time hierarchy (e.g. every 10 minutes, every one hour, every weekend, etc.). Figure 4.8 shows the average concentrations of NO<sub>2</sub> every 10 minutes for each day, including weekdays and weekends, for one participant without any restriction on their whereabouts. The graph shows a recurrent behaviour of the PM<sub>2.5</sub> depicted by high concentration values every weekday around 9am (i.e. morning rush hours) and between 5pm and 8pm (i.e., evening rush hours).

The analysis can go further by adding a GROUP BY clause on participants' micro-environments and compute the hourly average per micro-environment for all participants combined to uncover the micro-environment with the richest pollu-

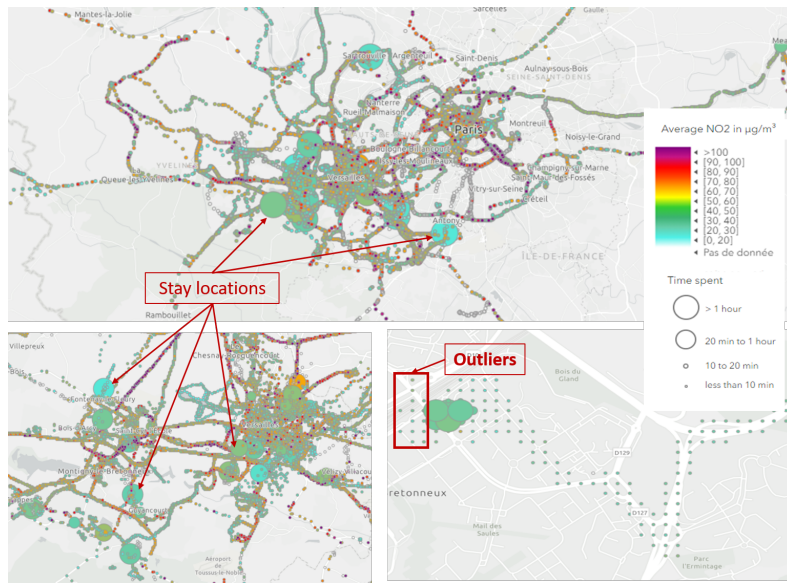


Figure 4.7: Concentrations of NO<sub>2</sub> for all participants combined in Paris and Versailles regions.

tion phenomena. Figure 4.9 depicts the maximums of hourly averages per micro-environment for all participants combined. That says participants are mostly exposed to the highest hourly average of NO<sub>2</sub> and particulate matters in the micro-environment “Office”. As for BC, participants are mostly exposed to the highest hourly average in indoor spaces.

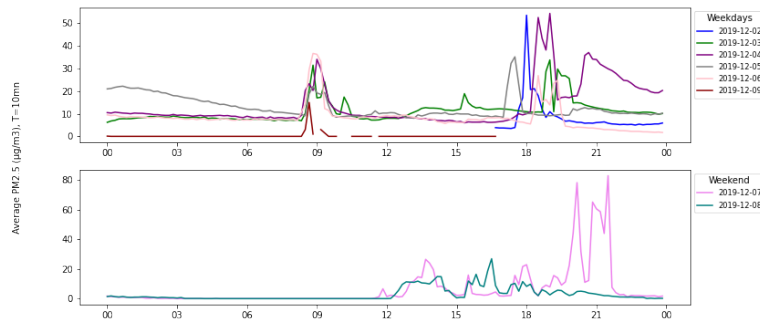


Figure 4.8: 10 minutes average concentrations of each day.

As we have said above, the MULTICS model can integrate external information that can be used in order to enrich the data with air quality information and/or to compare it to the collected data. In our context, and as an example, we added Airparif<sup>5</sup> air quality data as an external source into the model. Airparif provides the

<sup>5</sup>Air quality observatory in Paris region <https://www.airparif.asso.fr/>

	PM1.0(Max)	PM2.5(Max)	PM10(Max)	NO2(Max)	BC(Max)
Office	122.5	216.5	250.5	401.5	14089.2
Home	69.6	115.0	128.0	371.4	9943.2
Indoor	109.3	153.2	198.5	153.5	20917.8
Outdoor	130.1	213.0	236.4	147.9	12240.1
Transport	109.3	196.3	226.7	142.1	11246.6

Figure 4.9: Maximum of hourly averages per micro-environment for all participants combined.

same area pixelization as ours in the Paris region. It is set to 12.5x12.5 m<sup>2</sup> in Paris, 25x25m<sup>2</sup> in the inner suburbs and 50x50m<sup>2</sup> elsewhere in the rest of the region. Therefore, by matching the hilbert index and the timestamp, we can compare Airparif values with the collected data. Figure 4.10 depicts such comparison of NO<sub>2</sub> values throughout one participant trajectory. The two graphs preserve the same tendency, indicating the existence of a correlation between the two sources of data. However, we can remark that for some periods of time (denoted by the black dashed squares in Figure 4.10), the red and the blue lines do not share the same behaviour. After verifying the participant declared micro-environment, this difference is due to an indoor space where participant was in during these periods of time.

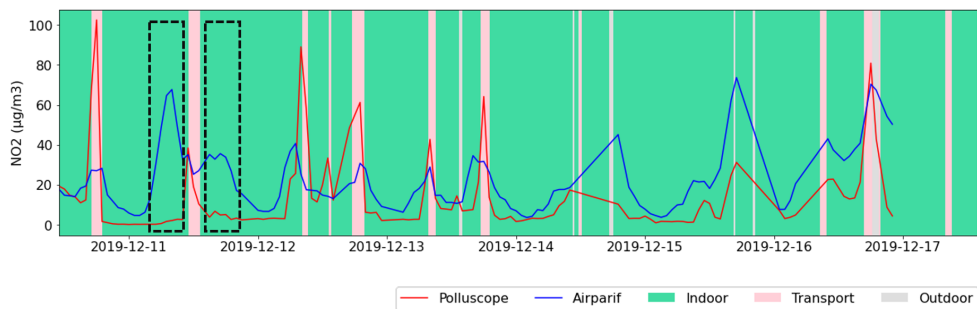


Figure 4.10: Airparif data versus the collected data for NO<sub>2</sub> in one trajectory.

#### 4.5.5 . Temporal Disaggregation

In this section, we illustrate an example of temporal disaggregation of Particulate Matter PM<sub>10</sub> using NO<sub>2</sub> time series data as proxy. As a matter of fact, Particulate Matters are available every minute. So, for the sake of the experiment, we compute the average of PM<sub>10</sub> every hour to get the low frequency time series, and then disaggregate these data into a higher frequency using NO<sub>2</sub> time series as a proxy and the maximum log likelihood estimates of Chow-Lin (Chow-Lin-Maxlog) method. Therefore, we have the ground truth data to compare the results with.

Figure 4.11 expresses the results of the temporal disaggregation process. NO<sub>2</sub>\_1minute illustrated by the green line, denotes the time series of NO<sub>2</sub> at 1 minute

frequency. `PM10_1hour` illustrated by the red curve, denotes the hourly average of the PM10 time series. `PM_1minute` illustrated by the orange curve, indicates the ground truth of the collected measurements of PM10 at the frequency of one minute. Lastly, `PM_predicted_1minute` illustrated by the blue curve, indicates the predicted values of the disaggregation process on PM10 data from a low frequency (i.e. every hour) to a higher frequency (i.e. every minutes).

As shown in Figure 4.11, the estimates of PM10 follow the trend of NO2 with respect to the aggregated values of PM10. More particularly, extreme values in NO2 pull the estimates values to their levels, which explains the occurrence of peaks in the estimates values of PM10. The results of temporal disaggregation show a clear match between the blue curve (prediction) and the orange (ground truth) PM10 value. The root mean square error (RMSE) was found to be equal to 4.079.

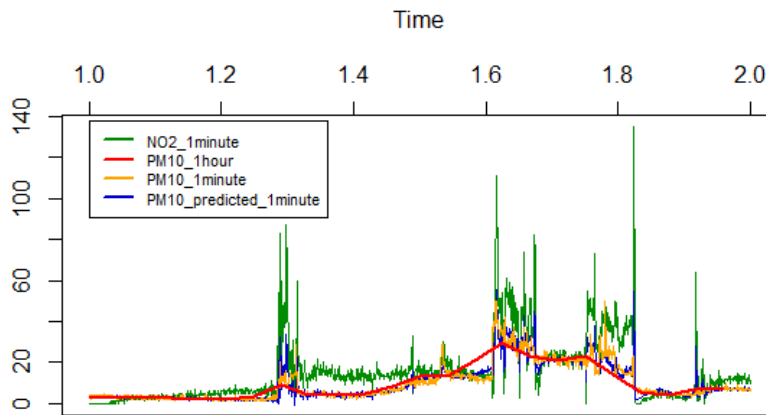


Figure 4.11: Example of the disaggregation of PM10 time series using NO2 data as proxy with the Chow-Lin-Maxlog method

#### 4.5.6 . Spatial Disaggregation

In our context, we train a Random Forest (RF) regression model on a resolution of  $200\text{ m}^2$  to estimate the values of NO2 at a finer level of  $50\text{ m}^2$ . For this purpose, the average values of NO2 collected over one week by 11 participants are computed over pixels of 200 m spatial resolution. These values are used as a response variable in the model.

As for the model features, pollution level is highly correlated with land use categories, so we integrate this information using Open Street Map (OSM) data sets. Appendix C expresses the land cover information types used in the RF regression model as explanatory variables.

The distance from pixel center to covariates included in the model is added to the data, so that it can be incorporated in the model. We conducted a colinearity

test between covariates to ensure the validity of the model. The RF model is then trained at a coarse level (i.e.  $200\text{ m}^2$ ). Thereby, to ensure the validity of our model, we use the data over a whole week of one participant in the testing phase. As inputs, we feed to the model the distance to covariates from finer level (i.e.  $50\text{ m}^2$  pixels). The model returns then as output the estimates values of NO<sub>2</sub> as shown in Figure 4.12. The blue line denotes the real values while the red line indicates the predicted values. Root mean square error (RMSE) was calculated to report the performance of the disaggregation model. Compared to the ground truth, the NO<sub>2</sub> model returns an RMSE of 8.29. We notice that the estimates of NO<sub>2</sub> follow the trend of the real values. With respect to the statistical values shown in Appendix C, the RMSE of NO<sub>2</sub> shows that the spatial disaggregation model performance is acceptable. Figure 4.12 also shows a reasonable fit in the predicted values (red) versus the real values (blue). Notice that the x-axis is the Hilbert index, which captures the variation amongst neighbouring pixels.

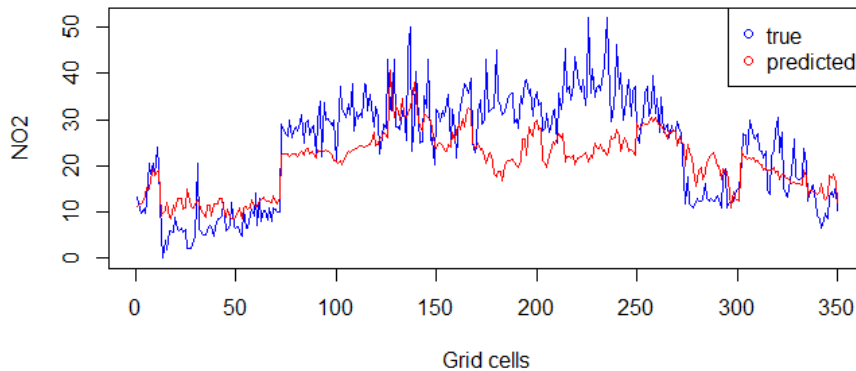


Figure 4.12: Real versus estimated values of NO<sub>2</sub> after the spatial disaggregation.

#### 4.5.7 . Computational Costs

In order to evaluate the performance of the MULTICS model in term of execution time, we have used a distributed system of 5 nodes, each node with a capacity of 32 Go. As the Polluscope deployment remains limited due to its exploratory project, the data warehouse does not reach the expected volume in MCS context in spite of its 4.4 millions tuples in the fact table `measurement`. Therefore, we opt for synthetic data to augment the data and achieve the desired volume. Furthermore, we selected six different queries with different complexity either by adding a `ROLLUP` clause to the queries, or computing different aggregated values with different conditions, or performing a `LEFT JOIN` operator on two fact tables and compute then the different aggregates. Therefore, we compare the performance of the model with respect to the data volume and query complexity. We express each query and its utility to remark the wide variety of queries that the model allows to mine data from different perspectives.

