University of Udine
Dept. of Mathematics, Computer Science, and Physics
PhD Course in Computer Science, Mathematics and Physics

Ph.D. Thesis

Fingerprint-based Indoor Positioning and Beyond: Modeling, Representation, and Learning

Candidate:
Nicola Saccomanno

Supervisor:
Prof. Angelo Montanari

Co-Supervisors:
Andrea Brunello, PhD
Andrea Dalla Torre, PhD

Year 2023

Author's e-mail: saccomanno.nicola@spes.uniud.it
nicola.saccomanno@uniud.it

Author's address:

Dipartimento di Scienze Matematiche, Informatiche e Fisiche
Università degli Studi di Udine
Via delle Scienze, 206
33100 Udine
Italia

To those who have always been there,
to those who are no longer there,
to those who will be there.

Abstract

Data science is composed of two distinct keywords, i.e., *data* and *science*. Hence, the present work lies within a discipline that addresses concrete problems (*data*) in a scientific manner, to extract new knowledge in terms of explanations and predictions. The concrete problem at this dissertation's heart is fingerprint-based positioning, particularly indoors. The work starts by addressing the joint *modelling* of indoor positioning and building topology. The goal is to provide a flexible framework, actualized in a relational database, that goes beyond the current manner of dealing with indoor positioning data, so as to meet the needs of both researchers and practitioners. Semantically linking positioning and building topology fosters reproducibility, the exploitation of shared information, and the development of new evaluation strategies. Motivated by the desire to study fingerprinting comprehensively, the thesis then focuses on the relationship between the vector space of fingerprints and the real-world 2D/3D space. Although a substantial body of literature studied metrics with respect to their positioning performance, almost none investigated their ability to capture, in the space of fingerprints, the spatial relationships between the associated positions: a foundational aspect useful to better understand the precise nature of positioning. Along the dissertation, it is shown how classical metrics used in positioning fail, to a large extent, to capture spatial information, even if some are better than others. Motivated by such findings and relying on machine learning, a new metric is developed, optimised to capture spatial relationships. As a by-product, the metric improves positioning performances, highlighting a link between the two tasks and laying the foundation for the development of a new family of fingerprinting approaches. Afterwards, the work focuses on exploiting deep learning to reduce the impact of some issues affecting fingerprinting. The cornerstone idea is to employ a ranking-based *representation* of the fingerprints paired with recurrent neural networks. Beyond providing several advantages, the approach supports a concept of interpretability that is useful to obtain new scientific and operational insights. The last part of the work lays the first steps towards the long-term research direction of achieving seamless indoor-outdoor positioning, concerning both data modelling and algorithmic solutions. First, a system dealing with the spatio-temporal information pertaining to cellular network-based outdoor crowdsourced fingerprints is proposed. Next, the focus turns to the topic of trajectories, i.e., sequences of positions/fingerprints, with the goal of understanding how different representations behave.

Contents

List of Figures

List of Tables

Introduction

Data science is a popular expression nowadays, composed of two distinct keywords, i.e., *data* and *science*. Far from being just a buzzword, it is a discipline that addresses concrete problems (*data*) in a scientific manner to extract new knowledge in terms of explanations and predictions [143].

Such a twofold nature of data science will be the main thread of this thesis, which is focused on the problem of *positioning*, i.e., identifying the location of something or someone within a given reference system; although such a task can be carried out both in an outdoor and an indoor setting, we mainly focus on the latter. Notably, people spend, on average, 90% of their time indoors in a variety of contexts, such as offices, shopping malls, airports, hospitals, and museums [70]. In all of them, the ability to accurately determine the location of a target device can significantly enhance the user experience, provide valuable data for businesses, and improve safety and security in emergency situations [170, 172]. Besides, location-based information and services are also seen as essential components in the development of the Internet of Things (IoT) and smart cities.

Over the years, multiple approaches have been proposed to tackle the problem of positioning, each with its pros and cons [177]. In this dissertation, we will focus on *fingerprinting*. Fingerprint-based positioning relies on the collection, at different locations, of radio-frequency signal measurements, called fingerprints, to infer the position of a device. Compared to other solutions, its general advantages are cost-effectiveness, wide availability, flexibility, scalability, and limited power consumption [103, 121].

The fingerprinting community has largely been focusing on proposing algorithms for position estimation, trying to overcome the challenges that naturally emerge in indoor environments, such as signal attenuation, multi-path propagation and changes in available radio emitters [262]. In this thesis, we take a different, more holistic approach. We investigate several challenges that arise in the domain, exploring the potential of the principles and methods typical of data science, such as data modeling, analysis, and machine learning.

In the most natural way, we begin our journey from the *data*, addressing the problem of *modeling* indoor positioning. Here, the lack of standards, the high heterogeneity of collected measurements, and the inherent variety of indoor premises are prominent challenges towards the development of viable solutions. Such difficulties led, in the literature, to an extensive research effort that resulted in the proposal of multiple algorithms, sampling strategies, benchmark datasets, and building modeling approaches [81, 150, 221, 256, 285]. While, on the one hand, this contributed towards the advancement of the field, on the other, it hindered evaluation, compa-

rability and interoperability. To cope with the aforementioned issues, we present a comprehensive and extensible framework that relies on a relational database to represent topological information of indoor premises, which can be seamlessly combined with fingerprint positioning data. As we will see, the flexibility of the system allows it to handle multiple indoor scenarios and support a wide range of tasks, both in terms of its industrial deployment and its use within the research community.

We then move towards *science*. One of the concepts at the root of fingerprinting is that of *metric*, since measuring the distances among fingerprints is at the base of all the deterministic positioning approaches [12]. Nevertheless, while a large number of studies examined, in different settings, distance functions in terms of their positioning performance, only a few works studied them per se, abstracting from the positioning task [261, 262]. Thus, we focus on the latter aspect, investigating the ability of classical metrics to preserve spatial information in the vector space of fingerprints. In other words, we ask ourselves whether there is a relationship between the distances measured in the vector space of fingerprints and those measured between the real-world locations that the fingerprints represent. Needless to say, being able to reason spatially based only on fingerprints would be a key element, for instance, in developing new families of algorithms, e.g., based on semi-supervised learning. We address this matter focusing on WiFi fingerprints due to the pervasiveness of such a technology, performing an extensive analysis involving multiple datasets, normalization functions, metrics, and granularity levels. The obtained results will then lead us towards the development of a new metric relying on machine *learning*.

As previously mentioned, numerous challenges affect indoor positioning based on radio signals, such as WiFi. Indeed, signal attenuation, multi-path propagation, changes in the environment, and the heterogeneity of the sensing equipment, are all aspects affecting what a device will observe in a given place at a given moment in time. A line of research pursued in the literature consists of relying on different *representations* of the fingerprints [54, 284], capable of fostering properties like, for instance, robustness to noise. However, a typical downside of such representations is that they convey a reduced amount of information, leading to sub-optimal positioning performance. This is precisely the case with ranking-based fingerprinting [164], where properties such as invariance to scale and bias are exploited to achieve a higher resistance to signal perturbations at the expense of localization accuracy [54, 136, 155, 254]. To limit the shortcomings of such a fingerprint representation, while retaining its advantages, we employ deep learning, specifically recurrent models. In this way, thanks to the inductive bias and the autoregressive nature provided by such types of models, we exploit the inherent features of both ranked fingerprinting and the hierarchical compositionality of indoor buildings. On top of that, motivated by the fact that interpretability in machine learning can also be used to extract novel scientific knowledge and operational insights about the application domain [191], we investigate whether this also holds in our context. This is by no means an easy task to accomplish, as interpretability has not been explicitly de-

fined in indoor positioning, and explanations cannot be extracted relying on just plug-and-play approaches. Rather, they require a careful and tailored investigation.

The last part of the thesis moves towards generalizing what we accomplished for the indoor positioning realm. A unique challenge in the field of positioning is to develop a system capable of dealing with seamless indoor-outdoor information [166]. Again, we focus on fingerprinting, as it constitutes, in the outdoor setting, a valid alternative to the Global Navigation Satellite System [269]: despite its overall reduced precision, it requires a significantly lower amount of energy, it is less expensive, it is not strongly affected by multi-path effects or urban canyons, and it is more widely available throughout the day of a typical user [152, 153, 162, 304]. Nevertheless, outdoor fingerprinting often heavily relies on a comprehensive and accurate knowledge of the cellular network configuration, which is not static and, above everything, is not disclosed by network operators [266]. In this respect, we show how employing a spatio-temporal database derived from appropriate modeling of the domain it is possible to reconstruct the configuration of the cellular network, relying exclusively on crowdsourced data collected by standard users, while providing support to cope with the evolution of the cellular network through space and time, also in case of reconfiguration phenomena. Another topic (among many) that we identify as fundamental to achieving the long-term research goal of a seamless indoor-outdoor positioning system is that of trajectories, i.e., sequences of points [172]. In fact, trajectories can play a pivotal role in managing the transition of users and devices between indoor and outdoor environments, which is one of the most critical aspects regarding the integration sought. Despite its importance, no prior study regarding fingerprinting specifically dealt with this theme, characterized by non-trivial challenges related to low-sampling rates and heterogeneity. We, therefore, investigate which is the best trajectory representation for our domain, paying particular attention to understanding which type of data, between spatial and cellular, and their combinations, to exploit in order to have meaningful similarity evaluations.

The thesis is structured as follows. In Chapter 1, we give an overview of the concept that we use throughout the dissertation, providing a general introduction to data science and machine learning, and going into the details of indoor positioning, with particular attention to WiFi-based fingerprinting. In Chapter 2, we discuss our framework that jointly models indoor positioning and building topology. The following part of the thesis consists of Chapter 3 and Chapter 4, where we study the relationship between distances measured in the fingerprint vector space and those among the location that they represent. First, we will focus on understating the behaviour of classical metrics, and then we will investigate how to learn a better metric tailored for fingerprinting. Chapter 5 describes our work regarding combining deep learning with ranked fingerprinting to obtain an accurate system robust to noise. Then, we show how to extend this latter approach to support a concept of interpretability that is useful to get new scientific and operational insights in Chapter 6. With the goal of seamless indoor-outdoor localisation, the last part of the work is devoted to outdoor fingerprinting. First, in Chapter 7, we discuss a

general and flexible yet complete database schema for cellular networks, modelled after the information available in signal fingerprints, and capable of fostering the crowdsourced collection of data. Next, we delve into the topic of trajectories and their similarity evaluation in Chapter 8. Finally, we provide an overall assessment of the work done and future research directions.

I.1 Publications

We hereby provide the references to the articles on which the results of this thesis have been published.

- The framework that jointly models indoor positioning and building topology (Chapter 2) has been presented in [39], *Andrea Brunello, Angelo Montanari, and Nicola Saccomanno.* ***A framework for indoor positioning including building topology****. IEEE Access, October 2022.*

- Results about our line of research concerning understanding the relationship between the vector space of fingerprints and the real world (Chapter 3), as well as the learning of a meta-metric through genetic programming (Chapter 4), have been published respectively in [224], *Nicola Saccomanno, Andrea Brunello, and Angelo Montanari.* ***What You Sense is Not Where You are: on the Relationships Between Fingerprints and Spatial Knowledge in Indoor Positioning****. IEEE Sensors Journal, March 2022*, and [40], *Andrea Brunello, Angelo Montanari, and Nicola Saccomanno.* ***A genetic programming approach to wifi fingerprint meta-distance learning****. Pervasive and Mobile Computing, August 2022.*

- The work about combining recurrent neural networks with ranked fingerprinting (Chapter 5) has been published in [223], *Nicola Saccomanno, Andrea Brunello, and Angelo Montanari.* ***Let's Forget About Exact Signal Strength: Indoor Positioning based on Access Point Ranking and Recurrent Neural Networks****. In Proceedings of the 17th EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services (MobiQuitous). ACM, December 2020*, while its extension regarding interpretability (Chapter 6) is currently under review.

- The database for outdoor fingerprinting (Chapter 7) has been presented in [36], *Andrea Brunello, Andrea Dalla Torre, Paolo Gallo, Donatella Gubiani, Angelo Montanari, and Nicola Saccomanno.* ***Crowdsourced reconstruction of cellular networks to serve outdoor positioning: Modeling, validation and analysis****. Sensors, January 2023.* In this dissertation, we are not going to provide a complete description of such a work. Instead, we outline the resulting model and its capabilities, mainly for what concerns the

temporal aspect. Such a choice stems from the fact that the publication in [36] builds on previous work by Gubiani et al. [100] (analysis of the context and outline of the framework), Viel, in his PhD dissertation (background, first modeling, and preliminary analysis) [265], and Andreussi, in his master thesis (contribution to the implementation of the revised and spatio-temporal extended database) [9], while our main contributions pertain to the refactoring of the entire model, its extension with the temporal dimension, and the subsequent analyses.

- Finally, the work about trajectories for outdoor fingerprinting (Chapter 8) has been published in [93], *Paolo Gallo, Donatella Gubiani, Angelo Montanari, and Nicola Saccomanno.* ***A new similarity measure for low-sampling cellular fingerprint trajectories****. In Proceedings of the 21st IEEE International Conference on Mobile Data Management (MDM), June 2020.*

In addition to these papers, throughout the PhD journey, we also worked in other directions related to applied artificial intelligence and data science, particularly in the domain of medicine and biology, leading to the following publications:

- [26] *Andrea Bernardini, Andrea Brunello, Gian Luigi Gigli, Angelo Montanari, and Nicola Saccomanno.* ***AIOSA: an approach to the automatic identification of obstructive sleep apnea events based on deep learning****. Artificial Intelligence in Medicine, July 2021.*

- [35] *Andrea Brunello, Marcello Civilini, Stefano De Martin, Antonella Felice, Marinella Franchi, Lucilla Iacumin, Nicola Saccomanno, and Nicola Vitacolonna.* ***Machine learning-assisted environmental surveillance of legionella: a retrospective observational study in friuli-venezia giulia region of italy in the period 2002–2019****. Informatics in Medicine Unlocked, January 2022.*

- [27] *Andrea Bernardini, Andrea Brunello, Gian Luigi Gigli, Angelo Montanari, and Nicola Saccomanno.* ***OSASUD: A dataset of stroke unit recordings for the detection of obstructive sleep apnea syndrome****. Scientific Data, April 2022.*

1
Background

1.1 Data science and machine learning

Machine learning is a subfield of artificial intelligence (AI) that involves the development and study of algorithms that can learn from and make predictions or decisions based on data. These algorithms are trained on a dataset, i.e., a collection of data, and can fulfil their task without being explicitly programmed. Machine learning is a powerful tool for analyzing and understanding complex data, and it has a wide range of applications in several fields such as finance, healthcare, e-commerce, and engineering.

Data science, on the other hand, is the use of scientific and mathematical techniques to extract insights and knowledge from data. It is a multidisciplinary field that combines elements of computer science, statistics and domain expertise. Data science is concerned with the harvesting, processing and analysis of large datasets and the employment of these insights to inform decision-making or solve real-world problems. Often, to achieve such goals, it involves the use of machine learning algorithms as well as other techniques.

Thus, data science and machine learning are strongly related fields, often providing their best when used in combination. Indeed, from a certain perspective, machine learning can be seen as a part of both data science and artificial intelligence, as data science encompasses a range of techniques for analyzing and understanding data, including machine learning.

In the remainder of this section, we give an overview of those topics, providing a general context for this dissertation. As the areas of machine learning, data science, and deep learning are nowadays quite broad, in the chapters that make use of specific techniques (e.g., genetic algorithms, or recurrent neural networks), a detailed account of the involved methods will be reported.

1.1.1 Data science

Data science is a relatively new field that has emerged in the last few decades, driven by the explosion of data availability and advances in computing power. It is an interdisciplinary area that uses scientific methods, processes, algorithms and systems

to extract knowledge and insights from structured (highly organized, e.g., tables) and unstructured data (no pre-defined organization e.g., text-heavy). Its origins can be traced back to the field of statistics, which has long been used to analyze and interpret data. However, the growth of the internet and the proliferation of electronic devices have led to a vast increase in the amount of data being generated and collected, making traditional statistical methods less suitable for the task at hand. As a result, data science has evolved to incorporate a wide range of tools and techniques from other disciplines such as computer science, machine learning, and data engineering, to analyze and make sense of these data, identifying patterns and trends, ultimately making predictions and recommendations. Besides finding application in several industries, its contribution is paramount in scientific research, where it fosters discoveries and advances in multiple domains.

Data science is a multi-centric field being, as we said, a combination of statistical analysis, programming, and domain expertise. Indeed, to deal with large and complex datasets, it is necessary to employ a variety of tools and techniques to clean, organize, and analyze information. For instance, a key step is exploratory data analysis, which requires exploring and visualising the data to better understand its structure and content, leading to the creation of charts, plots, and other figures to highlight possible underlying patterns and trends. This preliminary step is of great importance to guide all the following processing and analyses. Another task is data cleaning and preparation, where raw data, which is often messy and not organized, is pre-processed in order to make it suitable for subsequent analyses. Typical scenarios include handling missing values, fixing errors, and transforming the data into a usable format. Once data have been pre-processed, it is possible to analyze them. As previously mentioned, this is done by relying on multiple techniques, such as, for instance, hypothesis testing, building models, and running experiments. Besides taking actions following the knowledge derived through the analyses, data can also be used to make predictions, which requires developing appropriate inference models for the task at hand.

1.1.2 Machine learning

Machine learning is a sub-field of artificial intelligence about the design, development and usage of algorithms and models that can learn from data. The goal is to be able to make predictions and take decisions based on new data, without relying on systems explicitly programmed to fulfil such tasks. Everything revolves around data, as indeed the cornerstone principle is that systems can learn from them, identify patterns, and take action, all with minimal human intervention. But what is meant by learning? According to [181] *"A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E"*.

The history of modern machine learning can be traced back to the 1950s, when researchers began exploring the use of computers to perform tasks that normally

require human intelligence, such as learning, decision-making, and problem-solving. One of the early pioneers in the field was Arthur Samuel, who developed a program that could play a game of checkers and improve its performance over time by learning from its mistakes [228]. In the following decades, researchers developed novel algorithms for machine learning, such as decision trees, nearest neighbours, and artificial neural networks. These early approaches were limited in their capabilities, but they laid the foundation for more advanced algorithms that were later developed. In the 1980s and 1990s, the field experienced a resurgence of interest, as researchers began to explore it for tasks such as speech recognition, natural language processing, and image classification. In the 21st century, machine learning has become pervasive and ubiquitous and is now used in a wide range of applications, including search engines, self-driving cars, and conversational agents. Indeed, machine learning has undergone significant evolution and has become an essential tool in several fields, leading to the development of new technologies, such as deep learning, which have further transformed the discipline and enabled even greater insights to be gained from data.

There are several different types of machine learning paradigms, which mainly differ one from another by the level of supervision they require.

Supervised learning requires the highest level of supervision, as the algorithm is trained on a labelled dataset, where for each example in the training set, the correct output (i.e., label) is provided. The goal is to make predictions on new, unseen examples that are drawn from the same distribution as the training set. Such new examples are often referred to as test set. The kind of label also defines the type of task, leading to several types of supervised learning, including classification and regression. Classification involves predicting a categorical label, such as whether an email is spam or not. Regression concerns predicting a continuous numerical value, such as the price of a house. Concerning the algorithms, there are many possibilities. Linear regression is used to solve regression tasks by finding the best coefficients for the features appearing in the linear model, fitting the input data. A widely used technique for estimating such parameters of a linear regression model is the least squares method. A generalization of the linear regression model for binary outcomes, thus, a method suitable for classification tasks, is the logistic regression [25, 62], which linearly combines the input features, fitting a logistic function. K-nearest neighbors algorithm (k-NN) is a non-parametric method based on relying on the k closest elements of the training set for a given input instance, considering an appropriate metric/distance function (e.g., Euclidean distance, cosine similarity) [82]. It can be used for both classification and regression tasks: in the first case the outcome is determined using a (weighted) voting mechanism among the k neighbors, assigning the most common class; in the second case, a (weighted) average of the labels of the k neighbors is evaluated. Support Vector Machines work by finding the hyperplane (or a set of hyperplanes) in a high-dimensional space that maximally separates the nearest (w.r.t. to the hyperplane) training-data point of any class [61]. Decision trees are another type of model that can be used for both classification

and regression tasks [213]. The general idea is to use nodes to represent simple decision rules inferred from the data features. Among the most appreciable features of decision trees is that they are simple to understand and interpret, if the size of the three does not grow too much. Single models, often in the form of weak learners, can also be combined into a new, more performing model by means of ensemble learning approaches [72, 154]. For instance, this is the case of random forests [33], which we can simplify as a combination of several decision trees by means of bagging [32], an ensemble algorithm. Using boosting [18, 232] rather than bagging leads, for instance, to gradient-boosted trees [86, 87] instead. It is worth highlighting that the use of ensembles has its pros and cons. It is generally true that ensemble solutions have greater prediction capabilities compared to the base models, but some characteristics of the weak learner are lost. For instance, comparing decision trees against random forest and gradient-boosted trees, it is immediately noticed that more data, as well as careful fine-tuning, are required, and the model becomes black-box rather than white-box: they are not inherently interpretable anymore.

The lowest level of supervision is required by unsupervised learning. Here, the algorithm is not given any labelled training examples and must find patterns, structures, and relationships in the data in an unguided manner. Generally speaking, it finds application in tasks such as data compression, denoising, and generative modeling, although it is very hard to evaluate in practice: there is no clear understating of how to measure the quality of the learned relationships. One common task pertaining to unsupervised learning is clustering, where the goal is to group similar data points together. This can be useful for applications such as anomaly detection or customer segmentation. Clustering is an extremely complex and broad topic. It can be achieved with various algorithms that differ significantly in their understanding of what constitutes a cluster and how to find it efficiently. For instance, some clustering models rely on distance connectivity [281] or density of the data space [78]. Moreover, they often require to choose some parameters, like the number of clusters that should be made, or some threshold used to determine whether two objects should belong to the same group or not [115]. This results in the fact that determining the optimal clustering algorithm for a specific task is very difficult, often leading to an iterative process of knowledge discovery. Another unsupervised task is dimensionality reduction, where the goal is to find a lower-dimensional representation of the data that captures its essential structure, i.e., some meaningful properties of the original data are preserved. This can be useful for tasks such as visualization or feature selection. The most used type of dimensionality reduction is feature projection. It consists of transforming the data from its original high-dimensional space to one with a lower number of dimensions. Such mapping can be done in a linear way (e.g., Principal component analysis (PCA) [109, 201]) or by relying on nonlinear dimensionality reduction techniques (e.g., Kernel PCA [234], t-SNE [263]).

Intermediate levels in the supervision spectrum include but are not limited to reinforcement learning, where the machine is given only a numerical performance

score as guidance, and semi-supervised learning, where just a small portion of the data is tagged.

In reinforcement learning, the algorithm learns by interacting with its environment and receiving rewards or punishments for certain actions [249]. The objective is to learn a policy that maximizes the cumulative reward over time.

In semi-supervised learning, the algorithm is given a small number of labelled training examples and a large number of unlabeled examples. The goal is to make use of both types to improve the accuracy of the model. The rationale behind this family of approaches is to exploit the information content provided by the labels while reducing the number of labelled examples that should be obtained. As a matter of fact, dataset collection and its labelling are one of the most critical issues affecting modern AI solutions. It is generally recognized that with a larger amount of data it is possible to obtain more accurate predictions, but at the same time labelling them is a complex process, which is also time and money-consuming.

Nowadays, we cannot end this introduction without talking about deep learning [97], which is a branch of machine learning that is originally inspired by the structure and functioning of the brain. It builds around the notion of artificial neural networks, which are algorithms designed to recognize patterns and relationships in data. The term *deep* stems from the fact that multiple layers are involved in the algorithm, and each of them is in charge of applying a sequence of (non-linear) transformations to the input data. Such transformations are based on a set of weights, which are optimized according to a given loss function by means of back-propagation [222].

Compared to machine learning, the main advantage of deep learning is its capability to exploit large amounts of unstructured data, making it very suitable to work with images, text, and signals, to perform, for instance, computer vision, natural language processing and understating, speech recognition, and signal processing tasks. Dealing with such modalities is very complex for classical machine learning approaches as well as for other AI methods, like symbolic ones.

The capability of deep learning to achieve good results with these modalities is partially given by the fact that there exists specific architecture that can deal with each of them, acting as inductive bias [20]. For instance, this is the case of Convolutional Neural Networks (CNN) [139, 140] and Vision Transformers [71] for computer vision; Recurrent models like RNNs and LSTMs [107], and Transformers [68, 214, 264] for text and sequences. Of course, nothing comes for free. Deep learning algorithms require a significant amount of data and computational resources to train, and they can be difficult to interpret due to their complex structure.

It is worth concluding this brief overview by pointing out that having large amounts of data and being able to use machine and deep learning approaches are not a panacea. Especially when working in applied contexts and with real data, the right handling of data is the key to success in all subsequent tasks. Proper data pre-processing and the provision of appropriate mechanisms and solutions for organised data storage are fundamental to unlocking the full informational power of our data. Therefore, a new, almost symbiotic relationship between machine learning

and data science emerges, namely the fact that approaches and steps typical of the latter become functional for the former.

1.2 Positioning systems

As we already said, data science is a discipline that addresses concrete problems in a scientific manner, to extract knowledge in terms of explanations and predictions. The concrete problem we are going to focus on in this thesis is that of *positioning*. Indeed, identifying the position of an object or a person is a task of fundamental importance in everyday life. As a matter of fact, the number of applications and domains making use of such information are several, including transportation, urban planning, environmental studies, business, navigation, logistics, emergency management, contact tracing, access control, and public security [170, 172].

From a general perspective, positioning or localising someone means determining where it is placed within a specific reference system. The simplest, oldest and most adopted reference system is the one based on geographic coordinates: latitude, longitude, and possibly elevation. Such a system measures locations directly on the Earth regarded as a sphere, while other systems, such as the geocentric one, model the Earth as an object in a three-dimensional Cartesian space, representing locations as points characterized by three coordinates. Cartesian coordinates are employed also by planar modeling of the Earth, where map-projection techniques are used to minimise the distortion introduced by such approximation. Finally, a special case of the planar one is when a small portion of the Earth, or a small specific (local) area (e.g., a building), is considered. In such instances it is possible to rely on 2D/3D coordinate systems, ignoring the negligible effect of the Earth's curvature. To define an effective coordinate system, it is also necessary to specify an origin point, and a mapping between the system and the modelled object (e.g., the Earth, or a specific area). With all these ingredients, it is then possible to express someone's position relative to a real, reference place.

The flexibility needed by the availability of multiple reference systems is immediately motivated by the fact that we are interested in performing positioning not only in outdoor settings but also in indoor ones. Outdoor localization is generally performed relying upon global navigation satellite systems (GNSSs), such as the United States Global Positioning System (GPS), Russia's Global Navigation Satellite System (GLONASS), China's BeiDou Navigation Satellite System, and the European Union's Galileo [253]. Outdoor positioning systems are nowadays capable to provide very precise estimates, although some issues still affect them, such as the high battery consumption [152, 304], the expensive price of the module, the performance reduction in indoor areas and urban canyons [153, 162, 269].

Concerning the ability to identify people and objects' locations within a building, is still a challenging task. This is caused by the substantial differences characterizing indoor and outdoor spaces, such as severe multi-path effects, non-Line-of-Sight

conditions, high signal attenuation, temporal instability due to changes in the scenario, high accuracy demand, which, overall, make it impossible to rely on GNSS for indoor localization. Thus, the research on effective alternatives applicable to the mobile and Internet of Things domains and to perform positioning in indoor premises without relying on satellite signals is paramount [206], especially if we consider that people spend most of their time indoors [70].

In the remainder of this section, we are going to provide an account of indoor positioning, which is the main topic at the center of this work. As we will see, there exist a lot of different approaches to tackle such a problem. After a general overview of the state-of-the-art (Section 1.2.2), we will focus on WiFi-based fingerprinting, the technique we studied and investigated in the dissertation (Section 1.3).

1.2.1 Indoor localization

In the last twenty years, a lot of efforts have been made towards developing solutions alternative to GNSS, especially considering the indoor case. However, although several progress has been made, some common limitations still affect indoor positioning systems (IPS), such as, for instance, the lack of standards, the absence of global solutions applicable to heterogeneous devices, the often unsatisfactory position accuracy, and the strong dependency of a positioning system from the specific environment [121, 211].

The design of IPS becomes then a challenging task [91], exacerbated by the many desiderata such systems should satisfy, such as, for instance, high accuracy, coverage of a given environment (e.g., a single floor or an entire building), low cost, few ad-hoc infrastructures, privacy, satisfactory update rate, low battery consumption, robustness, availability, scalability (in terms of both new environment to cover and higher number of users) and many others [19, 75, 170, 227, 251, 292, 299]. Although indeed these aspects are fundamental from the user perspective, evaluation purposes, and they may be useful to understand which IPS is more suitable to a given application scenario, from a technical perspective an indoor positioning system is typically characterized according to three different dimensions [177].

First, the most obvious characteristic is the technology underlying the system. This crucial aspect is related to the type of data source exploited to perform the positioning rather than to a specific device or hardware. In the last twenty years, many different sources have been exploited in order to determine people and object positions inside a building such as, for instance, radio signals (cellular, WiFi, Bluetooth), RFID, magnetic fields, light intensities, information coming from sensors such as gyroscopes or accelerometers (inertial measurements), sound, vision, and so on [177]. Simultaneously, several techniques based on the exploitation of a single or a combination of the above sources have been developed.

It is then possible to look at whether the IPS requires the presence of devices on the person or object whose position is to be estimated, or whether it relies solely on an infrastructure, possibly even ad-hoc, installed in the environment. In the first

scenario, we speak of device-based (or active) localisation, while in the second of device-free (or passive). For instance, a system relying on a smartphone carried by a user belongs to the first type. Conversely, a system that detects through a fixed device the location of multiple objects by looking at whether the state of the surrounding environment changes (e.g., relying on light-related data or WiFi channel state information), is device-free.

Finally, one can categorise IPSs from the point of view of the hardware and its deployment in an environment. Precisely, whether specific devices need to be installed or the required knowledge about the infrastructure already in place is considered. A system that exploits the native infrastructure without requiring to know its details will be called infrastructure-free, alternatively infrastructure-based. For example, an IPS that utilises the WiFi network already deployed in an indoor scenario and does not require knowing the location of the access points will pertain to the first category, whereas if it does require knowing their locations, it will belong to the second type, even if no dedicated access point is installed for positioning. Again, a system that exploits Bluetooth Low Energy (BLE) and demands deploying some beacons will be infrastructure-based.

It can be seen from the above examples that a given IPS is typically characterised by looking at all three dimensions, as they convey orthogonal features.

1.2.2 An overview of IPS approaches

Throughout the years, multiple different approaches have been investigated to address indoor positioning tasks.

Light in the form of infrared or visible light communication can be exploited for indoor positioning purposes [3, 303]. Systems based on such technology are often capable to provide quite precise positioning estimates, in the order of centimetres. The main limitations of the light-based approaches, which may find application as both device-free and device-based systems, are the line of sight requirements and the large amounts of noise introduced when other light sources are present in the environment, daylight above all.

It is also possible to exploit visual information to perform positioning. A straightforward device-free approach, which requires to have dedicated hardware, is the recording of images about the environment and the target, with the task of identifying the latter position within the former. Device-based approaches are more variegate, and they range from solutions based on marker recognition to visual odometry [10].

Magnetic fields can be used to support IPS too, exploiting the idea of collecting a database reporting the variations of the magnetic field strength in the environment [199]. They can rely on natural magnetic fields or artificially generated ones. The key difference is that the firsts do not require the installation of dedicated hardware, but provides lower positioning performance than those of the second type [64].

Another information that can be exploited to perform positioning is the one

related to inertial measurement units. Such data are often used within dead reckoning solutions, that are based on the idea of exploiting the previously known position of the device, and the movements made from it [171]. In order to improve the performance of such approaches, it is not uncommon to pair movement data with additional information petering the environment, such as the map representing its layout and possible constraints. Dead reckoning IPS are inherently device-based and offer their best when used in combination with other technologies. They are capable to provide good positioning estimates, although their uttermost issue is the accumulative errors: as they strongly rely on the previous location, if such an estimate is not precise, it would result in having an even greater error for the subsequent position estimate [227].