**Query 1.** *Detection of stops based on cell density.*

The multidimensional model can be used for data enrichment such as deriving stops, which are defined as the locations where a participant spends most of their time. For this, the query calculates the densities of the cells of the predefined grid, simply by counting the number of measurements per cell and per participant in the period. The most dense cells (the density threshold can be given as a parameter; here 200) is a probable stop.

```

1 SELECT user_id, Location_id, count(*) as 'density'
2 FROM measurement
3 WHERE location_id is not null
4 GROUP BY user_id, location_id
5 Having count(*) > 200
6 ORDER BY density desc

```

**Query 2 (aggregations).** *Exposition to PM2.5 per participant.*

This query returns a view of the individual average value (alias Exposure) to PM2.5 as well as the average values of other participants, which allows them to compare their exposures to others, and discover if they are more or less exposed. The output of this query expresses the exposure to PM2.5 per participant, the encountered maximum value, and the exposure duration.

```

1 SELECT M.user_id, round(avg(M.measurement_value),2) as
   Exposure, count(*) as duration,max(M.measurement_value)
   as Peak_value
2 FROM measurement M, measurement_type MT
3 WHERE M.measurement_type_id=MT.measurement_type_id
4 AND MT.measurement_name="PM2.5"
5 GROUP BY M.user_id
6 ORDER BY Exposure_PM25

```

**Query 3 (aggregations, Roll-up).** *Longitudinal analysis: Exposure to PM2.5 per participant over time.*

Longitudinal analysis involves analyzing exposure per participant over time. This query requires aggregation along the temporal dimension, a Roll-up operation over the user's dimension, and a Dice operation to select the exposure for all participants combined at the Day and Hour levels, as well as the peak value.

```

1 SELECT M.user_id, day(T.time) AS Day, hour(T.time) AS Hour,
   round(avg(M.measurement_value),2) AS Exposure, max(M.
   measurement_value) AS Peak_value
2 FROM measurement M, measurement_type MT, time T
3 WHERE MT.measurement_type_id = M.measurement_type_id AND T
   .time_id=M.time_id AND MT.measurement_name='PM2.5'
4 GROUP BY M.user_id, day(T.time), hour(T.time)
5 WITH ROLLUP ORDER BY 1,2,3

```

**Query 4 (aggregations, Roll-up, two subqueries).** *Longitudinal analysis with participants' micro-environments.*

Longitudinal analysis can be considered to cover individual exposure by micro-

environment over time (i.e., for each time interval that the individual has spent in that micro-environment). The output of this query returns the start time of the activity of the participant in question, his micro-environment, the duration of time spent in this micro-environment, the exposure to PM2.5 during this duration as well as the recorded peak value.

```

1 SELECT r1.user_id, r2.time as start_time, r2.description,
   round(avg(r1.measurement_value),2) as Exposure, count
   (*) as Duration, max(r1.measurement_value) as Peak_val
2 FROM
3 (SELECT user_id, time, measurement_name, measurement_value
4 FROM measurement_type MT, measurement M, time T
5 WHERE MT.measurement_type_id=M.measurement_type_id
6 AND M.time_id=T.time_id
7 AND MT.measurement_name = "PM2.5"
8 AND M.user_id='9999946') as r1,
9 (SELECT MR.user_id, T.time, LEAD(T.time) over (ORDER BY T.
   time) AS next_row, ME.description
10 FROM micro_environment_record MR, time T,
   micro_environment ME
11 WHERE T.time_id = MR.start_time
12 AND ME.micro_environment_id=MR.micro_environment_id) as r2
13 WHERE r1.user_id=r2.user_id
14 AND r1.time between r2.time and r2.next_row
15 GROUP BY r1.user_id, r2.time,r2.description
16 WITH ROLLUP ORDER BY 1,2,3

```

**Query 5 (aggregations, left join).** Compare external data to the collected data.

This query allows to perform a comparison between the external data (e.g. Airparif) and the collected data. The purpose of this comparison is to see to what extent the two sources of data are consistent, either by calculating a correlation coefficient and/or plotting a graph of the two data sources to visually examine the difference between the two sources.

```

1 SELECT r1.*, r2.airparif
2 FROM
3 (SELECT user_id, time, location_id, measurement_value as
   VGP
4 FROM measurement M, time T
5 WHERE M.user_id = '9999915'
6 AND M.measurement_type_id = 3
7 AND T.time_id = M.time_id
8 AND M.location_id IS NOT NULL ) AS r1
9 LEFT OUTER JOIN
10 (SELECT time, location_id, measurement_value as airparif
11 FROM airparif, time
12 WHERE airparif.measurement_type_id = 3
13 AND airparif.time_id = time.time_id) AS r2
14 ON (r1.time = r2.time AND r1.location_id=r2.location_id)
15 ORDER BY r1.time

```



**Query 6 (aggregations, Roll-up, two subqueries).** *Individual exposure per micro-environment.*

*This query is similar to query 4. By removing the time dimension, this query returns the total time spent in each micro-environment, the average value of PM2.5 in each micro-environment, plus the recorded peak value.*

```

1  SELECT r1.user_id, r2.description, r1.measurement_name,
      round(avg(r1.measurement_value),2) as Exposure, count
      (*) as Duration, max(r1.measurement_value) as Peak_val
2  FROM
3  (SELECT user_id, time, measurement_name, measurement_value
4  FROM measurement_type MT, measurement M, time T
5  WHERE MT.measurement_type_id=M.measurement_type_id
6  AND M.time_id=T.time_id
7  AND MT.measurement_name = "PM2.5" ) as r1,
8  (SELECT MR.user_id, T.time, LEAD(T.time) over (ORDER BY MR
      .micro_environment_record_id) AS next_row, ME.
      description
9  FROM micro_environment_record MR, time T,
      micro_environment ME
10 WHERE T.time_id = MR.start_time
11 AND ME.micro_environment_id=MR.micro_environment_id) as r2
12 WHERE r1.user_id=r2.user_id
13 AND r1.time between r2.time and r2.next_row
14 GROUP BY r1.user_id, r2.description, r1.measurement_name
      WITH ROLLUP
15 ORDER BY 1,2,3

```

The results of the performance tests are shown in Figure 4.13. The curves show that the execution time seems to follow a linear behaviour along data volume. It can also be seen that the difference between execution times of the six queries is not uncanny, and it varies between 10 seconds and 20 seconds.

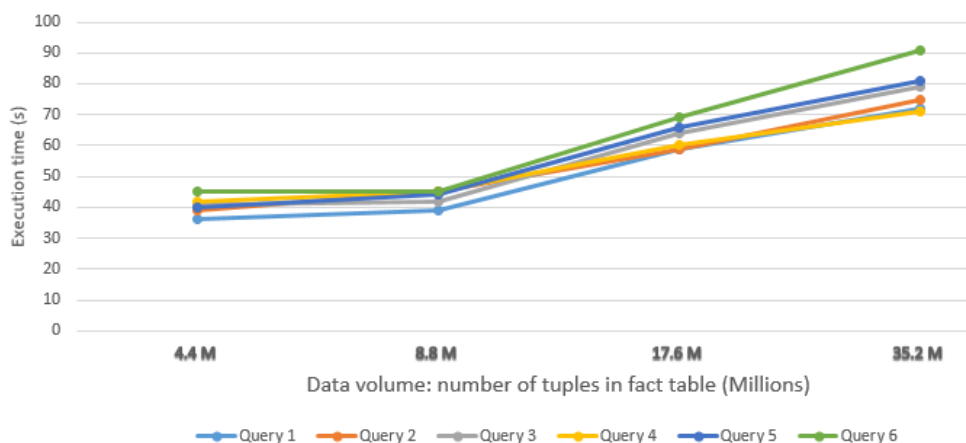


Figure 4.13: Execution time varying the data volume.

## 4.6 . Conclusion and Perspectives

This chapter tackles the exploration and analysis of geo referenced data series collected in the context of Mobile Crowd Sensing. Several works attempted to deal with the complex nature of such data. This chapter tries to fill the gap between raw data and usable information, by providing a multidimensional view on the data.

After analysing the requirements of multidimensional data modeling in MCS, this chapter introduces such multidimensional data model designed for processing and querying the different aspects of individual trajectories together with underlying pollution measures. The implementation of the model was based on the Spark SQL and Hadoop ecosystem for data analysis in order to consider all the aspects of the data. The core data model and the methodology considered is applied to urban mobility and pollution data but is generic enough to act as a reference model for other applications.

We intend to use OLAP model, on one hand, to detect anomalies by exploring the data and correcting it by applying for instance statistical smoothing methods such as the moving average and the exponential moving average. On the other hand, OLAP model can be utilised for data enrichment such as the derivation of stops (i.e. stay locations) according to the density of points per cell, or according to the sparsity of GPS readings over time. Finally, in the case of real life use, the volume increases continuously which could be a bottleneck. The question which arises, therefore, is *"with the incurred data volume, is it reasonable to store all the data in the data warehouse, or store only the fresh measurement stream, or use a hybrid model where historical data are aggregated to some extent to achieve a trade-off between utility and efficiency?"*. This leads to new challenges in terms of model and maintenance operation (to trigger the aggregation). The suggested disaggregation operators will be useful to estimate the original facts in this context.



## 5 - Automating Data Analysis Pipelines

### Contents

---

<b>5.1</b>	<b>Introduction . . . . .</b>	<b>130</b>
<b>5.2</b>	<b>Problem Statement . . . . .</b>	<b>131</b>
<b>5.3</b>	<b>Microservices Architectures: Related Work .</b>	<b>131</b>
<b>5.4</b>	<b>System Architecture . . . . .</b>	<b>132</b>
5.4.1	Data Processing . . . . .	133
5.4.2	Visualisation . . . . .	134
<b>5.5</b>	<b>Design of the Microservices . . . . .</b>	<b>136</b>
<b>5.6</b>	<b>Visualisation Demonstration . . . . .</b>	<b>139</b>
5.6.1	COMIC Demonstration Scenario . . . . .	139
5.6.2	Grafana Demonstration . . . . .	141
<b>5.7</b>	<b>Conclusion . . . . .</b>	<b>142</b>

---

## 5.1 . Introduction

In Chapter 1 of this dissertation, we have motivated our proposal for our sixth contribution, i.e. an end-to-end infrastructure based on micro-services for the whole data analytics lifecycle. Therefore, we present in this chapter a scalable infrastructure based on microservices for the implementation of the whole system lifecycle. The proposed architecture includes the discussed components in Chapters 3 and 4 in order to build a scalable and reliable ecosystem for automating data analysis pipelines.

As discussed in Chapters 3 and 4, we have come to the conclusion that the combination of databases with machine learning (ML) algorithms constitutes the backbone of our computing infrastructure. Therefore, a design methodology that takes into consideration the different challenges of MCS data is highly desirable. The solution design needs to (i) digest data coming from different sensors and process them in a way that guarantees the efficient management of different data formats and granularities, (ii) annotate the observed measurements and identify their meaning and context in order to proficiently analyse the collected data, and most importantly, (iii) orchestrate the whole process, rather than use monolithic approaches.

Therefore, to provide a robust and scalable embedded model design, such as automated data analytics pipelines, for data processing and analysis in MCS, we focus on the following key questions: What are the main components for extracting usable information from raw sensory data ? How can an analytics model be developed using microservices ? How can the identified components be orchestrated to achieve analytics services ? How can the analytics results be delivered for different users and decision makers?

Microservices architecture is a software development methodology that seeks to break down an application into core functions, each of which is referred to as a "service", which is designed to meet a specific and unique business need. So, in this work, we present an envisioned End-to-End microservices-based architecture for implementing and automating data analytics pipelines to extract usable information from MCS. We identify the main features for extracting information and deploy them into microservices [41]. The proposed architecture assigns to each service a specialised functionality which is scaled and operated independently of other services. The whole microservices pipelines are orchestrated using Apache Kafka.

The remainder of this chapter is organised as follows. In Section 5.2, we give a thorough problem description. Section 5.3 reviews the principles designing and automating data analytics pipelines based on microservices in MCS applications. Section 5.4 presents the proposed design methodology of our system for MCS applications while introducing two visualisation platforms. Section 5.5 discusses the design of microservices. Section 5.6 focuses on illustrating some demonstrations scenarios for data visualisation. Finally, Section 5.7 presents the conclusion.

## 5.2 . Problem Statement

As described above, designing an innovative system for data analytics pipelines built of MCS sensors is the main objective here. In MCS environment, machine learning-based approaches have become an important part for automatising data processing and knowledge discovery. This field has seen a great spread of technologies that support data mining analytics. These technologies support data querying and predictive models with systems such as SQL-based databases, No-SQL-based databases, machine learning based tools such as Python, R and Scala. While these technologies may operate independently from each other, it is natural to seek a single platform with connector to each system, that can automatise the data analytics workflow in an autonomous way.

Furthermore, even with the development of several machine learning-based solutions for data analysis, data analytic designs for MCS which can process heterogeneous MCS data and get meaningful insights from it are still limited. Particularly, due to the huge amount of the collected data nowadays, a MCS analytics system must be designed to be scalable and fault tolerant, so it is able to handle future developments, and to keep pace with the increase of data volume without losing any.

Moreover, since the deployment of machine learning models can be done through two possible ways: either by (i) pre-training the model off-line and integrating it directly into the workflow, or by (ii) training and using the model on the fly. The designed system should present high flexibility which allows to easily add or remove component from the system without compromising its robustness.

## 5.3 . Microservices Architectures: Related Work

In recent years, architectures based on microservices have become a popular technique for several platforms for the development of flexible applications. Dmitry and Manfred [36] demonstrate the benefit of using microservices in M2M developments to overcome the limitations of monolithic approaches in IoT applications. Ali *et al.* [3] propose a design methodology based on microservices to support predictive analytics for IoT applications. In their proposed framework, each part of the analytics process is embedded in a service and can handle a specialised functionality. Apache Kafka was chosen to process IoT data in the proposed design. Furthermore, the authors focus on ML based approaches to provide predictive analytics capabilities for IoT data, and validate their approach using two datasets with, at most 700 instances. However, MCS data is characterised by its large scale, and the usefulness and scalability of this approach on such huge data is not confirmed.

In the context of smart cities, Bellini *et al.* [11] propose a system for knowledge base construction process of smart cities related aspects, from data ingestion to knowledge base construction and validation. The proposed system allows managing large volumes of data coming from a variety of sources. However, the approach of

[11] did not consider embedding ML algorithms in the knowledge base construction pipelines, which is an important component in our case. Krylovskiy *et al.* [75] build a smart city IoT platform based on microservices architecture style for a variety of applications. The authors demonstrate the benefits of using microservices based architecture in large-scale cross-domain application development for smart city environment. As the authors claim, the proposed platform is at its early stages and a more thorough evaluation is needed to extract more objective conclusions.

In a more generic line of works, Hamilton *et al.* [63] introduce an Apache Spark based microservice orchestration framework that integrate intelligent and cognitive services provided by ML into big data applications. The authors demonstrate the scalability of their approach and its competitive quality for a variety of intelligence tasks data such as text, vision, face and numeric data. The applicability of this approach on heterogeneous sensory data needs to be verified.

In this work, we propose an architecture based on microservices and ML to process and analyse MCS data. The process is orchestrated by Apache Kafka, which allows the migration from monolithic approaches to an automatised and autonomous workflow.

#### 5.4 . System Architecture

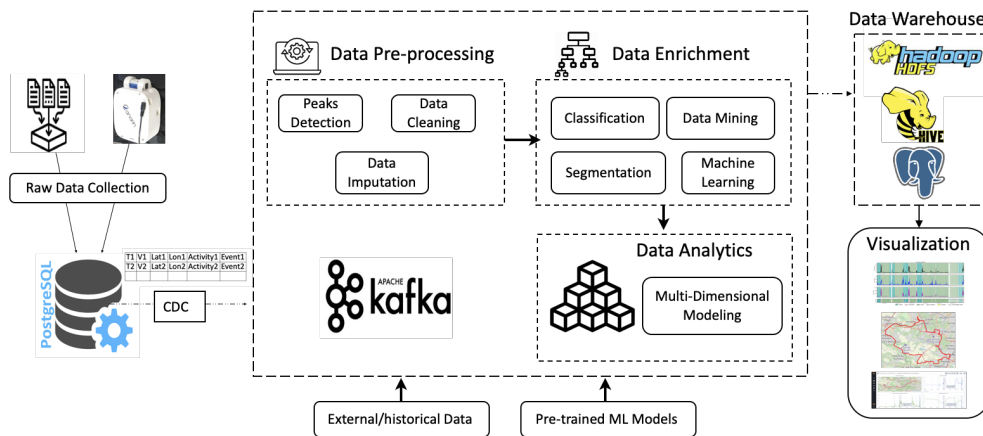


Figure 5.1: Design of microservices for automating data analytics pipelines in MCS.