A very commonly used family of technology is those based on radio signals. IPS based on WiFi can be partitioned into two classes [285]. The first one exploits spatial and/or temporal features associated with the signal emitted by a WiFi access point (AP) and received by a device. Approaches belonging to this group are (*i*) those based on Time of Arrival (TOA), that estimate the distance of the APs on the basis of the time of arrival at the device of their signals, (*ii*) those based on the Angle of Arrival (AOA), that look at the angle between a reference direction and the incident signal wave, and (*iii*) those making use of the Time Difference of Arrival (TDOA), that extend TOA by assessing the temporal difference between the signal arrival times of multiple APs and a reference signal chosen among them. For all these families of spatio-temporal techniques, at least two-to-three APs need to be visible, and their precise position must be known [121, 285]. Techniques in the second class rely on the Received Signal Strength (RSS), that is, the power of the signal received from a WiFi AP. One approach belonging to this group is signal propagation modeling, which consists of exploiting a large-scale path loss model that accounts for both free-space and loss due to obstructions. This model is then used to estimate the RSS observable at various positions, after a calibration procedure of the propagation model, which is a complex step, paramount to get good performance, often requiring human intervention [17]. The most popular technique based on the RSS is fingerprinting, also referred to as scene analysis. Fingerprinting is characterised by two phases. The first one is the so-called *offline* phase, during which a site survey is executed, aimed at the collection of RSS vectors at different predefined locations. In the subsequent *online* phase, a user samples an RSS vector at an unknown location, comparing it with those previously collected, finally providing an estimate of the user's position by means of specific algorithms.

Bluetooth Low Energy requires the deployment of ad-hoc beacons, which have the advantages of being dedicated for positioning purposes (i.e., they are not used for other purposes, like in the case of WiFi), the relatively low cost, and the very low battery consumption [64, 80]. However, the range of the beacons is lower than those of WiFi access points, posing scalability issues in the case of large scenarios. Bluetooth shares a lot of similarities with WiFi, and, in principle, the same approaches (e.g., TOA, fingerprinting) can be used in this setting. Nevertheless, given the need

to place the beacons, the uttermost advantage of fingerprinting, i.e., not requiring information on the infrastructure, is inherently lost, promoting the adoption of other techniques [7, 31, 42, 175].

RFID (Radio Frequency IDentification) are electronic tags that store some data which can be read by properly-equipped devices through radio frequency. Multiple types of tags exist (passive, semi-passive, and active), differing in the way in which they emit the data (periodically, or when a reader signal is detected). From a positioning perspective, the role of the tag and the reader can be interchangeable, and they depend on the application scenario. Tags are smaller than readers, thus, having the latter fixed naturally fits the case where there a many targets for the positioning, as, for instance, while tracking goods in a warehouse. However, the most classical scenario is where tags are placed at pre-defined locations, and the user reads them to retrieve its position. RFID approaches are infrastructure-based and use quite simple algorithms, which reported accuracy may vary considerably [34, 239].

Finally, also cellular data can be used for indoor localization, although they are also exploited for outdoor positioning, as GNSS replacement [135, 178, 229, 286]. In an indoor setting, where mainly fingerprinting and lateration approaches are used, the accuracy of cellular solutions is quite low compared to the other competitors, making it a poorly considered approach, unless combined with other technologies. However, 5G standards and beyond aim to incorporate specific information meant to support localization tasks, in both outdoor and indoor settings [184]. Altogether, this might lead in the near future to having a technology applicable to both environments while providing good performance, paving the way to a resurgence of cellular-based positioning and indoor-outdoor seamless localization.

1.3 A focus on WiFi-based fingerprinting

In this section, we first provide an overview of the methodologies used in WiFi fingerprinting to estimate the position of a user. Then, in the second part of the section, we introduce some fundamental concepts and materials that will be used multiple times throughout the entire thesis. Specifically, we will discuss and explain the motivations behind the adoption of some relevant metrics and normalization functions in the domain of indoor fingerprinting; then, we provide an overview of the datasets from literature exploited in the present dissertation.

1.3.1 Definition and methodologies

Fingerprinting is a widely exploited technique for positioning, which could be used for both indoor and outdoor localization, relying on different technologies, such as radio signals, visual clues, etc. Let's focus on the case of WiFi fingerprinting, which is the approach at the centre of this dissertation. Such a technique is generally

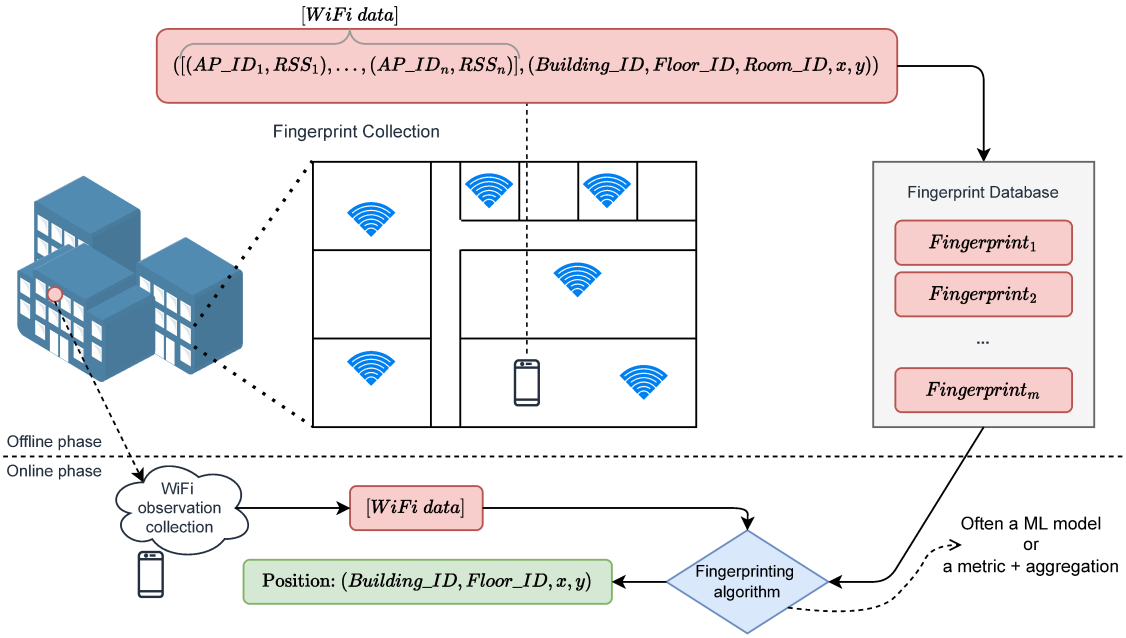


Figure 1.1: High-level description of WiFi-based fingerprinting.

composed of two different phases. The *offline phase* consists of the collection of fingerprints. Typically it is performed by standing at various predefined locations, called *reference points* (RPs), in the target environment, observing the status of the WiFi network for an extended amount of time. For each location, multiple fingerprints are obtained, in order to reduce the effect of temporal instability [284]. In the case of WiFi, a fingerprint is a vector reporting the RSS observed for each AP in the environment. The set of collected fingerprints with the corresponding RPs is stored in a database often referred to as *radio-map*. Note that the collection of the fingerprints can follow different strategies, for instance, relying on crowdsourcing [155, 203, 273]. The *online phase* is concerned with the actual usage of the systems. First, a new fingerprint is collected at an unknown position. Then, such observation is compared with those stored in the database, or processed by means of a model trained on the radio-map, in such a way as to determine the possible target's location. Figure 1.1 describes the high-level functioning of indoor WiFi fingerprinting.

Formally, let $\mathcal{L} = \{\mathbf{l}_1, \mathbf{l}_2, ..., \mathbf{l}_o\}$ be the set of o discrete locations at which the desired scenario has been sampled. Each location $\mathbf{l}_i$, $1 \leq i \leq o$, is represented as a tuple that encodes position-related information, i.e., $(building_i, floor_i, room_i, x_i, y_i) \in \mathcal{B} \times \mathcal{F} \times \mathcal{R} \times \mathbb{R} \times \mathbb{R}$, with $\mathcal{B}$, $\mathcal{F}$, and $\mathcal{R}$ being respectively the sets that include categorical labels regarding the building, the floor, and the room/area where the locations are sampled, while x_i and y_i are the latitude and longitude data[1]. The kth fingerprint

[1]Here and throughout the dissertation we are going to consider as standard granularity levels

associated with the location $\mathbf{l}_i$ is denoted as the vector $\mathbf{f}_{i_k} = [f_{i_k1}, f_{i_k2}, ..., f_{i_kq}] \in \mathbb{R}^q$, where q is the total number of APs appearing in the considered scenario (sensed all over the locations in $\mathcal{L}$) and f_{i_kj}, $1 \leq j \leq q$, is the RSS value related to the AP that is given the unique identifier j (or `null` if such AP is not detected).

In the literature, many surveys revise the approaches, the advancements, and the challenges related to the research on fingerprint-based indoor positioning, e.g., [103, 121, 221, 285]. Even though there is not a general consensus on a taxonomy of WiFi-based indoor positioning systems, the majority of existing surveys agree that at least two classes of solutions can be identified [103, 211]. The first, simplest one includes deterministic algorithms. A member of this class is RADAR [12], a k-Nearest-Neighbour (k-NN) solution which selects the fingerprint in the database closest to the measurement done at runtime. It makes use of Euclidean distance, but different metrics and similarity measures can be exploited. In [261], a comparative analysis of the performance of several measures applied in combination with k-NN is conducted. Many other solutions belong to this class. Their main advantages are being easy to implement and, often, featuring low computational complexity [103, 167]. The other class of methods is the probabilistic one. These model the positioning problem with an arbitrary complex formulation $\hat{\mathbf{l}} = \arg\max_{\mathbf{l}\in\mathcal{L}} P(\mathbf{l}|\mathbf{s})$, where $\mathbf{s}$ is the observed fingerprint sampled in the online stage and $\hat{\mathbf{l}}$ is the most likely location. Horus [295] is a representative of this class.

Many approaches, especially the most recent ones, are not classified into one of the two groups. Instead, very often the term "advanced techniques" is used to describe methods relying on more complex concepts, such as, for instance, sensor fusion, trajectories, and machine learning [5, 255]. Nevertheless, it is worth highlighting that, to some extent, even these solutions can be associated with deterministic and probabilistic definitions, or a mixture of the two. For instance, as we shall see in Chapter 6, classification-oriented deep learning algorithms return probabilities, fitting the probabilistic framework.

1.3.2 Metrics and normalization functions

As it is well known, a metric is a function that satisfies the following three axioms: *(i) identity of indiscernibles*, *(ii) symmetry*, and *(iii) triangle inequality* (*non-negativity* is also implied). Metrics, in the form of dissimilarity functions, are used to measure the distance between pairs of fingerprints in their own high-dimensional vector space. As pointed out in various contributions [261, 262], many different metrics can be used for indoor localization. In [261], the authors compare the performance of 52 of them, paired with (k-)Nearest Neighbor, in solving the positioning task; however, in a recent paper [262], it is observed that only a few of them are actually

for indoor localisation, building, floor, room, latitude and longitude (or any other 2D valid representation). Nevertheless, it is only an assumption, and it is possible to generalize all the definitions and outcomes to differently-arranged structures.

exploited in real-world positioning systems. Since we aim at comparing metrics of practical significance, our analysis will be carried out mainly on such a restricted set of dissimilarity measures.

Let $\mathbf{p}, \mathbf{q} \in \mathbb{R}^n$ be two vector column fingerprints, and let $p_i \in \mathbf{p}$, for all $i \leq n$, be the RSS value in fingerprint $\mathbf{p}$ for the access point with identifier i (keep in mind that RSS values are negative).

Five out of the ten considered functions are specializations of the *Minkowski* distance, which, in its general formulation, can be defined as follows:

$$d_{\text{minkowski}}(\mathbf{p}, \mathbf{q}; m) = \left(\sum_{i=1}^{n} |p_i - q_i|^m \right)^{\frac{1}{m}} . \tag{1.1}$$

The five variants are obtained by assigning a different value to the parameter m, namely, $m = 1, 2, 3, 4$, and 5. It is worth noticing that when $m = 1$ and $m = 2$ we respectively get the well-known *Manhattan* and *Euclidean* distances.

Similar to the *Euclidean* one, except for the application of the square root to the single elements, is the *Jeffries-Matusita* distance, which is defined as follows:

$$d_{\text{matusita}}(\mathbf{p}, \mathbf{q}) = \sqrt{\sum_{i=1}^{n} (\sqrt{p_i} - \sqrt{q_i})^2} . \tag{1.2}$$

Another generalization of *Euclidean* distance is the *Mahalanobis* one, which is defined as follows (in vectorial form):

$$d_{\text{mahalanobis}}(\mathbf{p}, \mathbf{q}, \mathbf{S}) = \sqrt{(\mathbf{p} - \mathbf{q})^T \mathbf{S}^{-1} (\mathbf{p} - \mathbf{q})} , \tag{1.3}$$

where $\mathbf{S}^{-1}$ is the inverse of the covariance matrix $\mathbf{S}$ computed over all fingerprint pairs. Observe that, for $\mathbf{S}$ being invertible, the number of linearly independent observations (fingerprint pairs) has to be larger than the number n of dimensions (considered APs)[2].

We also analyze the behaviour of *Sørensen* distance (aka Bray-Curtis coefficient). It is a semi-metric, that is, the triangle inequality axiom is relaxed, which is defined as follows:

$$d_{\text{sorensen}}(\mathbf{p}, \mathbf{q}) = \frac{\sum_{i=1}^{n} p_i - q_i}{\sum_{i=1}^{n} p_i + \sum_{i=1}^{n} q_i} . \tag{1.4}$$

[2]To cope with such an issue, when used in practice in this dissertation, we first apply principal component analysis to reduce the number of variables/dimensions while preserving 95% of the explained variance.

Similar in spirit to *Sørensen* distance is the *Cosine* one, which is defined as follows:

$$d_{\text{cosine}}(\mathbf{p},\mathbf{q}) = 1 - \frac{\sum_{i=1}^{n} p_i \cdot q_i}{\sqrt{\sum_{i=1}^{n} p_i^2} \cdot \sqrt{\sum_{i=1}^{n} q_i^2}} \,. \tag{1.5}$$

Notice that *Cosine* distance gives a judgment of orientation, not a magnitude. This violates the identity of indiscernibles axiom. It also lacks the triangle inequality.

A different approach is taken by *Neyman chi-square divergence*, which can be exploited to measure the distance between pairs of fingerprints in the following way:

$$d_{\text{neyman}}(\mathbf{p},\mathbf{q}) = \sum_{i=1}^{n} \frac{(p_i - q_i)^2}{p_i} \,. \tag{1.6}$$

Note that it satisfies neither the symmetry nor the triangle inequality axiom.

Finally, we took into consideration the *Signal Weighted Euclidean Distance (SWED)* [271], which is defined as follows:

$$d_{\text{SWED}}(\mathbf{p},\mathbf{q}) = \sqrt{\frac{1}{m} \sum_{i \in \pi} \frac{|r_{d_0} - \frac{p_i+q_i}{2}|}{\sum_{j \in \pi} |r_{d_0} - \frac{p_j+q_j}{2}|} (p_i - q_i)^2} \tag{1.7}$$

where $\pi = \langle i \in \{1,\ldots,n\} \mid p_i \neq \texttt{null} \wedge q_i \neq \texttt{null} \rangle$, with $|\pi| = m \leq n$, is the sequence of (the identifiers of) the APs commonly observed (i.e., not `null`) between $\mathbf{p}$ and $\mathbf{q}$, and r_{d_0} is a reference value denoting the hypothetical RSS of an AP measured at the predefined distance $d_0 = 1$. In this work, just a single r_{d_0} set to -31.7 *dbm* (on the basis of [271]) is used for all the APs. It is worth pointing out that *SWED* is not defined when the two fingerprints have no AP in common, i.e., $|\pi| = 0$. In such a case, the resulting distance will correspond to the lowest RSS recorded in the database.

Observe that some dissimilarity functions are not available for negative values as input. This is the case, for instance, with *Matusita* distance, where the square root is applied to the vector values. Moreover, APs not detected by fingerprints need to be dealt with as well. Thus, before computing the distance between pairs of fingerprints, a normalization step is usually applied to their RSSs. In the following, we consider the effects of 4 normalization functions proposed in the literature: (*i*) *positive values*, (*ii*) *zero-to-one normalized values*, (*iii*) *exponential values*, and (*iv*) *powed values*.

Given a fingerprint, the *positive values* normalization subtracts from each AP the minimum recorded RSS, selected from all the observations, after subtracting 1 from it (such a value is called r_{min}). As a result, all fingerprint values f_i are made positive in the interval $[1, |r_{min}| + 1]$, and the lowest admissible value 0 is used to

represent those AP which have not been detected in a fingerprint (i.e., $f_i = \texttt{null}$). Let $\mathbf{p}$ be a fingerprint and i be an AP, with RSS value p_i. The value of the latter is transformed as follows:

$$Positive_i(\mathbf{p}) = \begin{cases} (p_i - r_{min}) & \text{if AP } i \text{ is detected,} \\ 0 & \text{otherwise.} \end{cases} \tag{1.8}$$

Zero-to-One normalization is a variant of the previous one, which maps all RSS values in the range $[0, 1]$:

$$Zero\text{-}to\text{-}One_i(\mathbf{p}) = \frac{Positive_i(\mathbf{p})}{-r_{min}} \ . \tag{1.9}$$

Both the above normalization strategies are linear. However, since RSS values are given in *dBm*, they follow a logarithmic scale. Thus, it makes sense to apply a transformation that breaks the trends in such values. *Exponential* and *Powed* normalizations do exactly that. The first one is formally defined as follows:

$$Exponential_i(\mathbf{p}) = \frac{\exp(Positive_i(\mathbf{p}) \cdot \frac{1}{\alpha})}{\exp(-r_{min} \cdot \frac{1}{\alpha})} \ , \tag{1.10}$$

while the second one has the following form:

$$Powed_i(\mathbf{p}) = \frac{Positive_i(\mathbf{p})^\beta}{-r_{min}^\beta} \ . \tag{1.11}$$

Parameters α and β must be tuned according to the specific use case. Since their optimization is out of the scope of the present work, the values $\alpha = 24$ and $\beta = e$ are assumed, as suggested in the literature [261].

1.3.3 Datasets

In this dissertation, we will mainly consider 16 public datasets, which are recognized as valuable testbeds by the indoor positioning community.[3] All of them come from complex real-world scenarios and differ from each other in various respects, including the structural properties of the buildings, the sampling strategy, and basic features like the number and the distribution of RPs (and fingerprints). An overview of their distinctive characteristics can be found in Table 1.1[4]. In the following, we give a short account of each of them.

[3] Acronyms and data (except for `UTS`) are taken from [257]. `UJI 2` has not been included, since it represents an extension of `UJI 1` with additional test set data for which information about positions is unavailable.

[4] For the sake of the proposed analyses, sometimes we will distinguish between training and test data, and other times we will consider each dataset as a whole. Thus, dataset properties are detailed taking into account both cases.

DSI 1 and DSI 2 [188] have been collected in the same university building at the Department of Information Systems of the University of Minho, Portugal. The second is obtained from the first one by removing fingerprints (FPs) sampled at the same RP. LIB 1 and LIB 2 [176] model the Universitat Jaume I's two-floor library

Table 1.1: Characteristics of the considered datasets.

DB	Dim/Area	Setting	#u	#d	#b	#f	#r	split	#FP	#RP	FP x RP	#AP	FPρ	Valid APs
DSI 1	100 m x 18 m	university floor	1	1	1	1	25	train	1369	230	1 to 9	152	0.73 ± 0.27	24.8 ± 8.6
								test	348	-	-	125	0.70 ± 0.27	23.6 ± 7.8
								whole	1717	578	1 to 9	157	0.91 ± 0.32	24.6 ± 8.5
DSI 2	100 m x 18 m	university floor	1	1	1	1	25	train	576	230	1 to 5	152	0.31 ± 0.11	24.8 ± 8.6
								test	348	-	-	125	0.29 ± 0.12	23.6 ± 7.8
								whole	924	578	1 to 5	157	0.49 ± 0.17	24.3 ± 8.4
LIB 1	15 m x 10 m	two-floor library	1	1	1	2	2	train	576	48	12 or 36	77	2.42 ± 0.59	14.2 ± 4.6
								test	3120	-	-	174	2.41 ± 0.70	21.0 ± 6.3
								whole	3696	212	12 or 36	174	7.87 ± 2.21	20.0 ± 6.5
LIB 2	15 m x 10 m	two-floor library	1	1	1	2	2	train	576	48	12 or 36	119	2.42 ± 0.59	21.6 ± 7.0
								test	3120	-	-	165	2.41 ± 0.70	18.8 ± 5.1
								whole	3696	212	12 or 36	197	7.87 ± 2.21	19.3 ± 5.5
MAN 1	50 m x 36 m	office hallway	1	1	1	1	-	train	14300	130	10 or 110	27	20.88 ± 4.48	10.3 ± 2.5
								test	460	-	-	24	20.55 ± 5.12	10.5 ± 2.5
								whole	14760	176	10 or 110	28	21.47 ± 4.81	10.3 ± 2.5
MAN 2	50 m x 36 m	office hallway	1	1	1	1	-	train	1300	130	10	26	1.90 ± 0.41	14.1 ± 2.9
								test	460	-	-	25	1.87 ± 0.47	14.1 ± 2.7
								whole	1760	176	10	28	2.57 ± 0.60	14.1 ± 2.8
SIM	50 m x 20 m	simulated	-	-	1	1	-	train	10710	1071	1 or 10	8	8.86 ± 1.80	8.0 ± 0.0
								test	1000	-	-	8	8.86 ± 1.50	8.0 ± 0.0
								whole	11710	2071	1 or 10	8	9.70 ± 1.85	14.1 ± 2.8
TUT 1	124 m x 57 m	university building	1	1	1	4	< 822	train	1476	1476	1 to 17	309	0.33 ± 0.13	32.9 ± 12.1
								test	490	-	-	183	0.41 ± 0.10	25.0 ± 7.3
								whole	1966	1928	1 to 17	309	0.19 ± 0.09	31.0 ± 11.6
TUT 2	145 m x 88 m	university building	1	1	1	3	NA	train	584	584	1	353	0.11 ± 0.05	47.9 ± 19.0
								test	176	-	-	248	0.12 ± 0.05	21.9 ± 7.1
								whole	760	760	1	354	0.07 ± 0.03	41.9 ± 20.2
TUT 3	130 m x 62 m	crowdsourced	8	21	1	5	822	train	697	694	1 to 14	779	0.17 ± 0.07	48.0 ± 38.4
								test	3951	-	-	989	0.16 ± 0.08	49.7 ± 38.7
								whole	4648	4511	1 to 14	992	0.41 ± 0.19	49.5 ± 38.7
TUT 4	130 m x 62 m	crowdsourced	8	21	1	5	822	train	3951	3843	1 to 14	989	0.96 ± 0.50	49.7 ± 38.7
								test	697	-	-	779	0.91 ± 0.48	48.0 ± 38.4
								whole	4648	4511	1 to 14	992	0.41 ± 0.19	49.5 ± 38.7
TUT 5	85m x 145 m	university building	NA	NA	1	3	NA	train	446	446	1	489	0.08 ± 0.03	42.5 ± 21.1
								test	982	-	-	296	0.07 ± 0.02	34.8 ± 13.5
								whole	1428	1428	1	489	0.19 ± 0.09	37.2 ± 16.7
TUT 6	135 m x 62 m	university building	NA	NA	1	4	< 822	train	3116	3116	1	562	0.65 ± 0.31	34.9 ± 15.9
								test	7269	-	-	566	0.63 ± 0.31	34.7 ± 15.9
								whole	10385	10385	1	489	1.06 ± 0.53	34.7 ± 15.9
TUT 7	88 m x 137 m	university building	NA	NA	1	3	NA	train	2787	2787	1	767	0.48 ± 0.30	27.0 ± 11.4
								test	6504	-	-	770	0.47 ± 0.29	27.1 ± 11.1
								whole	9291	9291	1	801	0.88 ± 0.51	27.1 ± 11.2
UJI 1	108703 m^2	university campus	19	25	3	4 or 5	$\gg 933$	train	19861	933	1 to 80	465	2.46 ± 1.84	18.1 ± 7.3
								test	1111	-	-	367	2.45 ± 1.62	16.5 ± 6.9
								whole	20972	1995	1 to 80	520	0.73 ± 0.43	18.0 ± 7.3
UTS	44000 m^2	university building	1	1	1	16	NA	train	9108	1466	1 to 35	557	0.46 ± 0.29	35.1 ± 13.5
								test	388	-	-	500	0.51 ± 0.3	0.51 ± 0.3
								whole	9496	1852	1 to 35	589	0.49 ± 0.3	35.4 ± 13.5

DB = dataset name; Dim/Area = size of the premises; Setting = collection environment; #u = number of involved users; #d = number of employed devices; #b = number of buildings; #f = number of floors; #r = number of rooms; #FP = number of fingerprints; #RP = number of distinct positions (x, y, z); FP x RP = fingerprints sampled per position; #AP = number of APs seen at least one time; FP_ρ = average FP density within a 5 meter radius from each RP; Valid APs = average number of detected APs per FP; - denotes meaningless information, NA denotes not available information.

(Spain). Their main difference is the acquisition date: 2016 and 2017, respectively. Both MAN 1 and MAN 2 [125] have been collected in the corridors of a single floor in a building of the University of Mannheim, Germany. With respect to MAN 1, MAN 2 applies an additional post-processing step by which FPs are averaged in 10 blocks of 10 fingerprints each. SIM is the only artificially generated dataset that we took into consideration. It has been produced according to a simple path loss model with additive Gaussian noise (see [262] for details). All TUT datasets have been collected at Tampere University, Finland. TUT 1 [241], TUT 3 [155], TUT 4 [155] (which is TUT 3 with inverted training and test sets), and TUT 6 [156] all consider the same five-floor building, and differ from one another in their collection strategies. TUT 1 relies on cell averaging with a grid size of 1 meter, TUT 3 is based on crowdsourcing, and TUT 6 is collected in a classical way (like, for instance, DSI 1). As for TUT 2 [241], TUT 5 [218], and TUT 7 [156], they have been independently collected in the same three-floor university building. TUT 5 follows an approach similar to TUT 1, since it considers cell averaging with a 5 meter grid size, while the way in which TUT 2 and TUT 7 have been collected is similar to that of TUT 6. UJI 1 [256] has been collected at the Universitat Jaume I, and it is usually referred to as UJIIndoorLoc. Among the considered datasets, it is definitely the largest and most complex, being the only one that considers a multi-building scenario. This dataset is the most exploited in the literature for comparison purposes. Finally, UTS [245] has been collected at the FEIT Building of the Sidney University of Technology, Australia. It refers to a single building with a high number of floors.

I

Modeling Indoor Positioning

2

A Framework for Indoor Positioning including Building Topology

The lack of standards, the high heterogeneity of collected data, and the inherent variety of indoor premises are major challenges to the development of indoor positioning solutions. This led to an extensive research effort that resulted in the proposal of several localization algorithms [81, 103, 121, 177, 221, 285], sampling strategies [155, 176, 256], benchmark datasets [125, 155, 156, 176, 188, 218, 241, 245, 256, 262], and building modeling approaches [15, 95, 119, 150, 190, 289].

On the positive side, such an effort resulted in a deeper understanding of indoor positioning and the achievement of reasonably accurate estimations. On the negative side, the research has been faced with a number of issues, including: *(i)* the difficulty in retrieving a comprehensive collection of datasets for the experiments; *(ii)* the need of reconciling data representations using heterogeneous formats and conventions; and, *(iii)* the problems in comparing existing contributions, that used the same datasets in different ways, e.g., with respect to the training/test split selection. The matter is even more serious in an industrial setting, where the design and deployment of a positioning system involve solving non trivial and time demanding tasks such as determining the extent of information that should be modeled and the best way to accommodate needs that may change over time. Last but not least, there are cross-cutting themes which are halfway between research and industry as the effective modeling and integration of advanced elements, like, for instance, device trajectories and information about the topology of buildings. These data can contribute significantly to the realisation of state-of-the-art positioning systems; however, combining them in a uniform framework, taking into account all the aforementioned aspects, is not trivial at all.

The work described in this chapter lies at the intersection of all the above dimensions: we propose a comprehensive, yet general and extensible, framework that poses as a tool easily adoptable by both the research community and industrial practitioners, which allows one to jointly handle fingerprint and building information. We first provide an abstract model of the considered domain, and then we turn it into a concrete relational database. A relational database stores information by means of fixed-length records, that are collected within a set of tables. Operationally, the de-

velopment of a database begins with the definition of a conceptual schema, typically formalized by means of the Entity-Relationship (ER) modeling language, which is used to explicitly represent all domain requirements. The ER schema is then translated, using well-established mapping rules, into a logical schema, containing the definition of the tables in terms of attributes and their domains, and constraints, including primary and foreign keys. Finally, the tables are implemented into a physical RDBMS (Relational DataBase Management System) instance by making use of suitable SQL (Structured Query Language) instructions [242]. The choice of relying on such a DBMS is based on a number of reasons, including: *(i)* its ease of deployment into an industrial setting, thanks to the widespread mastery of this technology, *(ii)* the availability of *SQL*, an easy-to-learn language that supports a user-friendly interaction with the system, *(iii)* the existence of a streamlined design process, from the conceptual design to the physical implementation of the database, and, *(iv)* the possibility of natively handling domain constraints, so as to guarantee data quality requirements.

The source code needed to deploy the proposed solution is available online [37]. In addition, we provide access to an implementation of the system [38], already populated with data coming from well-known indoor positioning datasets, that demonstrates the full potential of the proposed solution; the idea is that of evolving over time the latter system into a centralised, open repository of indoor positioning data available to the research community. Overall, we believe that the combination of the above two elements fosters the wide adoption of the framework: a company or a research group can first become familiar with its online implementation, at no cost; then, the source code provides a quite straightforward manner to set up a production-ready running local instance. In addition, as we will see, the proposed solution does not force any strong constraint on the type of (fingerprint-based) localization system to possibly employ on top of it. Thus, it poses as a general backbone approach to support indoor positioning, capable of providing a clear, structured, customizable, and unified interface to access and exchange information [90].

The chapter is organised as follows. Section 2.1 discusses the motivations and challenges of the work. Section 2.2 outlines an Entity-Relationship conceptual schema of the proposed framework. The logical schema of the database is given in Section 2.3. Section 2.4 illustrates some notable use case scenarios. Section 2.5 briefly analyses related systems/framework, and provide a summary of the main differences across them.

2.1 Motivations and challenges

In fingerprint-based indoor positioning, modeling is commonly recognized as a very complex activity (see, e.g., [224]). In this section, we discuss the most pressing issues we addressed in developing the proposed framework, that aims at supporting all tasks involved in the offline and online phases of a localization system.

The main challenge is the intrinsic dynamic nature of the domain. As an example, in WiFi fingerprinting, access points may be added, removed, or replaced. As for cellular data, mobile cells can be merged or relocated [266]. In addition, new wireless technologies are continuously being developed. Even the indoor premises themselves can be modified in their arrangement and architectural characteristics over time. Alongside, not only the sources of information may change, but also the way in which they are perceived and recorded by devices. This is true for both the kinds of sensors they are equipped with and the effectiveness of their sensing capabilities. As a consequence, information stored in radio maps undergoes constant evolution: new fingerprints are added, and old ones are updated or even discarded. For these reasons, the framework must be designed to grant ongoing and long-term support to the collection and maintenance of radio map fingerprint data.

A second aspect pertains to the high heterogeneity of the domain. As shown in the literature [150], an indoor scenario can be described at different levels of detail. Determining the right abstraction level is not trivial, as it involves reasoning over the possible kinds of premises and the topological relationships among them, including reachability aspects. The heterogeneity also applies to fingerprint data under two dimensions. First, fingerprints may consider several types of signal sources, such as WiFi, Bluetooth, cellular information, data from inertial sensors and GNSS receivers, as well as their combinations. A well-designed solution should offer comprehensive support to multiple kinds of observation, and allow for an easy extension to new ones. Second, fingerprints may be collected according to several strategies, e.g., following a well-planned survey plan, or relying on a crowdsourced effort. In addition, different sampling strategies may coexist within a given premise, e.g., as a result of repeated survey campaigns performed over the same building.

This brings to light the issue of information sharing: if the same object appears in more than one dataset, it makes sense to store its data only once in the system, providing pointers to the original sources so as to maintain data lineage. This may be the case, for instance, with a room, a mobile network cell, or an access point, that has been considered in more than one study. From a practical point of view, this may also increase the overall amount of available information. Let us assume that, within the same building, on two adjacent floors, two independent positioning systems are deployed. Suppose that they sense WiFi data, and since they are close, they might be based on the same access points (detected through the ceiling/floor). However, as the two systems are fully disjoint, there is no way to combine the data related to one floor and those related to the other. Organising data by means of the proposed framework allows one to recognize that the same access points are used by both systems, and to fruitfully exploit such a knowledge, e.g., by producing a richer radio map. In fact, information sharing goes beyond sensors/emitters as, for instance, it may also support the combination of data about the topology of a building.

It is worth pointing out that the problem raised by redundant, and possibly inconsistent, information is not only tied to the combination of several data sources.

Such inconsistencies may, indeed, be already present within a single dataset: modeling topological aspects is quite complex, and requires the knowledge of and adherence to a large set of domain constraints. As an example, a floor should not - typically - be contained into multiple buildings. Thus, the framework must also provide a simple and uniform manner to enforce such constraints, so as to guarantee data quality requirements.