Dragoni *et al.* [38] define microservice as a cohesive, independent process interacting via messages. This view suggests that microservices are independent components. Each component is conceptually deployed in isolation and targets one single task. Starting from this point of view, we define the main tasks for extracting usable information and deployed them as microservices embedded in an envisioned architecture for data processing in MCS applications. The proposed architecture, as shown in Figure 5.1, provides a roadmap from data collection and

ingestion, to data preprocessing (such as data cleaning, noise removal, and data imputation), data enrichment with the context of the measured observations, data storage and analytics in a multidimensional schema, and data visualisation.

#### 5.4.1 . Data Processing

In Chapters 3 and 4, we discussed thoroughly the contents of the three micro-services, i.e. *data pre-processing*, *data enrichment*, and *data analytics*. Section 5.4.2 tackles the purpose and content of the last micro-service, i.e. *visualisation*.

In summary, in the first layer of the design, (i.e. data collection and ingestion), the raw data is pushed and stored into the database (e.g. PostgreSQL). We have chosen Apache Kafka to deploy our data analytics pipelines. Kafka is a powerful event-streaming platform, which can process huge amounts of real time data. Plus, it is scalable and fault tolerant. Using Kafka connect plug-in, we set a change data capture (CDC) model, which is the process of capturing any change in the database in real time, that listens to the database and pulls new events in the preprocessing services. Capturing changes can help in synchronizing data to other applications on the fly without passing through scheduler or batch process. Kafka will act as an orchestrator between the different microservices and components.

The collected time series data are segmented into segments of length  $n$ , where  $n$  can be a specified window length depending on the needs (i.e.,  $n=5$  minutes,  $n=1$  hour, or  $n=1$  week, etc.). Obtained data segments are then pushed to our first microservice, the pre-processing micro-service. Such as micro-service is responsible of improving the quality of the unprocessed data by applying data cleaning, noise removal with peak detection, while filling missing values with appropriate data imputation techniques.

The preprocessing micro-service will publish back the data into Kafka, which will be pushing them to a higher level microservice (i.e., data enrichment microservice). Such a data enrichment microservice, is in charge of segmenting and enriching data with the context based on some data mining techniques and by applying some pre-trained ML algorithms. The results of such an enrichment will be published back through Kafka connect to Kafka, which will forward data to the next microservice of data analytics. The data analytics microservice is in charge of performing analysing on the enriched and ready-to-use data, which are stored in a multidimensional data warehouse. Furthermore, external sourced data is integrated to be compared with MCS and further enrich them if needed. Since the two sources of data do not necessarily have the same scale, we introduce spatial and temporal disaggregation to extract, based on ML, finer grained data from coarse data and handle the problem of low MCS coverage, and facilitate the comparability of both data sources. The results are delivered to the data vizualisation microservice, which is discussed in the following section.

The proposed architecture offers many advantages to our data flow pipeline:

- Fault tolerance, since Kafka is fault tolerant. If a microservice is down, we



will not lose data, and when it is back, it starts from where it stops.

- Testing new microservices in run time without affecting the whole pipeline (i.e. testing new classification model with new features on real data, while keeping the old model running in the pipeline).
- Kafka will allow adding new microservices if needed or removing existing ones while the system is up and running.
- The system decouples microservices from data; they do not need to run processes to check if there is new data, they will be notified when their data is ready.
- Kafka Connect can help in transforming data while passing them from and into Kafka (i.e. we can change some values if needed, or even rename some fields to be coincident and understandable by other microservices)

#### 5.4.2 . Visualisation

We provide two frameworks for data visualisation of the enriched trajectory data, which presents our fifth contribution, i.e. interactive data visualisation platforms. The first platform consists of a Graphic User Interface (GUI), called **COMIC** (**Context Of Mobile Crowdsensing**), that we implement to show the different recognized micro-environments vis-à-vis the declared one, and allow the user to customise the learning algorithms. The second visualisation tool, which is based on Grafana<sup>1</sup>, displays all the components of the enriched trajectory data through time, including AQ data, trajectory, and contexts.

#### COMIC system overview

We addressed previously in Chapter 3 the usefulness of identifying automatically the micro-environments and its value in understanding personal exposure to pollution. We presented a robust hybrid model for micro-environment recognition based on multi-view learning that outperforms state of the art baselines, i.e. kNN-DTW and MLSTM-FCN.

In this section, we present a full-fledged implementation **COMIC** that can be used in real-world applications to infer the micro-environment from environmental crowd sensing data. It allows to tune the learning algorithms and compare the results to the baselines. The proposed system consists of a Graphic User Interface (GUI) and four layers: (1) The declared micro-environments by the participants, (2) the environmental data measurements over time, (3) the detected change points, and (4) the detected context.

- **Interface.** The GUI of COMIC is a front-end interactive interface which has three functionalities. It provides a list of all the participants. The

---

<sup>1</sup><https://grafana.com>

user is invited to choose one participant's data then the interface provides the plotted measurements with the corresponding user annotations (i.e., declared). The user is also allowed to choose the first-level and meta learners among a predefined learning models list. COMIC back-end then applies the classification for this participant's data and shows the predicted context for each single view and for the multi-view.

- **Declared micro-environments.** This information, which is also called self-reporting, characterizes the declared micro-environment by participants. This information is not always accurate as participants do not bother to fill thoroughly this information. Our GUI permits to show to what extent this information is accurate or not compared to our detected contexts and vice versa.
- **Data measurements.** Data measurements consist of the collected data by the participants. These data is used as input for the multi view learning model to detect automatically the context of the participants.
- **Change points.** Change points consist of the exact timestamps of the change in participants' micro-environments. Based on participants measurements, our GUI calls on the back-end the detected change points based on multidimensional change point detection [42], and displays the results.
- **Micro-environment recognition.** Our micro-environment recognition model is based on multi view learning modeling. The user is allowed to choose which model to use for the first-level and the meta learners. The GUI displays then the detected micro-environment for the selected participant.

COMIC leverages our idea in Chapter 3 and improves it. Instead of being restricted to *RF* as first-level learner and meta-learner, we carried several extensive experiments trying different classifiers (including SVM, and kNN) and added a new multi-view model with *kNN* as first-level learner and *RF* as a meta learner to show the effect of the first-level learner and meta-learner on the results. Plus, even in the case of missing dimensions, COMIC maintains its robustness and detects the context with the existing dimensions. Furthermore, we have implemented and included in COMIC the results of the Multivariate Long Short Term Memory with Fully Connected Network model (MLSTM-FCN) as well as KNN-DTW classifier for the aggregated data (all views are aggregated together) which are considered as state-of-the-art by the time series classification community [48].

Additionally, COMIC includes another component which consists of segmenting the multivariate time series into coherent segments, each segment represents a micro-environment by resorting to multivariate time series change point detection. Our change point detection approach consists of applying the CUSUM algorithm as first-level learner on each dimension separately. Each dimension generates a set

of detected change points. The output is then fed to a second-level learner to learn the weights of every dimension in proportion to the performance of individual learners using a gold set of annotated data as ground truth.

### Grafana dashboard

We took advantage of the visualisation platform of Grafana<sup>2</sup> to offer to participants an interactive experience with their enriched trajectory data. This involves exploring and analyzing the collected data along all their dimensions (spatial, temporal, quantitative or semantic measures, etc.) and at different levels of granularity. The visualisation platform Grafana is used for interactive visualisation of pollution levels encountered throughout participants' micro-environments and trajectories. Figure 5.2 presents an example of a trajectory with a dynamic link between the location of the participant (the blue dot on the map) and the concentration of pollutants according to the red sliding line on the measurement curves. Section 5.6.2 gives more details about the functionalities of Grafana.

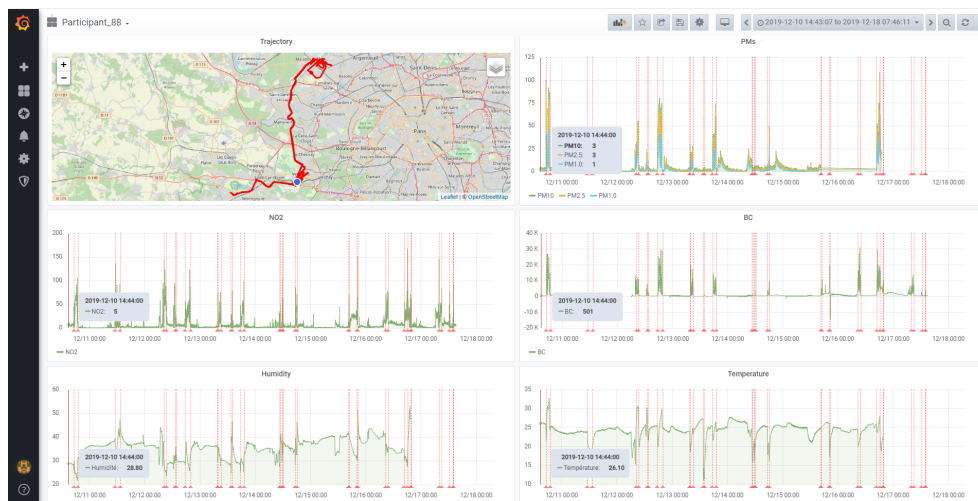


Figure 5.2: Visual analysis of a trajectory and its associated measurements.

## 5.5 . Design of the Microservices

To realize the proposed architecture design of microservices for automating data analytics in MCS, an implementation prototype has established as shown in Figure 5.3. A virtual machines on Debian system has been prepared with deployed docker images. Each instance of the docker hosts a microservice. A machine with a configuration of Xeon Gold processor and 32 GB of RAM, running Linux OS

<sup>2</sup>Open source data visualisation and analysis platform: <https://grafana.com>

hosts the Docker setup. The docker images together form a cluster of containers, which contains Zookeeper, Kafka, Kafka Connect, PostgreSQL and Python. The data pipeline is described in Figure 5.3.

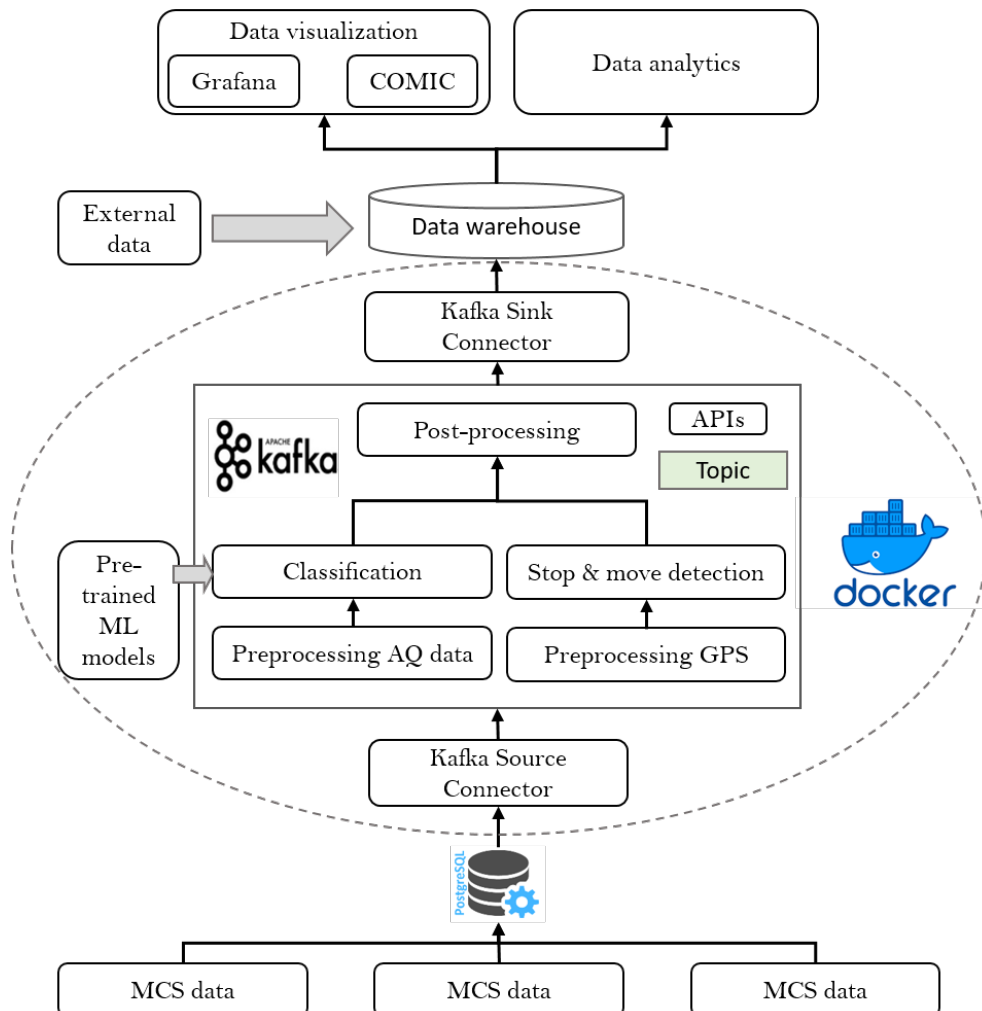


Figure 5.3: The system implementation prototype.

In this proposed architecture, data is extracted from different data sources and stored in PostgreSQL. Using Kafka connect plug-in, we set a change data capture (CDC). It provides the capability to capture any changes in the database in real time. Kafka Connect is connected to the data source (i.e. PostgreSQL) to fetch new tuples as they arrive. Since Kafka allows publishing and subscription of data into topics, Kafka Connect pushes the fetched data into a Kafka topic for any application to consume it directly using the consumer API. In our case, the preprocessing microservices read from the topic using a consumer API (Python in our case), and publish the results of the preprocessing back into another Kafka topic. In addition, these results are also sent to a target datastore (PostgreSQL

here). We set in parallel two preprocessing microservices. The first one pre-processes AQ data, by cleaning it from noise and outliers, and interpolating missing values. The second microservice pre-processes GPS data, which consists of (i) cleaning it from noise, (ii) rasterizing it into a grid, and finally (iii) assigning a Hilbert index to each grid cell as shown in Sections 4.4.2 and 4.4.4.

Thereafter, the classification microservice reads from the Kafka topics written by the AQ pre-processing microservices, and publishes the results of classification into another topics. In parallel, the stop & move detection microservices based on GPS data only, as discussed in Section 3.3.5, is running. It reads from the Kafka topics written by the GPS pre-processing micro-services, and publishes the results into another topic. The two results is combined with a post processing microservices, which post-processes the results of classification with the results of stop detection, and publishes the results, i.e. the detected micro-environment into Kafka. Thereafter, Kafka Sink Connector write data from the Kafka cluster to a new data warehouse (postgreSQL here) for other purposes such as data mining and visualisation. At the end, both the preprocessing results and the detected micro-environments are stored in the target database (PostgreSQL here) for data mining, analysis and visualisation. We define new schema for each result. The preprocessed data follows the schema of the input data collected by Kafka Source Connect, and stored in the target database. As for the detected micro-environments, the schema is first declared in the target database for the results of the post-processing microservice to take into account the participant ID, its micro-environment and the start time of entering the micero-environment.

Furthermore, the microservice, i.e. data analytics, allows to apply some data mining techniques to get insights from the data by analyzing it and/or compare it to external-sourced data (e.g. Airparif).

Last but not least, the microservice visualisation is directly connected to the data warehouse. On the one hand, it permits to inspect data and interact with it using Grafana. On the other hand, COMIC GUI allows to investigate the detected micro-environments and compare them with the declared micro-environments. The user can also switch the learning method and examine its impact on the results. Some visualisation scenarios are discussed in Section 5.6.

In this proposed architecture, when there is a new update, microservices can publish events and read them from Kafka without the need to communicate with each other. Indeed, communication between microservices is asynchronous and they are unaware of each other. If one microservice crashes, the system does not follow and continues to operate, which makes it efficient and easy to maintain on a large scale.

## 5.6 . Visualisation Demonstration

In this section, we demonstrate the user interaction experience with our two visualisation tools, i.e. COMIC and Grafana dashboard.

### 5.6.1 . COMIC Demonstration Scenario

In this section, we illustrates the user interaction with COMIC interface. The experiments are carried out on different environments. The multi-view learning model was implemented in Python 3.6 using scikit-learn 0.23.2 and tslearn [116]. The deep-learning model MLSTM-FCN [72] was trained using Keras 2.2.4. Our GUI (graphical user interface) was implemented using python 3.6, Plotly <sup>3</sup>, and Dash <sup>4</sup> framework. A real-world environmental data collected in the context of Polluscope project is used as a benchmark of environmental crowdsensing data. In this context, participant collect air quality measurements (NO<sub>2</sub>, PM<sub>1.0</sub>, PM<sub>2.5</sub>, PM<sub>10</sub>, BC, Temperature, Humidity) plus GPS locations which are used to derive participants speed. The recruited participants where given a mobile app in order to annotate their micro-environments whenever it changes. Micro-environments are grouped into five categories: Home, Office, Indoor, Outdoor and Transport. Indoor spaces incorporate all closed spaces except home and office, such as restaurants, stores and stations, while outdoor spaces, as its name indicates, consist of open spaces such as park and street. Users can load our interface and start enjoying its appealing functionalities. We emphasize three main scenarios of COMIC:

#### *Data visualisation:*

User can visualize the collected data during the campaign period. Each dimension is plotted in a separate graph. Along with each plot, the corresponding declared activity at that time is shown. Thus users can easily see visually how much the changes in participant's context and the changes in data are correlated. Figure 5.4 shows the different dimensions plots with the corresponding declared activities. Users can choose the participant id from the drop down in order to navigate through the data of different participants.

*Classification and CPD over all dimensions:* Users are able to perform classification and change point detection algorithm over the data of a specified participant. The users need only to specify the participant id, then they can choose the learner in the first-level learner (i.e., KNN-DTW or Random Forest), and by clicking on the classify button, classification and change point detection (CPD) will be applied on each view.

As shown in Figure 5.5a, each dimension is plotted with its declared micro-environments versus the ones detected by the first level learner on this dimension. Moreover, three plots appear showing the aggregated view using the KNN-DTW, the MLSTM-FCN algorithm, and another one showing the results of the multi-view learner. We can also notice that in the absence of some dimensions, KNN-DT and

---

<sup>3</sup><https://plotly.com/>

<sup>4</sup><https://dash.com/>

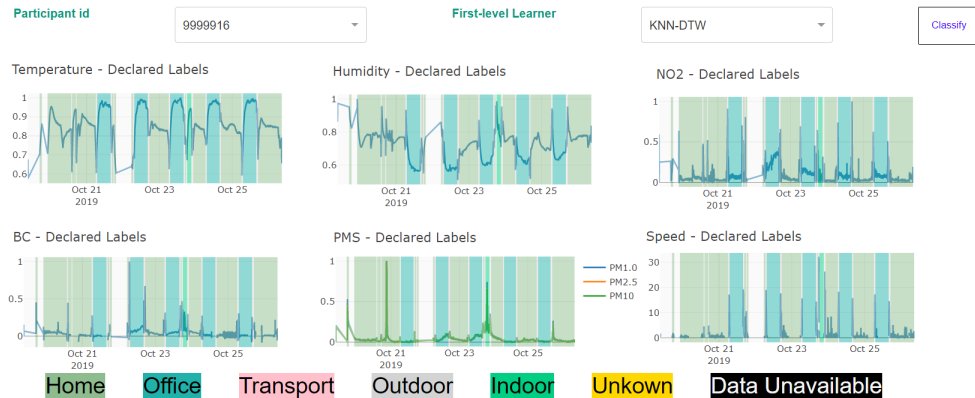
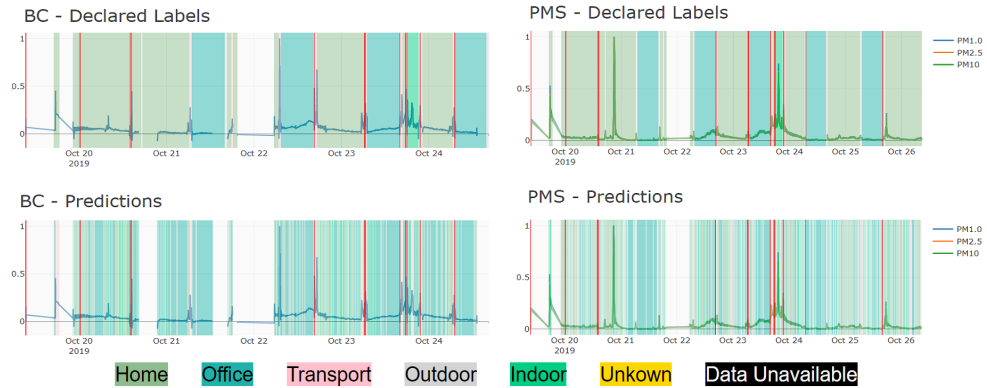


Figure 5.4: COMIC visualisation GUI.

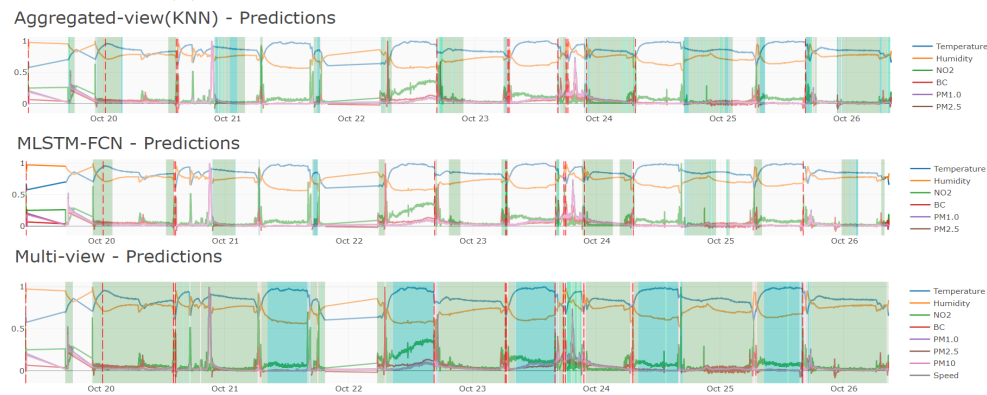
MLSTM-FCN will fail to detect the micro-environment, while the multi-view learner keeps detecting the micro-environments. Furthermore, the detected changes by the CPD algorithm are plotted also as red dash lines. This interface allows users to see to what extent the results of COMIC are accurate vis-a-vis the declared micro-environments and vice versa, by comparing the declared micro-environments in Figure 5.5a (e.g., BC - Declared labels) and the predicted micro-environments (i.e., Multi-view - Predictions).

*Classification and CPD over a specified dimension:* Another functionality of COMIC allows users to focus on one dimension and plot the classification result of the first-level learner, which in most the cases is not accurate. Users only need to specify the participant id, the dimension, and the first level learner from the drop-down lists, then they are invited to click on the classify button. The output of this functionality shows three plots: (1) a plot of the specified dimension with the declared micro-environment, (2) another plot showing the results of the chosen first level learner on the specified dimension, and (3) a third plot showing the results of the multi-view learner. All the plots include the Change points detected as a vertical dashed red line.

Figure 5.6 shows the comparison between the declared micro-environments, the ones predicted from the single dimension in question, and the ones detected by the COMIC model. On the one hand, the first graph indicates that the participant stayed outdoor for two consecutive days, meaning that this participant did not thoroughly annotated their data. Yet, our multi-view model (third plot) can detect successfully the micro-environment of this participant during this two days. On the other hand, the second plot shows the results of the first level learner. While this learner fails to detect all the micro-environments correctly, we recognize that the multi-view approach does a good job in detecting the participant's micro-environments. It is worth mentioning that when no data is collected whatsoever, our multi-view model can not detect the micro-environment.



(a) First Level Classification versus Declared Classes.



(b) Multi-view Classification versus Declared Classes.

Figure 5.5: Classification and CPD Dashboard

*Note that all the plots of the different scenarios are interactive plots, thus users have flexibility to select some areas or to zoom in/out over the plot. Moreover, users can download these plots as a PNG images.*

### 5.6.2 . Grafana Demonstration

In this section, we illustrate the functionalities of Grafana that we offer to participants an interactive visualisation experience with their data. The dashboard as shown in Figure 5.2 lays out the different dimensions of the collected data (i.e. AQ, trajectory, and micro-environments) in several panels. Five panels are reserved for AQ data, i.e. particulate matters (PMs), NO<sub>2</sub>, black carbon (BC), plus the temperature and relative humidity. Another panel is created for the participant trajectory which is dynamically linked to the concentrations of pollutants. For instance, participants can zoom in on only one panel and visualize the data of all the sixth panel at the same time range. Figure 5.7 depicts a zoom in on the same data as in Figure 5.2. We emphasize that no contribution has been suggested here. We simply took advantage of Grafana's functionalities to provide a user-friendly



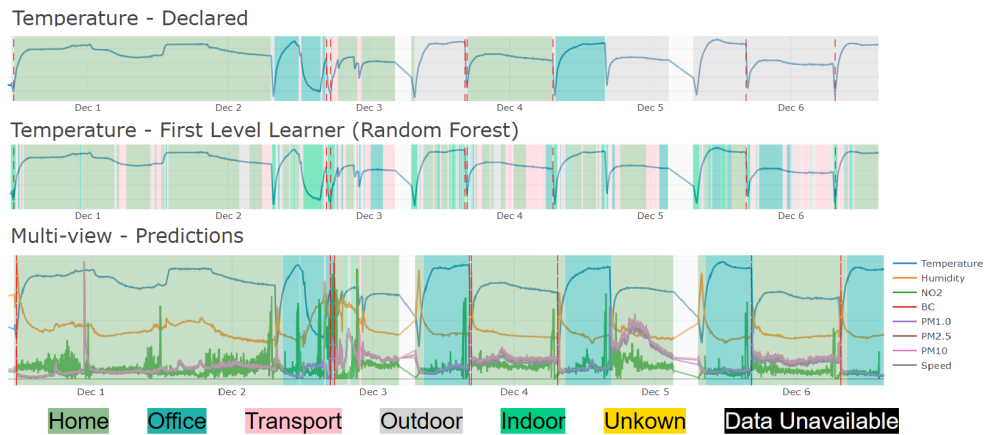


Figure 5.6: Comparison between single-view and multi-view models.

visualisation experience to the participants.

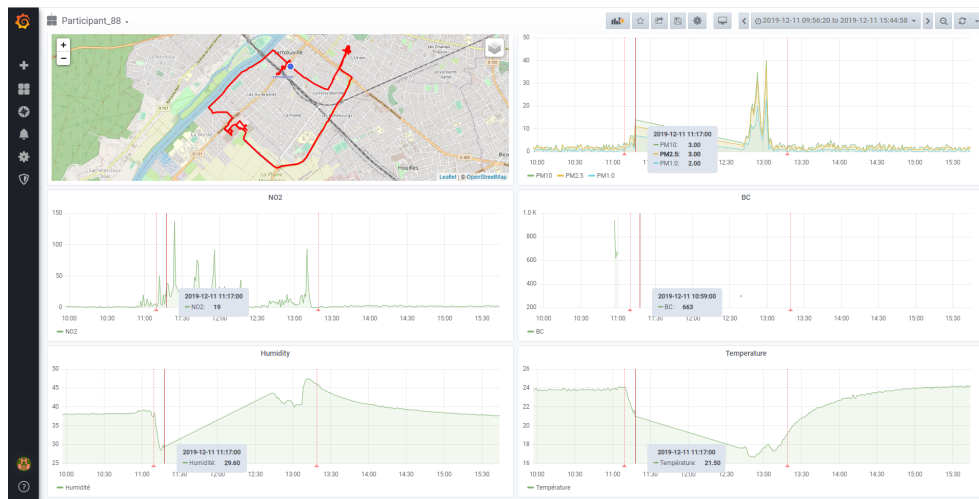


Figure 5.7: A zoom in on the collected data.

## 5.7 . Conclusion

In this chapter, we have presented an envisioned system for implementing data analytics pipeline based on micro-services architecture and Kafka. We identified the main features for extracting usable information from raw MCS data and deploy them into microservices. Kafka will act as an orchestrator that have the knowledge of the services, and takes care of communication between micro-services. Our proposed architecture will allow the automation of processing raw data into meaningful information in a scalable, and fault tolerant manner.

Furthermore, we developed COMIC, a visualisation GUI to show the different

recognized context and highlight the importance of multi-view learning compared to single view learning and other baseline approaches. COMIC covered also time series segmentation based on change point detection to demonstrate the change points in participants' contexts which are detected automatically by a multi-dimensional CPD model.