The remaining issues pertain to the online usage of the system. In the most general case, a user may submit a single fingerprint to a positioning system to obtain a position estimate. Thus, a first question is how fingerprints can be related to information about the structure of buildings to support localization. The problem becomes more complex when the positioning algorithm relies on a sequence of fingerprints, that is, a trajectory, arising from user navigation within the premises, as dealing with both single points and trajectories is far from being simple. Finally, the framework should be able of supporting the (possibly concurrent) usage of multiple prediction algorithms, which may generate different outputs. As an example, the exact position coordinates may be estimated, or the fingerprint can be matched to a single logical location, like a room, as well as to multiple ones, e.g., by means of a probability distribution.

As we will see, all the above issues have been taken into account in the design of the proposed solution.

2.2 Domain modeling

In this section, we provide a high-level modeling of the considered domain. First, we propose a way to represent information about indoor premises that can be easily paired with positioning data. Then, we integrate the resulting model with all the aspects that are relevant to a positioning system by means of an Entity-Relationship diagram, which is the cornerstone of the relational database at the core of the proposed framework.

2.2.1 Pairing indoor topology and fingerprints

To represent information about indoor premises, we rely on a relatively simple representation, that allows one to describe the topology of indoor environments without explicitly encoding elements like walls, windows, and objects in the rooms. The guiding principle is that of building a system as general as possible so that researchers and practitioners may use it regardless of the adopted fingerprinting methodology and with only a few, possibly none, data about the premises.

To this end, we conceptually model the indoor setting as a heterogeneous (directed) graph. This choice is supported by at least three arguments: *(i)* in the indoor setting, there are different types of elements, like buildings, floors, and rooms, which have different properties and, thus, are better modeled by different types of node,

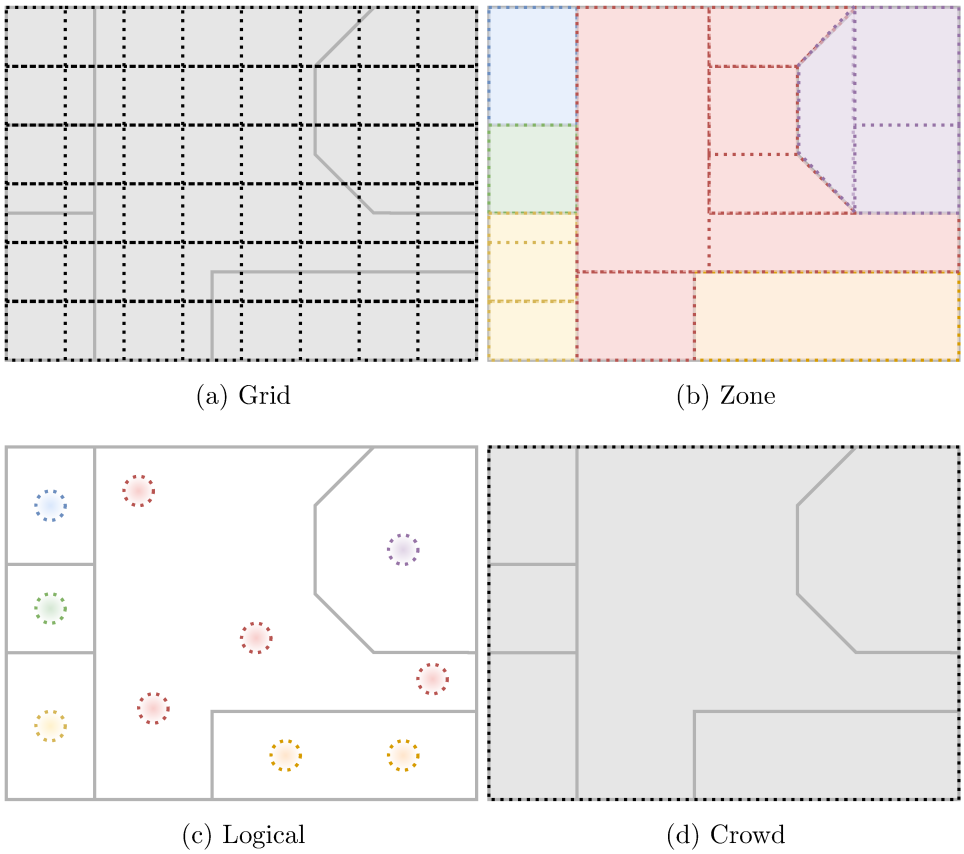
(ii) there are relationships among elements of the same type which are worth modeling, e.g., the relations of adjacency or walkability, that is, adjacency + traversability, over places, and *(iii)* there are relationships among elements of different type which are worth modeling as well, like, e.g., the intrinsic hierarchical organisation of indoor environments (a building consists of multiple floors, which are in turn composed of multiple rooms). Such a modeling is highly flexible: it makes it easy to capture relevant topological properties that may be useful for positioning purposes, while allowing to customize the detail at which to encode the structure of a given indoor scenario (information about most of the relationships is optional). In the following, we describe the characteristics of each element of the model as well as their relationships.

Building is the top object of the hierarchy. It can be adjacent to other buildings and contains a set of floors. We consider a building as a structurally independent element of the modeled domain, e.g., a separate construction possibly connected to others by indoor elements, such as bridges or underground tunnels. Each *floor* can be related to other floors to model the vertical ordering among them. Since the vertical dimension is fundamental in indoor positioning and navigation, information about it must be as sound and complete as possible. To this end, it is worth including in the model information about all the floors of a building, even if they are not explicitly involved in the positioning process. This is not a mandatory requirement, although the number of floors in a building is generally easy to obtain. A floor may consist of various elements, such as rooms, stairs, elevators, corridors, and so on, that we collectively name sites. A *site* is (a portion of) an indoor environment that one has decided to model explicitly. The core element linking position information (*fingerprints*, described next) and topology is the *tile*. A set of tiles in a floor is the result of a tessellation procedure, which defines the granularity at which the positioning task is performed. Different types of tessellation are possible:

- *Grid*: a fixed size regular grid is superimposed on the floor map, generating a set of tiles. Here the notion of tile is not associated with a site, since a grid-based partitioning does not take these pieces of information into account, and a grid cell can cross site boundaries. Thus, a grid-generated tile is linked directly to a floor (Figure 2.1a).

- *Zone*: an irregular grid is defined to partition a site or a floor. Each of the resulting areas is a tile. Each tile can be associated with at most one site or floor, and its coverage area can be arbitrarily large (Figure 2.1b).

- *Logical*: each tile can be considered as a semantic label associated with a meaningful location of the considered site or floor (Figure 2.1c); the tile has no geometrical shape, although it can optionally be characterised by a single pair of coordinates identifying a specific point in space for instance, the geometrical centre of a considered site, or a point of interest.



(a) Grid (b) Zone

(c) Logical (d) Crowd

Figure 2.1: Different tessellations for an indoor scenario. Dashed lines denote tiles; Tiles of the same colour refer to the same parent place (e.g., a site or a floor) in the hierarchy.

- *Crowd*: a single tile, devoid of geometrical data, is directly associated with a floor, acting as a general container of fingerprints in order to support crowdsourcing tasks (Figure 2.1d).

The relationships of adjacency and walkability over the set of places are encoded at the tile level, except for the building case, that relies on an ad-hoc relationship. The latter allows us to model the adjacency between buildings that are structurally separated and not connected in any manner, without making use of "outdoor" tiles. On the basis of information about tiles, it is possible to derive complex notions, such as paths (possibly traversing different floors) and adjacency relationships between rooms and corridors, according to the level of granularity of the chosen representation. A graphical account of an indoor scenario modeled by means of the proposed heterogeneous directed graph is reported in Figure 2.2.

Turning to the *fingerprint* data, we must bear in mind (see Section 1.3) that a fingerprint can be acquired during both the offline and the online phase of a

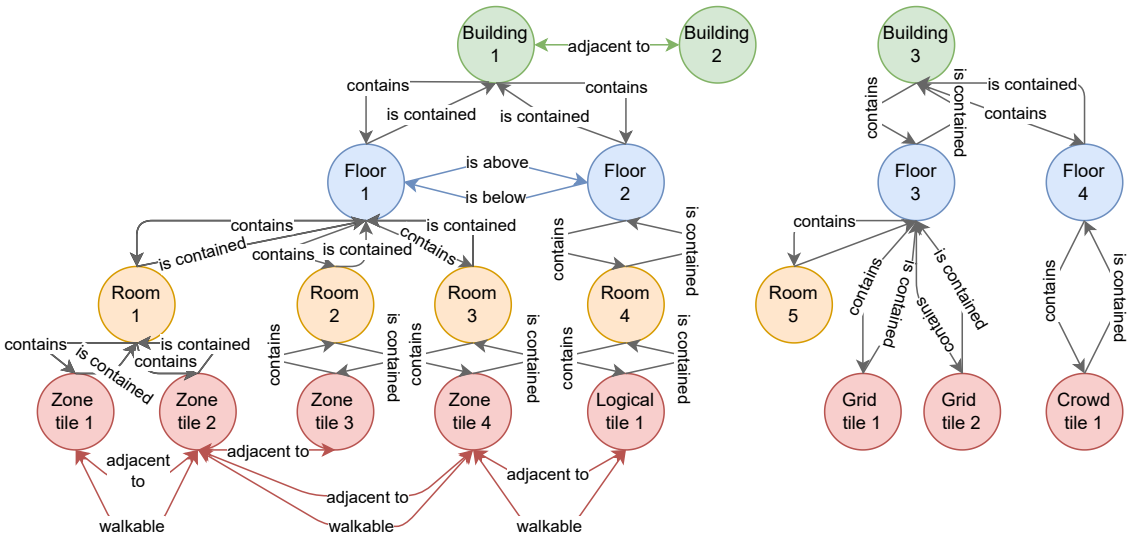


Figure 2.2: Proposed heterogeneous directed graph modeling a generic multi-building, multi-floor scenario using different types of tessellation.

positioning system. In the first case, it may be collected either via a planned survey conducted by experts or by users in a crowdsourced fashion. Depending on the collection modality, position information associated with a fingerprint may take on different forms. In the most standard setting, the position of a fingerprint is given as a vector of coordinates in a given reference system. In addition, a fingerprint can be logically associated with a specific element of the indoor environment, such as a floor, a site, or a portion of it, through the notion of tile.

Here, it is worth noticing the different semantics associated with *grid*, *zone*, and *logical* tiles and with *crowd* tiles. In the first case, each tile groups fingerprints related to a specific, predefined area. This is the case, for instance, when multiple fingerprints have been sampled for the same location. In the second case, a tile contains fingerprints collected from a given floor without any specific constraint. Note that, although *crowd* and *logical* tessellations may look similar, they play a very different role: the only way to model a crowdsourced scenario by a logical tessellation would be that of generating a distinct logical tile for each collected fingerprint, which is both counterintuitive and inefficient.

Finally, we also deal with the case in which no position information at all is associated with a fingerprint. This happens, for instance, with surveys where data gets labeled at a later stage, or when fingerprints are collected and exploited in an unsupervised fashion, or simply when the fingerprint has been collected during the online phase, e.g., no position estimation algorithm has been applied yet, or its results have not been stored in the system.

As a last remark, we observe that tessellations serve a dual purpose: on the one hand, they define a fingerprint collection strategy; on the other, they allow one to

model topological relationships. As an example, a given floor may rely on a *crowd* tessellation for the former purpose, and on a *zone* tessellation for the latter one.

Overall, the proposed model is highly flexible with respect to the type and the amount of data that a user is required to add, especially with respect to building topology. As an example, little to none information may be inserted about the latter, e.g., only the labels representing the building and the floor, and a crowdsourced approach can then be followed for the radio-map construction, with a minimum implementation effort. On the contrary, a person could be interested in modeling the indoor topology in its full detail: in such a case, complete information about the building components and their relationships can provided, and a fine grade tessellation, using logical or zone tiles, can be adopted.

To conclude, we would like to observe once more that we do not provide any means to model furniture and similar objects, as we aim at providing an account of building topology and integrating it with positioning data.

2.2.2 Overall conceptual schema

The Entity-Relationship diagram of Figure 2.3 provides a conceptual representation of information of interest about fingerprints, building topology, and their connection.

The schema consists of four distinct sub-schemas, each one focused on a specific portion of the domain, namely:

- the *Data source* sub-schema, which is responsible for preserving the data lineage, that is, it keeps track of the original sources of the data (fingerprints, observations, places) inserted in the database;

- the *Fingerprint* sub-schema, that records information about the fingerprints acquired by users/devices. Each fingerprint is collected at a given place and holds a set of observations, that is, the actual measurements performed by the device at the specific place. In addition, this sub-schema supports position estimation tasks;

- the *Observation* sub-schema, that stores detailed information about the data sensed by devices, which can be of various forms, like, e.g., the received WiFi or Bluetooth signal);

- the *Place* sub-schema, that models topological information about indoor scenarios, used for fingerprint collection and positioning tasks.

As we will see, the proposed design is general and flexible enough to be used in several contexts and scenarios.

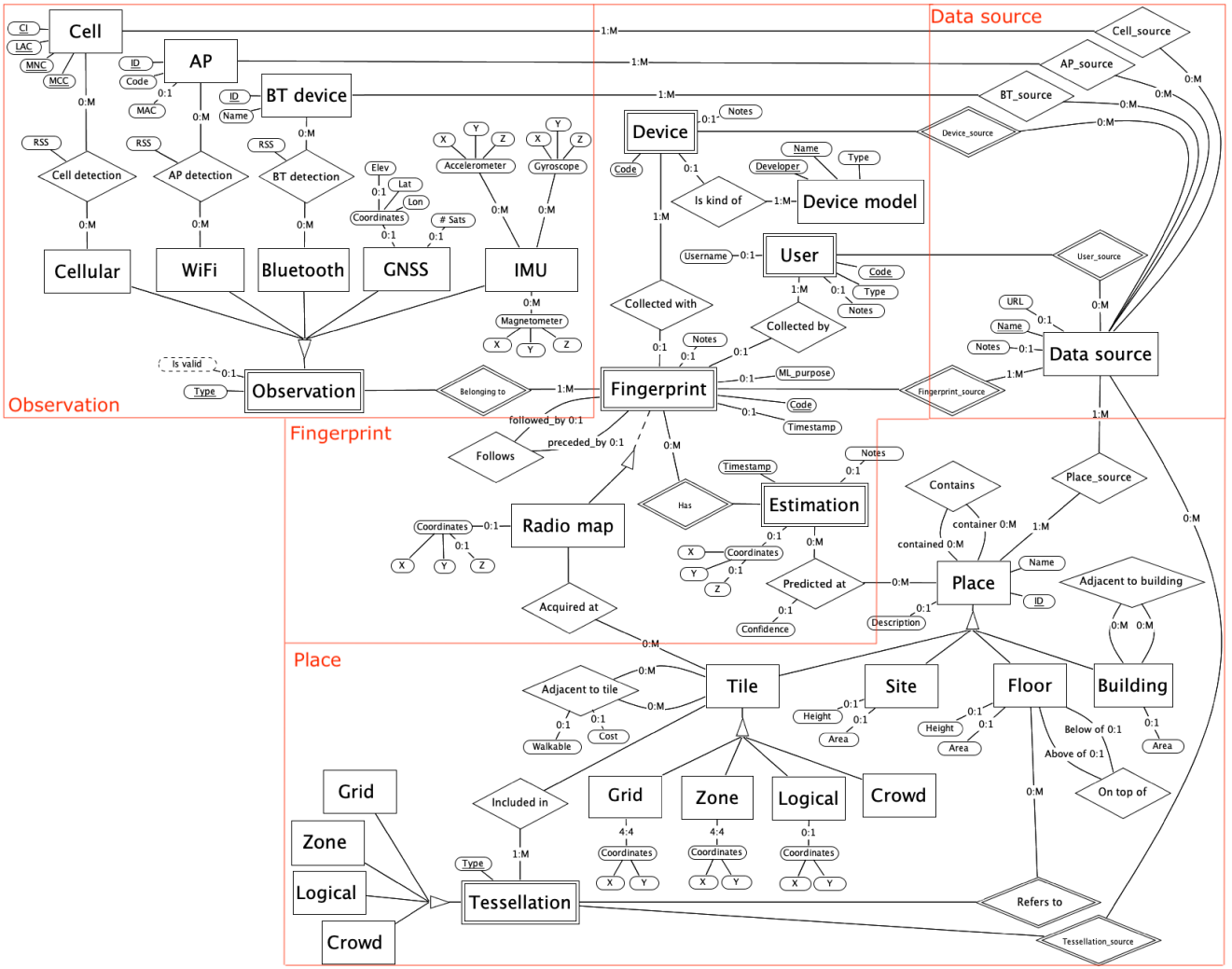


Figure 2.3: The Entity-Relationship diagram. The notation is very close to the one originally proposed by Chen [50].

Data source sub-schema

This sub-schema provides data lineage capabilities, that is, it allows one to track the original data sources of all pieces of information stored in the system. Each *Data source* is uniquely identified by a *name*, e.g., the name of the original dataset collecting the data (for instance, UJIIndoorLoc [256]). Optional information includes the *URL* of the original dataset and some free textual *Notes*. Several many-to-many relationships connect the data source to other schema entities, namely, *Cell_source*, *AP_source*, *BT_source*, *Device_source*, *User_source*, *Fingerprint_source*, and *Place_source*. In such a way, for instance, we can record information about the relationships between a data source and several access points. *Data_source* has an optional participation to all such relationships (for instance, no access point may be present in the dataset), except for *Fingerprint_source* and *Place_source*. In a given data source, indeed, at least information about the fingerprints and the premises where they were collected must be present.

Fingerprint sub-schema

This sub-schema stores information about fingerprints. Each *Fingerprint* has a *Code*, which uniquely identifies it within its data source, that is, *Fingerprint* is a weak entity with respect to *Data source*. In addition, it features the attribute *Timestamp*, that records the date and time at which the fingerprint was collected, the optional attribute *ML_purpose*, that encodes the intended use of the fingerprint in the original dataset, i.e., training, validation, or test purposes, and, possibly, some free textual *Notes*. A fingerprint can be preceded or followed by another fingerprint, as in the case of trajectories. This piece of information can be expressed by means of the relationship *Follows*. Some of the fingerprints may belong to the radio map, which is modeled through a partial specialization. For fingerprints belonging to the radio map, we store the following data about position information: the optional coordinates X, Y, and Z of the point of collection (Z is in its turn optional, as in many datasets only 2D spatial coordinates are considered); and the *Tile* where the fingerprint was acquired.

Each fingerprint is collected by means of a single physical *Device*, although information about it may be not known. A device is uniquely identified within a dataset by its *Code*. With a device we may associate further information which is stored within *Device model*, including *Developer* and *Name*, that together identify a device model, and *Type*, e.g., smartphone, tablet, or other. As an example, in the dataset there may be two distinct smartphones, characterized by codes A and B, both of the same device model *Samsung S22*.

Each fingerprint is collected by a *User*. As it happens with the device, information about the user may be missing. A user is uniquely identified within a dataset by its *Code*, and characterized by its *Type*, e.g., *trusted*, for known users that assembled the radio map, or *online*, for those using the system for localization purposes. In

addition, a user may have a *Username*.

To take into account the online usage of the positioning system, we introduce the entity *Estimation*. A fingerprint may be related to zero or more estimations, e.g., produced by different algorithms. In turn, an estimation is linked to a specific fingerprint, and may be discerned from the others that are associated with the same fingerprint thanks to its *Timestamp*. Each estimation may have some free textual *Notes*, and at least one among the following position data: the coordinates X, Y, and (possibly) Z of the point of prediction; and the *Place*(s) where the fingerprint was predicted. Observe that we can associate more than one place with a given estimation: this is the case with an algorithm that provides a probability distribution, where the single probabilities can be encoded by means of the optional attribute *Confidence*.

Observation sub-schema

This sub-schema deals with information about observations, that is, the data sensed about the environment, that compose a fingerprint. A *Fingerprint* may be associated with one or more observations, and an observation refers to one and only one fingerprint. Each *Observation* has an attribute *Type*, that specifies the kind of observation among those included in the specialization. Since such an attribute is a partial identifier with respect to *Fingerprint*, the latter may have at most one observation for each type. In addition, an observation may have a *Validity* value, that indicates whether its data are still to be relied upon, e.g., for positioning tasks. Observational data may indeed lose their reliability over time. This is the case, for instance, with WiFi fingerprints, that collect information about the detected access points, which may be turned off or relocated. Since the validity of an observation can be computed from the related *Fingerprint*'s *Timestamp*, it is actually a derived attribute.

An observation may be of different kinds (total and disjoint specialization). Think of the fact that, for instance, a typical smartphone is able to simultaneously collect several types of data. A *Cellular* observation consists of zero or more *Cells*, each one detected with its own *RSS* (Received Signal Strength). A cell is identified by the combination of *CI* (Cell Identifier), *LAC* (Local Area Code), *MNC* (Mobile Network Code), and *MCC* (Mobile Country Code). Similarly, a *WiFi* observation consists of zero or more *AP*s (Access Points), each one identified by an *ID* and characterized by a *code* and, possibly, a *MAC* number. Finally, a *Bluetooth* observation consists of zero or more detected *BT devices*, each one identified by an *ID* and characterized by a *Name*. Note that each cell, access point, and Bluetooth device may belong to one or more data sources. Again, this is quite natural, as the same cellular antenna may be detected in several scenarios. To accommodate for that, we introduced the surrogate key *ID* both for AP and Bluetooth devices. Such an attribute also allows us to discriminate, e.g., between two different APs that have been given the same *Code* in different datasets. The surrogate key is not necessary

for *Cell*, since such an entity set already features a global identifier.

The modeling of GNSS and IMU data undergoes a different logic, as they do not store information about a signal pattern received from external beacons. Specifically, a *GNSS* observation has an associated optional set of latitude (*Lat*), longitude (*Lon*), and elevation (*Elev*) coordinates (the latter is optional, as it depends on the number of available satellites, *# Sats*). As for *IMU* observation data, they can refer to different kinds of device, typically hosted on a same module: *Accelerometer* information, tracking the acceleration along axes X, Y, and Z; *Gyroscope* information, detecting the angular velocity with respect to axes X, Y, and Z; and *Magnetometer* information, measuring the magnetic field for axes X, Y, and Z. Note that an IMU observation may have several sets of data associated with it. This is the case, for instance, with a single fingerprint which contains multiple accelerometer samples that have been sensed several times over a period of time. In this case, information is modeled in a differential manner with respect to a previous fingerprint in a trajectory. Thus, a fingerprint with an associated IMU observation should also participate into the relationship *Follows*, in order to keep track of the preceding one.

To conclude, we observe that the above design choice allows us to store empty observations related to a fingerprint. This is the case, for instance, with a cellular scan which detected no cells (this may happen when the device is in an underground location).

Place sub-schema

This sub-schema models topological information about indoor scenarios. As previously pointed out, the intended goal is not that of storing extremely detailed data with which to reconstruct the exact appearance of the considered premises, but, rather, to keep track of information that may be useful for positioning purposes.

The main entity of the sub-schema is *Place*, which represents a generic spatial concept. Each instance of place is uniquely identified by a surrogate key *ID* (following the same reasoning pattern as that of *AP* and *BT device*), and characterized by a *Name* and, possibly, a *Description*. A place may belong to more than one data source (this is the case, for instance, with several datasets collected over the same premises at different times). A place is then partitioned (total and disjoint specialization) into *Building*, *Floor*, *Site*, and *Tile*. The relationship *Contains* allows one to keep track of a hierarchical structure among places.

Building represents the coarsest level of the hierarchy. Each building may have an associated *Area*, and can be adjacent to zero or more other buildings (relationship *Adjacent to building*). A building may include one or more floors.

A *Floor* may have an associated *Area* and a *Height*, and is contained in a single building. To preserve the vertical ordering of floors, we make use of the relationship *On top of*. It is worth pointing out that such a modeling decision requires one to specify all intermediate floors between any two given levels, in order to correctly and completely maintain the vertical relationships. However, such a constraint does

not limit the flexibility of the model as if some floors are not present in a dataset, they can still be added as empty (dummy) levels, without any sites or tiles included in them. A floor may contain zero or more sites, or, directly, tiles, as in the case of grid and crowd tessellations.

A *Site* represents a specific spatial area in an indoor scenario, such as a room, a corridor, a bridge between buildings, or some stairs between floors. A site may have a *Height* and an *Area*, and may belong to one or more floors. An example of the second case is an auditorium, that may have entrances on two or more different floors. A floor being used for positioning purposes always contains at least one tile.

A *Tile* represents the most basic piece of spatial information that can be stored within the system, and it acts as a bridge between the topological knowledge of the premises and the fingerprints. In accordance with the discussion in Section 2.2, a tile can be of four different types: (i) *Grid*, characterized by the four 2D *Coordinates* of its associated regular grid cell, (ii) *Zone*, characterized by the four 2D *Coordinates* of its polygon, (iii) *Logical*, possibly characterized by a pair of 2D *Coordinates*, and (iv) *Crowd*, with no other associated information. Information about adjacency is captured by the relationship *Adjacent to tile*, which may possibly track the walkability between tiles (attribute *Walkable*) and the associated traversing cost (attribute *Cost*).

Each tile is included in one and only one *Tessellation*, which should be of the same kind as that of the tile. The specialization of tessellation is total and disjoint, and thus a given tessellation contains only one kind of tiles. Note that, similarly to the case of *Observation*, *Tessellation* has an attribute *Type* as its partial identifier, and the entity set is weak with respect to *Floor* and *Data source*. The overall result is that, within a floor belonging to a given data source, we may have at most one tessellation for each kind. Thus, a floor may have a grid and a zone tessellation associated with it, each with its own tiles, but it cannot have two distinct grid tessellations.

2.3 Relational database development

In this section, we focus on the development of the relational database for indoor positioning. In particular, Section 2.3.1 discusses its logical schema, which has mainly been derived from the Entity-Relationship diagram of Section 2.2.2. Then, Section 2.3.2 deals with the problem of representing fingerprint ground truth information that can be useful for the evaluation of indoor positioning systems, an aspect which, as we will see, deserves specific attention. Although we do not describe here all the details regarding the physical implementation of the database, which include the definition of a set of data consistency constraints not directly expressible in the logical schema, the interested reader may find the source code on GitHub [37].

2.3.1 Logical schema

The conceptual schema illustrated in Section 2.2.2 can be translated into a relational schema by applying the standard mapping rules [242]. The resulting schema, depicted in Figure 2.4, includes all the tables obtained via such rules, as well as some additional tables that were required to store, for instance, type information. It is worth observing that the relational counterparts of the four ER sub-schemas can be easily identified: *Data source*, *Fingerprint*, *Observation*, and *Place*. In the following, we will describe the most important choices we took during the logical design process. In general, all composite attributes were handled by keeping their components, and we tried to reduce the size of foreign keys as much as possible in order to avoid complex join conditions and unnecessary usage of space.

The translation of the Data source sub-schema does not present any particular problem.

In the Fingerprint sub-schema, the *Fingerprint* specialization was translated by keeping just the parent entity and adding the Boolean attribute *is_radio_map* to it. As for the attributes *acquired_at_tile_place_id*, *coordinate_x*, *coordinate_y*, and *coordinate_z*, they pertain just to those instances belonging to the radio map (*is_radio_map* = *True*). The attribute *ml_purpose* has a dedicated domain, just including the strings *'training'*, *'validation'*, and *'test'*. As for the primary key, we introduced the surrogate *id*, which allows us to use simpler foreign keys when referring to fingerprint instances. Then, to enforce data consistency, we placed a uniqueness constraint over the pair of attributes *code* (the fingerprint identifier in the original dataset) and *data_source_id*. Similar considerations were made for the primary key of *estimation*, where we introduced the surrogate *id* and imposed a uniqueness constraint over the pair of attributes *timestamp* and *fingerprint_id*. In the table *device*, we find the usual surrogate key *id* and a uniqueness constraint defined over the pair of attributes *code* and *data_source_id*. Information about the device model and its type was recorded by means of two dedicated tables, to avoid unnecessary data replication and to allow for an easy extension of their allowed values. The same logic was followed for the table *user*, with the uniqueness constraint placed over the pair *code* and *data_source_id*, and the provision of a dedicated table used to store the user type.

Turning to the Observation sub-schema, we translated the *Observation* specialization by keeping just the children entities. In the table *cell*, we introduced the surrogate key *id* to avoid to deal with foreign keys consisting of four attributes. Multi-valued attributes belonging to the entity *IMU* were handled by introducing three separate tables, each one with the pair of attributes *fingerprint_id* and *epoch* as the primary key. The latter attribute can be used to sort multiple data coming from the same sensor within a single *IMU* observation. Being numeric, it can be used both as a kind of timestamp (thus implicitly conveying information about the sensor sampling frequency) or as a simple ordering integer. As for the domains of the attributes, *rss* stores negative numerical values corresponding to the decibel-milliwatts (dBm) of the received signals. Along each axis, acceleration in *observa-*

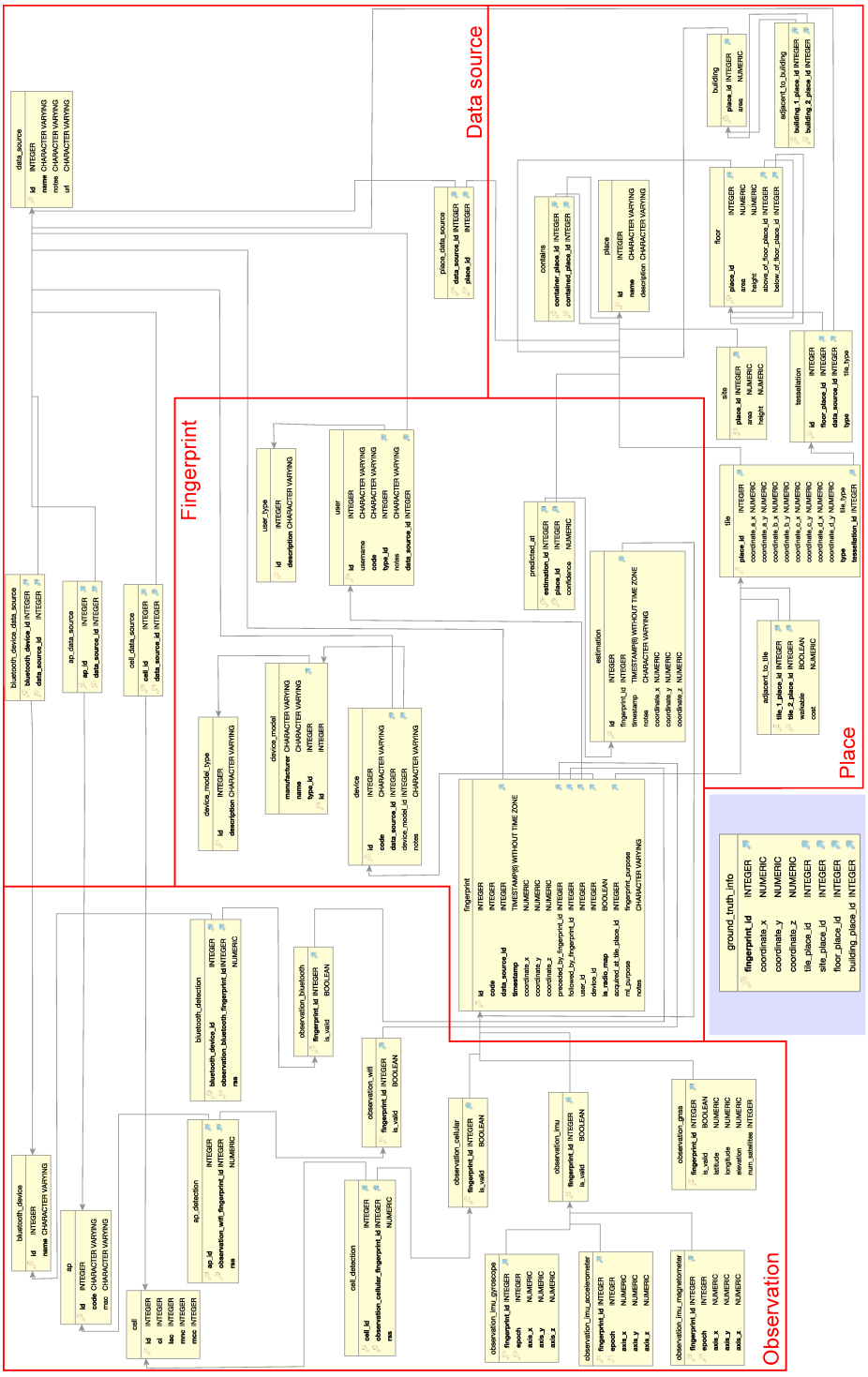


Figure 2.4: Logical schema of the indoor positioning relational database. Arrows represent foreign key directions. The red sub-schemas make up the *public* schema. The blue area depicts the *evaluation_support* schema (consisting of one table only).

tion_imu_accelerometer is measured in metre per second squared (m/s^2), angular velocity in *observation_imu_gyroscope* is encoded in radian per second (rad/s), and magnetic field intensity in *observation_imu_magnetometer* is recorded in microtesla (μT).

As for the Place sub-schema, we kept all entities involved in the *Place* specialization, given the presence of several distinct relationships involving them. As for the specialization of *Tile* and *Tessellation*, we instead kept just the parent entity and relied on the attribute *type*. The latter has the same custom domain both when used in the table *tile* and in the table *tessellation*, consisting of just the strings *'grid'*, *'zone'*, *'logical'*, and *'crowd'*. In the table *tile*, we explicitly listed all four pairs of coordinates. Such a decision allows us to have fine-grained control over the consistency of their usage. In the table *tessellation*, we introduced the surrogate key *id*, and we placed a uniqueness constraint on the triplet of attributes *type*, *floor_place_id*, and *data_source_id*. As for the attribute types, *height* and *area* are encoded in meters and square meters, respectively.

2.3.2 Management of ground truth information

We conclude the section with a short note on the management of fingerprint ground truth information. In contrast with our design choices, indeed, it may happen that a non-radiomap fingerprint is associated with some partial spatial information. As an example, this is the case with test set fingerprints in the dataset UJIIndoorLoc [256] that, although being considered as "online" fingerprints, still possess ground truth coordinate data, but are not associated with their tile or site of collection. Since information of this kind may be useful for the evaluation of indoor positioning systems, we decided to accommodate it into a separate schema, called *evaluation_support*. Within such a schema, we defined the table *ground_truth_info*, depicted in Figure 2.4, blue shaded area. No constraint on the absence of null values was enforced on the table, and thus, given a fingerprint, it allows us to store any kind of (possibly fragmentary) spatial data associated with it. Primary key attribute *fingerprint_id* is a foreign key with respect to the attribute *id* of the table *fingerprint*, while attributes *tile_place_id*, *site_place_id*, *floor_place_id*, and *building_place_id* are foreign keys pointing to the attribute *place_id* of tables *tile*, *site*, *floor*, and *building*, respectively. Such a table is particularly useful to foster reproducibility in indoor positioning experiments, as it allows one to collect and highlight the data which should be used for evaluation purposes.

2.4 Usage of the system

In this section, we first show how the proposed model is flexible enough to accommodate various indoor scenarios. Then, we discuss how the system can support non trivial tasks in the indoor positioning domain, other than promoting the research

on and the deployment of novel localization approaches.

The source code of our implementation, which includes the definition of some useful SQL queries and user defined functions (UDFs), is available on the GitHub page of the project [37]. Examples of UDFs that easily allow one to retrieve complex information are `MinimumShortestPath` (computed for pairs of heterogeneous elements, e.g., a tile and a floor, or a fingerprint and a room), `FPDistances` (measuring the distance between two fingerprints or estimations with several metrics), and `CharacterizingFP` (that determines the average fingerprint of a tile).

An online, freely accessible [38] implementation of the system has been developed in PostgreSQL [98]. Users may submit custom queries through the PGAdmin (web) interface [252] as well as relying on well-known database connectivity APIs (e.g., JDBC [60]). The database already stores more than 15 well-recognized datasets for indoor positioning [155, 156, 176, 188, 218, 241, 245, 256, 262], which can be retrieved and used as pleased. We remind all the potential users of the tool to give proper credit also to the original collectors of the datasets.

2.4.1 Representation of notable indoor scenarios

In this section, by means of a series of use cases, we show how the flexibility of the proposed model allows us to represent various indoor scenarios.

Logical tiles

Assume that we want to perform a simple tessellation of a given floor of a conference building, featuring different rooms, without resorting to the definition of complex grid or zone tiles. In such a case, we opt for a much lighter logical approach. A first solution might be that of associating with each of the rooms its logical tile, without any pair of coordinates. As an example, we may have a single logical tile representing "conference room A", acting as a purely logical link between the place (for which the semantics is defined by the tile label) and the fingerprints associated with it. However, if conference room A is quite large and has a stage far from the audience space, it might be sensible to further refine our tiling approach, for instance, to support more precise positioning tasks. We can thus specify two distinct logical tiles, the first one representing the audience space, and characterised by a pair of coordinates equal to the geometrical centre of such a space, and the second one tied, for instance, to the coordinates of the main tribune on the stage.

Inter-building connectivity

Let us consider two separate buildings connected by a bridge-like structure. The latter can be modeled as a building on its own, with a single floor, and some tiles adjacent and walkable with respect to tiles of the other two buildings. This solution can be easily generalized to a set of buildings connected by a network of underground

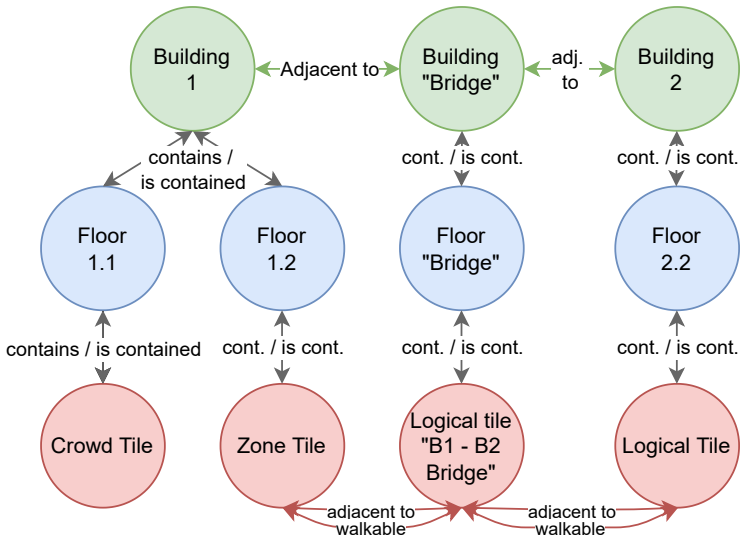


Figure 2.5: Inter-building connectivity modeling example (for simplicity, only one tile per floor is depicted).

tunnels, possibly structured on more than one level. A possible model of such a scenario is depicted in Figure 2.5.

Premises spanning over several floors

A theatre hall may have entrances on different floors. Here, also the tiles representing the hall belong to different floors. This is true for the tessellations of type *zone* and *logical*, where the tiles are connected to the *site* that corresponds to the hall (which in turn belongs to more than one floor), as well as for the tessellations of type *grid* and *crowd*, where the tile(s) are paired directly with the floors spanned by the hall. A possible representation of the resulting scenario for both cases is given in Figure 2.6.

Connectivity involving different tessellation strategies

Let $A.1$ and $A.3$ be two zone-tessellated floors of a building A, interleaved by a floor $A.2$ that follows a grid tessellation approach. Despite the heterogeneity of tessellations, it is still very easy to model adjacency and walkability relationships among tiles, so as to represent a possible path starting from a tile u on floor $A.1$ and ending at a tile z on floor $A.3$. More precisely, the path would start from the tile u; then, it would follow the walkability relationships among zone tiles till it reaches a tile v which is walkable with respect to a grid tile w belonging to the floor $A.2$. In the most general case, each grid tile belonging to the floor $A.2$ is adjacent (and walkable) to each of its neighbouring tiles, and thus the walkway would then follow the shortest path till a given tile x, vertically connected to a tile y of floor $A.3$, and

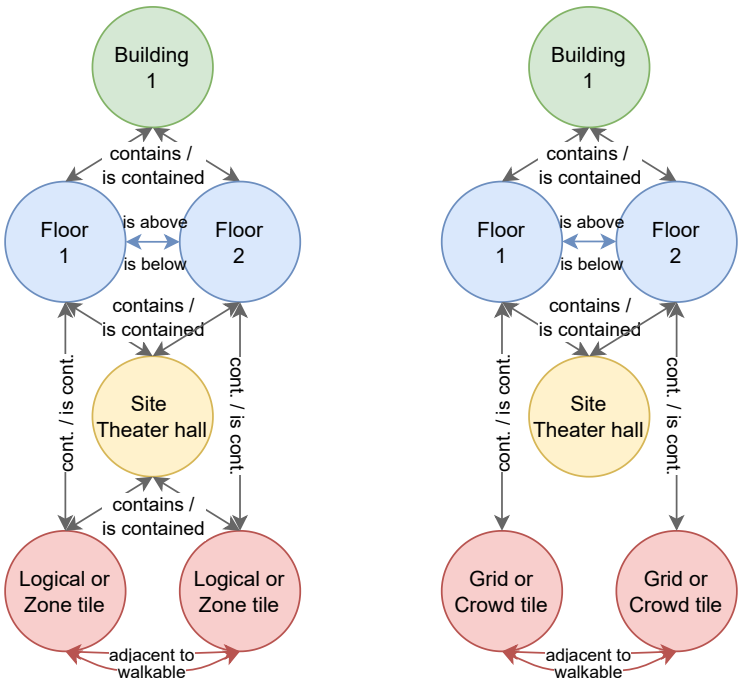


Figure 2.6: Premises spanning over several floors (theater hall) modeling example. Left-hand side denotes the case of logical or zone tessellations; right-hand side reports the case for crowd or grid ones (for simplicity, only one tile per floor is depicted).

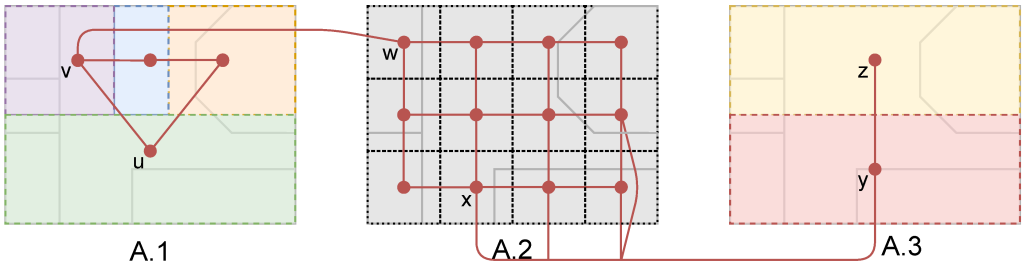
finally reach the tile z according to the relationships defined over the zone tiles of floor $A.3$. An example of topological information for such a scenario is reported in Figure 2.7.

Crowdsourced sampling

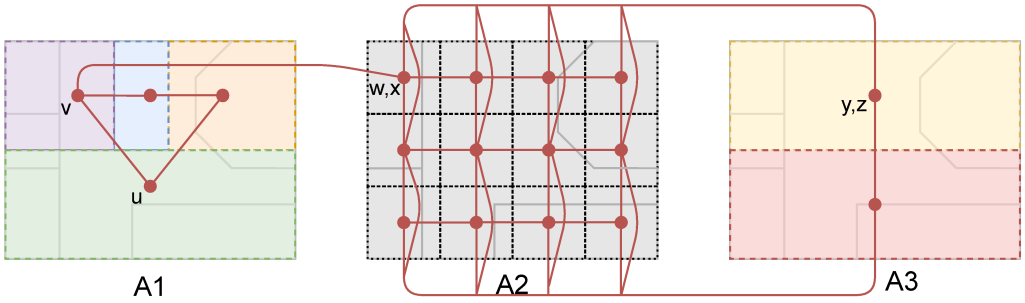
A crowdsourced collection of fingerprints performed on a floor can be supported by defining a tessellation of type *crowd* on it, consisting of just a single tile that acts as a general container. Then, collected fingerprints would all be connected to such a tile, and possibly be complemented by information about their position coordinates.

2.4.2 Support for indoor positioning tasks

The proposed modeling of the indoor positioning domain makes it possible to support a large array of interactions, ranging from very simple queries to rather advanced use cases. As we did in the previous section, we introduce some notable use case scenarios that demonstrate the potential of the system.



(a) Grid tiles have only specific inter-floor connections



(b) All grid tiles are connected with those physically adjacent

Figure 2.7: Two connectivity scenarios involving different tessellation strategies across three different floors. Floors $A.1$ and $A.3$ use zone tiles, while floor $A.2$ uses grid tiles. Red links denote adjacency and walkability relationships at the tile level (some have been compacted for simplicity).

Multi-source compositionality

Starting from the information stored within the *Place* sub-schema, it is rather easy to extract knowledge about the structure of an indoor positioning scenario in terms of, for instance, composition and adjacency relationships among sites, floors, and buildings. In doing that, information coming from different data sources can be merged (while still preserving data lineage) in order to obtain a complete picture of an indoor scenario, that may have been considered in a fragmented fashion within different studies. Information coming from multiple sources can be exploited also with respect to fingerprints. As an example, different radio maps pertaining to the same places can be merged to provide a higher number of (possibly temporally updated) observations, that can be used during the online phase of the system. In this respect, Figure 2.8a shows, for two datasets collected at the same premises one year apart from each other, how the available APs and their average RSS may change dramatically.

Multi-sensor positioning

Following the proposed approach, it is quite easy to combine WiFi and Bluetooth signals, which are the most useful information sources for fingerprinting. Such sensors, together with other point-based ones, are efficiently managed even if they are sampled at different rates and in the presence of IMU data and trajectories. This one allows to seamlessly exploit a rich set of information for positioning purposes.

More robust localization

Given a fingerprint submitted by a device, it is possible to filter the radio map so as to consider only fingerprints collected by similar devices or within a specific time window. In principle, such an ability has the potential to improve positioning performance, since devices may sense and record observation data in different manners, due to hardware or software differences. Figure 2.8b shows, for the UJIIndoorLoc dataset, how different devices indeed exhibit, for the same location, different average RSS patterns. In addition, focusing on WiFi fingerprinting, signal propagation is likely to vary in a cyclical fashion over a day or a week, due to the distribution of people within premises, which may have a perturbation effect on the signals. Restricting the attention to fingerprints recorded within specific time windows is also likely to allow for a more sensible comparison among observation data.

Complex and more expressive metrics

Storing topology information supports the development and usage of error metrics for indoor positioning that are more advanced than the commonly considered 2D distance. As an example, predicting the location of a user with a 3 meter error radius is way more serious if this brings to an uncertainty over the floor the user belongs to, with respect to an uncertainty over the position of the user within a single room. To account for that, reachability information stored by means of the *adjacent_to_tile* relation could be leveraged. As an example, it is possible to apply the UDF `FPDistances` to the two fingerprints with identifiers 525373 and 525373 (belonging to a version of the dataset UJIIndoorLoc enriched with adjacency information) to calculate, in addition to the 2D and 3D Euclidean distances, the minimum shortest path between them, that might represent the traveling effort for a user from a wrong to a right position estimate.

```
SELECT * FROM FPDistances(525373, 525405);          [178 msec, Fig. 2.8c]
```

Topology-aware positioning algorithms

Topology information also fosters the research on state-of-the-art positioning algorithms, based, for instance, on Graph Neural Networks or on other machine learn-

ing techniques, like, e.g., Hidden Markov Models, that could effectively leverage the graph structure of indoor premises in order to provide better position estimates.

Trajectory and personalised localisation

The system allows one to store subsequent online fingerprints provided by the same device or user over time. In such a way, it actually generates a trajectory of predictions, which can be exploited by a positioning algorithm to reduce the prediction error, discard outlier fingerprints, or reason about users' typical patterns. For instance, employing the UDF `TrajectoriesInPlace`, one can retrieve in an array-like format all the trajectories, that is, sequences of fingerprints, passing through a given place, e.g., tile, site, or floor. Below, we consider the tile with id 520815 (IPIN 2021 Competition Track 3 dataset).

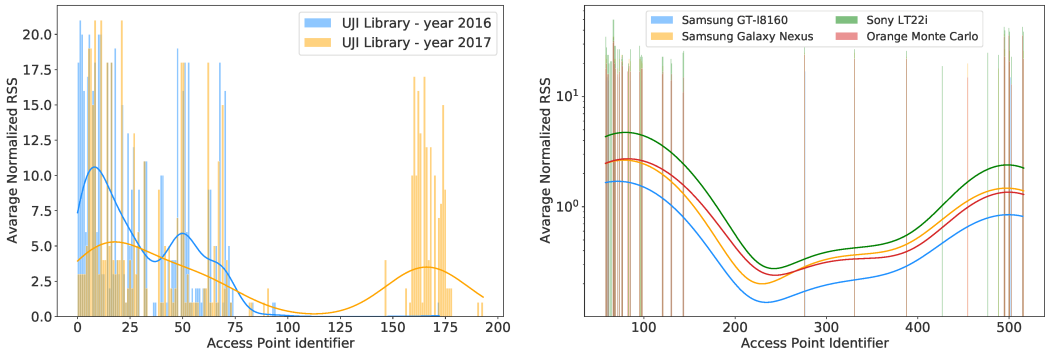
```
SELECT * FROM TrajectoriesInPlace(520815);          [130 msec, Fig. 2.8d]
```

Navigation

The system supports navigation tasks as well. As an example, a user may find the shortest path (in terms of the number of traversed tiles or associated traversability costs) to a specific location within the indoor premises, starting from his/her predicted location. This comes down to finding the shortest path between two nodes in the graph that represents the indoor scenario.

Knowledge discovery

The richness of information stored in the system, in terms both of the data and the relationships among them, makes it possible to develop a large number of unsupervised analysis tasks. For instance, usage patterns of the system could be investigated in order to discover regularities with respect to specific classes of users or time windows. Similar analyses may bring to the discovery of issues within a radio map, e.g., due to outlier fingerprints, that can then be corrected or removed. As a final example, consider the case of WiFi fingerprints collected within the same premises at different time points by different devices. By analyzing their signal patterns, it may be possible to discover regularities and differences in the access points or in the propagation of their signals. At this stage, a step further could be that of developing a normalization strategy to cope with such variations, so to consider a "time-and-device-corrected" version of fingerprints which may bring to a better online phase usage of the radio map.



(a) Average RSS observed for the same premises one year apart.

(b) Average RSS patterns of different devices for the same tile.

	euclidean2d numeric	euclidean3d numeric	msp numeric
1	63.9531078212779	[null]	18

(c) `FPDistances(525373, 525405)`. The 3D Euclidean distance is *null* as z-coordinates are not provided for the two involved fingerprints.

	trajectory integer[]
1	{520938,520939,520940,520941,520942,520943,520944,520945,520946,520947,520948,520949,520950,520951}
2	{520952,520953,520954,520955,520956,520957,520958,520959,520960,520961,520962,520963,520964,520965}
...	...
9	{520924,520925,520926,520927,520928,520929,520930,520931,520932,520933,520934,520935,520936,520937}
10	{520910,520911,520912,520913,520914,520915,520916,520917,520918,520919,520920,520921,520922,520923}

(d) `TrajectoriesInPlace(520815)`.

Figure 2.8: Outcomes of some interactions with the system.

2.5 Related systems/frameworks

A great amount of research has been done on indoor positioning systems, their applications, and related tools. Here, we focus on two areas that are closely related to the developed framework, i.e., modeling indoor premises, and comparing positioning systems. A comparative summary of notable features, with a ranking of their importance and support capabilities by the considered systems/frameworks, is given in Table 2.1.

Table 2.1: Summary of the notable features and their importance and support capabilities by the considered systems/frameworks

System / Framework	Compare IPSs	Model Indoor Positioning	Multi-sensor Support	Building Topology	Indoor Mapping	Detailed Scenarios	Enforce Data Consistency
IndoorGML [59]		*	*	***	***	*	
IFC [41]					***	***	
EvAAL [16]	***		***				
LSR [215]	***		*				
Web tool [146]	**						
Web tool [186]	**						
our	*	***	***	***			***

*** = prominent feature of the system, full/good support capabilities; ** = feature of the system but there are better competitors, some support capabilities; * = not an aim of the system, which anyway provides limited support capabilities; IPS = Indoor Positioning System.

Modeling indoor premises

Several proposals have been made on how to model indoor premises [2, 141, 190], although a general consensus is still missing. Many of the most advanced formats and standards, e.g., IFC – Industrial Foundation Classes [41] or CityGML [58], pay a special attention to the description of scenarios with all their details [150]. IndoorGML [59, 119], instead, mainly focuses on the description of the structure of a premise, with a focus on the arrangement of spaces and their relationships. All these approaches (especially the latter) are quite flexible and provide some support to positioning tasks through additional application layers.

Our contribution differs from them in several respects. First, its main goal is to store and support (fingerprint-based) positioning, and only in accordance with that to provide topological information about the premises. From this point of view, the proposed solution is close in spirit to IndoorGML, which also has the latter capability. However, our modeling goes in the direction of what should be implemented within the prospective IndoorGML 2.0, allegedly designed to also work with topological information only [69]. Another key difference with respect to existing solutions is the possibility to easily interact via SQL (in fact, the proposed one is the first and only relational database-centric system). For the sake of completeness, it is worth remarking that IndoorGML is better than our approach in handling indoor modeling/mapping and navigation tasks. However, as already remarked, it is important to keep in mind that our motivations and goal are quite different.

Comparing positioning systems

Mainly motivated by the high diversity in metrics, datasets, and result reporting approaches in the literature, several authors have studied how to enable a fair comparison of indoor positioning systems, also implementing several tools [209, 258]. The EvAAL framework [16] has been largely adopted by indoor positioning compe-

titions [207, 208, 216, 260]. Its main goal is to enable fair, realistic, and systematic comparison of positioning solutions, especially in the case when they rely on different methodologies and sensors data. It achieves that through its core principles: natural movement of an actor, realistic environment, realistic measurement resolution, and, third quartile of point Euclidean error. Localisation Systems Repository [215] aims at supporting continuous, reliable, and accurate positioning on smartphone devices, providing a large benchmark suite and a repository for localisation systems source code. Web platforms for the evaluation and comparison of indoor localisation algorithms have been proposed in [146] and [186], respectively focusing on radio frequency-based and fingerprinting data. Our framework largely differs from all previous ones, that are basically oriented towards storing tabular datasets and comparing algorithmic performances as: (*i*) it supports the industrial-level deployment of indoor positioning systems, (*ii*) it stores data in a normalised way, highlighting their relationships and making them easier to exploit, (*iii*) it embeds information on building topology, and (*iv*) it supports advanced concepts such as trajectories and multi-sensor data.

II

On the Relationships Between Fingerprints and Spatial Knowledge

3

What You Sense Is Not Where You Are

As we already highlighted in Section 1.3, over the years, many different algorithms and data representations have been used in WiFi fingerprint-based indoor positioning (see, for instance, [12, 167, 193, 223, 261, 295]). Some of them follow traditional paths, others are more original (for a survey, see [103]). Classical solutions include the so-called deterministic algorithms like, e.g., (k-)Nearest Neighbor [12], as well as probabilistic approaches, which, given a fingerprint observation, determine its most likely location in a stochastic way [295]. Advanced solutions, based on machine and deep learning, have also been employed [193]. All these families of algorithms have been paired with different kinds of fingerprint representation, including ranking-based ones [223], and with different normalization functions [261].

However, the vast majority of the work done so far focused on achieving the best possible performance in the positioning task, without analyzing in detail the behaviour of the chosen fingerprint representations and of the related comparison methodologies. Indeed, also when metrics have been taken into account in a systematic way [261], the main objective has been that of understanding which was the best performing one in position estimation tasks.

In this chapter, we do something different, abstracting by the problem of positioning, with the goal of understanding what amount of spatial information is incorporated by and we can extract from the fingerprints. To such an extent, we systematically study the relationship between the high dimensional vector space of the fingerprints and the real-world geometrical space, where the user/device position is meant to be found. Thus, our main research question is more profound and fundamental than the actual problem of positioning, as it pertains to understanding whether the fingerprint vector space, paired with classical metrics used in fingerprinting, encodes enough spatial information to effectively represent the real world and if it is possible to exploit such a kind of knowledge. Even though the addressed problem and the positioning task might appear to be very similar, we will show that they are surprisingly different.

To find adequate answers to our questions, we characterize the behaviour of normalization functions and metrics at different granularity levels, that is, with respect

to the whole scenario, or focusing on a single building rather than a single floor. Our main finding is that conventional normalization functions and metrics do not allow one to establish a meaningful relationship between fingerprints and the underlying real-world locations, although some of them (e.g., cosine distance) are better than others. Such discoveries might have also strong practical implications, such as, for instance, in the choice of positioning algorithm to be used, or in explaining why deterministic approaches employing classical metrics are effective, but recent improvements are somewhat limited. With the aim of making our analyses as comprehensible and generalisable as possible, and also understanding to which extent our problem is linked with the surrounding environment, we take into account 15 of the datasets introduced in Section 8.3.1, which largely differ from each other in contents and size.

The chapter is organized as follows. In Section 3.1 we provide an overview of the few works that tried to address questions similar to ours, so as to define the starting point of our journey. In Section 3.2 the technical aspects at the core of the analysis are illustrated. Finally, the outcomes of the experiments are thoroughly discussed in Section 3.3.

3.1 Fingerprinting beyond positioning accuracy: the point so far

Even though most contributions about fingerprinting focus on issues related to the accuracy of the localization, there are some exceptions where the problem is addressed from a broader perspective. This is the case, for instance, with [117], where the authors analyze some meaningful properties of fingerprints, like RSSs distribution and APs interference, pointing out their impact on the pattern recognition techniques exploited to solve the positioning problem. In [118], various phenomena that influence the performance of fingerprinting systems are identified through an extensive statistical investigation of RSS features. An analysis of the sources of large positioning errors in (deterministic) fingerprinting has been recently made in [262]. In [189] the authors investigate the performance of simultaneous collection of WiFi data through multiple interfaces, leading them to observe that observations are weakly correlated, allowing them to exploit such differences to improve positioning accuracy. Finally, the quality and robustness of crowd-sensed WiFi fingerprints in the context of indoor positioning have been investigated in [203], where the authors report statistics on this kind of data and discuss the accuracy of positioning based on them. The take-home message is that databases similar to those created by trained personnel can be obtained by collecting a sufficiently high number of crowd-sensed fingerprints.

No one of the above studies, however, explicitly investigates the relationships between geometrical and fingerprint spaces. To the best of our knowledge, the only

existing contributions are the following ones. An application of Manhattan distance to a relatively small dataset has been described in [67]. The outcome of the experimentation is that the two spaces are weakly connected and their relationship can be modelled by a quadratic function. Such a work has been later extended to Euclidean and infinity norms, limiting the attention to very short spatial distances [182]. The additional experimentation confirmed the difficulty in achieving a satisfactory mapping between the two spaces. The infinity norm emerges as the best performing one, according to a criterion based on standard deviation, and, in general, short distances better preserve the relationship between the two spaces, thus allowing algorithms based on the (k-)Nearest Neighbor one to be successfully applied to indoor positioning. The same conclusion was reached in [183], where information about the relationships between the two spaces is exploited as a training phase estimate for the expected positioning error. Finally, an application of positioning algorithms based on Nearest Neighbor to fingerprints belonging to either a real-world dataset or a simulated one (using both a quantized and an optimistic signal representation) is described in [262]. The analysis of the correlation between the distance in the fingerprint space and the positioning error shows that it is lower in the quantized (i.e., classical WiFi RSS values) than in the optimistic (decimal precision RSS) scenario. The situation is even worse when real-world data are considered, as in that case there is no correlation at all between the two.

Our work, described in this chapter, significantly differs from all the above ones. Indeed, there is some evidence from the literature that there is not a linear proportion between fingerprint and real-world distances. However, existing results make use only of some specific metrics, such as Manhattan and Euclidean, to measure the distance between pairs of fingerprints, and they do not analyze in detail the influence of meaningful parameters like, for instance, the applied normalization functions, the area where data are collected (a set of buildings, a single building, or a portion of a building), and the considered range of distances.

In the following, we investigate the relationships between the fingerprint and the real-world space in a comprehensive and systematic way, and, for the first time, we study the problem per se, in order to understand whether (and how) reliable spatial knowledge can be obtained from fingerprint data, irrespective from the positioning task. To this end, we take into consideration a variety of metrics and normalization functions, as well as multiple independent environments (datasets) at different granularity levels (see Section 1.3.2, and Section 1.3.3).

3.2 Experimental setting

We first carry out and carefully analyze a series of correlation-based experiments on the dataset `UJI 1` (the most complex one); then, we generalize the findings to

14 other datasets[1]. We consider a number of elements that may have an impact on the relationships between the fingerprint and the geometrical space: (*i*) the normalization function applied to the fingerprint data; (*ii*) the distance function used in the fingerprint space; (*iii*) the distance used to compare real-world locations (2D or 3D spatial Euclidean distance); and (*iv*) the granularity level at which we analyze the scenario (we can either refer to the whole dataset or restrict our analyses to points belonging to the same building or even to the same floor).

To determine whether one or more combinations of the above elements are good enough to represent the real world in the fingerprint space in terms of distances between pairs of locations, we make use of Pearson correlation. More precisely, we asked ourselves if there exists a distance function to exploit in the fingerprint space which is (linearly) proportional to the one used in the 2D/3D geometrical space.

Such a problem is clearly much harder than just exploiting metrics to find the real-world location closest to a given fingerprint (the classical indoor fingerprint-based positioning with (k-)Nearest Neighbor algorithms), but easier than reconstructing the actual geometrical distances in the fingerprint space. As for its relevance, it suffices to observe that the existence of such a proportionality would allow one to derive spatial knowledge on the application domain from the fingerprint space, without making use of any explicit geometrical representation of it. Such an ability would relieve the burden of tasks like radio map construction and maintenance, and even provide the basis for some new approaches to positioning.

Let us describe the pipeline of the experimental evaluation. First, we randomly selected 400 samples from the dataset `UJI 1`, at a given granularity level, to roughly check the ability of the fingerprint space to preserve spatial properties. Then, for each viable pair of fingerprints[2], we computed the real-world spatial 2D/3D distance, and all possible variations of fingerprint normalization functions and metrics. Finally, we evaluated the Pearson correlation between all the resulting fingerprints and spatial distances. For the sake of proper evaluation, for each granularity level, the whole process was repeated 100 times, allowing us to synthesize the findings by means of statistical summaries like centrality and dispersion measures.

The second test aimed at looking in more detail at the emerged correlations. As a matter of fact, the goodness of a result may depend on an overall well-behaved trend, or the high linearity of the considered relationship may be confined to some spatial distance ranges. To avoid misinterpretations, we studied how the correlation changes as the maximum allowed real-world spatial distance is shortened.

Third, we assessed the influence of the 2D vs. 3D spatial distances on the studied relationships. Non-trivial changes are to be expected when different granularity levels are considered; in addition, some normalization functions/metrics might be better in reconstructing the 2D rather than the 3D world.

[1]Note that since we consider datasets as whole `TUT 3` and `TUT 4` are identical; thus, we keep only the former.

[2]Given two fingerprints (identified by) $FP_1, FP_2 \in \mathbb{N}$, we only considered the pair (FP_1, FP_2) with $FP_1 < FP_2$.

Finally, for the sake of generalizability, we repeated the first experiment on all 15 datasets considering 100 randomly-selected samples. On the one hand, since the number of observations per granularity level available for some datasets is very low, we reduced the sample size from 400 to 100 to adopt a common setting for all of them; on the other hand, using 400 samples for the evaluation of the dataset `UJI 1` allowed us to better assess some complex phenomena. Then, to check whether the results are consistent across the datasets, we used Spearman, that is, ranking, correlation.

3.3 Results

In this section, we describe and thoroughly analyze the outcomes of the experimentation. We start with some remarks on the ability of the various combinations of metrics and normalization functions to preserve the spatial distance relationships in the fingerprint space. As we shall see, none of them has a satisfactory performance; however, some interesting patterns emerge, notably when dealing with different granularity levels. Then, we shift the focus to the analysis of the behaviour of metrics at different spatial distances. Some non-trivial facts are uncovered, like, for instance, the fact that metrics exhibit different representational abilities at different distances. Finally, we discuss the generalizability of the analysis and of its outcomes to all the datasets: while the relationships among normalization functions and metrics are mostly preserved, there are a couple of significant exceptions.

3.3.1 Quantitative correlation analysis

Figure 3.1 shows a boxplot of the correlation performance for each combination of metric and normalization function, obtained from 100 runs on the dataset `UJI 1` at three different granularity levels. Each box extends from the first to the third quartile values of the data, with a line at the median. Whiskers extend to the smallest and largest observations which are not outliers (considering 1.5 times the interquartile range). We report only the results for the 3D case, being almost identical to those for the 2D one. Such a similarity brings us to our first conclusion: the task at hand is equally challenging for both kinds of spatial distance. All metrics are far distant from a perfect correlation, suggesting that they all fail to capture the characteristics of complete and complex spaces. Nevertheless, we observe that, as the environment becomes more and more constrained, e.g., with measurements taken from the same floor, the correlation improves. This confirms what is typically observed in indoor positioning: the best estimations are often provided by hierarchical approaches.