## 6 - Conclusions and Future Work

### Contents

---

<b>6.1</b>	<b>Summary of Contributions . . . . .</b>	<b>146</b>
<b>6.2</b>	<b>Future Work . . . . .</b>	<b>149</b>
6.2.1	Map-matching Based Enrichment . . . . .	149
6.2.2	Events Processing . . . . .	149
6.2.3	Exposure Profiles . . . . .	149
6.2.4	Privacy and Participants' Incentives . . . . .	149

---

This is the last chapter of the dissertation. We will summarize the work presented throughout the previous chapters, and emphasize our achievements against the research questions presented in Chapter 1 in Section 6.1. Thereby, we will highlight possible future research directions in Section 6.2.

## 6.1 . Summary of Contributions

We started this dissertation by emphasizing the need of a holistic approach for data management and analysis of spatio-temporal data series produced by moving objects in the context of Mobile Crowd Sensing (MCS). While the literature proposes different solutions for handling moving objects such as moving objects data management in the database community and several data mining techniques for analytical purposes, an overall approach that fills the gap between data collection and comprehension is missing. Therefore, and as stated in Chapter 1, the primary research question this dissertation was prepared to address is:

***Problem Definition & Research Question #1: What is the gap between raw enriched trajectory data and usable knowledge, and how to bridge it ?***

To address this problem and based on a thorough investigation of the existing research proposals in the literature, we have identified the main structural components of extracting comprehensive and usable information from enriched trajectory data. We have broken down these components into a set of research questions that this thesis work answers.

### ***Research Question #2***

*What are the fundamental preprocessing steps for spatio-temporal enriched trajectory data ? How to find spatio-temporal noise and outliers ? How to differentiate between an artifact peak and noise ? What is the gap between raw enriched trajectory data and clean enriched trajectories, and how to bridge such gap ? How to achieve purified enriched trajectories ?*

To answer this question, we started by differentiating between systematic errors and random errors. We evaluated then state of the art methods that focus on reducing the effect of random errors on sensor readings. Therefore, we designed pre-processing layer that can bridge the gap between raw trajectories and purified enriched trajectories. Specifically, in Chapter 3, we presented a computing approach for reconstructing the enriched trajectory data in terms of time series and GPS data cleaning from outliers and noise, as well as interpolating missing values.

### ***Research Question #3***

*Can we provide a more comprehensive and semantically enriched representation of enriched trajectory data ? Any intermediate models are necessary to achieve the semantically enriched representation aforementioned ? Can we contextualise the data and enrich it with the type of activity and movement (i.e. micro-environment) ? What are the spatio-temporal requirements to characterize the micro-environment and summarize their observed properties ? Can we combine different sensors (i.e.,*

*both GPS and AQ sensors) to automatically infer people's context ? Which types of algorithms and computational solutions need to be designed for this purpose ? Do data mining (e.g. feature representation) or statistical summary techniques have the ability to provide solutions for such recognition tasks ?*

To answer this question, we have investigated a wide range of state of the art proposals in the area of data segmentation and activity recognition from GPS trajectory and time series. Although these two subjects are chained to each other, yet, they have been studied distinctly and together in the literature depending on the application domain. Chapter 2 presents the finding of this investigation depending on the origin of data (i.e. GPS data, wearable sensors) and exhibits some generic methods independently of the origin of the data.

Therefore, we developed a model for multidimensional data segmentation based on change point detection (CPD). The proposed model divide the cleaned enriched trajectories into a set of coherent segments, where each segment represent a micro-environment. We contrast the proposed approach with a traditional CPD model and show the effectiveness of our approach. We further complete the semantic enrichment by designing a hybrid model for context recognition which can integrate geographic and multivariate time series views to annotated enriched trajectories with the type of activity and movement. The geographic view adds semantic annotations to segments (i.e stop and move annotation, transportation mode annotation) from GPS tracks only. The multivariate time series view detect the exact label of segments (e.g. *home, office, store, metro, park*, etc.). The designed model combines data from heterogeneous sensors, and has the ability to infer efficiently the label even if one (or more) dimension is missing.

#### **Research Question #4**

*How to further enrich sensory data ? Does such semantic enrichment need additional external sources, such as the traditional network of fixed stations and models ? How to merge sensory data with external data ? How to align both sources of data with such different spatial and temporal scales and very low MCS coverage while taking into account micro-environments ? How to handle the problem of missing value provoked by merging two data sources with different spatial and temporal scales ? How to integrate and compare external data to sensory data ? Can we provide a generic model for external data integration and comparison with sensory data ?*

To allow data mining and analysis of the collected data, we proposed an adaptive and flexible system for data management and exploration in MCS. The proposed model captures every facet of the multidimensional data and allows its exploration from several perspectives (i.e., longitudinal, spatial and temporal perspectives) and multi-scale. In addition to this, we adopted the discretization method of the spatial and temporal dimensions. Therefore, the integration of external data from fixed stations and the alignment of the spatial and temporal granularities of the two data sources in order to match and compare them has become possible.

Moreover, we introduced new operators of spatial and temporal disaggregation to extract, based on machine learning, finer grained data from coarse data and handle the problem of missing values and low MCS coverage. This spatial and temporal disaggregation allowed us to provide the complete exposure story of participants even if their sensors stopped collecting data for a while.

#### ***Research Question #5***

*Can we provide an interactive visualisation platform to explore every facet of the data, including GPS tracks and measurements ? Is it possible to visualise the difference between the detected and the declared micro-environments ? To what extent the computation model can affect the results of micro-environment's detection ? Can we visualise that effect ? Which sensory data contribute more in this inference ?*

To answer this question, we presented a two-faces visualisation framework of the enriched trajectories. First, we developed a visualisation Graphic User Interface (GUI) to show the different recognized context and highlight the importance of multi-view learning compared to single view learning and other baseline approaches. It allows users to customise the learning methods for detecting micro-environments and displays the detected micro-environments vis-à-vis the declared micro-environments. This functionalities permit to discover visually the best model for micro-environment detection and, thereby, choose the most suitable model for the data.

The second visualisation tool was based on Grafana to offer to participants an interactive experience with their enriched trajectory data. The Grafana dashboard plots the different dimensions of the collected data plus the GPS trajectory on a map. Participants can zoom in and zoom out on the data. They can select periods of time with high exposure to pollution and discover the micro-environment related to this phenomena. Furthermore, an interactive link between the map and the AQ plots is created so the participants can see visually the location of their collected data.

#### ***Research Question #6***

*Can we automatise the aforementioned process of bridging the gap between data collection and data comprehension ? How to enable the pipelines to work properly and efficiently without human involvement ? Which technologies are best suitable for this purpose ?*

An envisioned system for implementing and automating data analysis pipelines has been proposed to answer this question. We deployed a scalable infrastructure based on micro-service for the whole model lifecycle. Kafka was chosen to act as an orchestrator that has the knowledge of the services, and to take care of communication between micro-services. The proposed prototype will allow the automation of processing raw data into meaningful information in a scalable, and fault tolerant manner.

## 6.2 . Future Work

As discussed in Chapter 1, the ultimate goal of this thesis work is the ability to extract usable information from raw MCS data. The novel approaches for analyzing enriched trajectory data and better understanding personal exposure proposed in this thesis open many opportunities for future research.

### 6.2.1 . Map-matching Based Enrichment

The first perspective of this thesis work is to extend the proposed enrichment model proposed in Chapter 3 for further enhancements of micro-environment recognition. More specifically, we intend to add other external contextual data that might be of interest such as map-matching with Point of Interest (POIs) and transportation network. Map-matching refers to the procedure of matching trajectory's GPS points with road networks, public transportation system, and POIs within the city. This process will further enhance the performance of the micro-environments recognition model when the detected micro-environments GPS tracks are matched with their exact locations on the map.

Therefore, the expected model will be established on a dimension multiplicity process depicted by the underlying micro-environment (e.g., home, office, restaurant, etc.), contextual data (e.g., POIs), as well as sensors data.

### 6.2.2 . Events Processing

The next important action item in our future work agenda is the incorporation of pollution-related events in the MCS data analysis. Typically, some events where the air quality is a marker (i.e., opening a window, smoking, cooking, etc.) might be of interest to understand personal exposure.

### 6.2.3 . Exposure Profiles

On another perspective, we are interested in mining MCS data in order to define an exposure profile. Particularly, we are interested in answering the following question: *Are there any regularities or patterns (typical profiles) that could allow us to generalize to an unobserved population ?*

Therefore, taking advantage of the multidimensional model proposed in Chapter 4 and OLAP functionalities will allow us to mine participants trajectories and extract meaningful patterns. Typically, we are curious about discovering similar patterns shared by participants in terms of exposure and time spent per micro-environment. This problem falls into the classical sequential data mining. The exposure profiles can then be derived from as clusters of participants that share similar patterns.

### 6.2.4 . Privacy and Participants' Incentives

The human (i.e. crowd) involvement in collecting MCS data naturally brings privacy concerns. People may not be encouraged to volunteer in collecting sensory data, because it may contain private and sensitive information, such as their where-



abouts all the time. Privacy is an important issue for real-life application domains such as MCS. While this thesis work did not cover the privacy and security issues of enriched trajectories, but it will definitely be an important research topic in the future. While participating in MCS, the volunteers engage themselves to meet a strict protocol by carrying the sensors continuously, charging them, sometimes carrying out some maintenance (such as changing the filter every day), self-reporting their activities or filling a diary. These constraints may discourage potential participants. Therefore, incentive procedures are necessary to provide the volunteers with rewards for their participation, and for holding the kit and collecting data for the whole campaign period. The main incentive is their own benefit in terms of insights in their exposure. This has been observed in the Polluscope project, where a personalized report has been produced and distributed to the participants, in combination with a general workshop on the project intermediary results. Another way is to gamify the collection protocol, or to foster participants' interaction via a social network.

## 7 - Bibliography

- [1] M. Abboud, H. E. Hafyani, J. Zuo, K. Zeitouni, and Y. Taher. Micro-environment recognition in the context of environmental crowdsensing. *Proceedings of the Workshops of the EDBT/ICDT 2021 Joint Conference*, 2841, 2021.
- [2] A. Ali and J. K. Aggarwal. Segmentation and recognition of continuous human activity. *Proceedings IEEE Workshop on Detection and Recognition of Events in Video*, pages 28–35, 2001.
- [3] S. Ali, M. A. Jarwar, and I. Chong. Design methodology of microservices to support predictive analytics for iot applications. *Sensors*, 18(12):4226, 2018.
- [4] T. Alsahfi, M. Almotairi, and R. Elmasri. A survey on trajectory data warehouse. *Spatial Information Research*, 28(1):53–66, 2020.
- [5] Ambiciti. <http://ambiciti.io/>, Last accessed May 2022.
- [6] S. Aminikhanghahi and D. J. Cook. A survey of methods for time series change point detection. *Knowledge and Information Systems*, 51:339–367, 2016.
- [7] S. Aminikhanghahi and D. J. Cook. Using change point detection to automate daily activity segmentation. *2017 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, pages 262–267, 2017.
- [8] S. Aminikhanghahi and D. J. Cook. Enhancing activity recognition using cpd-based activity segmentation. *Pervasive and Mobile Computing*, 53:75 – 89, 2019.
- [9] S. Aminikhanghahi, T. Wang, and D. J. Cook. Real-time change point detection with application to smart home time series data. *IEEE Transactions on Knowledge and Data Engineering*, 31:1010–1023, 2019.
- [10] A. Antoniou, A. Storkey, and H. Edwards. Data augmentation generative adversarial networks. *arXiv preprint arXiv:1711.04340*, 2017.
- [11] P. Bellini, M. Benigni, R. Billero, P. Nesi, and N. Rauch. Km4city ontology building vs data harvesting and cleaning for smart-city services. *Journal of Visual Languages & Computing*, 25(6):827–839, 2014.

- [12] D. J. Berndt and J. Clifford. Using dynamic time warping to find patterns in time series. In *KDD workshop*, volume 10, pages 359–370. Seattle, WA, USA:, 1994.
- [13] C. Bettini, X. S. Wang, and S. Jajodia. A general framework for time granularity and its application to temporal reasoning. *Annals of mathematics and artificial intelligence*, 22(1-2), 1998.
- [14] C. Bettini, X. S. Wang, S. Jajodia, and J.-L. Lin. Discovering frequent event patterns with multiple granularities in time sequences. *IEEE Transactions on Knowledge and Data Engineering*, 10(2), 1998.
- [15] C. Bettini, S. Jajodia, and S. Wang. *Time granularities in databases, data mining, and temporal reasoning*. Springer Science & Business Media, 2000.
- [16] S. Bimonte, A. Tchounikine, and M. Miquel. Spatial olap : Open issues and a web based prototype. In *10th AGILE International Conference on Geographic Information Science*, page 11, 2007.
- [17] S. Bimonte, L. Ren, and N. Koueya. A linear programming-based framework for handling missing data in multi-granular data warehouses. *Data & Knowledge Engineering*, page 101832, 2020.
- [18] A. Blázquez-García, A. Conde, U. Mori, and J. A. Lozano. A review on outlier/anomaly detection in time series data. *ACM Computing Surveys (CSUR)*, 54(3):1–33, 2021.
- [19] A. Bonavita, R. Guidotti, and M. Nanni. Individual and collective stop-based adaptive trajectory segmentation. *Geoinformatica*, pages 1–27, 2021.
- [20] J. C. G. Boot, W. Feibes, and J. H. C. Lisman. Further methods of derivation of quarterly figures from annual data. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 16(1):65–75, 1967.
- [21] J. Bournay and G. Laroque. Réflexions sur la méthode d’élaboration des comptes trimestriels. *Annales de l’INSEE*, 36:3–30, 1979.
- [22] L. Breiman. Stacked regressions. *Machine learning*, 24(1):49–64, 1996.
- [23] E. Camossi, M. Bertolotto, and E. Bertino. A multigranular object-oriented framework supporting spatio-temporal granularity conversions. *International Journal of Geographical Information Science*, 20(05), 2006.
- [24] S. Campora, J. A. F. de Macedo, and L. Spinsanti. St-toolkit: A framework for trajectory data warehousing. In *14th AGILE Conference on Geographic Information Science. Utrecht, Holanda*, 2011.