As for the normalization functions, *Positive* and *Zero-to-One*, whose overall performances are very close to each other, are the best ones. The *Exponential* normalization causes a relevant downgrade of the correlation when combined with *Sørensen*,

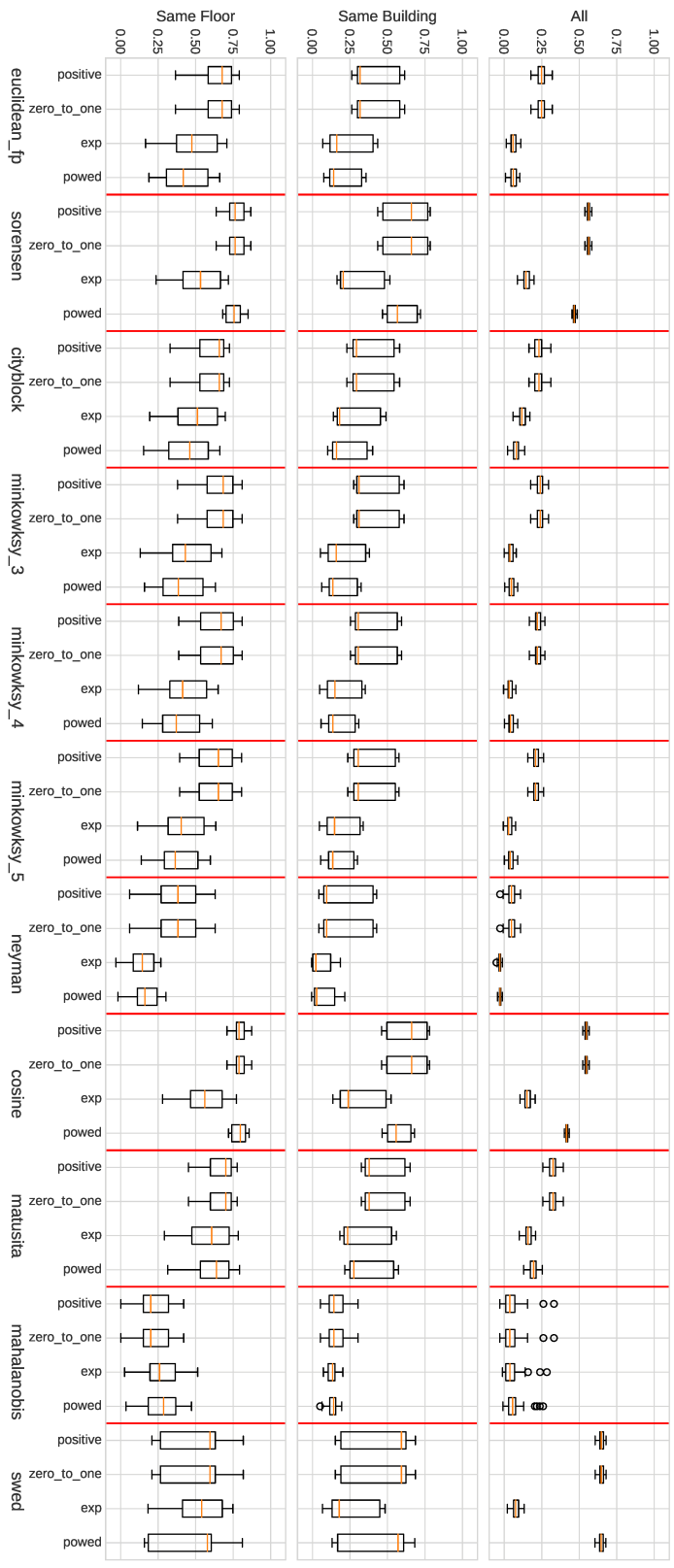


Figure 3.1: Correlation performance on the dataset UJI 1 for all the combinations of normalization functions, metrics, and granularity levels (3D case). Each boxplot is determined by the correlation values obtained from 100 different runs.

Cosine, and *SWED* metrics, while just a slight reduction is observable in the case of *Matusita*. As for *Minkowsky*-based measures, *Exponential* behaves like *Powed* normalization. *Powed* performance is somehow in between those of *Exponential* and *Positive*.

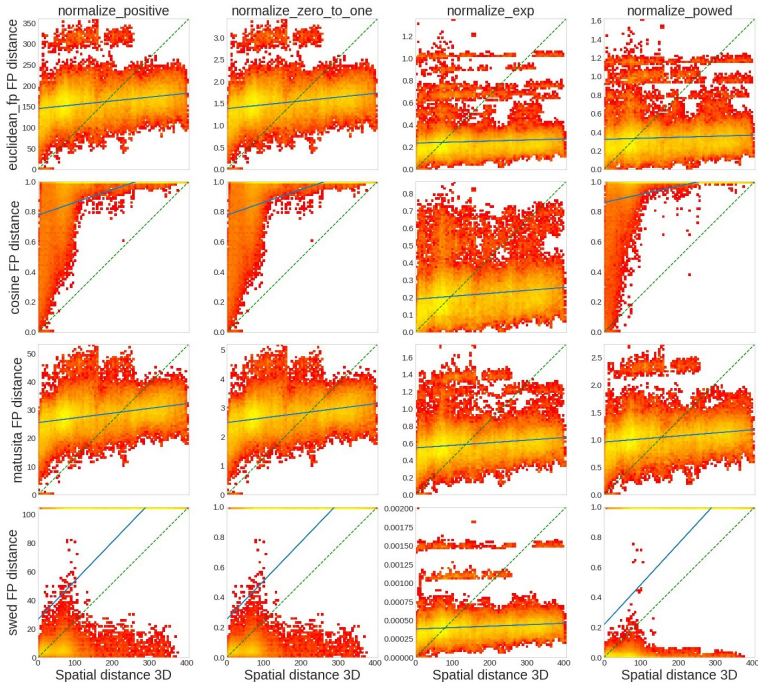
As for the metrics, *Sørensen* and *Cosine* have the best performance, followed by *SWED*, *Matusita* (which is the least affected by normalization functions, excluding *Mahalanobis*), and *Minkwosky*-based ones (which all exhibit a similar behaviour); *Neyman* and *Mahalanobis* are drastically worse. The best overall performance is that of *Cosine* distance in combination with *Positive* and *Zero-to-One* normalization functions; such an outcome contrasts with the one reported in the indoor positioning literature for this dataset, which suggests a superiority of *Sørensen* with *Powed* normalization.

Figure 3.1 also allows us to analyze the variability of the performance across the runs. If the behaviour of a combination is stable, that is, weakly dependent on the chosen fingerprints (and, thus, on the corresponding environment), the associated boxplot should collapse around the median. While this is actually the case when we consider the whole dataset, the situation changes when we restrict our attention to the same building or floor, where boxplots encode information coming from different areas/environments (the given floor or building is chosen independently at each run). Overall, looking at the boxplots, we can conclude that almost all the metrics exhibit an unstable behaviour (especially *SWED*); once more, *Sørensen* and *Cosine* appear to be the most performing, notably when restricting to a single floor, rather than to the same building.

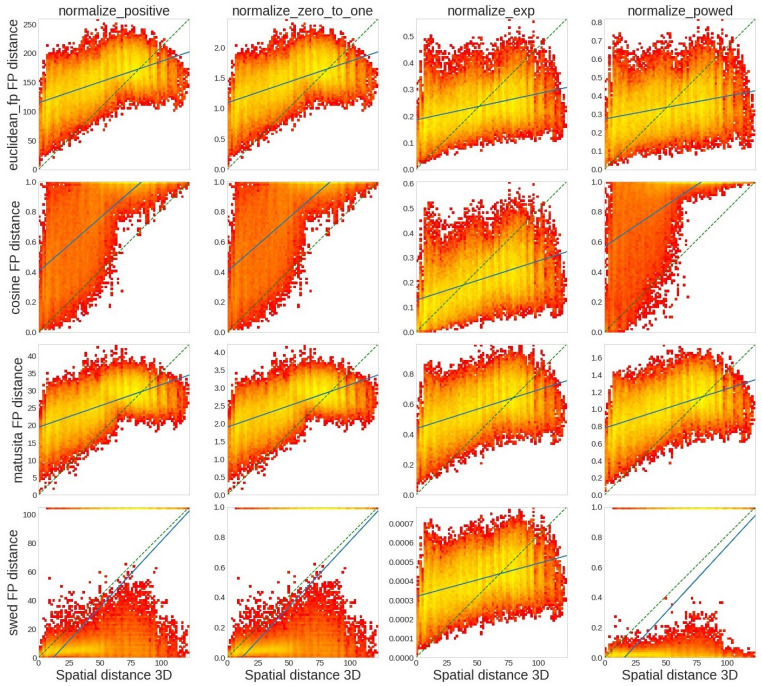
3.3.2 Qualitative correlation analysis

The results of the quantitative analysis allowed us to gather general knowledge of the performance of the considered combinations. Now, we take a closer look at the outcomes of a representative run to check whether some patterns linked to spatial distances emerge. As a matter of fact, a correlation result may be good thanks to a global behaviour, or because of the considered relationship being highly linear within some spatial distance ranges. Figure 3.2 shows the behaviours in the 3D case when instances are sampled across the entire dataset (Figure 3.2a), only from the same building (Figure 3.2b), and only from the same floor (Figure 3.2c). Brighter colours denote a higher point density. For the sake of clarity, we report the results for the most representative metrics only.

Looking at Figure 3.2, we first notice that, in general, as the granularity level decreases, the plots show a lower dispersion. Whatever the granularity, comparing the overall and optimal trends, it emerges a tendency to an overestimation of the fingerprint dissimilarity approaches, especially for what concerns shorter distances. As for the normalization functions, we observe that the *Exponential* one tends to make all trends similar, across the metrics. As for the metrics themselves, the high correlation results for *Cosine* and *SWED* seem to be largely determined by

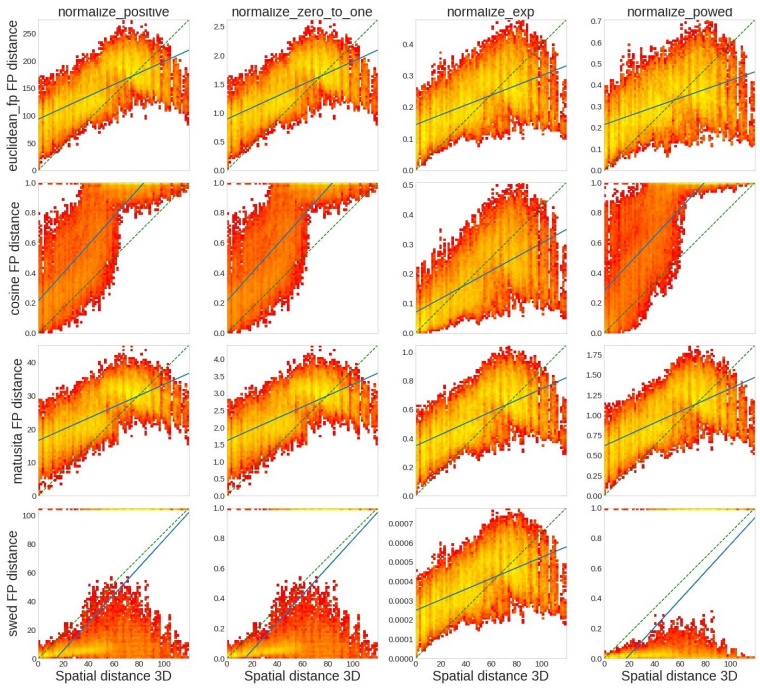


(a) All



(b) Same building

Figure 3.2: *Euclidean*, *Sørensen*, *Matusita*, and *SWED* behaviours at 3 granularities. Brighter color for higher density; correlation trend (solid line); ideal case (dashed).



(c) Same floor

Figure 3.2: *Euclidean*, *Sørensen*, *Matusita*, and *SWED* behaviours at 3 granularities (cont'd). Brighter color for higher density; correlation trend (solid line); ideal case (dashed).

a saturation behaviour (especially on distant observations) at all granularity levels (the same holds for *Sørensen*). This counterbalances the high dispersion of points visible from medium to low distances, and it can be attributed to the comparison of fingerprints without common APs. If we focus on observations taken from the same building or floor, the trend seems to change around a spatial distance of 70 meters for all the metrics. While each metric has its own peculiarities, e.g., *Euclidean*, *Matusita*, and *SWED* ones show an inversion, and *Cosine* one has a stepped curve, the underlying cause is most probably the same, although difficult to state exactly. A possible explanation can be given by measurements that are spatially distant and characterized by overall low RSSs: in evaluating fingerprint dissimilarity, their contribution might be similar to that of missing APs, misidentifying the differences and leading to an underestimation.

3.3.3 Impact of spatial distance

To better understand the patterns that emerged from Figure 3.2, Figure 3.3 presents the correlation trends when different upper bounds on the spatial distance are applied. We focus on the case of the *same building*, which turned out to be the most interesting one, and we omit the results for *Minkovsky_3*, *Minkowsky_4*, and

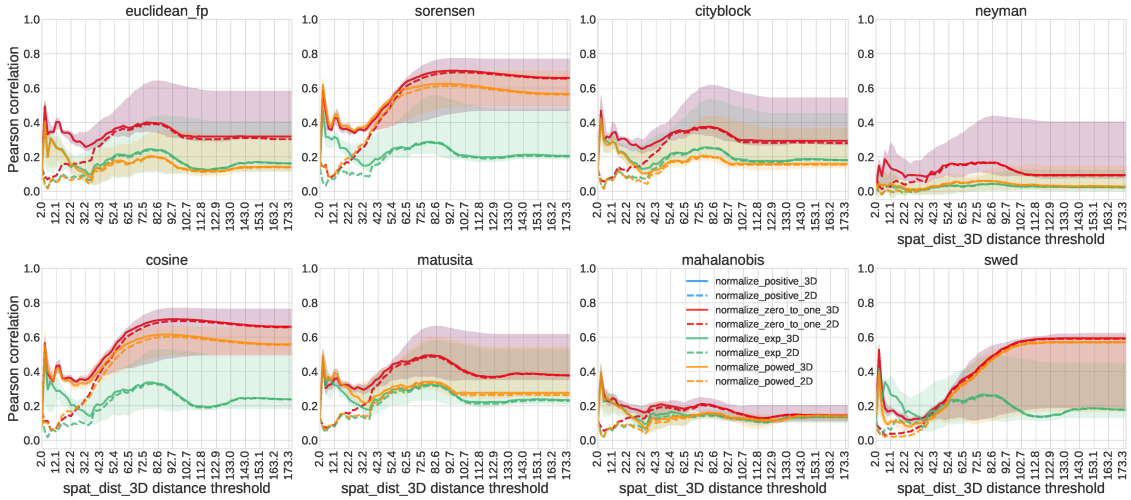


Figure 3.3: Correlation values at different maximum spatial distances (samples from same building). Values for the median (3D: solid line, 2D: dashed line), 25th and 75th percentiles (shaded area, just for the 3D case) are reported. Curves for *Positive* (blue) and *Zero-to-One* (red) perfectly overlap.

Minkowsky_5, which are very similar to those for *Euclidean*.

As for the normalization functions, *Positive* and *Zero-to-One* show again an overlapping behaviour, and the impact of normalization functions on the correlation results is negligible for extremely low spatial distances. Overall, a common trend emerges. As the maximum spatial distance threshold is reduced, the correlation slightly increases till the former reaches 80 meters, except for *SWED*, which shows a steady curve. Thereafter, a substantial decrease follows. At this stage, variability decreases as well. Then, just looking at the 3D case, when the spatial distances become lower than 30/40 meters, the correlation tends to increase again, to finally collapse towards 0 for points which are spatially very close. As for the 2D case, we observe that the second increase in correlation is missing, if not with an extremely low magnitude and for very low threshold values. Thus, in the case of a single building, the behaviour of 3D is better than that of 2D. Of course, the fact that differences between 2D and 3D are perceivable on the left side of the graphs should come as no surprise, since, proportionally, information about the altitude has a higher impact on 2D-closer points, as confirmed by Figure 3.4. Altogether, the observed trend suggests that, as spatial distance decreases, there is not a linearly proportional reduction of the distance values computed among fingerprints: metrics seem to be able of characterizing points that are far away from or very close to each other, while struggling with medium distances.

To further analyze such a phenomenon, look at Figure 3.5, where a zoomed version of *Cosine* graph is depicted. The peak in correlation values on the left clearly occurs in conjunction with a 3D distance of around 4 meters. As each floor

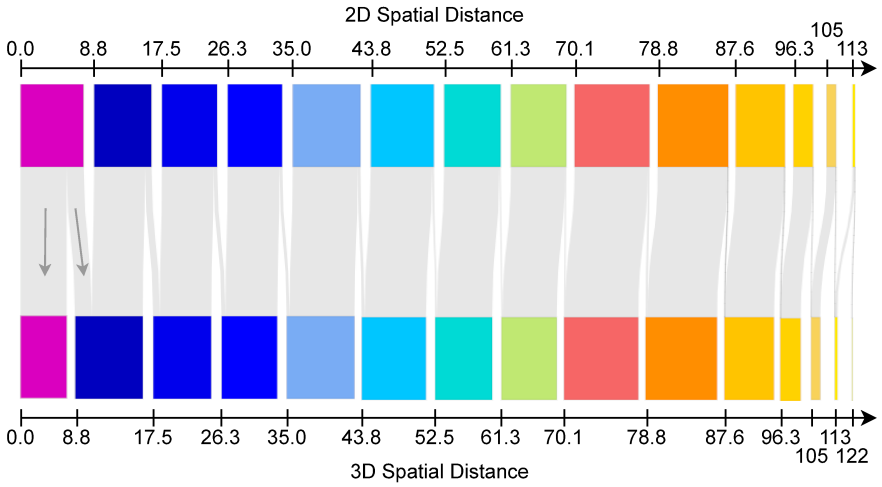


Figure 3.4: Changes to the distances distribution when 3D instead of 2D spatial distance is considered. Observations are binned considering their spatial 2D and 3D distances. Distances increase in the 3D case.

is 3.7 meters high, we can hypothesize an effect brought by the altitude coordinate. Obviously, information about the altitude contributes to an increase in the spatial distance between measurements, and the effect is the same as that of moving the related points towards the right in Figure 3.2 (fingerprint distance is unchanged). Given the tendency of fingerprint metrics to overestimate, the net result is that such points are actually pushed closer to the ideal case, boosting the performance. As shown in Figure 3.6, this seems to be due to the fact that locations at a 3D spatial distance around 3.7 meters are almost vertically aligned, thus belonging to two contiguous floors. Interestingly, for both shorter (fingerprints belonging to the same floor) and longer (mixed situation) distances the correlation drops. In conclusion, the introduction of 3D distance allows one to better preserve the spatial relationship in the fingerprint space, also suggesting an ability of metrics to characterize fingerprints floor-wise.

3.3.4 Dealing with fingerprints with disjoint access points

The influence of pairs of fingerprints without common APs on the saturation behaviour of some metrics, like, for instance, *Cosine*, requires a closer look. Figures 3.7, 3.8, and 3.9 are exactly the same as Figures 3.1, 3.2, and 3.3, respectively, but for the removal of totally unrelated fingerprint pairs.

Figure 3.7 shows that metrics behave differently on the pruned dataset, depending on whether or not they exhibited a saturation behaviour on the original unpruned dataset. On the one hand, saturating metrics show a decrease in the correlation, that is more pronounced at higher granularity levels. This is true, in particular, for *SWED*, that becomes the less performing distance in the most complex scenario.

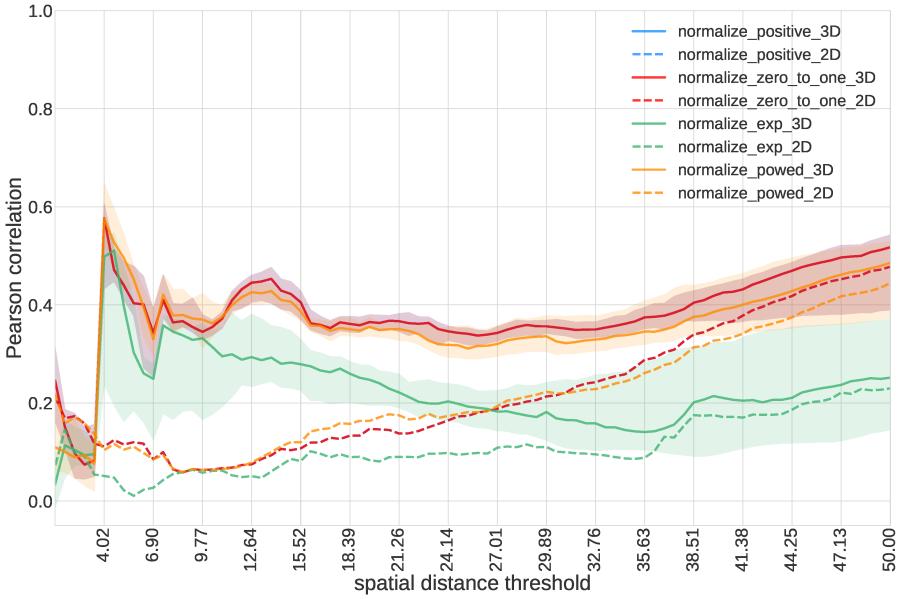


Figure 3.5: Correlation values for *Cosine* metric for low spatial distances. Curves for *Positive* (blue) and *Zero-to-One* (red) perfectly overlap.

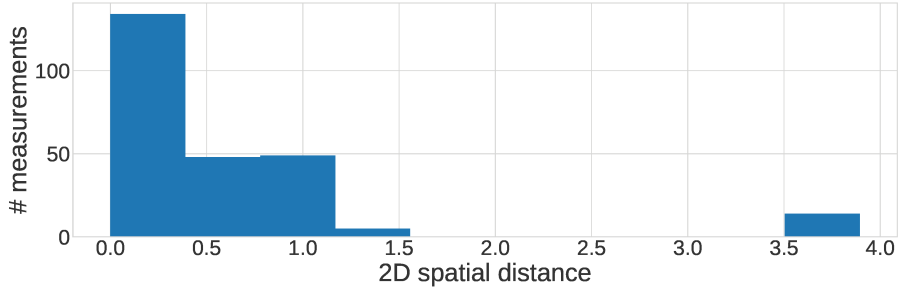


Figure 3.6: Distribution of 2D distances when 3D distances $\in [3.6, 3.8]$.

On the other hand, metrics without a saturation behaviour on the original dataset show a slight increase in their overall correlations. When only data from a single floor are considered, a decrease of the variability can be detected for all the metrics.

The same picture emerges from Figure 3.8. Compared to Figure 3.2, all clusters of points in the top part of the graphs that were exhibiting a saturating behaviour are missing. No other significant change to the shape of the graphs or to the overall trends are perceptible, except for a rescaling of *SWED* and a rarefaction of the right side of the graphs for *Euclidean* and *Matusita* in the most complex scenario.

Figure 3.9 confirms the performance drop at all thresholds for *SWED*. Moreover, metrics exhibit a behaviour smoother than before for large spatial distances, and the peak in the correlation for very small spatial distances is less marked.

The latter analysis shows that some measures, like, e.g., *SWED*), are heavily

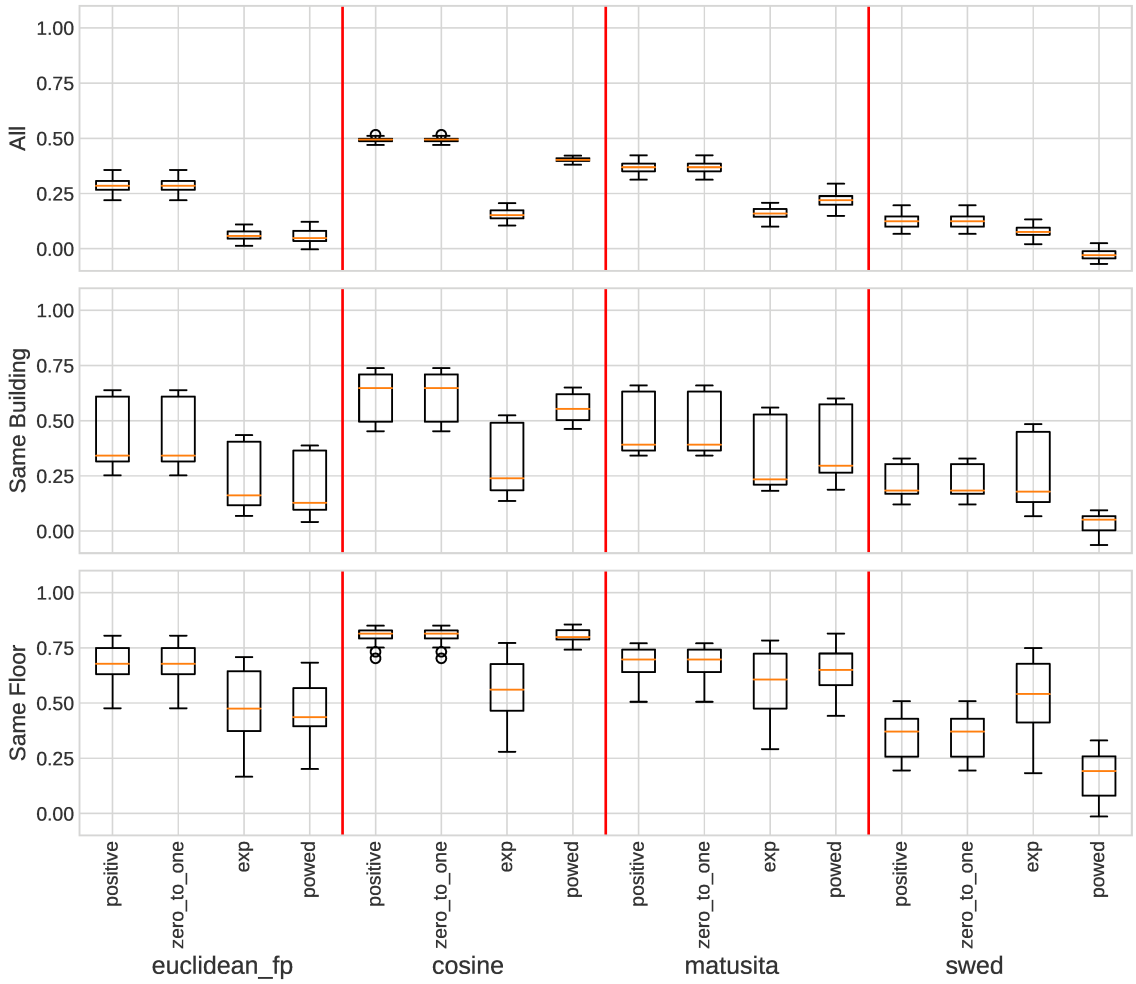


Figure 3.7: Correlation results on FP pairs with at least a common AP.

affected by the presence of pairs of fingerprints devoid of common APs, while others have a more uniform behaviour. Other factors, like the normalization functions, do not seem to play a relevant role. The take-away message for applications is that metrics with a homogeneous behaviour in different settings should be preferred.[3]

3.3.5 Generalizability of the achieved results

We conclude the section by checking whether the relationships between metrics and normalization functions that we observed for the dataset `UJI 1` can be generalized to the other 14 datasets. To this end, we make use of Spearman ranking correlation: if the ranking is preserved across normalization functions/metrics/datasets, then we conclude that their behaviour resembles that of `UJI 1`.

[3]It is clearly worth investigating extensions of existing metrics that allow them to effectively manage fingerprint pairs devoid of common APs.

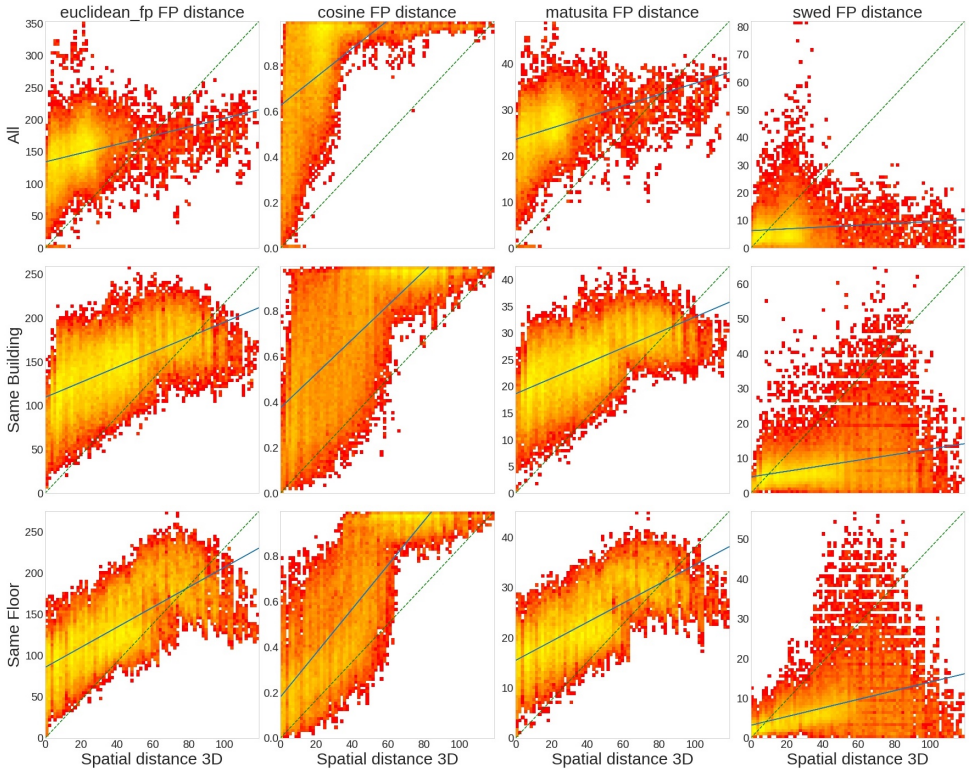


Figure 3.8: Metrics behaviours on FP pairs with at least a common AP. Only *Zero-to-One* normalization function has been considered.

To assess the stability of the combinations of metrics and normalization functions, we proceeded as follows. For each combination, we evaluated its median correlation value in each dataset on 100 runs. Then, we ranked the combinations from the best (rank = 1) to the worst (rank = 44, given 4 normalizations and 11 metrics) performing one, for each dataset. Next, for each combination, we computed the mean and standard deviation of its ranks across the datasets, under the assumption that if a combination has a stable behaviour, then its rank is more or less the same in all datasets. The outcomes of the analysis are reported in Figure 3.10. Overall, results tend to be preserved across the datasets, with a slightly higher stability in the *same floor* case. *SWED* looks problematic as, despite its apparently good performance, it turns out to be highly unstable (its behaviour seems to strongly depend on the environment). Apart *SWED*, the most unstable and worst performing metric is *Neyman* in the *same building* scenario, that nevertheless does not exceed a standard deviation of 9 (over rankings of length 44). *Sørensen* and *Cosine* (especially the latter), combined with *Positive* and *Zero-to-One*, are, by far, the best performing combinations and among the most stable ones.

Figure 3.11 shows the ranking correlations for normalization functions across the datasets. They have been established as follows: first, for each normalization function, we computed its median correlation value across the metrics for each dataset;

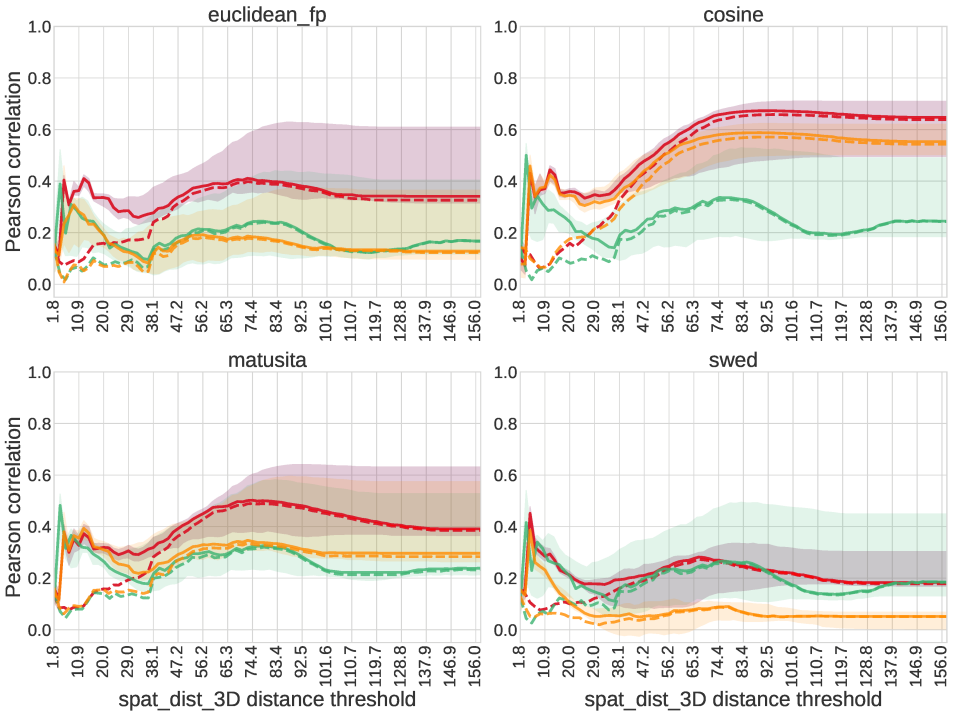


Figure 3.9: Correlation values for FP pairs with at least a common AP at different maximum spatial distances (same building).

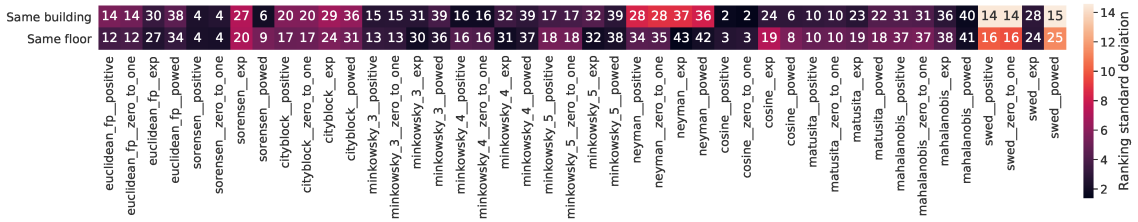


Figure 3.10: Ranking mean (value in the box) and standard deviation (color of the box) for each combination of normalization functions and metrics, across the datasets. Results for the granularities *same building* and *same floor* are reported.

then, we evaluated the ranking correlation between all possible pairs of datasets. Due to its very high instability, *SWED* has been excluded in the analysis to provide more accurate, general, and unbiased results. A strong correlation is indeed present in both the *same building* (upper triangular part) and the *same floor* (lower triangular part) scenarios. Thus, normalization functions tend to preserve their relative good and bad behaviour across the datasets.

As for the metrics (again, *SWED* excluded), we followed a similar pattern, the only exception being that the median was calculated for each metric across all the normalization functions. As shown in Figure 3.12, an almost perfect correlation

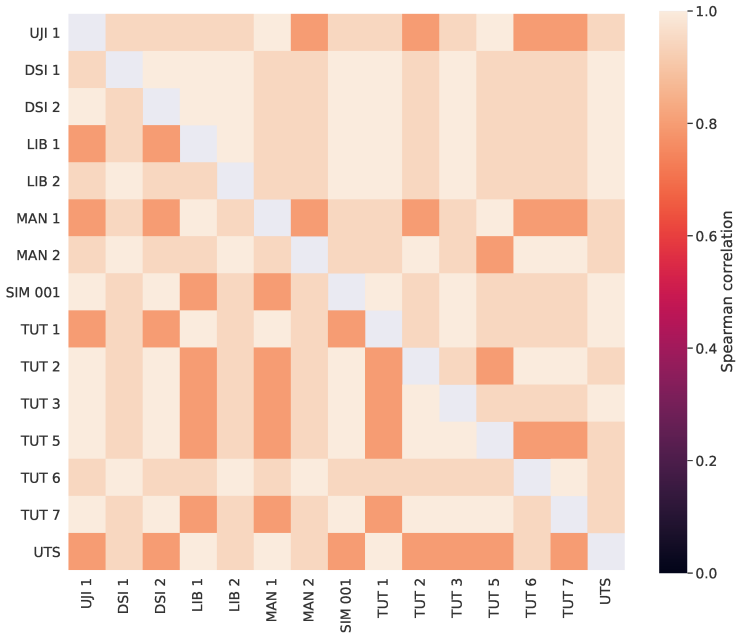


Figure 3.11: Ranking correlation for normalizations across datasets. Upper triangular matrix: *same building*; lower triangular matrix: *same floor*.

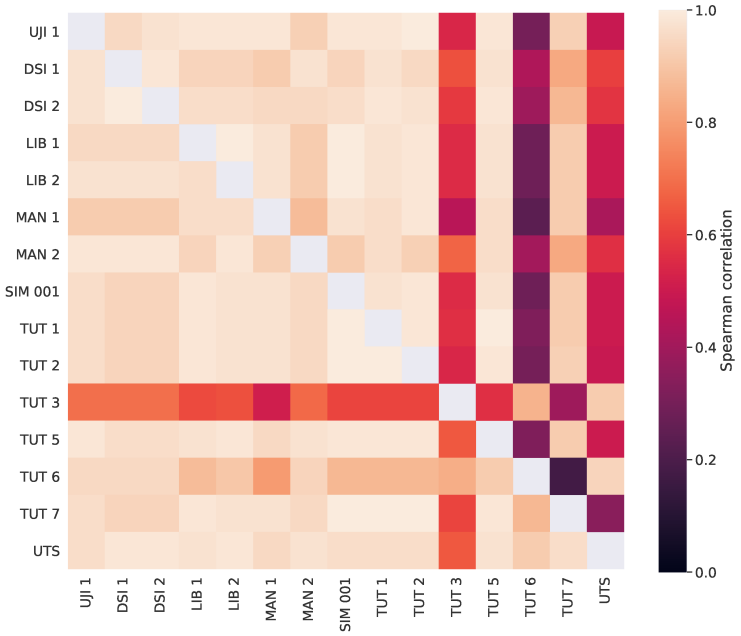


Figure 3.12: Ranking correlation for metrics across datasets. Upper triangular matrix: *same building*; lower triangular matrix: *same floor*.

among the majority of the datasets emerges, suggesting that also metrics tend to preserve their relative performance. Notable exceptions are datasets `TUT 3`, `TUT`

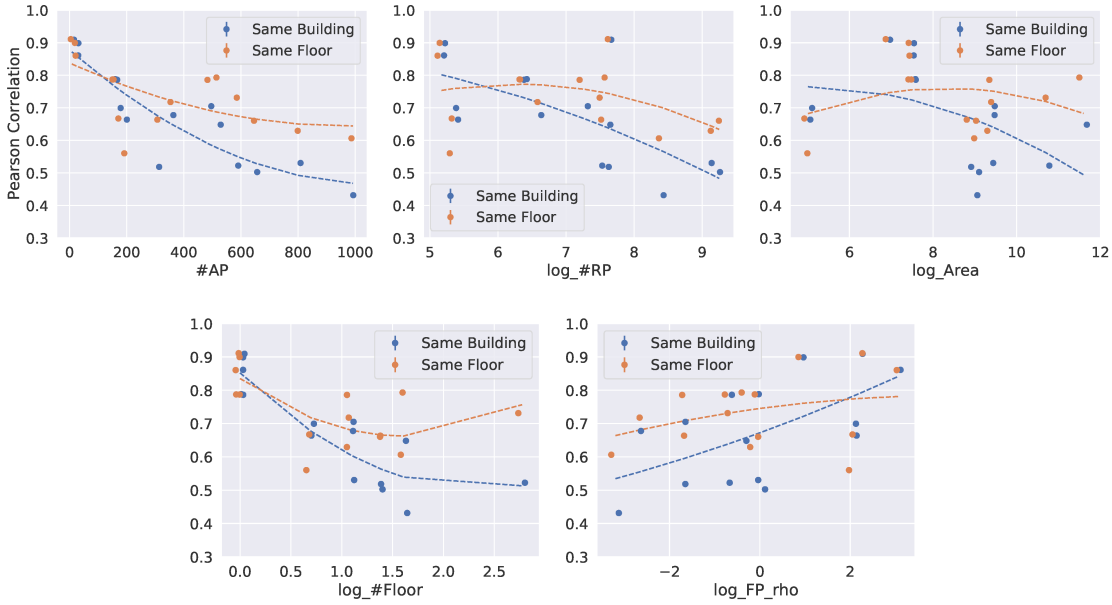


Figure 3.13: Median correlation for *Zero-to-One*-normalized *Cosine* metric compared against some datasets features (all but #AP in logarithmic scale). For the sake of clarity, some jitter has been applied on the x axis.

6, and UTS. As for TUT 3, its anomalous behaviour is detected in both scenarios (single building and single floor), and it may be caused by the crowdsourcing-based sampling strategy. As for TUT 6 and UTS, in the single building case, an overall flattening of the metric performance towards low correlation values was observed (except for *Cosine* and *Sørensen*), leading to a situation where their differences are less marked and, thus, the ultimate rankings are possibly not meaningful. In any case, in the single building case, the three datasets are highly correlated, suggesting that their anomalous behaviour is somehow consistent and possibly due to some common, but not clearly identifiable, features.

It is worth noticing that the two latter experiments do not provide any result about the expected absolute value of correlations, which may largely depend on the properties of the considered scenario. To shed light on this, we relate the median correlation value achieved by *Zero-to-One*-normalized *Cosine* metric to some features of the datasets (and, thus, of the environments). In Figure 3.13, some interesting patterns emerge. The number of APs seems to strongly influence the expected correlation value, suggesting an inverse proportionality between the two quantities. This may be caused by some noise introduced by the APs. The number of reference positions (#RP) appears to influence only the building granularity, where the presence of many RPs has a negative effect. The number of floors suggests that it is possible to obtain very high correlation values if we have just a single floor, while no clear pattern emerges for other values. Finally, neither the area of the premises

nor the average fingerprint density within a 5-meter radius from each RP seem to be relevant.

3.4 Discussion

Spatial knowledge about a domain may help a lot in tasks like, e.g., radio map construction and maintenance, and may provide the basis for more advanced approaches to indoor positioning. In this chapter, we systematically studied the relationships between the high dimensional fingerprint space and the underlying real-world one at multiple granularity levels. To this end, we considered several fingerprint normalization strategies and dissimilarity functions, and we thoroughly evaluated them over 15 public indoor positioning datasets with respect to their capability of preserving real-world distances. On the positive side, we stated some interesting, general properties of them; on the negative one, we showed that, at least in conventional approaches, there is not a (linear) proportionality between distances in the fingerprint space and those between the corresponding spatial locations. This implies that it is not possible to gain reliable spatial knowledge from fingerprint data only, by making use of commonly adopted methods and techniques.

The work done so far can be viewed as the *pars destruens* of a two-phase research. As for the *pars construens*, the core of the following chapter, we investigate whether a better performance can be achieved by suitably combining existing metrics via evolutionary algorithms, and, more generally, machine learning.

4

Fingerprint Meta-distance Learning with Genetic Programming

In the previous chapter, we showed that the relationship between distances in the high dimensional space of the fingerprints and those in the real-world geometrical space, where the user/device position is meant to be found, is not captured by classical metrics used in isolation, and that each of them exhibits a radically different behaviour: some are definitely better than others in preserving such *spatial information*, although being far from the optimal case.

Notably, being able to preserve spatial information would be of great importance within indoor positioning tasks. For example, it would make possible, for the position estimation of a user, to rely on offline fingerprints that are located farther away compared to the classical scenario, meaning that more sparse (and easily maintainable) radio maps can be considered. Note that, in principle, to train a model capable of preserving such a spatial relationship, knowing the exact coordinates of the fingerprints sampling points is not required. In fact, an estimate of their spatial distance is enough, which may be obtained by means of other sensors (e.g., inertial measurement unit). In turn, this would promote both crowdsourcing-based collection strategies (since exact position labels are not needed) and foster the development of semi-supervised approaches.

In light of the aforementioned aspects, two research questions naturally arise: *(i)* is it possible to combine the best characteristics of different fingerprint distance functions to improve spatial information preservation (in terms of the relationship between distances in the fingerprint space and those in the real-world space) and/or positioning accuracy? and, *(ii)* is there any connection between the ability of a fingerprint distance function of preserving real-world spatial relationships and its performance in positioning tasks?

In this chapter, we address both of them. First, we combine classical fingerprint distance functions into a meta-distance. To this end, we frame the problem as a symbolic regression one, and we exploit a genetic programming algorithm to solve it. Such a learning paradigm allows us to avoid model biases that affect classical regression models, as the structure of the model is learnt, too. Our approach optimizes the relationship between fingerprint distances and those of the underlying

real-world locations, modeled by means of Pearson correlation, relying only on one well-known, publicly available dataset (`UJI 1`). The ability of the resulting distance of preserving real-world spatial information in the fingerprint domain is then thoroughly discussed: an improvement with respect to classical approaches can indeed be achieved in that respect. Second, in order to uncover the possible connection between spatial information preservation and localization performance, we evaluate the synthesized distance within a positioning task. The outcome of the latter analysis is that such a relation indeed exists. Moreover, the novel distance provides a better localization accuracy with respect to classical ones, despite not being expressly designed for such a purpose. All the results are compared to those achieved by classical machine learning regression approaches and fingerprint distance functions. Moreover, also their generalizability is evaluated by testing the performance on other datasets commonly used in the literature without any kind of model retraining or adaptation. As a final take-home message, the difficulties we encountered in developing an effective meta-distance suggest that classical fingerprint distance functions, even in combination, are not able of preserving spatial information in its entirety, paving the way for the development of more advanced metric learning approaches.

The rest of the chapter is organized as follows. Section 4.1 gives an account of the usage of genetic programming in indoor positioning. Section 4.2 recalls the basics of symbolic regression, evolutionary computation, and genetic programming. The setting of the experiments is outlined in Section 4.3. Section 4.4 presents the outcomes of the experiments. In Section 4.5, a critical evaluation of the work done and the collected results is conducted, and future research directions are outlined. As a matter of fact, in Section 4.6 we provide some preliminary results regarding one of this research directions: the usage of deep metric learning for continuous similarity.

4.1 Genetic programming in indoor positioning

In the following, building on the framework and results provided in Chapter 3, we investigate the possibility of improving the amount of spatial knowledge that can be obtained from fingerprint data by combining canonical fingerprint distance functions into a meta-distance. To this end, we rely on Genetic Programming (GP), a particular kind of evolutionary algorithm that will be illustrated in detail in Section 4.2.2.

Only a few attempts at employing GP in indoor positioning have been made in the past. Here is a short account of them. In order to estimate the distances at which the observable APs are placed, Gualda et al. [99] define a function that suitably combines a set of arithmetical operators (propagation model). The resulting pieces of information are later used to perform positioning by means of spherical trilateration with the Gauss-Newton algorithm. The focus of the work by Lembo et al. [145] is indoor positioning based on cellular LTE fingerprints. GP is exploited

for both the radio map construction and representation, in the offline phase, and the position estimation, in the online phase. In the offline phase, a symbolic regression method is used to construct a function that, given a position, is able to determine the received signal strengths of each observed base station. In the online phase, GP is employed to optimize a cost function comparing the measured signal strengths with those associated with each base station at a given position. Song et al. [246] rely on GP to solve the task of extracting the best set of candidates from the radio map on which to base the position calculation for a given fingerprint. GP is also exploited by Eldeeb et al. [76], although with a quite different goal. They make use of GP to solve the problem of placing a set of access points in a building in such a way that the diversity among fingerprints collected at a set of test locations is maximized (optimization problem). Finally, Le et al. [138] combine support vector machine regression with GP to reduce the overall computational complexity of WiFi fingerprint-based positioning.

Our contribution differs from previous ones in various respects. Its focus is not on developing a novel positioning algorithm, but, rather, on investigating the relationship between fingerprints and real-world spatial distances. More precisely, we would like to understand whether or not a proportionality between fingerprint distances and those between the corresponding real-world locations can be recovered by suitably combining classical fingerprint distance functions into a meta-distance. As we shall see, such a task is quite hard, and the outcomes of our analysis suggest that no further improvement, besides those achieved in the present work, can be obtained as long as we rely on classical approaches. As a by-product, we shed light on the connection between the preservation of spatial distances and position estimation, showing that the proposed meta-distance, even if trained with a different objective, achieves better results also in the latter task.

4.2 Symbolic regression and GP

In this section, we give a short account of the basics of symbolic regression, evolutionary computation, and genetic programming, that represent the building blocks of the forthcoming investigation.

4.2.1 Symbolic regression

Symbolic regression (SR) is an optimization task where the space of mathematical expressions is searched to find the best performing model, in terms, for instance, of prediction capabilities, on a given dataset. The main difference between SR and classical regression is that the latter tries to optimize the parameters of a pre-defined model structure, such as the coefficients of a linear combination of input variables, while in SR both the model structure and its parameters are learnt [194]. No specific model is usually provided as the starting point for an SR algorithm; rather,

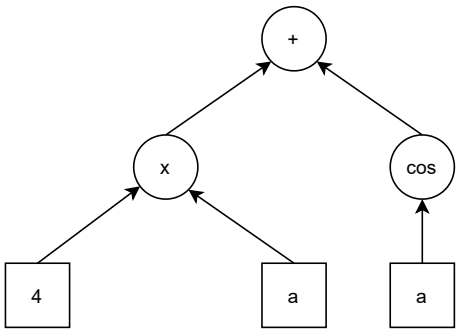


Figure 4.1: Mathematical expression $(4 \times a) + cos(a)$ represented in a tree-like fashion.

the search starts from a set of random expressions built on a combination of mathematical operators, constants, and variables. Each expression can be represented in a tree-like fashion, as shown in Figure 4.1.

The main advantage of SR is that it is not affected by any kind of human bias or partial knowledge of the considered domain. In addition, several metrics can be exploited to guide the search in the expression space including, e.g., the predictive performance and the simplicity of a given solution. As a downside, the search space of an SR task is much larger than that of classical regression approaches. Thus, advanced optimization techniques, such as those based on evolutionary computation, are required to explore it.

4.2.2 Evolutionary computation and genetic programming

In the following, we will solve an SR task following an approach based on *Evolutionary Algorithms* (EAs), i.e., population-based metaheuristics inspired by the process of biological evolution and genetics, that have been shown to excel in the solution of combinatorial optimization problems [74]. Unlike blind random search, EAs exploit historical information to direct the search into the most promising regions of the search space, relying on methods that mimic the processes that in natural systems lead to adaptive evolution.

In nature, a population of individuals tends to evolve in order to adapt to its environment. In a similar way, EAs are characterized by a population, where each individual represents a possible solution to a given optimization problem, and any solution is evaluated with respect to its degree of "adaptation" to the problem through a single- or multi-objective *fitness* function.

The EA population iteratively undergoes a series of *generations*. At each generation step, individuals picked by a *selection strategy* go through a process of reproduction. The selection strategy is the key factor that distinguishes one evolutionary-based approach from another, although, typically, individuals with a high degree of adaptation are more likely to be chosen (*elitism*). In this way, the elements of the

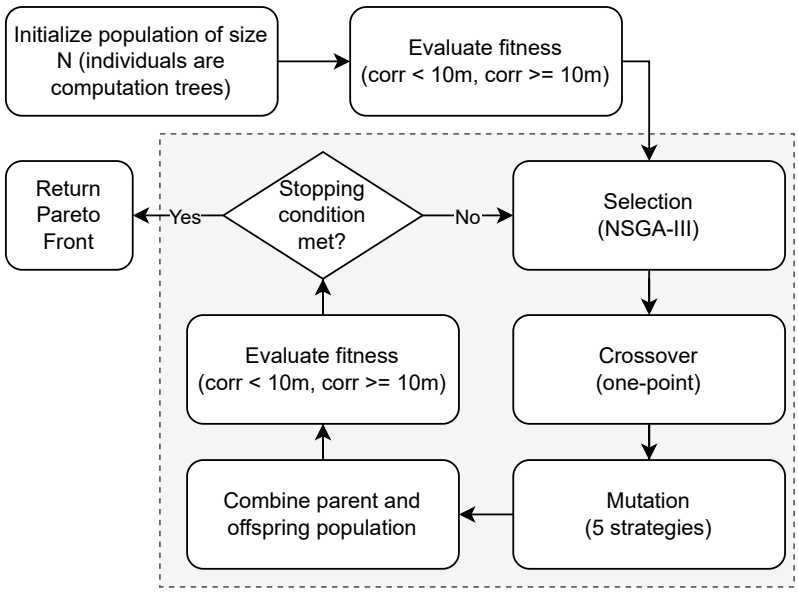


Figure 4.2: Overview of the evolutionary algorithm, where the term *corr* stands for the Pearson correlation between distances in the fingerprint and real-world space. The shaded area includes the steps performed during a generation.

population iteratively evolve toward better solutions. The reproduction process involves the application, with a certain degree of probability, of suitable *crossover and mutation operators*. As a result, an offspring is generated, which is finally merged with the previous population, and the cycle repeats until a stopping condition is met. Such a condition can be based, for instance, on population diversity or on a given fitness threshold.

Crossover is the EA counterpart of natural reproduction, by which the characteristics of two individuals are combined by generating one of two offspring. As a general rule, high crossover rates tend to pull the population towards a local minimum or maximum. Mutation applies random changes to the encoding of the selected solution, with the goal of maintaining genetic diversity in the individuals. It prevents premature convergence of the algorithm to a local optimum, thus allowing it to explore the search space more broadly. A schematic account of the whole process is given in Figure 4.2.

As a matter of fact, EAs tend to produce solutions that are as good as possible for a given training set, against which the fitness function is evaluated, without considering the performance on possible new cases. This may raise overfitting issues. An open research question thus pertains to generalization in evolutionary computation, and several efforts are being made to solve it [63]. As an example, Gonçalves and Salva [96] make use, through the generations, of a small and frequently changing random subset of the training data as a way of reducing overfitting and improving generalization. We will adopt a similar strategy.

In the context of EA, *Genetic Programming* (GP) is a technique that allows one to evolve programs starting from a population of random solutions [205]. An individual can be represented as a computation tree, where each internal node encodes a primitive and leaves represent input values. The output value is produced by the primitive encoded in the root. Typical crossover/mutation operations performed on computation trees include subtree exchange and node/leaf addition/removal/replacement. As we will see, GP can also be used in the context of symbolic regression tasks.

4.3 Experimental setting

In this section, we first discuss the proposed EA implementation, detailing the set up of the initial population, the definition of the genetic operators, and the employed fitness function; then, a description of the experimental workflow is provided, which involves training the EA on the dataset `UJI 1`, while considering the other repositories for test purposes. All experiments have been carried out on a Google Cloud Platform virtual machine instance equipped with 32 cores and 40 GB of main memory.

4.3.1 The evolutionary algorithm and its implementation

The proposed solution relies on DEAP (Distributed Evolutionary Algorithms in Python) framework [84], whose aim is to provide practical tools for the rapid prototyping of custom evolutionary algorithms. Its core consists of two simple structures: a *creator* and a *toolbox*. The first module allows the creation of individuals and populations that may include any data structure, facilitating the implementation of any kind of EA. The *toolbox* is a container of operators that the user can exploit, for instance, to perform genetic recombination and individual selection. Means to perform an efficient distribution and parallelization of specific parts of user code are provided as well. In the following, we give a detailed account of the proposed instantiation of the framework to build an evolutionary algorithm suitable for our purposes.

Population and initialization. Each individual belonging to the population is a computation tree that combines the values provided by *Euclidean*, *Matusita*, and *Cosine* fingerprint distances. The population is initialized following DEAP's *genHalfAndHalf* method, which generates random computation trees with a maximum height of 6, as suggested by Koza in his seminal work [127]. Specifically, half of the time a tree whose leaves have all the same depth is generated; in the other cases, each leaf may have a different depth.

Primitives. A node of a computation tree may represent a single value, that can be one of the considered fingerprint distance functions, a DEAP's *EphemeralConstant*, that is, a random value chosen in the interval $[-10, 10]$ according to uniform

sampling, or a constant in the set $\{-1, 0.5, 1, 2, 3\}$. A node may also encode a unary function, applied to a value x, in the set $\{log_e(x), \sqrt{x}, x^2, |x|, -x, 1/x, sin(x), cos(x)\}$. Finally, a node may compute a binary function, applied to a pair of values x, y, in the set $\{x + y, x - y, x \cdot y, x/y, x^y, max(x, y), min(x, y)\}$. To avoid problems with invalid calculations, like, for instance, $x/0$, each function is implemented in its "protected" version, where *NaN* values are replaced by 0, positive infinite values are replaced by 1, and negative infinite values are replaced by -1.

Training data. Pairs of fingerprints are extracted from the official training split of dataset UJIIndoorLoc. For each pair, *Euclidean*, *Matusita*, and *Cosine* fingerprint distance functions are computed, as well as the real-world 3D Euclidean distance between the locations associated with them. As pointed out in Chapter 3, fingerprint distance functions tend to show specific behaviours at different granularity levels. For this reason, we generated three datasets following a random sampling approach: the first one (*Whole*) contains pairs coming from all the datasets, the second one (*Same Building*) contains pairs belonging to the same building, and the third one (*Same Floor*) contains pairs belonging to the same floor. To make the implementation of counter-overfitting measures easier, taking also into account the constraints posed by the hardware at our disposal, we partitioned each of the three generated datasets into 4000 sub-datasets, each one consisting of all distinct pairs that can be obtained from 100 points, that is, 4950 instances. Of course, only a subset of all possible pairs of fingerprints is considered in this manner, which should still be representative of the whole scenario. This is a quite natural approach, being UJIIndoorLoc composed of 19861 training fingerprints, leading to almost 400 million pairs (clearly intractable from a memory and computational point of view).

Fitness function. A bi-objective fitness function is employed. Each objective tries to maximize Pearson correlation between the fingerprint distance values computed by a given individual and the corresponding real-world *Euclidean* distances. The first objective focuses on fingerprint pairs whose real-world distance is in the range $[0, 10)$ meters; the second deals with those whose distance is in the range $[10, +\infty)$ meters. Such a choice takes into account the results reported in Chapter 3, where the three considered fingerprint distance functions showed a rather different behaviour in the chosen intervals. Since two objectives are used, no single best-performing solution can be directly selected in a given population. Rather, a *Pareto front* of optimal solutions can be identified, containing all *non-dominated* solutions.[1]

Counter-overfitting and termination criteria. Some methods to improve the generalization capability of the generated solutions have been employed. First, each of the previously-described $3 \cdot 4000$ sub-datasets were further partitioned into 90% training and 10% validation data. The reason is as follows. At each EA generation,

[1] A set $\mathcal{S}$ of solutions for an n-objective problem with fitness function $f = \langle f_i, \ldots, f_n \rangle$ is said to be *non-dominated* if and only if for each $x \in \mathcal{S}$, there exists no $y \in \mathcal{S}$ such that (i) $f_i(y)$ improves $f_i(x)$ for some i, with $1 \leq i \leq n$, and (ii) for all j, with $1 \leq j \leq n$ and $j \neq i$, $f_j(x)$ does not improve $f_j(y)$.

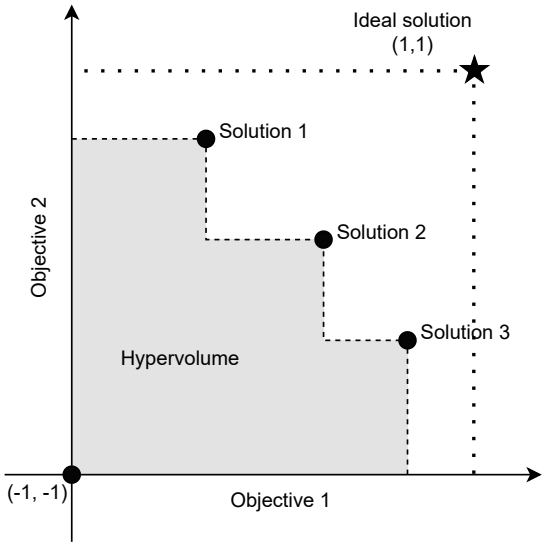


Figure 4.3: Hypervolume for our bi-objective maximization problem. The reference (worst) point is $(-1, -1)$, while $(1, 1)$ is the ideal solution.

the fitness of the individuals is established with respect to a new random selection of 90 sets, uniformly sampled from *Whole*, *Same Building*, and *Same Floor* training data. The idea is that good solutions should generalize across different selections, resulting in a greater chance to survive over the generations, while unstable solutions would be penalized. Then, to provide us with a fixed dataset against which to consistently evaluate the behaviour of the individuals along the generations, a random selection of 90 sets is also sampled from the validation data, only at the beginning of the EA execution.

As for the termination criteria of the algorithm, even though an upper bound on the number of generations is given, an *early stopping* strategy also has been defined, based on the *hypervolume* measure. Intuitively, the hypervolume of a Pareto front measures the volume of the search space, bounded by a given reference point, that is weakly dominated by the points on the Pareto front [43]. A graphical account of the hypervolume of a given front in the context of our bi-objective maximization problem is provided in Figure 4.3. The assumption is that populations of heterogeneous and well-performing solutions are characterized by a high hypervolume. Turning to the specific early stopping criterion, at each generation, the fitness of the individuals is evaluated not only on the training set data, but, as previously mentioned, also against a fixed (through the generations) selection of 90 sets, uniformly sampled from *Whole*, *Same Building*, and *Same Floor* validation data. This allows to assess the generalization capability of the solutions on data which are different from those used to guide the evolution. The training is interrupted when no sufficient improvement over the validation hypervolume is observed for a given number of generations. At the end of the EA execution, to foster generalization, the population with the highest

validation hypervolume is returned.

Crossover. Given two parent solutions, that are, two computation trees, two individuals are generated by one-point crossover (DEAP'S *cxOnePoint*). The operator randomly chooses a node in each individual and exchanges the subtrees rooted at it. To avoid bloat, that is, an excessive increase in mean program size without a corresponding improvement in fitness, we placed a static limit of 17 on the children height (DEAP's *staticLimit*), following once more a suggestion from Koza [127]. When an invalid (over the height limit) child is generated, it is simply replaced by one of its parents, randomly selected.

Mutation. Various operators are considered, among those available in DEAP, each one chosen with uniform probability: *mutUniform*, which selects a node in a given individual, and exchanges it with a randomly generated (mathematical) expression; *mutInsert*, that inserts a new subtree at a random position in an individual; *mutNodeReplacement*, that replaces a randomly chosen node in the individual; *mutEphemeral*, that changes the value of a single constant used within an individual; *mutShrink*, that randomly chooses a subtree in an individual and replaces it with one of the subtree's arguments (also randomly chosen). As before, to control bloat we impose a *staticLimit* constraint with a tree height of 17.

Selection. As for the selection criterion, in order to promote population diversity, we rely on the elitist strategy implemented in NSGA-III [66], based on the concepts of *reference points* and *niche preservation*.

The following hyperparameters are used by the EA: Population size = 1000, Crossover probability = 0.7, Mutation probability = $0.3/\sqrt[2]{num\_gen}$, Max generations = 500, Hyp window (early stopping) = 100, Min hyp increment (early stopping) = 0.001. Optimal values for population size, crossover probability, and mutation probability have been determined through grid search over validation set data, respectively considering the intervals $[500, 1000, 5000]$, $[0.5, 0.6, 0.7, 0.8]$, and $[0.1, 0.2, 0.3]$ (the latter to be divided by the squared root of the EA generation index). Observe that mutation probability is rather high at the beginning to ensure an effective *exploration* of the search space; it then rapidly decays with the number of generations, to allow the *exploitation* of the most promising solutions that have been found. As for the running time of the algorithm, 500 is a rather conservative upper bound to the number of generations of the EA [205], although the evolutionary process may be interrupted earlier (given the early stopping strategy) if an increment of less than 0.001 in the validation hypervolume is observed over the last 100 generations.

4.3.2 Workflow of the experiments

We executed the proposed evolutionary algorithm with the tuned hyperparameters. During its run, we kept track of the Pareto front at each generation, ultimately selecting the one that achieved the highest hypervolume with respect to the validation

split of the dataset UJIIndoorLoc. We then filtered the front keeping just the points exhibiting a fitness higher than a baseline provided by *Cosine* (with respect to both the objectives), being it the fingerprint distance function most capable of preserving spatial information according to Chapter 3 outcomes. Among the remaining points, we selected the final solution according to the criterion suggested by De La Fuente et al. [65], that is, by computing the hypervolume covered by each point in the Pareto front and selecting the one associated with the highest hypervolume.

The extracted solution (i.e., the resulting meta-distance) was evaluated over the official test set of UJIIndoorLoc. First, we considered its capability of preserving spatial information, again, establishing the Pearson correlation value with respect to real-world location distances, at the usual three different granularity levels (pairs from all the dataset/same building/same floor). Then, the meta-distance performance was determined within a position estimation task, assessing its accuracy when used as the similarity metric within a (k-)Nearest Neighbor approach. Note that parameter k, i.e., the number of neighbors which are used to determine the final prediction, must be properly set in order to obtain optimal results. Since in this work we are not interested in providing best-in-class positioning performance, but rather in evaluating the relative difference among several fingerprint distance functions, we simply set $k = 3$.

For the sake of comparison, we also confronted the behavior of the proposed meta-distance and those provided by *Euclidean*, *Matusita*, and *Cosine* fingerprint distance functions, both in isolation and combined through machine learning algorithms. In the latter case, we modeled the problem as a regression task, with the three distance functions as the predictors, and the real-world Euclidean distance among the locations as the target. To avoid learning biases, the training and validation sets were the same as those generated for the evolutionary algorithm (90-10 split). The considered models were *XGBoostRegressor* [51] and Scikit-learn's [202] *LinearRegression*, tuned with respect to the validation set by means of *Hyperopt* library [24]. The choice of relying on such approaches is motivated by: *(i)* the state of the art performance still offered by XGBoost on tabular data [30]; *(ii)*, the simplicity of the linear regression model, that makes it another quite natural baseline; and *(iii)*, the fact that our choice allows us to compare the performance provided by symbolic and conventional regression approaches.