- [25] B. Chaix, Y. Kestens, K. Bean, C. Leal, N. Karusisi, K. Meghrief, J. Burban, M. Fon Sing, C. Perchoux, F. Thomas, et al. Cohort profile: residential and non-residential environments, individual activity spaces and cardiovascular risk factors and diseases—the record cohort study. *International journal of epidemiology*, 41(5):1283–1292, 2012.
- [26] B. Chaix, Y. Kestens, C. Perchoux, N. Karusisi, J. Merlo, and K. Labadi. An interactive mapping tool to assess individual mobility patterns in neighborhood studies. *American journal of preventive medicine*, 43(4):440–450, 2012.
- [27] L. Chatzidiakou, A. Krause, M. Kellaway, Y. Han, Y. Li, E. Martin, F. J. Kelly, T. Zhu, B. Barratt, and R. L. Jones. Automated classification of time-activity-location patterns for improved estimation of personal exposure to air pollution. 2022.
- [28] N. Chawla, K. Bowyer, L. Hall, and W. Kegelmeyer. Smote: Synthetic minority over-sampling technique. *J. Artif. Intell. Res. (JAIR)*, 16:321–357, 06 2002. doi: 10.1613/jair.953.
- [29] K. Chen, D. Zhang, L. Yao, B. Guo, Z. Yu, and Y. Liu. Deep Learning for Sensor-based Human Activity Recognition: Overview, Challenges and Opportunities. *arXiv:2001.07416 [cs]*, Jan. 2020. URL <http://arxiv.org/abs/2001.07416>. arXiv: 2001.07416.
- [30] T. Cheng and Z. Li. A multiscale approach for spatio-temporal outlier detection. *Transactions in GIS*, 10(2):253–263, 2006.
- [31] H. Cho and S. M. Yoon. Divide and conquer-based 1d cnn human activity recognition using test data sharpening. *Sensors*, 18(4):1055, 2018.
- [32] D. Choujaa and N. Dulay. Tracme: Temporal activity recognition using mobile phone data. In *2008 IEEE/IFIP International Conference on Embedded and Ubiquitous Computing*, volume 1, pages 119–126. IEEE, 2008.
- [33] G. Chow and A.-I. Lin. Best linear unbiased interpolation, distribution, and extrapolation of time series by related series. *The Review of Economics and Statistics*, 53(4):372–75, 1971.
- [34] S. Dabiri and K. Heaslip. Inferring transportation modes from gps trajectories using a convolutional neural network. *Transportation research part C: emerging technologies*, 86:360–371, 2018.
- [35] H. Deng, G. Runger, E. Tuv, and M. Vladimir. A time series forest for classification and feature extraction. *Information Sciences*, 239:142–153, 2013.

- [36] N. Dmitry and S.-S. Manfred. On micro-services architecture. *International Journal of Open Information Technologies*, 2(9), 2014.
- [37] T. M. T. Do and D. Gatica-Perez. The Places of Our Lives: Visiting Patterns and Automatic Labeling from Longitudinal Smartphone Data. *IEEE Transactions on Mobile Computing*, 13(3):638–648, Mar. 2014. ISSN 1558-0660. doi: 10.1109/TMC.2013.19.
- [38] N. Dragoni, S. Giallorenzo, A. L. Lafuente, M. Mazzara, F. Montesi, R. Mustafin, and L. Safina. Microservices: Yesterday, Today, and Tomorrow. In M. Mazzara and B. Meyer, editors, *Present and Ulterior Software Engineering*, pages 195–216. Springer International Publishing, Cham, 2017. ISBN 978-3-319-67425-4. doi: 10.1007/978-3-319-67425-4\_12.
- [39] C. L. Eicher and C. A. Brewer. Dasymetric mapping and areal interpolation: Implementation and evaluation. *Cartography and Geographic Information Science*, 28(2):125–138, 2001.
- [40] H. El Hafyani. Big data series analytics in the context of environmental crowd sensing. In *2020 21st IEEE International Conference on Mobile Data Management (MDM)*, pages 246–247. IEEE, 2020.
- [41] H. El Hafyani. Big data series analytics in the context of environmental crowd sensing. In *2020 21st IEEE International Conference on Mobile Data Management (MDM)*, pages 246–247. IEEE, 2020.
- [42] H. El Hafyani, K. Zeitouni, Y. Taher, and M. Abboud. Leveraging change point detection for activity transition mining in the context of environmental crowdsensing. *Actes de la conférence BDA 2020*, 1:64, 2020.
- [43] H. El Hafyani, K. Zeitouni, Y. Taher, and M. Abboud. Leveraging change point detection for activity transition mining in the context of environmental crowdsensing. *The 9th SIGKDD International Workshop on Urban Computing*, 2020.
- [44] H. El Hafyani, M. Abboud, J. Zuo, K. Zeitouni, and Y. Taher. Tell me what air you breath, i tell you where you are. In *17th International Symposium on Spatial and Temporal Databases, SSTD '21*, page 161–165, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450384254. doi: 10.1145/3469830.3470914. URL <https://doi.org/10.1145/3469830.3470914>.
- [45] E. Elnahrawy and B. Nath. Cleaning and querying noisy sensors. In *Proceedings of the 2nd ACM international conference on Wireless sensor networks and applications*, pages 78–87, 2003.

- [46] M. Ester, H. P. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. *In Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining (KDD'96)*, pages 226–231, 1996.
- [47] M. Etemad, A. Soares Júnior, and S. Matwin. Predicting transportation modes of gps trajectories using feature engineering and noise removal. *In Advances in Artificial Intelligence: 31st Canadian Conference on Artificial Intelligence, Canadian AI 2018, Toronto, ON, Canada, May 8–11, 2018, Proceedings 31*, pages 259–264. Springer, 2018.
- [48] H. I. Fawaz, G. Forestier, J. Weber, L. Idoumghar, and P.-A. Muller. Deep learning for time series classification: a review. *Data Mining and Knowledge Discovery*, 33(4):917–963, 2019.
- [49] H. I. Fawaz, B. Lucas, G. Forestier, C. Pelletier, D. Schmidt, J. Weber, G. I. Webb, L. Idoumghar, P.-A. Muller, and F. Petitjean. Inceptiontime: Finding alexnet for time series classification. *ArXiv*, abs/1909.04939, 2020.
- [50] R. Fernández. A methodological note on the estimation of time series. *The Review of Economics and Statistics*, 63(3):471–76, 1981.
- [51] R. K. Ganti, F. Ye, and H. Lei. Mobile crowdsensing: current state and future challenges. *IEEE communications Magazine*, 49(11):32–39, 2011.
- [52] E. Garcia-Ceja, C. E. Galván-Tejada, and R. Brena. Multi-view stacking for activity recognition with sound and accelerometer data. *Information Fusion*, 40:45–56, Mar. 2018. ISSN 1566-2535. doi: 10.1016/j.inffus.2017.06.004. URL <http://www.sciencedirect.com/science/article/pii/S1566253516301932>.
- [53] S. Gharghabi, C.-C. M. Yeh, Y. Ding, W. Ding, P. Hibbing, S. LaMunion, A. Kaplan, S. E. Crouter, and E. Keogh. Domain agnostic online semantic segmentation for multi-dimensional time series. *Data mining and knowledge discovery*, 33(1):96–130, 2019.
- [54] L. Gong, T. Yamamoto, and T. Morikawa. Identification of activity stop locations in gps trajectories by dbscan-te method combined with support vector machines. *Transportation research procedia*, 32:146–154, 2018.
- [55] M. F. Goodchild and N. S. N. Lam. Areal interpolation: A variant of the traditional spatial problem. *Geo-Processing*, 1:97–312., 1980.
- [56] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

- [57] B. Guo, Z. Yu, X. Zhou, and D. Zhang. From participatory sensing to mobile crowd sensing. In *2014 IEEE International Conference on Pervasive Computing and Communication Workshops (PERCOM WORKSHOPS)*, pages 593–598. IEEE, 2014.
- [58] M. Gupta, J. Gao, C. C. Aggarwal, and J. Han. Outlier detection for temporal data: A survey. *IEEE Transactions on Knowledge and data Engineering*, 26(9):2250–2267, 2013.
- [59] F. Gustafsson. *Adaptive filtering and change detection*, volume 1. Citeseer, 2000.
- [60] R. H. Güting and M. Schneider. *Moving objects databases*. Elsevier, 2005.
- [61] R. H. Güting, V. Almeida, D. Ansorge, T. Behr, Z. Ding, T. Hose, F. Hoffmann, M. Spiekermann, and U. Telle. Secondo: An extensible dbms platform for research prototyping and teaching. In *21st International Conference on Data Engineering (ICDE'05)*, pages 1115–1116. IEEE, 2005.
- [62] T. Hägerstrand. What about people in regional science? *Papers of the Regional Science Association*, 24:6–21, 1970.
- [63] M. Hamilton, N. Gonsalves, C. Lee, A. Raman, B. Walsh, S. Prasad, D. Banda, L. Zhang, L. Zhang, and W. T. Freeman. Large-scale intelligent microservices. In *2020 IEEE International Conference on Big Data (Big Data)*, pages 298–309. IEEE, 2020.
- [64] A. Hendawi, J. Shen, S. S. Sabbineni, Y. Song, P. Cao, Z. Zhang, J. Krumm, and M. Ali. Noise patterns in gps trajectories. In *2020 21st IEEE International Conference on Mobile Data Management (MDM)*, pages 178–185. IEEE, 2020.
- [65] S. Host, T. Cardot, A. Saunal, V. Gherzi, and F. Joly. Mortalité attribuable à la pollution atmosphérique en Île-de-france. quelle évolution depuis 10 ans et quels bénéfices d'une amélioration de la qualité de l'air dans les territoires ? *Observatoire régional de santé Île-de-France*, 2022.
- [66] N. Iftikhar and T. B. Pedersen. Schema design alternatives for multi-granular data warehousing. In *DEXA'10*, pages 111–125. Springer, 2010.
- [67] J. Iglesias, J. Cano, A. M. Bernardos, and J. R. Casar. A ubiquitous activity-monitor to prevent sedentariness. In *2011 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, pages 319–321. IEEE, 2011.

- [68] J. Jacobs. 'Dividing by 4': A Feasible Quarterly Forecasting Method? Cite-seer, 1994.
- [69] L. C. Jatoba, U. Grossmann, C. Kunze, J. Ottenbacher, and W. Stork. Context-aware mobile health monitoring: Evaluation of different pattern recognition methods for classification of physical activity. In *2008 30th annual international conference of the IEEE Engineering in Medicine and Biology Society*, pages 5250–5253. IEEE, 2008.
- [70] S. K. Jensen, T. B. Pedersen, and C. Thomsen. Time series management systems: A survey. *IEEE TKDE*, 29(11):2581–2600, 2017.
- [71] W. Jiang and Z. Yin. Human Activity Recognition Using Wearable Sensors by Deep Convolutional Neural Networks. In *Proceedings of the 23rd ACM international conference on Multimedia*, MM '15, pages 1307–1310, New York, NY, USA, Oct. 2015. Association for Computing Machinery. ISBN 978-1-4503-3459-4. doi: 10.1145/2733373.2806333. URL <https://doi.org/10.1145/2733373.2806333>.
- [72] F. Karim, S. Majumdar, H. Darabi, and S. Harford. Multivariate lstm-fcns for time series classification. *Neural Networks*, 116:237–245, 2019.
- [73] C. Koncilia, T. Morzy, R. Wrembel, and J. Eder. Interval olap: Analyzing interval data. In *DaWaK'14*, pages 233–244. Springer, 2014.
- [74] I. Kontopoulos, K. Chatzikokolakis, K. Tserpes, and D. Zisis. Classification of vessel activity in streaming data. In *Proceedings of the 14th ACM International Conference on Distributed and Event-based Systems*, pages 153–164, 2020.
- [75] A. Krylovskiy, M. Jahn, and E. Patti. Designing a smart city internet of things platform with microservice architecture. In *2015 3rd International Conference on Future Internet of Things and Cloud*, pages 25–30. IEEE, 2015.
- [76] B. Languille, V. Gros, N. Bonnaire, C. Pommier, C. Honoré, C. Debert, L. Gauvin, S. Srairi, I. Annesi-Maesano, B. Chaix, and K. Zeitouni. A methodology for the characterization of portable sensors for air quality measure with the goal of deployment in citizen science. *Science of The Total Environment*, 708:134698, 11 2019. doi: 10.1016/j.scitotenv.2019.134698.
- [77] B. Languille, V. Gros, N. Bonnaire, C. Pommier, C. Honoré, C. Debert, L. Gauvin, S. Srairi, I. Annesi-Maesano, B. Chaix, et al. A methodology for the characterization of portable sensors for air quality measure with the goal of deployment in citizen science. *Science of the Total Environment*, 708:134698, 2020.



- [78] O. D. Lara and M. A. Labrador. A survey on human activity recognition using wearable sensors. *IEEE communications surveys & tutorials*, 15(3): 1192–1209, 2012.
- [79] L. Leonardi, G. Marketos, E. Frentzos, N. Giatrakos, S. Orlando, N. Pelekis, A. Raffaetà, A. Roncato, C. Silvestri, and Y. Theodoridis. T-warehouse: Visual olap analysis on trajectory data. In *2010 IEEE 26th international conference on data engineering (ICDE 2010)*, pages 1141–1144. IEEE, 2010.
- [80] L. Leonardi, S. Orlando, A. Raffaetà, A. Roncato, C. Silvestri, G. Andrienko, and N. Andrienko. A general framework for trajectory data warehousing and visual olap. *Geoinformatica*, 18(2):273–312, Apr. 2014. ISSN 1384-6175. doi: 10.1007/s10707-013-0181-3. URL <https://doi.org/10.1007/s10707-013-0181-3>.
- [81] Q. Li, Y. Zheng, X. Xie, Y. Chen, W. Liu, and W.-Y. Ma. Mining user similarity based on location history. In *Proceedings of the 16th ACM SIGSPATIAL international conference on Advances in geographic information systems*, pages 1–10, 2008.
- [82] S. Li, Y. Li, and Y. Fu. Multi-view time series classification: A discriminative bilinear projection approach. *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, pages 989–998, 2016.
- [83] J. Lines, S. Taylor, and A. Bagnall. HIVE-COTE: The Hierarchical Vote Collective of Transformation-based Ensembles for Time Series Classification. In *2016 IEEE 16th international conference on data mining (ICDM)*, pages 1041–1046, 2016.
- [84] R. B. Litterman. A random walk, markov model for the distribution of time series. *Journal of Business & Economic Statistics*, 1(2):169–173, 1983.
- [85] L. Liu, Y. Peng, S. Wang, M. Liu, and Z. Huang. Complex activity recognition using time series pattern dictionary learned from ubiquitous sensors. *Inf. Sci.*, 340-341:41–57, 2016.
- [86] E. Lo, B. Kao, W.-S. Ho, S. D. Lee, C. K. Chui, and D. W. Cheung. Olap on sequence data. In *Proceedings of the 2008 ACM SIGMOD MOD’08*, pages 649–660, 2008.
- [87] J. Lu and R. H. Güting. Parallel secondo: Practical and efficient mobility data processing in the cloud. In *2013 IEEE International Conference on Big Data*, pages 107–25. IEEE, 2013.

- [88] J. D. Mazimpaka and S. Timpf. Trajectory data mining: A review of methods and applications. *Journal of Spatial Information Science*, 2016(13):61–99, 2016.
- [89] H. J. Miller. Tobler’s first law and spatial analysis. *AAG*, 94(2):284–289, 2004.
- [90] H. J. Miller. A measurement theory for time geography. *Geographical analysis*, 37(1):17–45, 2005.
- [91] H. J. Miller. Time geography. In *Handbook of Behavioral and Cognitive Geography*. Edward Elgar Publishing, 2018.
- [92] J. Monteiro, B. Martins, P. Murrieta-Flores, and J. M. Pires. Spatial disaggregation of historical census data leveraging multiple sources of ancillary information. *ISPRS International Journal of Geo-Information*, 8(8):327, 2019.
- [93] B. Moon, H. V. Jagadish, C. Faloutsos, and J. H. Saltz. Analysis of the clustering properties of the hilbert space-filling curve. *IEEE TKDE’01*, 13(1):124–141, 2001.
- [94] G. Nayak, V. Mithal, X. Jia, and V. Kumar. Classifying multivariate time series by learning sequence-level discriminative patterns. In *Proceedings of the 2018 SIAM International Conference on Data Mining*, pages 252–260. SIAM, 2018.
- [95] J. K. Nidzwetzki and R. H. Güting. Distributed secondo: an extensible and scalable database management system. *Distributed and Parallel Databases*, 35(3):197–248, 2017.
- [96] OGC. <http://www.ogc.org/>, Last accessed May 2022.
- [97] OpenRadiation. <https://openradiation.org/>, Last accessed May 2022.
- [98] F. J. Ordóñez and D. Roggen. Deep convolutional and lstm recurrent neural networks for multimodal wearable activity recognition. *Sensors*, 16(1):115, 2016.
- [99] E. S. Page. Continuous inspection schemes. *Biometrika*, 41:100–115, 1954.
- [100] G. Palshikar et al. Simple algorithms for peak detection in time-series. In *Proc. 1st Int. Conf. Advanced Data Analysis, Business Analytics and Intelligence*, volume 122, 2009.

- [101] L. Pappalardo, F. Simini, G. Barlacchi, and R. Pellungrini. scikit-mobility: a python library for the analysis, generation and risk assessment of mobility data, 2019.
- [102] C. Parent, S. Spaccapietra, C. Renso, G. Andrienko, N. Andrienko, V. Bogorny, M. L. Damiani, A. Gkoulalas-Divanis, J. Macedo, N. Pelekis, et al. Semantic trajectories modeling and analysis. *ACM Computing Surveys (CSUR)*, 45(4):1–32, 2013.
- [103] N. Pelekis, E. Frentzos, N. Giatrakos, and Y. Theodoridis. Hermes: A trajectory db engine for mobility-centric applications. *International Journal of Knowledge-Based Organizations (IJKBO)*, 5(2):19–41, 2015.
- [104] J. Pärkkä, M. Ermes, P. Korpipää, J. Mäntyjärvi, J. Peltola, and I. Korhonen. Activity classification using realistic data from wearable sensors. *IEEE transactions on information technology in biomedicine: a publication of the IEEE Engineering in Medicine and Biology Society*, 10(1):119–128, Jan. 2006. ISSN 1089-7771. doi: 10.1109/titb.2005.856863.
- [105] K. Rehl, S. Gröchenig, and S. Kranzinger. Why did a vehicle stop? a methodology for detection and classification of stops in vehicle trajectories. *International Journal of Geographical Information Science*, 34(10):1953–1979, 2020.
- [106] S. Rivest, Y. Bédard, M.-J. Proulx, M. Nadeau, F. Hubert, and J. Pastor. Solap technology: Merging business intelligence with geospatial technology for interactive spatio-temporal exploration and analysis of data. *ISPRS*, 60(1):17 – 33, 2005. ISSN 0924-2716.
- [107] A. P. Ruiz, M. Flynn, and A. Bagnall. Benchmarking Multivariate Time Series Classification Algorithms. *arXiv:2007.13156 [cs, stat]*, July 2020. URL <http://arxiv.org/abs/2007.13156>. arXiv: 2007.13156.
- [108] A. P. Ruiz, M. Flynn, J. Large, M. Middlehurst, and A. Bagnall. The great multivariate time series classification bake off: a review and experimental evaluation of recent algorithmic advances. *Data Mining and Knowledge Discovery*, 35(2):401–449, 2021.
- [109] A. Sadri, Y. Ren, and F. D. Salim. Information gain-based metric for recognizing transitions in human activities. *Pervasive Mob. Comput.*, 38:92–109, 2017.
- [110] C. Sardanios, I. Varlamis, and G. Bouras. Extracting user habits from google maps history logs. In *2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, pages 690–697. IEEE, 2018.

- [111] L. Savary, T. Wan, and K. Zeitouni. Spatio-temporal data warehouse design for human activity pattern analysis. In *Proceedings. DEXA, 2004.*, pages 814–818. IEEE, 2004.
- [112] C. Sax and P. Steiner. tempdisagg: Methods for temporal disaggregation and interpolation of time series. *The R Journal*, 5/2:88–87, 2013.
- [113] P. Smyth and D. Wolpert. Stacked density estimation. In M. I. Jordan, M. J. Kearns, and S. A. Solla, editors, *Advances in Neural Information Processing Systems 10*, pages 668–674. MIT Press, 1998. URL <http://papers.nips.cc/paper/1353-stacked-density-estimation.pdf>.
- [114] J. Snell, K. Swersky, and R. Zemel. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems*, volume 30, pages 4077–4087. Curran Associates, Inc., 2017.
- [115] F. R. Stevens, A. E. Gaughan, C. Linard, and A. J. Tatem. Disaggregating census data for population mapping using random forests with remotely-sensed and ancillary data. *PLOS ONE*, 10(2):1–22, 02 2015.
- [116] R. Tavenard, J. Faouzi, G. Vandewiele, F. Divo, G. Androz, C. Holtz, M. Payne, R. Yurchak, M. Rußwurm, K. Kolar, and E. Woods. Tslearn, a machine learning toolkit for time series data. *Journal of Machine Learning Research*, 21(118):1–6, 2020. URL <http://jmlr.org/papers/v21/20-091.html>.
- [117] W. R. Tobler. Smooth pycnophylactic interpolation for geographical regions. *Journal of the American Statistical Association*, 74(367):519–530, 1979.
- [118] E. Toch, B. Lerner, E. Ben-Zion, and I. Ben-Gal. Analyzing large-scale human mobility data: a survey of machine learning methods and applications. *Knowledge and Information Systems*, 58(3):501–523, 2019.
- [119] A. Vaisman and E. Zimányi. What is spatio-temporal data warehousing? In *International Conference on Data Warehousing and Knowledge Discovery*, pages 9–23. Springer, 2009.
- [120] A. Vaisman and E. Zimányi. Mobility data warehouses. *ISPRS International Journal of Geo-Information*, 8(4):170, 2019.
- [121] J. E. van Engelen and H. Hoos. A survey on semi-supervised learning. *Machine Learning*, 109:373–440, 2019.
- [122] R. Wagner, J. A. F. de Macedo, A. Raffaetà, C. Renso, A. Roncato, and R. Trasarti. Mob-Warehouse: A Semantic Approach for Mobility Analysis

- with a Trajectory Data Warehouse. In J. Parsons and D. Chiu, editors, *Advances in Conceptual Modeling*, pages 127–136. Springer International Publishing, 2014. ISBN 978-3-319-14139-8. doi: 10.1007/978-3-319-14139-8\_15.
- [123] T. Wan. *Modélisation et implémentation de systèmes OLAP pour des objets mobiles*. PhD thesis, Versailles-St Quentin en Yvelines, 2007.
- [124] T. Wan, K. Zeitouni, and X. Meng. An olap system for network-constrained moving objects. In *Proceedings of the 2007 ACM SAC*, pages 13–18, 2007.
- [125] B. Wang, T. Jiang, X. Zhou, B. Ma, F. Zhao, and Y. Wang. Time-Series Classification Based on Fusion Features of Sequence and Visualization. *Applied Sciences*, 10(12):4124, Jan. 2020. doi: 10.3390/app10124124. URL <https://www.mdpi.com/2076-3417/10/12/4124>.
- [126] J. Wang, Y. Chen, S. Hao, X. Peng, and L. Hu. Deep Learning for Sensor-based Activity Recognition: A Survey. *Pattern Recognition Letters*, 119: 3–11, Mar. 2019. ISSN 01678655. doi: 10.1016/j.patrec.2018.02.010. URL <http://arxiv.org/abs/1707.03502>. arXiv: 1707.03502.
- [127] L. Wei and E. Keogh. Semi-supervised time series classification. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '06, pages 748–753, New York, NY, USA, Aug. 2006. Association for Computing Machinery. ISBN 978-1-59593-339-3. doi: 10.1145/1150402.1150498. URL <https://doi.org/10.1145/1150402.1150498>.
- [128] W. W. S. Wei and D. O. Stram. Disaggregation of time series models. *Journal of the Royal Statistical Society: Series B (Methodological)*, 52(3): 453–467, 1990.
- [129] O. Wolfson, B. Xu, S. Chamberlain, and L. Jiang. Moving objects databases: Issues and solutions. In *Proceedings. Tenth International Conference on Scientific and Statistical Database Management (Cat. No. 98TB100243)*, pages 111–122. IEEE, 1998.
- [130] O. Wolfson, P. Sistla, B. Xu, J. Zhou, and S. Chamberlain. Domino: Databases for moving objects tracking. *ACM SIGMOD Record*, 28(2):547–549, 1999.
- [131] D. H. Wolpert. Stacked generalization. *Neural Networks*, 5(2):241 – 259, 1992. ISSN 0893-6080. doi: [https://doi.org/10.1016/S0893-6080\(05\)80023-1](https://doi.org/10.1016/S0893-6080(05)80023-1). URL <http://www.sciencedirect.com/science/article/pii/S0893608005800231>.

- [132] D. H. Wolpert. Stacked generalization. *Neural networks*, 5(2):241–259, 1992.
- [133] Z. Yan, D. Chakraborty, C. Parent, S. Spaccapietra, and K. Aberer. Semitri: a framework for semantic annotation of heterogeneous trajectories. In *Proceedings of the 14th international conference on extending database technology*, pages 259–270, 2011.
- [134] L. Ye and E. Keogh. Time series shapelets: A New Primitive for Data Mining. *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '09*, page 947–956, 2009.
- [135] C.-C. M. Yeh, Y. Zhu, L. Ulanova, N. Begum, Y. Ding, H. A. Dau, D. F. Silva, A. Mueen, and E. Keogh. Matrix profile i: all pairs similarity joins for time series: a unifying view that includes motifs, discords and shapelets. In *2016 IEEE 16th international conference on data mining (ICDM)*, pages 1317–1322. IEEE, 2016.
- [136] J. Yoon, D. Jarrett, and M. van der Schaar. Time-series generative adversarial networks. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alchê-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL <https://proceedings.neurips.cc/paper/2019/file/c9efe5f26cd17ba6216bbe2a7d26d490-Paper.pdf>.
- [137] N. J. Yuan, Y. Zheng, X. Xie, Y. Wang, K. Zheng, and H. Xiong. Discovering urban functional zones using latent activity trajectories. *IEEE Transactions on Knowledge and Data Engineering*, 27(3):712–725, 2014.
- [138] M. Zhang and A. A. Sawchuk. Motion primitive-based human activity recognition using a bag-of-features approach. In *Proceedings of the 2nd ACM SIGHIT International Health Informatics Symposium, IHI '12*, pages 631–640, New York, NY, USA, Jan. 2012. Association for Computing Machinery. ISBN 978-1-4503-0781-9. doi: 10.1145/2110363.2110433. URL <https://doi.org/10.1145/2110363.2110433>.
- [139] X. Zhang, Y. Gao, J. Lin, and C.-T. Lu. TapNet: Multivariate Time Series Classification with Attentional Prototypical Network. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 6845–6852, 2020.
- [140] Y. Zheng. Trajectory data mining: An overview. *ACM Trans. Intell. Syst. Technol.*, 6(3), May 2015. ISSN 2157-6904. doi: 10.1145/2743025. URL <https://doi.org/10.1145/2743025>.

- [141] Y. Zheng. Trajectory data mining: An overview. *ACM Trans. Intell. Syst. Technol.*, 6(3), May 2015. ISSN 2157-6904. doi: 10.1145/2743025. URL <https://doi.org/10.1145/2743025>.
- [142] Y. Zheng, Q. Li, Y. Chen, X. Xie, and W.-Y. Ma. Understanding mobility based on GPS data. In *Proceedings of the 10th international conference on Ubiquitous computing*, pages 312–321. Association for Computing Machinery, New York, NY, USA, Sept. 2008. ISBN 978-1-60558-136-1. URL <https://doi.org/10.1145/1409635.1409677>.
- [143] Y. Zheng, L. Liu, L. Wang, and X. Xie. Learning transportation mode from raw gps data for geographic applications on the web. In *Proceedings of the 17th international conference on World Wide Web*, pages 247–256, 2008.
- [144] Y. Zheng, X. Xie, W.-Y. Ma, et al. Geolife: A collaborative social networking service among user, location and trajectory. *IEEE Data Eng. Bull.*, 33(2): 32–39, 2010.
- [145] Y. Zheng, L. Zhang, Z. Ma, X. Xie, and W.-Y. Ma. Recommending friends and locations based on individual location history. *ACM Transactions on the Web (TWEB)*, 5(1):1–44, 2011.
- [146] Z.-H. Zhou. *Ensemble Methods: Foundations and Algorithms*. CRC press, 2012.
- [147] E. Zimányi, M. Sakr, and A. Lesuisse. Mobilitydb: A mobility database based on postgresql and postgis. *ACM Transactions on Database Systems (TODS)*, 45(4):1–42, 2020.
- [148] J. Zuo, K. Zeitouni, and Y. Taher. Incremental and adaptive feature exploration over time series stream. In *2019 IEEE International Conference on Big Data (Big Data)*, pages 593–602, 2019.
- [149] J. Zuo, K. Zeitouni, and Y. Taher. Exploring interpretable features for large time series with se4tec. In *Proc. EDBT*, pages 606–609, 2019.