To assess the generalization capability of our meta-distance function, as well as that of the contenders, we applied them also to the test sets of the other 15 datasets, without any kind of re-training or adaptation.

All previous experiments were instrumental for a final, important investigation, that is, determining the relationship between the capability of a fingerprint distance function of preserving spatial information and its performance in the positioning task.

The source code needed to replicate the experiments is available at the website `https://github.com/dslab-uniud/genetic_indoor`.

4.4 Results

In this section, we first illustrate the meta-distance generated by the proposed EA. Then, the ability of the synthesized meta-distance of preserving spatial relationships as well as its accuracy in a position estimation scenario are discussed and compared to those of the contenders. In addition, we run an analysis of the generalization capabilities of the approaches taken into consideration for both tasks. We conclude the section with an examination of the correlation between the two tasks, that are, the capability of a fingerprint distance function of preserving spatial information and its positioning performance.

4.4.1 The generated meta-distance

The raw meta-distance function generated by the proposed genetic programming solution was originally encoded by a tree of height 17, with 88 nodes. After a post-processing step, that removed all redundant operators, we ended up with the computation tree shown in Figure 4.4, which has a height of 15 and consists of 60 nodes. Notice that the computation tree makes use of all three considered fingerprint distance functions, with a large prevalence of *Cosine* and *Matusita*. Observe also that the output of the function may be negative. This is not a problem, as it can still be relied upon to identify the closest neighbors of a given fingerprint.

4.4.2 Correlation of spatial and fingerprint distances

To perform an objective comparison with the results reported in Chapter 3, we set up a similar evaluation framework, determining the correlation of the considered distances over 100 randomly-selected samples of UJIIndoorLoc's test split, each one consisting of 400 fingerprints. For each pair of fingerprints, we computed the real-world spatial 3D distance, as well as the fingerprint distances computed according to the approaches described in Section 4.3.2. Then, we evaluated the Pearson correlation between the spatial distance and each fingerprint distance. The experiments were repeated considering fingerprints taken from the entire dataset, as well as from the same building and the same floor. It is worth pointing out that the random sampling approach allowed us to test the average performance of each fingerprint distance function, as well as its stability across the samples, that intuitively represent different indoor scenarios.

Figure 4.5 shows a boxplot of the correlation performance for the different fingerprint distance functions, obtained from the 100 samples at the three considered granularity levels. Notably, the GP-based meta-distance function consistently scores among the best performing ones. Indeed, its median correlation is only surpassed by *XGBoostRegressor* at the *Whole* granularity level (0.624 vs 0.641), and it is almost matched by *Cosine* at the *Same Floor* granularity level (0.826 vs 0.817). This is confirmed by a permutation test with a significance level of 0.05, with the null

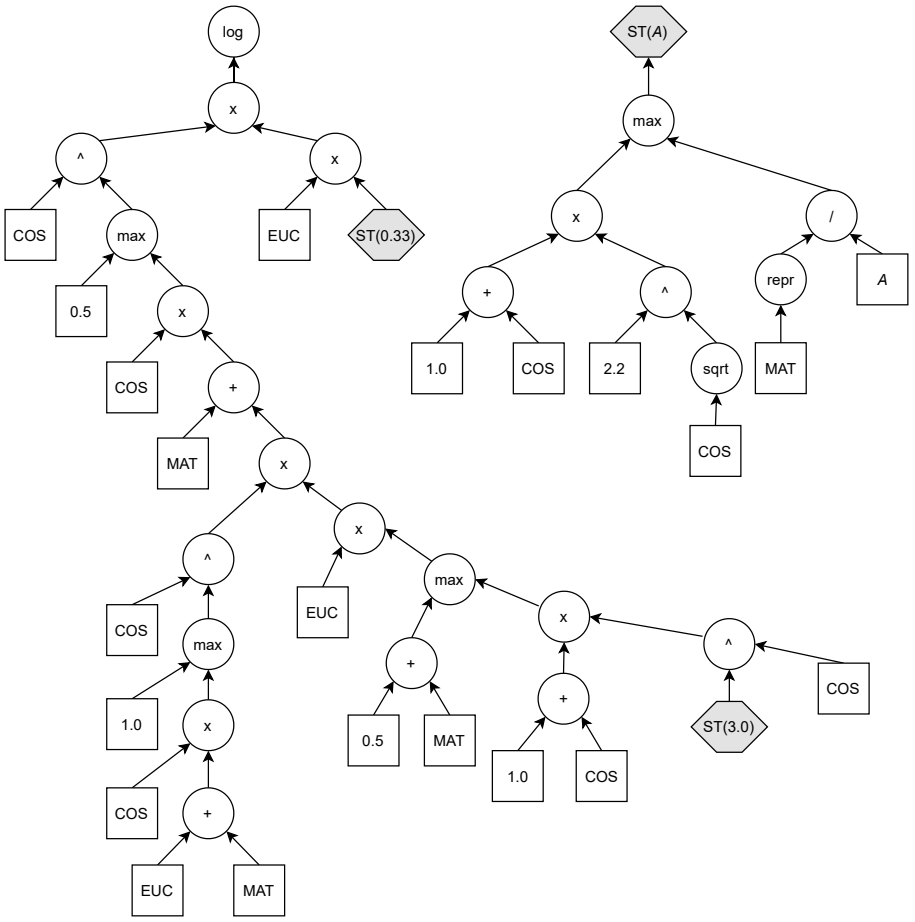


Figure 4.4: The computation tree generated by the evolutionary algorithm, where $ST(\cdot)$ nodes represent parameterized instantiations of the $ST(A)$ subtree. The meta-distance makes use of all three considered fingerprint distance functions.

hypothesis that the median of our approach is not greater than that of the given competitor. Its outcomes are shown in Table 4.1. Each cell of the table reports the p-value of the genetic distance with respect to the other approaches. As for the meta-distance stability, the widths of the boxplots are on par with the competitors with respect to the *Whole* and *Same Floor* granularity levels, while it shows a higher instability (slightly less than *XGBoostRegressor*) in the *Same Building* scenario.

The boxplots allowed us to gather a general understanding of the performance of the fingerprint distance functions. To establish if some specific correlation patterns emerge at different spatial distance intervals, let us analyze a single representative run. Figure 4.6 and Figure 4.7 show the distance function behaviours at different granularities, respectively considering classical and machine learning-based approaches.

Classical fingerprint distance functions have a tendency towards overestimation, especially for what concerns shorter distances. *Euclidean* and *Matusita* exhibit a

Table 4.1: Permutation test results (one-sided p-value).

	Euclidean	Cosine	Matusita	LinearRegression	XGBoostRegressor
Whole	0.0	0.0	0.0	0.0	1.0
Building	0.0	0.0111	0.0	0.0059	0.0
Floor	0.0	0.1589	0.0	0.0052	0.0002

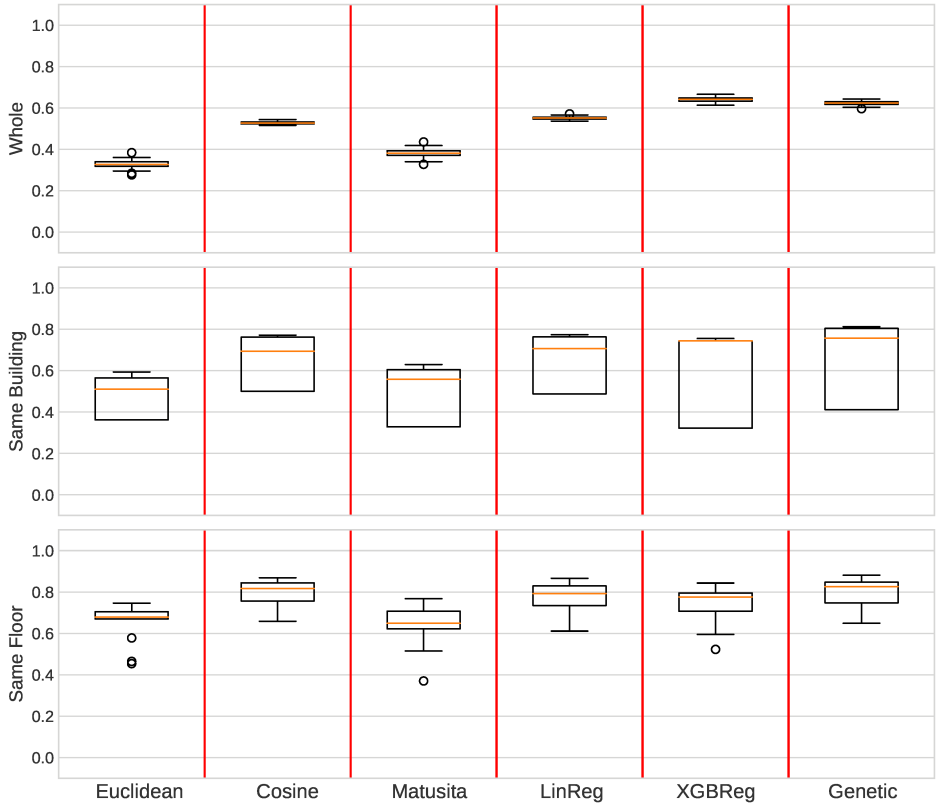


Figure 4.5: Correlation performance for all the considered fingerprint distance functions and granularity levels. Each boxplot is determined by the correlation values obtained from 100 different test runs. Boxes extend from the first to the third quartile values of the data, with a line at the median. Whiskers extend to the smallest and largest observations which are not outliers (considering 1.5 times the interquartile range). The GP-based meta distance function consistently scores among the best performing ones.

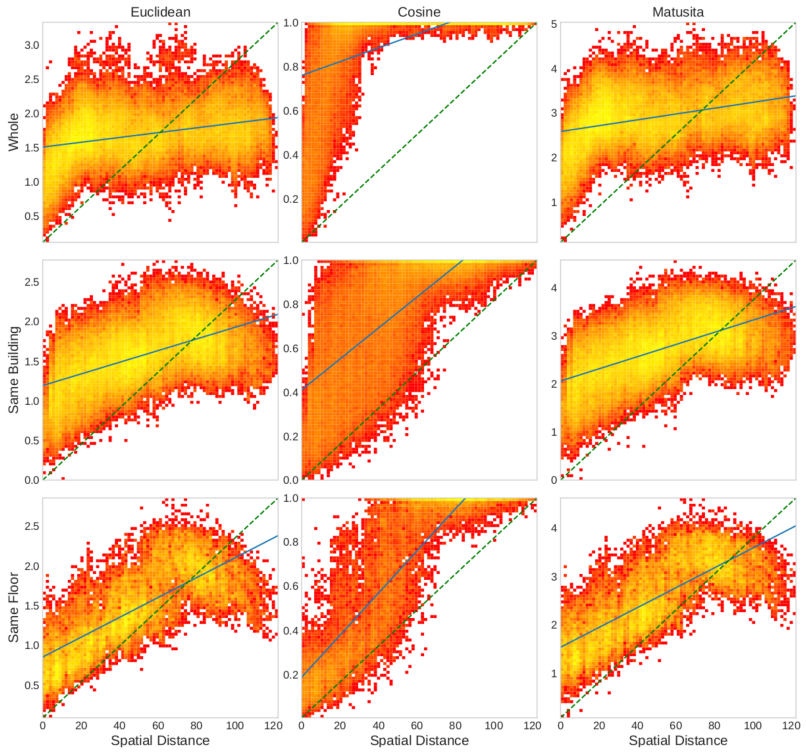


Figure 4.6: Classical fingerprint distances (y-axis) plotted against the spatial distance (x-axis) when instances are sampled from the entire dataset (top), only from a same building (centre), and only from a same floor (bottom). Brighter colour denotes a higher density; correlation trend (solid line); ideal case (dashed). The considered distances exhibit rather different behaviours.

similar behaviour, with a trend inversion around a spatial distance of 75m. *Cosine*, instead, shows a stepped curve, with a clear saturation phenomenon. The latter effect, which is more evident on distant observations, largely determines the high correlation results of such a distance function.

As for the machine learning-based distances, the trend shown by *LinearRegression* is similar to that of *Cosine*, especially for what concerns *Whole* and *Same Floor* granularity levels. The saturation phenomenon is still present, although much less pronounced, a fact that is confirmed by the slightly higher median correlation obtained by *LinearRegression* with respect to *Cosine* in the *Whole* and *Same Building* scenarios. Turning to *XGBoostRegressor*-based distance, it typically exhibits a good behaviour till a spatial distance of 25/30m, especially at the *Whole* granularity level, after which its sparsity greatly increases.

The behavior of the solution based on genetic programming resembles that of *LinearRegression* at the *Whole* granularity level, but with an even less pronounced saturation effect. At the *Same Building* granularity level, the distance shows a larger dispersion, as it was already evident in the boxplots; nevertheless, it still has

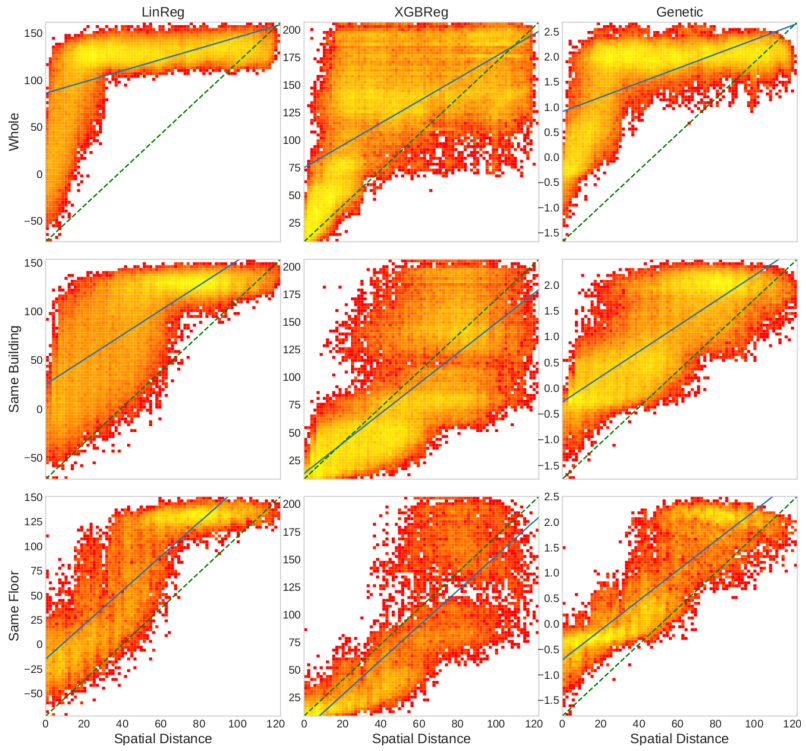
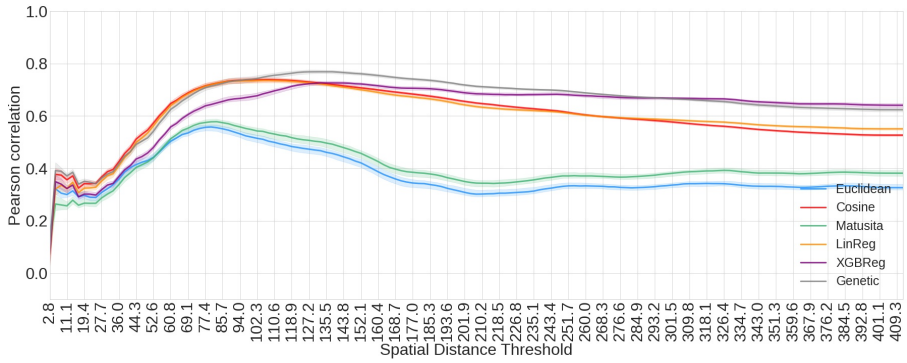


Figure 4.7: Machine learning-based fingerprint distances (y-axis) plotted against the spatial distance (x-axis) when instances are sampled from the entire dataset (top), only from the same building (centre), and only from the same floor (bottom). Brighter colour denotes a higher density; correlation trend (solid line); ideal case (dashed). While all approaches seem to provide an improvement in correlation with respect to Figure 4.6, the best case is that of the GP-based meta distance.

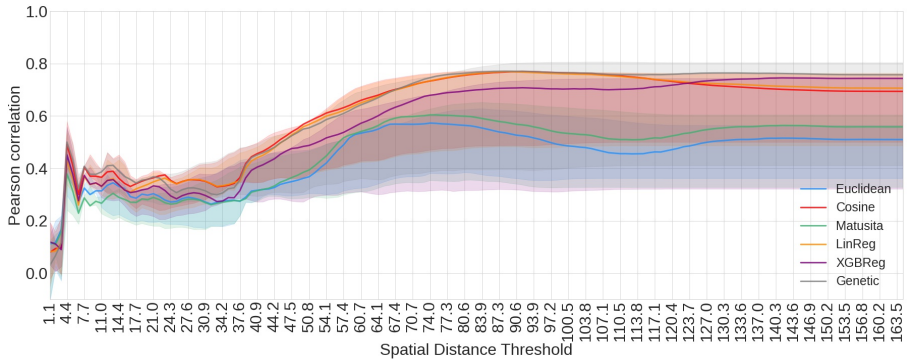
a very high correlation with the spatial distance. Finally, in the *Same Floor* case, it manages to slightly ameliorate the stepped/inversed curve behaviour typical of the competitors.

In analogy to Chapter 3, Figure 4.8a, Figure 4.8b, and Figure 4.8c report the correlation trends when different upper bounds on the spatial distance are applied, at the three granularity levels. As a general rule, the genetic programming-based distance obtains higher correlations over the different scenarios and distance thresholds, a fact confirmed by Table 4.2, that reports the area under the different curves. It is worth pointing out that, in general, fingerprint distance functions show a high correlation for large spatial distances, while the correlation decreases as lower and lower thresholds are considered.

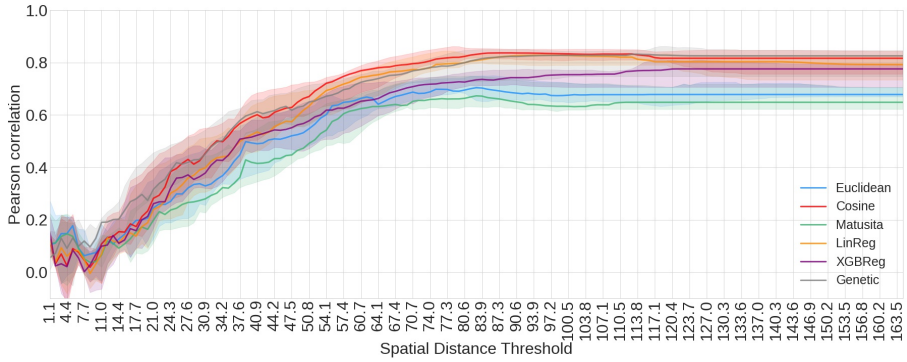
As a final step, we generalize the results obtained on the dataset UJIIndoorLoc to the other 15 datasets. To this end, for each dataset, given a fingerprint distance function, we determined its median correlation value for each granularity level, proceeding in the same way as for UJIIndoorLoc. This allowed us to rank the



(a) All



(b) Same building



(c) Same floor

Figure 4.8: Correlation values at different maximum spatial distances and granularity levels. Each point x represents the correlation value looking at spatial distances smaller than x only. Each curve reports the median correlation value over 100 iterations (solid line), while its shaded area refers to the interquartile range. Overall, the GP-based meta-distance obtains higher correlations over the different scenarios and distance thresholds.

Table 4.2: Area under the curves of Figure 4.8; higher is better.

	Euclidean	Cosine	Matusita	LinearRegression	XGBoostRegressor	Genetic
Whole	152	245	165	245	257	**267**
Building	72	100	76	99	95	**102**
Floor	91	109	85	105	99	**110**
Average	105	151	109	150	150	**160**

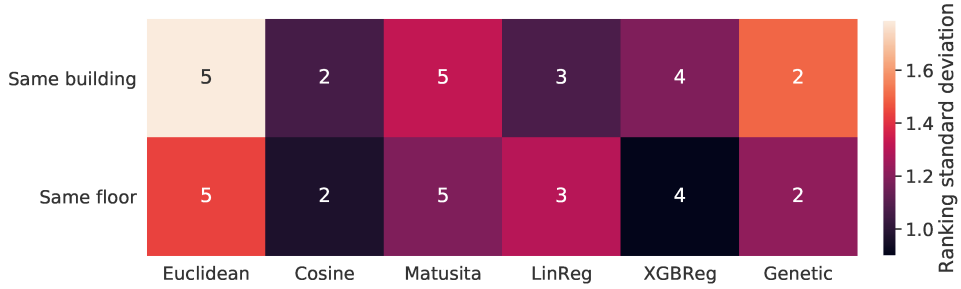


Figure 4.9: Generalization performance of the correlation results of the considered fingerprint distance functions. For each distance, the average rank of its performance across all datasets is reported. Lower value and darker color are better. The best results are provided by *Cosine* and the GP-based meta distance.

fingerprint distance functions from the best (rank = 1) to the worst (rank = 6), in each dataset and for *Same Building* and *Same Floor* scenarios (except for UJIndoorLoc, in all datasets *Whole* is equivalent to *Same Building*). With the idea that a highly-generalizing function ought to perform consistently well in all scenarios, we then computed the rank mean and standard deviation across the datasets (again, by distinguishing the two granularity levels). From the results reported in Figure 4.9, it emerges that the genetic programming-based meta-distance and *Cosine* one have the overall best position in the ranks, suggesting that they typically achieve a high correlation; however, *Cosine* exhibits a lower standard deviation of its rank across the datasets, indicating a more stable behavior and, thus, a higher generalization capability.

4.4.3 Positioning performance

Let us focus now on the positioning performance of the different fingerprint distance functions. Prediction tasks are performed employing several (3-)Nearest Neighbor models based on the UJIIndoorLoc training set, each relying on a different fingerprint distance function. For each test instance, we determined its building, floor, and position coordinates, following the approach of most of the other contributions (see, for instance, [155, 261]), where additional information, such as floor and position, are considered significant if and only if the building prediction is correct.

Table 4.3: Positioning performance, where the Avg Positioning Error must be evaluated together with the Success rate.

Dataset	Success rate (higher is better)						Avg Positioning Error (lower is better)					
	Euclidean	Cosine	Matusita	LinReg	XGBReg	Genetic	Euclidean	Cosine	Matusita	LinReg	XGBReg	Genetic
DSI 1	-	-	-	-	-	-	5.5738	5.4187	6.1188	5.9691	5.6679	5.1711
DSI 2	-	-	-	-	-	-	5.1398	4.7864	5.3984	5.3557	5.1261	4.7817
LIB 1	0.9984	0.9990	0.9974	0.9984	0.9974	0.9990	2.7868	2.7940	2.9448	2.9753	2.9736	2.7330
LIB 2	0.9878	0.9997	0.9840	0.9990	0.9939	0.9997	3.7722	2.8403	3.8380	3.1351	3.1647	2.8286
MAN 1	-	-	-	-	-	-	2.7508	2.7242	3.1164	4.2938	7.6868	2.5517
MAN 2	-	-	-	-	-	-	2.5739	2.6138	2.8530	5.4599	10.4590	2.0456
SIM 001	-	-	-	-	-	-	2.7576	2.9739	2.7622	16.1441	27.8726	2.9579
TUT 1	0.8776	0.9061	0.8122	0.8408	0.8939	0.9102	7.6248	6.4019	8.1684	7.8799	7.4410	6.2686
TUT 2	0.7330	0.8864	0.6420	0.7330	0.7727	0.9318	12.2685	11.1418	14.5394	13.2493	12.0263	9.0762
TUT 3	0.8841	0.8932	0.8659	0.8656	0.8102	0.8881	8.2669	7.8789	8.6782	8.7538	11.2988	8.3370
TUT 4	0.9527	0.9498	0.9397	0.9225	0.9067	0.9555	5.5533	5.5748	5.8400	6.3594	7.5121	5.6289
TUT 5	0.8065	0.8921	0.7424	0.7933	0.8024	0.9593	6.6869	6.3685	7.2852	7.7890	9.8523	6.4486
TUT 6	0.9994	0.9986	0.9989	0.9950	0.9833	0.9996	2.6439	2.4867	2.9754	3.5034	4.1593	2.7035
TUT 7	0.9903	0.9906	0.9860	0.9729	0.9427	0.9903	2.5399	2.5727	2.8096	3.7454	5.1171	2.6483
UJI 1	0.8884	0.9199	0.8326	0.8704	0.9163	0.9226	7.7856	7.5601	9.0035	9.2227	7.7501	6.8471
UTS	0.9261	0.9235	0.9314	0.9182	0.8865	0.9261	7.5346	7.4480	8.0462	8.1390	8.8102	7.8682
Borda*	49	38	66	65	67	**29**	43	32	73	80	79	**29**

*A lower value denotes an overall better position in the rank.

More precisely, for any given instance, the building is predicted first, following a distance-weighted voting procedure within the (3-)Nearest Neighbor model. Then, in a similar way, the floor is determined, though considering only instances belonging to the predicted building. Finally, the position coordinates are computed as a weighted centroid, considering only those neighbors that belong to the estimated building and floor.

Table 4.3 reports the results obtained on the official test splits of all the considered datasets, taking into account the success rate, that is, the fraction of instances for which both building and floor were accurately predicted, and the average positioning error (in meters). Notably, in the considered setting, the latter metric cannot be considered in isolation, as the positioning error is evaluated only for those instances where building and floor have been correctly identified. This is a pretty natural approach, since whenever a device is associated with the wrong floor or building, information at a finer granularity becomes meaningless.

The main outcome is that the proposed meta-distance matches and often surpasses the positioning performance provided by the contenders. This is clear from the (inverse) Borda counts [29] (ranked voting system) reported at the bottom of Table 4.3. They have been obtained by summing the ranking position of each distance over the different datasets (rank 1 = best, rank 6 = worst), and show that the genetic distance cumulatively has a lower (thus, better) position in the ranks.

4.4.4 On the correlation-positioning relationship

In Section 4.4.2, we discussed the ability of distance functions of preserving spatial information, in terms of the correlation between fingerprint distances and those of the associated real-world locations. Then, in Section 4.4.3, we illustrated their

positioning performance. In both cases, the genetic-based meta-distance turned out to be the overall best performing solution. In this section, we explore the relationships between the two tasks, assessing the correlation between the area under the curves of Figure 4.8 and the corresponding positioning accuracy. The conjecture to be verified is that a larger area should be associated with a lower positioning error. To assess it, we performed some experiments based on the UJIIndoorLoc dataset test split, which have been designed as follows.

First, we determined the 95th percentile of the real-wold spatial distances of all 3rd neighbors in the considered (k-)Nearest Neighbor models, which resulted in 21.6 meters. Then, following the same approach as in Figure 4.8, we built a different set of curves for each building/floor combination, considering only spatial distances in the interval $[0, 21.6]$. Next, for each distance function and building/floor combination, we established its average positioning error and the area under its previously-generated curve. As a result, we obtained a list of positioning errors and a corresponding list of areas under the curve, containing values that span all distances and building/floor combinations. Finally, to establish whether there exists a meaningful relationship between the two quantities, we computed Spearman's rank-order correlation between the two lists (negating the areas).

Results are shown by the boxplot in Figure 4.10. The median correlation value is well above 0.6, indicating that, in general, distance functions that exhibit a higher (negated) area also lead to better positioning accuracies. The rather high interquartile range should not come as a surprise, as the correlation values refer to buildings/floors that may be very different from one another. Last but not least, note how the lower whisker, corresponding to the first quartile of the data, extends below -0.4. This is a striking result as it means that, in some cases, a higher overall correlation actually corresponds to a higher positioning error. The reason behind this phenomenon, that upon further analysis seems to characterize two specific building/floor combinations, is still unclear and will be investigated in further studies. To such an extent, we plan to collect a new fingerprint dataset, designed to represent specific indoor scenarios in a controlled fashion.

4.5 Discussion

In this section, we discuss the strengths and limitations of the proposed solution. First of all, as shown by the pictures in Section 4.4.2, we would like to point out that combining classical fingerprint distance functions into a meta-distance is an approach that it is worth pursuing. The patterns exhibited by the three learning methodologies are, indeed, clearly different from those of classical functions, as well as from each other. This latter point suggests that the choice of the underlying learning algorithm is of primary importance. In the present case, according to the performed analyses, the genetic-based meta-distance behaves the best, both from a qualitative and a quantitative point of view.

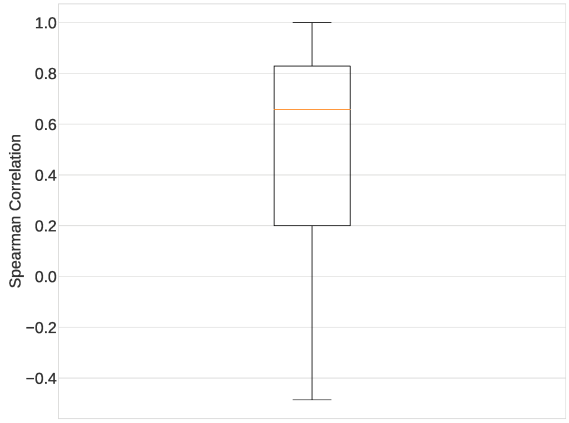


Figure 4.10: Spearman correlation between the average positioning errors and the (negated) area under the curves. The plot must be interpreted as those in Figure 4.5. Overall, distance functions that exhibit a higher (negated) area lead to a lower positioning error.

Having said that, its performance, as shown in Figure 4.5, is far from being optimal, as it only slightly improves those exhibited by the other distance functions. This is not completely unexpected, since the designed EA simply learns how to combine classical fingerprint distances and it does not directly exploit fingerprints themselves. On the one hand, this suggests that there is no way of significantly improving performance by just relying on currently-available fingerprint distance functions. On the other hand, it raises two important points. First, other approaches should be considered, perhaps still focusing on the correlation task, but learning directly from the raw fingerprints. Second, building upon classical distances, which can be computed for any generic pair of fingerprints, grants an *implicit* form of generalization: it allows one to directly apply a given meta-distance to any dataset with no further adaptation, and irrespective of the number of underlying observed APs. Moreover, Figure 4.9 shows that when the meta-distance is based on an effective learning strategy, like the genetic programming one, it is also possible to achieve an *explicit* form of generalization, witnessed by the overall high correlation results.

The same outcome is true for the positioning task, where the genetic approach unquestionably achieves the top performance. This is a notable result, given that such a meta-distance was learned and optimized within the EA according to a proxy correlation metric and not minimizing the positioning error, as commonly done in the literature.

On the basis of this last evidence, we shall now discuss in some detail the relationship between the correlation and the positioning task. The experiment conducted in Section 4.4.4, based on Spearman correlation, confirmed the existence of such a connection. Still, note that, as evident from Figure 4.4.4 (and as expected), we did

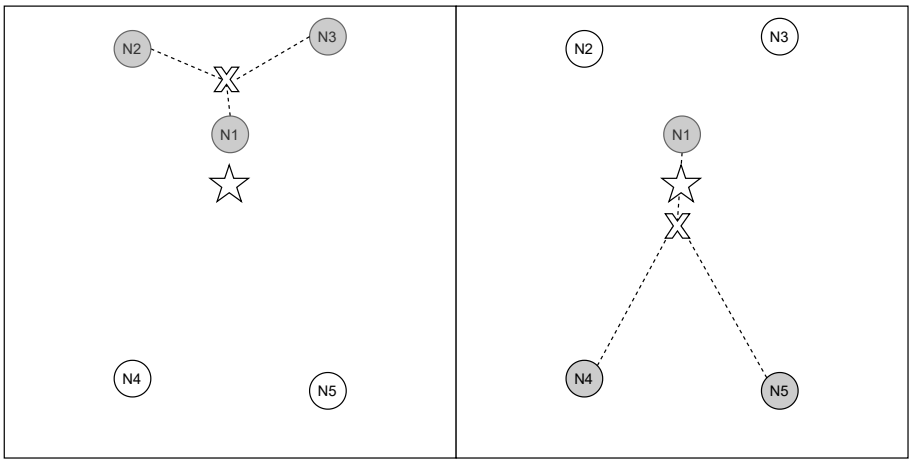


Figure 4.11: Positioning task scenarios within the (k-)Nearest Neighbor framework. Left: weighted centroid (X symbol) calculated with the 3 instances closest to the real position (star symbol). Right: best-performing triplet.

not achieve a perfect correlation. A very large area is indeed not sufficient to achieve the best performance in localization. Consider, for instance, a fingerprint distance function which achieves a Pearson correlation value of 1.0 with respect to spatial distances, meaning that the real-world distance proportionalities are perfectly mapped in the fingerprint space. In that case, running a (k-)Nearest Neighbor model in the fingerprint domain would extract the same neighbors as if it was run on the corresponding real-world location coordinates. Figure 4.11 (left) shows a situation where perfectly knowing the 3 nearest neighbors does not lead to the smallest positioning error, if a weighted centroid is considered for the prediction (as in Section 4.4.3). The best candidates to determine the centroid would have been, instead, those reported in the scenario on the right. As already pointed out, we did not look for the best possible algorithm to employ for localization, meaning that other algorithms or variants of the used (k-) Nearest Neighbor model could lead to further improvements when paired with the proposed meta-distance.