## A - Appendix A

In this appendix, we describe the data collection protocol in the Polluscope project, the objective of which is to estimate and analyze the personal exposure to air pollution in the Paris region, as well as its health effects, using individual sensors measuring the concentrations of several atmospheric pollutants [77]. During three campaigns, more than one hundred participants have been recruited to collect environmental measurements along with geo-location for one week, 24 hours a day, while performing their daily activities.

### A.1 . Data Collection Campaigns

The sensors used for the collection of quantitative data are identical, geo-located and deployed by a total of 103 participants. These voluntary individuals participated in the two RECORD (Residential Environment and CORonary heart Disease) cohorts and the VGP (Versailles Grand Parc) cohort. Table A.1 presents the general characteristics of the three campaigns, i.e. VGP, RECORD1 and RECORD2. More specifically, 27 participants from the RECORD1 cohort, 63 participants from the VGP cohort and 13 participants from RECORD2 cohort wore identical sensors continuously for 7 days over three different seasons of 2019 and 2020. It is the summer season (June-September) for the RECORD1 cohort, the fall-winter season (October-December) for the VGP cohort and the winter-spring season (January-March) for the RECORD2 cohort.

In addition to this temporal dimension, the spatial distribution, particularly with regard to the place of residence, distinguishes the three cohorts. As shown in Figure A.1, the RECORD cohorts are mainly concentrated in Paris and its inner suburbs, while the VGP cohort is spread over the Community of Versailles Grand Parc, in the outer suburbs, with a significant part located in Versailles.

Table A.1: General characteristics of the two campaigns VGP and RECORD.

Campaign	Number of Participants	Measurement period	Sensor's wearing time	Used Sensors
RECORD1	27	June - September 2019	7 days	
VGP	15	October 2019	7 days	Canarins for PMs, AE51 for BC and Cairsens for NO2
	12	November 2019		
	09	November 2019		
	15	December 2019		
	12	December 2019		
RECORD2	13	January - March 2020	7 days	



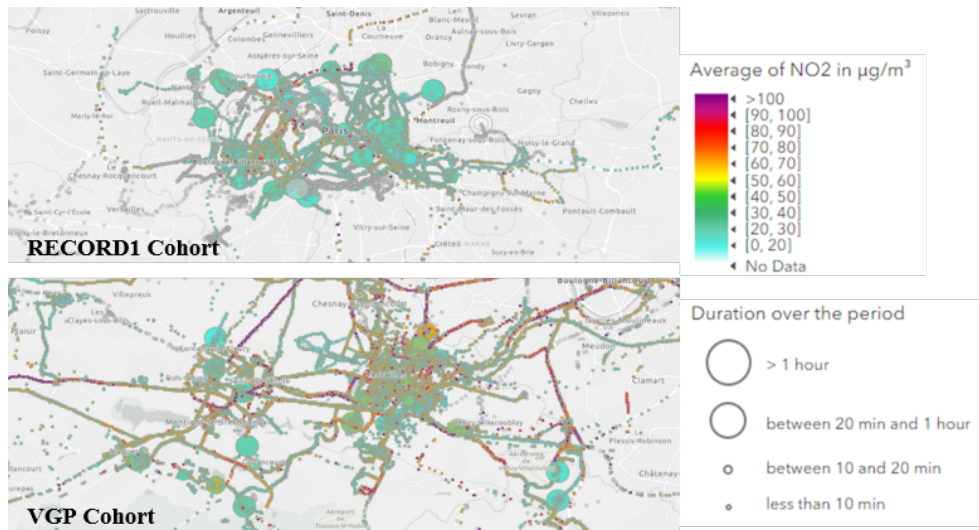


Figure A.1: Illustration of the spatial distribution of participants in the two cohorts according to their place of residence.

## A.2 . Data Collection

A kit comprising three sensors plus a tablet was carried by the participants in a backpack during the cohorts. The deployed sensors operate with a time step of one minute. The tablet collects GPS data at a frequency of 1 to 30 seconds. Fifteen kits have been prepared for deployment by rotation in several waves of one week each, separated by a week of verification and qualification. Thus, the collection was spread over ten to twelve weeks for each of the cohorts.

The sensors allow to estimate in real time the exposure to NO<sub>2</sub>, PM<sub>1.0</sub>, PM<sub>2.5</sub>, PM<sub>10</sub> and Black Carbon (BC) pollutants. They also measure the temperature and relative humidity in real time. The tablet equipped with a GPS allows participants to annotate on a mobile application the change of their micro-environments, which is also called space-time budget. More specifically, the participant indicates any change in micro-environment by selecting the new one from among several categories of frequented places (home, work, transport, other), means of travel (car, metro, train, tram, bus, motorbike, bicycle, walking), but also between various activities carried out (sport, rest, walking, dog walking, catering, cinema, shopping, etc.). In addition, this mobile application allows participants to select certain events or activities that may impact pollutants' concentrations (cooking, opening or closing a window, lighting the fireplace, smoking, walking, etc.).

A new campaign will be conducted in VGP. Part of the participants are from the previous cohort, which will allow to compare the seasonal effect. Besides, a new sensor has been introduced for particulate matters. Indeed, among the objectives of Polluscope was the evaluation of environmental sensors as the technology advances, and the selection of the most adapted solutions for the campaigns. The adopted

solution is PMSCAN (also called AirDIAMS in the DIAMS project<sup>1</sup>).

---

<sup>1</sup><https://www.airdiums.eu/tutoriel-microcapteurs-airdiums>



## B - Appendix B

### B.1 . Longitudinal Analysis

**Example B.1.1.** What is the individual exposure to the PM2.5 pollutant and the maximum exposure periods ?

```
1 SELECT M.user_id,day(T.time) AS Day,hour(T.time) AS Hour ,
   avg(M.measurement_value) AS Exposure , max(M.
   measurement_value) AS Peak_value
2 FROM measurement M, measurement_type MT, time T
3 WHERE MT.measurement_type_id = M.measurement_type_id AND T
   .time_id=M.time_id AND MT.measurement_name='PM2.5'
4 GROUP BY M.user_id, day(T.time), hour(T.time) WITH ROLLUP
```

We imply by exposure the average concentration of pollutants. The duration of exposure makes it possible to generate the received dose. Thus, the query of Example B.1.1 returns the individual exposure to PM2.5 over time which is illustrated in Figure B.1. The query output shows the individual exposure to PM2.5 from user\_id drilling down to the Day, drilling down to the Hour. The aggregates at the participant level are denoted by null in the Hour attribute, and all participants combined aggregates are denoted by null in the user\_id attribute.

user_id	Day	Hour	Exposure	Peak_value
null	null	null	22.0	5010.0
99999M	null	null	143.0	5010.0
99999M	30	null	7.0	455.0
99999M	30	23	5.0	6.0
99999M	30	22	4.0	6.0
99999M	30	21	3.0	6.0
99999M	30	20	2.0	3.0
99999M	30	19	3.0	4.0
99999M	30	18	2.0	7.0
99999M	30	17	3.0	12.0

Figure B.1: Longitudinal analysis over time hierarchy.

### B.2 . Spatial Analysis

As shown in Example B.2.1, the spatial indexing representation allows to answer queries such as : *What is the level of pollution of frequently visited locations by participants at different levels of the spatial hierarchy?* The P64 and P16 columns indicate coarse levels of hierarchy where each pixel contains respectively a grouping

of 64 and 16 finer pixels. The query creates a subtotal of the hierarchy levels of P64 drilling down to P16. This is equivalent to computing the aggregates for the following grouping sets: (P64, P16), (P64) and (i.e. all). The query returns also the exposure to PM2.5 at coarse levels of hierarchy (i.e. 16 and 64 pixels). An excerpt of the query's output is displayed in Figure B.2 which shows the frequency of visiting the coarse levels of hierarchy as well as the exposure level to PM2.5 in these levels. Thus, in order to get the most visited places, an ORDER BY Frequency clause at the end of the query is sufficient.

**Example B.2.1.** What is the level of pollution at the different levels of the spatial hierarchy ?

```

1 SELECT FLOOR(M.location_id/64) AS P64, FLOOR(M.location_id
   /16) AS P16, count(*) AS Frequency, avg(M.
   measurement_value) AS Level
2 FROM measurement M, measurement_type MT
3 WHERE MT.measurement_type_id = M.measurement_type_id AND
   MT.measurement_name='PM2.5'
4 GROUP BY FLOOR(M.location_id/64), FLOOR(M.location_id/16)
   WITH ROLLUP

```

P64	P16	Frequency	Exposure
null	null	532545	14.0
129017	null	36	8.0
129017	516071	28	8.0
129017	516068	8	6.0
129016	null	783	5.0
129016	516067	27	4.0
129016	516066	52	4.0
129016	516065	45	7.0
129016	516064	659	5.0
127807	null	542	5.0

Figure B.2: Exposure at coarse levels of the spatial hierarchy.

## C - Appendix C

Type	Features	Description
Traffic	Point	Turning circle Parking Mini roundabout Crossing Traffic signals Fuel
Transport	Point	Tram stop Bus stop Railways halt Bus station Taxi Railways station
Landuse	Polygon	Forest Park
Railways	Line	Subway rail
Roads	Line	motorway trunk secondary tertiary primary Max speed = 30km/h Max speed = 50km/h Max speed = 70km/h Max speed = 100km/h Max speed = 130km/h

Table C.1: Description of the used covariates for the spatial disaggregation of NO<sub>2</sub> values.

Pollutant	Min	Max	Mean	Standard Deviation
NO <sub>2</sub>	0	52	24.06	11.4

Table C.2: A statistical summary of NO<sub>2</sub> values.



## D - Appendix D

### Résumé en Français

La surveillance et la mesure de la qualité de l'air constituent un enjeu actuel majeur pour les politiques urbaines visant à lutter contre les pollutions atmosphériques et à mettre en œuvre des actions d'adaptation. En effet, la mauvaise qualité de l'air est l'un des principaux facteurs de risque pour la santé humaine. Elle est responsable de près d'un décès sur dix en Île-de-France en 2019 et d'environ 7 millions de décès prématurés chaque année dans le monde d'après l'OMS. Par conséquent, la qualité de l'air doit être surveillée afin de réduire le développement de pathologies chroniques graves liées à l'exposition à la pollution de l'air, qui se traduisent par une augmentation de la mortalité, une diminution de l'espérance de vie et un recours accru aux soins.

La collecte participative - Mobile Crowd Sensing (MCS) en anglais - constitue un nouveau paradigme basé sur la technologie émergente des micro-capteurs connectés. Elle offre la possibilité de mesurer l'exposition individuelle à la pollution de l'air n'importe où et n'importe quand. Elle offre une opportunité unique pour estimer et analyser l'exposition individuelle selon les habitudes de vie, d'activité et de déplacement propres à chacun et qui est mal connue jusque-là. La particularité de ce paradigme de collecte est la combinaison de la localisation spatiale avec des mesures et des annotations continues dans le temps. Dans le contexte de cette thèse, des participants ont été recrutés et équipés d'un kit de capteurs et d'un appareil mobile pour collecter des mesures de la qualité de l'air telles que les particules fines de différentes tailles, le dioxyde d'azote, le carbone suie, la température et l'humidité. L'appareil mobile est utilisé pour collecter les traces GPS. De plus, une application mobile est mise à la disposition des participants afin qu'ils puissent indiquer le type de lieu (appelé micro-environnement) tel que le domicile, le bureau, le métro, le bus, etc., et ce à chaque fois qu'ils en changent. Cela amène à générer en continu des séries de données géo-localisées, qui finissent par former une grande masse de données. Celle-ci constitue une mine d'information pour des analyses variées et une opportunité unique d'extraction de connaissances sur l'exposition, sa variation temporelle, spatiale, mais aussi la caractérisation de la qualité de l'air par micro-environnement et de la fréquence et durée de leur fréquentation.

Par ailleurs, pour interpréter l'exposition individuelle des participants à la pollution, les données de capteurs doivent être contextualisées non seulement par la localisation, mais aussi par le micro-environnement où elles ont été mesurées. Sans ces informations, les mesures collectées sont difficilement exploitables pour analyser et comprendre l'exposition individuelle et les risques engendrés.

Toutefois, cette analyse est loin d'être simple, car il y a un gap important



entre les séries de données brutes des capteurs et les informations exploitables. En effet, les données brutes sont imparfaites. Elles sont souvent bruitées, comportent des anomalies de mesures et parfois des pertes de données, ce qui nécessite un nettoyage et un prétraitement minutieux. De telles imperfections de données, qui affectent à la fois les données de séries chronologiques et la géolocalisation, doivent être prises en compte dans le processus de traitement et d'analyse. En outre, les annotations des changements de micro-environnement sont elles aussi imparfaites et incomplètes, ce qui constitue un autre défi. Par exemple, certains participants indiquent qu'ils sont dans leur bureau à 3h du matin ou allument une cheminée dans une rue, d'autres ignorent cette tâche d'annotation. Il y a donc un grand intérêt à détecter automatiquement le contexte des participants pour compenser les annotations manquantes. Par ailleurs, une telle automatisation permettrait d'alléger le protocole de collecte pour les futurs participants en se passant de leur renseignement systématique.

Une autre caractéristique des données collectées par campagnes de MCS est que la couverture spatiale est très irrégulière. Certains endroits sont couverts par une forte densité spatiale, tandis que d'autres n'ont aucune mesure. A contrario, les réseaux de stations fixes de surveillance réglementaire de la qualité de l'air fournissent un modèle de couverture spatio-temporelle complet pour la zone d'étude. Cependant, la résolution spatio-temporelle de ces stations fixes est très différente des données de capteurs mobiles. Fusionner et comparer ces deux sources de données, de résolution spatio-temporelle et de couverture territoriale si différentes, est difficile.

Le défi majeur que cette thèse cherche à relever est de combler le gap entre les séries de données brutes des capteurs et les informations exploitables en proposant une approche holistique d'analyse et d'extraction de connaissance des données collectées dans le contexte du MCS. Plus précisément, nous mettons en œuvre un processus analytique de bout en bout qui comprend le prétraitement des données, leur enrichissement avec des informations contextuelles, la modélisation et le stockage de ces données, ainsi que la mise en œuvre d'un tableau de bord interactif pour la visualisation des données en temps réel. Nous avons implémenté notre proposition en veillant à automatiser son déploiement. Les approches proposées sont appliquées sur des données réelles collectées au sein du projet ANR Polluscope.