To summarize, the developed meta-distance is able to combine classical fingerprint distance functions effectively, exploiting their best characteristics both to preserve higher spatial information in the fingerprint space and to achieve a better performance in the positioning task. In addition, the achieved results strengthen the idea that the correlation between spatial and fingerprint distances must be further investigated, in particular concerning its relationship with the positioning error and the possibility to use the correlation as a proxy task for localization. Nonetheless, given the complexity and generality of the proposed approach, we believe that no further improvement can be achieved by combining classical fingerprint distance functions only. This paves the way for a new research direction, that is, the development of novel, generalizable distance functions by means of metric learning approaches.

4.6 A glimpse of the road ahead: deep metric learning for continuous similarities

Before deviling into our proposal to solve the goal of obtaining an optimal metric for fingerprinting, which incorporates spatial information, we would like to discuss why in light of the conclusions we just witnessed this line of research could be promising.

As we said, our investigation of genetic programming showed that we can learn a metric with good positioning performance by an optimization procedure that maximizes the correlation between the distances measured in the fingerprint vector space and the corresponding distances observed in the real world. Thus, to learn the metric we do not need to know the location of each fingerprint, but rather just the spatial distances among a set of pairs of fingerprints. However, the spatial distance between two fingerprints can be derived by relying on other sensors, such as the inertial measurement units (IMU). This suggests that if we have a system passively collecting fingerprints and IMU data, we can obtain all the information we need to learn a metric in crowdsourcing, without requiring the users to input any knowledge, just wardriving. Of course, this is not enough to perform positioning, since we will have a (possibly very good) metric to compare fingerprints, whose output values are strongly aligned with the spatial domain. Nevertheless, if we have a very good metric, we only need very few labelled fingerprints, as we can reliably estimate the distances (short or long) from them. This pipeline would lead us towards having semi-supervised indoor localization, reducing by a large margin the effort needed to build and maintain the radio-map. Moreover, the metric could be easily updated continuously at the online time, limiting issues arising from changes in the environment.

Given the result we saw in this chapter, we know that learning a metric starting from the effects of another set of metrics does not seem to be an approach capable to solve optimally our goal. Thus, to improve in this respect, we make use of deep metric learning (DML), which, informally, aims to learn similarity metrics in an end-to-end fashion with deep neural networks. However, deep metric learning has mostly [124] been defined for binary supervision and rankings [89, 278], making it unfit for our setting. In this section, we are going to provide an overview of how to extend it for continuous similarity, reporting some preliminary results applied to fingerprinting.

4.6.1 Deep metric learning in a nutshell

Let's consider a batch of multivariate training data $\mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$, such that $\mathbf{x}_i \in \mathbb{R}^d$, and the corresponding set of labels $\mathcal{Y} = \{\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_N\}$, such that $\mathbf{y}_i \in \mathbb{R}^b$. The objective of deep metric learning is to learn the parameters θ of a neural network $\phi(\cdot;\theta) : \mathbb{R}^d \to \mathbb{R}^z$, according to a given loss function L which constrains to map closer elements that are similar and more distant those that are

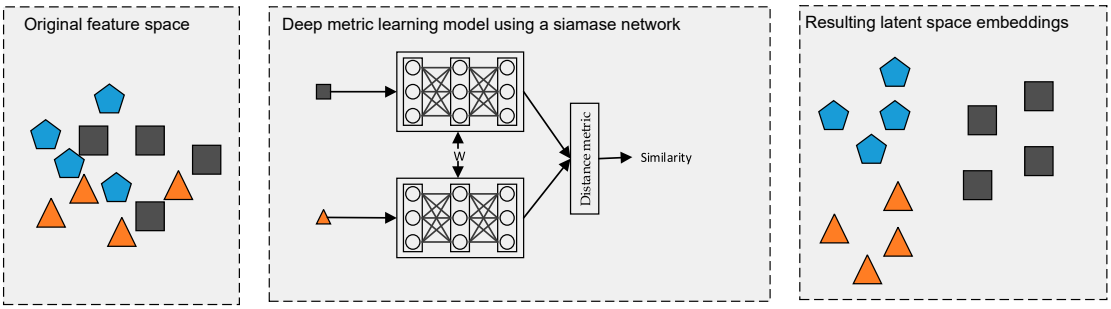


Figure 4.12: Graphical account of deep metric learning general workflow (adapted from [120]).

dissimilar from each other, in an z-dimensional space. A graphical account of DML is reported in Figure 4.12. Observe that the final layer of ϕ has dimension z and its output, an embedding, is often L_2-normalized with a unit norm, i.e., $\mathbf{z}_i = \frac{\phi(\mathbf{x}_i)}{\|\phi(\mathbf{x}_i)\|_2}$, with $\mathcal{Z} = \bigcup_{i=1}^{N} \mathbf{z}_i$.

DML has been developed primarily by the computer vision community for tasks such as face recognition, person re-identification, zero-shot and self-supervised learning. The differences among approaches are given by two fundamental elements: the loss function and the sampling strategy.

Starting from the latter, DML is based on comparing elements. As a consequence, we recognise different types of pairs (or, more generally, n-uplets) of examples: positive, i.e., two elements are similar; negative, i.e., two elements are dissimilar. The embedding space is learned by comparing the distance measured between pairs, minimizing a loss function that penalizes large distances between positive pairs and small distances between negative ones. As a result, it is paramount to avoid useless comparisons and construct a set of pairs composed only of meaningful elements; as a matter of fact, if the model already maps the relationship among a set of elements correctly it will not learn anything more from them. This a complex issue to deal with during training, and if not treated correctly might lead to very slow learning convergence and insufficient performance on those sets of elements that are poorly represented.

Many loss functions have been developed for DML, starting from the contrastive [57, 272] and triplet loss [233]. To give an idea about DML losses, let's consider the latter, which is based on comparing the distances between an anchor point $\mathbf{z}^a$, a positive point $\mathbf{z}^p$ (which is similar to the anchor point), and a negative point $\mathbf{z}^n$ (which is dissimilar to the anchor point). Thus, given a set of triplets $\{(\mathbf{z}^a, \mathbf{z}^p, \mathbf{z}^n)_i | \mathbf{z}^a, \mathbf{z}^p, \mathbf{z}^n \in \mathcal{Z}\}_{i=0}^{M}$ computed over the batch, the triplet loss can be defined as follows:

$$L_{triplet} = \sum_{i=0}^{M} [\|\mathbf{z}_i^a - \mathbf{z}_i^p\| - \|\mathbf{z}_i^a - \mathbf{z}_i^n\|]_+ \ , \tag{4.1}$$

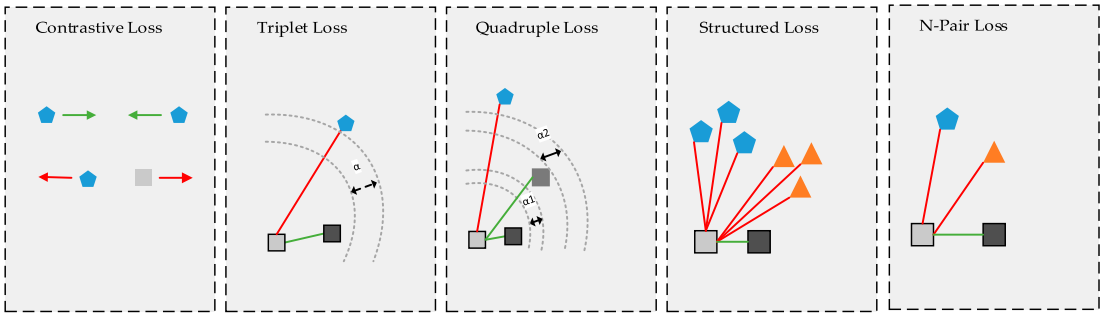


Figure 4.13: Graphical example of some DML loss functions (adapted from [120]).

where $[\cdot]_+ = \max(0, \cdot)$. The concept behind this loss has been extended in the literature in various directions (see Figure 4.13 for a graphical insight), in order to cope with the fact that it is unproductive to just consider all the possible triplets that we can construct from the batch and that the gradient contribution brought by each sample is constant irrespective of the error [123, 244, 277, 279, 296].

4.6.2 A preliminary proposal for continuous similarities

Let's suppose to have a meaningful semantic way, i.e., a function $s(\cdot, \cdot) : \mathbb{R}^b \times \mathbb{R}^b \to \mathbb{R}$, to evaluate the distance between a pair of elements $\mathbf{x}_i$ and $\mathbf{x}_j$ by looking at their labels $\mathbf{y}_i$ and $\mathbf{y}_j$, respectively. From a general perspective, our objective is to find an embedding mapping ϕ such that

$$\frac{s(\mathbf{y}_i, \mathbf{y}_j)}{s(\mathbf{y}_k, \mathbf{y}_h)} \propto \frac{\|\mathbf{z}_i - \mathbf{z}_j\|_2^2}{\|\mathbf{z}_k - \mathbf{z}_h\|_2^2} \,. \tag{4.2}$$

The formula in Eq. 4.2 exactly encodes the notion of proportionality we are interested in: the ratio (proportion) between the distances of the embedded elements in the latent space shall resemble the original one between the similarities computed on the labels. Note that s is a domain-dependent function (e.g., the Euclidean distance if we are dealing with labels representing spatial coordinates). Moreover, note that rather than the normalized Euclidean distance, the (angular) distance in the latent space could be computed using the cosine one, since $\|\mathbf{z}_i - \mathbf{z}_j\|_2^2 = 2 - 2\langle \mathbf{z}_i, \mathbf{z}_j \rangle$, with $\langle \cdot, \cdot \rangle$ denoting the dot product.

To achieve the goal of having a loss function preserving the proportionality of the distances between elements while moving between target and embedding space, two key properties have to be satisfied. The first one is that similarities of the embedded elements need to keep the relative ordering of the similarities evaluated across the labels. That is, for each training element $\mathbf{x}_i \in \mathcal{X}$, given the set of similarities involving it $\mathcal{S}_i = \{s(\mathbf{y}_i, \mathbf{y}_j) \,|\, j \in \{1, \ldots, N\}, i \neq j\}$, and the corresponding set of distances in the latent space $\hat{\mathcal{S}}_i = \{\|\mathbf{z}_i - \mathbf{z}_j\|_2^2 \,|\, j \in \{1, \ldots, N\}, i \neq j\}$, if we sort $\mathcal{S}_i$

and $\hat{\mathcal{S}}_i$ then the two elements originated by the pair (i, j) must be located in the same position in both the sorted sequences. Mathematically, what we would like to have is similar to an order-embedding, i.e., a function f both order-preserving ($s(\mathbf{y}_i, \mathbf{y}_j) \leq s(\mathbf{y}_i, \mathbf{y}_k) \Rightarrow \|f(\mathbf{x}_i) - f(\mathbf{x}_j)\|_2^2 \leq \|f(\mathbf{x}_i) - f(\mathbf{x}_k)\|_2^2$) and order-reflecting ($\|f(\mathbf{x}_i) - f(\mathbf{x}_j)\|_2^2 \leq \|f(\mathbf{x}_i) - f(\mathbf{x}_k)\|_2^2 \Rightarrow s(\mathbf{y}_i, \mathbf{y}_j) \leq s(\mathbf{y}_i, \mathbf{y}_k)$).

This is the first main difference with respect to deep metric learning working only with categorical labels. Although our approach may fit in the ranking-based category of deep metric learning ([219]), in the classical setting the mapping imposes a weaker constraint, since it needs to preserve similarity defined in a binary way, i.e., $s(\mathbf{y}_i, \mathbf{y}_j) = \{0, 1\}$. Additionally, while in a categorical setting $s(\mathbf{y}_i, \mathbf{y}_j) = 0$ and $s(\mathbf{y}_i, \mathbf{y}_k) = 0$ is satisfied for both $y_j = y_k$ and $y_j \neq y_k$, in the continuous case it is not true, suggesting as inter-instances relationship must be learned, resulting in a more complex manifold and thus in a harder optimization problem. It immediately follows that a binary version of similarity is not suitable for the continuous case[2], and also the definition of positive or negative instances/tuples does not find application, rather we have tuples characterized by strong or weak similarity.

It is worth pointing out that ranking is enough to have a deep metric learning approach for continuous data. In fact, with a function preserving ordering between distances when mapping the elements from the target to the latent space, we might be able to determine the correct relative similarity among any arbitrary set of elements in the latent space. However, much of the semantic that characterizes the continuous labels and the distances among them is lost. Let's suppose that, for instance, the distance between a pair of elements is double compared to another, i.e., $s(\mathbf{y}_i, \mathbf{y}_j)/s(\mathbf{y}_k, \mathbf{y}_h) = 2$. This is very characterizing information of the continuous setting that we would like to keep while measuring distances in the latent space (i.e., $\|\mathbf{z}_i - \mathbf{z}_j\|_2^2 / \|\mathbf{z}_k - \mathbf{z}_h\|_2^2 = 2$). Reasoning only with rankings does not allow us to do that.

From an mathematical perspective, let's define $\mathbf{S} \in \mathbb{R}^{N \times N}$ the matrix obtained evaluating all the pairwise similarity across $\mathcal{Y}$ (i.e., all the possible combination of $s(\mathbf{y}_i, \mathbf{y}_j)$), and $\hat{\mathbf{S}} \in \mathbb{R}^{N \times N}$ the pairwise normalized Euclidean distance matrix on $\mathcal{Z}$. Let's consider the matrix version of $argsort : \mathbb{R}^{N \times N} \to \mathbb{R}^{N \times N}$ which in each output row produces the indices that would sort the original input row. We can define the proportional deep metric loss function as:

$$\begin{aligned}\mathcal{L}_{prop} = \frac{1}{N} \sum_{i=1}^{N} \sum_{j=2}^{N} \frac{1}{j-1} \sum_{k=2}^{j} \left(1 - sign\left(\left[\frac{\hat{\mathbf{S}}^{\mathbf{\Pi}}_{i,k}}{\hat{\mathbf{S}}^{\mathbf{\Pi}}_{i,j}} - 1\right]_+\right)\right) \cdot \left|\frac{\mathbf{S}^{\mathbf{\Pi}}_{i,k}}{\mathbf{S}^{\mathbf{\Pi}}_{i,j}} - \frac{\hat{\mathbf{S}}^{\mathbf{\Pi}}_{i,k}}{\hat{\mathbf{S}}^{\mathbf{\Pi}}_{i,j}}\right| + \\ sign\left(\left[\frac{\hat{\mathbf{S}}^{\mathbf{\Pi}}_{i,k}}{\hat{\mathbf{S}}^{\mathbf{\Pi}}_{i,j}} - 1\right]_+\right) \cdot \left(2 - \frac{1}{\hat{\mathbf{S}}^{\mathbf{\Pi}}_{i,k} / \hat{\mathbf{S}}^{\mathbf{\Pi}}_{i,j}} - \frac{\mathbf{S}^{\mathbf{\Pi}}_{i,k}}{\mathbf{S}^{\mathbf{\Pi}}_{i,j}}\right),\end{aligned} \tag{4.3}$$

[2]Unless thresholds are introduced. However, thresholds might completely depend on the domain, thus being rather arbitrary.

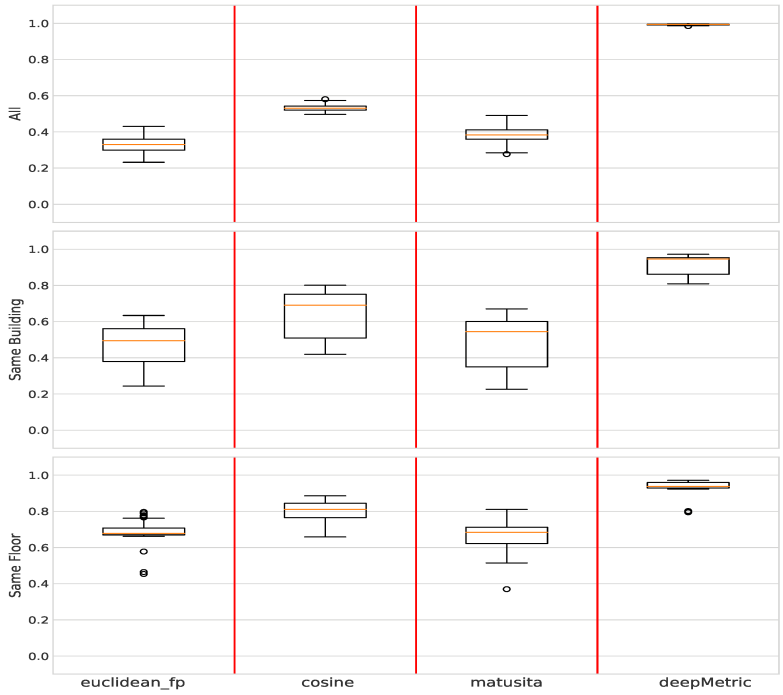
where $[\cdot]_+ = \max(0, \cdot)$, $\mathbf{\Pi} = argsort(\mathbf{S})$, $\mathbf{S}^{\mathbf{\Pi}}$ and $\hat{\mathbf{S}}^{\mathbf{\Pi}}$ are the matrixes $\mathbf{S}$ and $\hat{\mathbf{S}}$ sorted according to $\mathbf{\Pi}$, respectively[3]. Observe that the iterative sequential scanning of the ordered information given by the chain of summations ensures all the properties we are interested in. By construction, $0 < \mathbf{S}^{\mathbf{\Pi}}_{i,k}/\mathbf{S}^{\mathbf{\Pi}}_{i,j} \leq 1$, since the numerator is always smaller or equals to the denominator. $\hat{\mathbf{S}}^{\mathbf{\Pi}}_{i,k}/\hat{\mathbf{S}}^{\mathbf{\Pi}}_{i,j}$ has a different behaviour depending on whether latent space distances of the considered elements are in the correct order or not. If the ordering is correct, i.e., similar elements are closer than dissimilar ones, $0 < \hat{\mathbf{S}}^{\mathbf{\Pi}}_{i,k}/\hat{\mathbf{S}}^{\mathbf{\Pi}}_{i,j} \leq 1$ and $sign([\hat{\mathbf{S}}^{\mathbf{\Pi}}_{i,k}/\hat{\mathbf{S}}^{\mathbf{\Pi}}_{i,j} - 1]_+) = 0$. Thus, when ordering is satisfied, the loss optimizes the absolute error between the true and the actual proportions of the distances, leading to $0 \leq |(\mathbf{S}^{\mathbf{\Pi}}_{i,k}/\mathbf{S}^{\mathbf{\Pi}}_{i,j}) - (\hat{\mathbf{S}}^{\mathbf{\Pi}}_{i,k}/\hat{\mathbf{S}}^{\mathbf{\Pi}}_{i,j})| \leq 1 - (\mathbf{S}^{\mathbf{\Pi}}_{i,k}/\mathbf{S}^{\mathbf{\Pi}}_{i,j})$. On the other hand, when the ordering is incorrect, $\hat{\mathbf{S}}^{\mathbf{\Pi}}_{i,k}/\hat{\mathbf{S}}^{\mathbf{\Pi}}_{i,j} \geq 1$ and $sign([\hat{\mathbf{S}}^{\mathbf{\Pi}}_{i,k}/\hat{\mathbf{S}}^{\mathbf{\Pi}}_{i,j} - 1]_+) = 1$, leading to the loss component being $1 - (\mathbf{S}^{\mathbf{\Pi}}_{i,k}/\mathbf{S}^{\mathbf{\Pi}}_{i,j}) < 2 - 1/(\hat{\mathbf{S}}^{\mathbf{\Pi}}_{i,k}/\hat{\mathbf{S}}^{\mathbf{\Pi}}_{i,j}) - (\mathbf{S}^{\mathbf{\Pi}}_{i,k}/\mathbf{S}^{\mathbf{\Pi}}_{i,j}) < 2$. Such a formulation ensures two different properties. First, when dealing with unordered samples the loss contribution is always strictly greater than the ordered case, prioritizing correct ordering over the preservation of the proportionality, as indeed the former is a prerequisite of the latter. Second, the resulting function is continuous.

4.6.3 First results, known issues, and current limitations

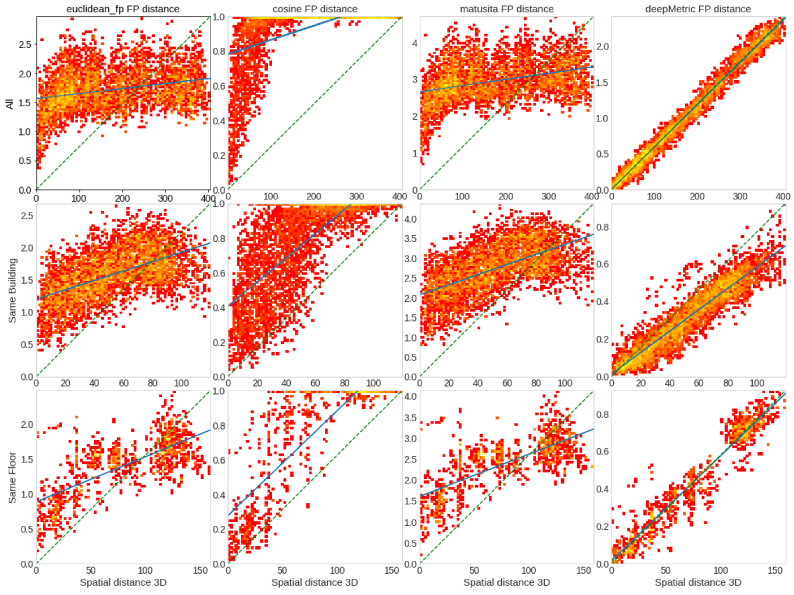
In Figure 4.14, we report some very preliminary results applied to `UJI 1` dataset, following the same protocol used for the already discussed analysis.

We can see some remarkable facts. First, there is an overall improvement in the quantitative outcomes, with an average correlation higher than the classical metrics (Figure 4.14a). This result is also confirmed qualitatively by Figure 4.14b, where the points lie in a line following the optimal case scenario. However, there is still a major challenge that needs to be addressed. Looking closely at both Figure 4.14c and Figure 4.14d, we can see that the performances for short distances, i.e., the left-hand side of the figures, are lower than those of the contenders. Beyond making Figure 4.14a and Figure 4.14b partially overoptimistic, this fact must be addressed in order to have an optimal metric, as those distances are the most informative from a point of view of positioning. We are currently working on addressing such an issue, and we conclude by giving an idea about why it is happening. First, our problem is affected by a strong imbalance in the data distribution. To understand why let's consider a random distribution of points. Given a single point, if we take into account all the pairwise distances, we immediately see that there are a few points very close to our anchor and a considerable amount of medium-far ones. Second, when we focus on short distances, the discriminative capacity of the fingerprints decreases. Thus, it becomes difficult for the neural network to discriminate between two instances representing similar locations but with different fingerprints due to

[3]Note that, in practice, a small quantity $\epsilon = 1e^{-10}$ is added to those values that might lead to numerical issues.



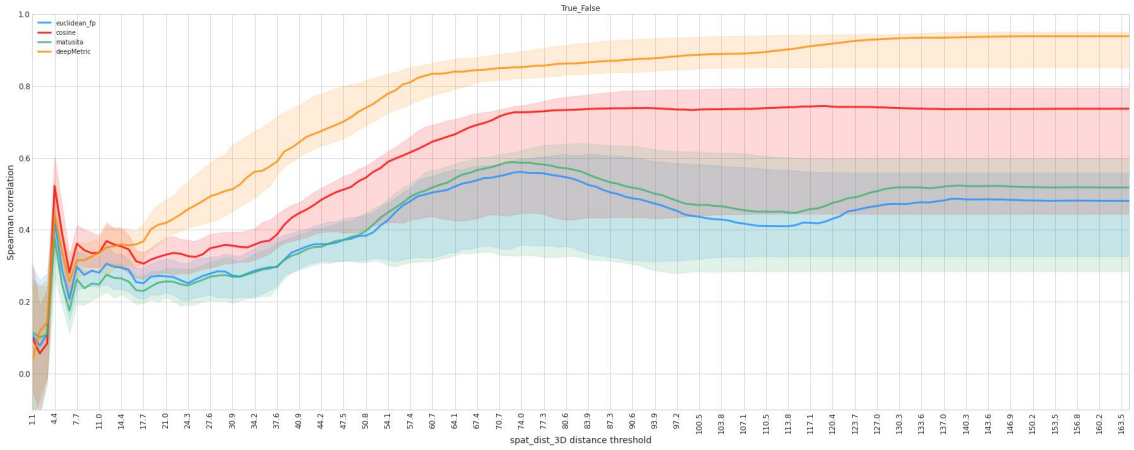
(a) Correlation performance



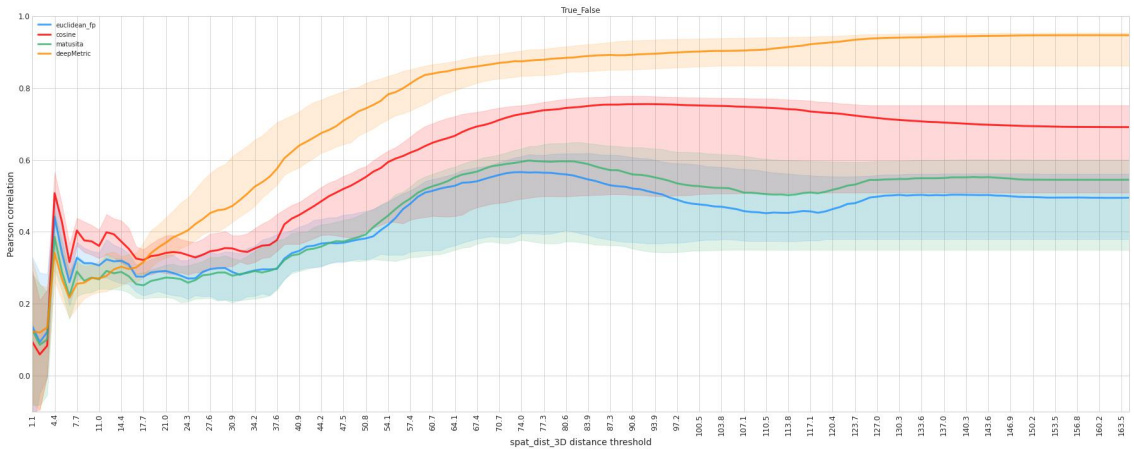
(b) Distance functions behaviours at 3 granularities

Figure 4.14: Preliminary results of proportional DML applied to `UJI 1` dataset.

noise, and two instances that may be similar under the fingerprint perspective, but different in terms of location, again due to noise. Finally, the loss function, in its current form, does not produce gradients proportional to the error (i.e., no implicit



(c) Ranking correlation values at different maximum spatial distances (same building)



(d) Correlation values at different maximum spatial distances (same building)

Figure 4.14: Preliminary results of proportional DML applied to `UJI` dataset (cont'd).

mining). Given what we said, it is evident that these hard samples are exactly those pertaining to short distances, the focus of our future work.

III

Effective Ranking-based Indoor Fingerprinting

5

Let's Forget About Exact Signal Strength

The popularity of indoor positioning based on WiFi fingerprints over other solutions is largely due to its low cost and high precision [285]. In particular, there is no need to deploy any special infrastructure, as WiFi APs are nowadays widely adopted. Fingerprinting can exploit them directly [121], and knowledge of the positions of APs is not necessary [103, 285]. On the other hand, some well-know issues affect this technique, preventing its wide adoption as the indoor counterpart of GNSS. The most significant challenge is related to radio map construction and maintenance. These tasks are labour intensive, time-consuming, and changes in the environment may force their repetition [121, 177, 211]. Another major challenge pertains to the uneven indoor propagation of WiFi signals, which negatively affects the accuracy of position estimation. As an example, RSS fluctuations are typically observed, caused by multipath effects resulting from obstacles, random noise, body attenuation, changes in the WiFi network, and device heterogeneity [177, 262].

A possible workaround to deal with some of the above-described issues, only partially explored in the literature, involves relying on specific representations of fingerprints. For instance, the authors of [284] propose *fingerprint spatial gradient*, that exploits spatial features of fingerprints from multiple adjacent locations to reduce spatial ambiguity and temporal instability of classical fingerprinting. Another alternative is to make use of ranking-based fingerprints, where the RSS information is only exploited to sort the APs detected at each location in decreasing order [54, 164]. As a result, sequences of AP identifiers, without any explicit information on their RSSs, are considered. This allows to better deal with problems such as device heterogeneity and signal perturbations, even though, in the past, the general performances brought by this kind of representation were lower than full-fledged fingerprint-based approaches [54, 136, 155, 163, 164, 223, 254].

As a matter of fact, it emerges that in order to reduce the negative impact of the issues affecting fingerprinting, it is often necessary to trade off other characteristics, frequently its accuracy. In the work described in this chapter, we move a step forward in this area, by exploiting ranked-based representation of the fingerprints jointly with deep learning. The aim is to retain the advantages of ranked fingerprinting, without

getting a reduction in performance. To the best of our knowledge, no previous attempts have been done in this direction. Specifically, the motivation for such an approach is to reduce as much as possible the influence of the above RSS issues, that severely affect position estimation. The ranking-based representation provides a more stable (and extremely compact) source of information for the model. In fact, the relation encoded by the rankings has already been shown to reduce errors related to RSS [164]. Consider, for instance, a scenario with multiple, heterogeneous devices, which is quite common in indoor positioning. While most probably two devices have a different view of the RSSs associated with the same APs for the same location, e.g., due to their different hardware, the corresponding two ranking representations are typically close to each other, e.g., thanks to the fact that rankings are invariant to bias and scaling [155, 164].

The true novelty of our proposal is that of determining the relationship between rankings and indoor locations by means of a Recurrent Neural Network (RNN). Such a model is commonly used to handle sequences of data, while here it is exploited to learn ordinal signal strength relations between APs. The input of the network is a single ranking representing a fingerprint, where each ranking position assumes a role equivalent to a time point. The output is the location of the user, that is determined by training the RNN to solve a classification-like problem.

In order to thoroughly explore this novel idea, two models are developed. One considers a flattened representation of the target location, e.g., a string consisting of the concatenation of the identifiers of the building, the floor, and the room; the other uses a structured/hierarchical representation. Moreover, tests evaluating both the number of ranking positions that should be considered and the robustness of the solution to signal noise perturbation are carried out.

The proposed approach is evaluated on three of the well-known publicly available indoor datasets described in Section 1.3.3. As we will see, the achieved results turned out to be comparable to those of state-of-the-art systems that rely on full-fledged fingerprints with RSSs (and possibly far more complex models), performing better than several solutions developed in the last years and tested on the same reference datasets. In addition, robustness to RSSs fluctuations is strongly supported by the results obtained by noise-added performance evaluation. Last but not least, the proposed solution can be applied straightforwardly to any fingerprint-like data, neglecting the complex preprocessing phases that characterise other approaches.

The rest of the chapter is organised as follows. In Section 5.1 we present an overview of some relevant contributions in the context of deep learning and fingerprinting. Section 5.2 describes the proposed approach. It also provides the necessary background knowledge on Recurrent Neural Networks, and ranked fingerprinting representation. In Section 5.3, an account of the experimental evaluation pipeline is given. Finally, in Section 5.4 a comprehensive and detailed analysis of the achieved results is carried out.

5.1 Deep learning for indoor positioning systems

Recently, a huge amount of work tackled the positioning problem through deep learning [81]. Here we report some examples. In [193], a building/floor classification deep neural network is proposed, aimed at reducing the complexity of parameter tuning and filtering. It combines a stacked autoencoder with a two-layer classifier. In [122], a scalable deep learning model consisting of a stacked autoencoder combined with a feed-forward neural network is presented. The autoencoder is used to reduce the dimension of the feature space, while the feed-forward component focuses on the classification task for building, floor, and room prediction in a hierarchical fashion. In [247], a comparison between a single neural network stacked autoencoder and an ensemble of multiple different networks is carried out. The authors argue that the ensemble, whose performance is shown to be better than that of the single model, can reduce the effects of RSS fluctuations. In [245], a new indoor dataset is introduced, and three deep learning models for multi-level positioning are developed. A stacked autoencoder with a classification layer is exploited to handle building identification, while two slightly different models, based on the combination of a stacked autoencoder with a one-dimensional convolutional neural network, are used to handle floor classification and position prediction. In [112], multiple convolutional neural networks are employed to predict a position starting from RSS time series encoded as multi-channel images. Each network deals with a specific hierarchical level of the domain, e.g., building or floor, and the choice of which model to use is driven by the prediction result for the preceding level of the hierarchy. Relying on sequences of signals allows one to reduce the noise and randomness generally affecting single point fingerprinting. In particular, the authors show that 100% accuracy is achieved in predicting building and floor for a well-known public indoor dataset. In [238] WiFi and magnetic field are encoded together as high-resolution images, so as to tackle the positioning problem, which is considered as a computer-vision one, by automatically learning the mapping between ground-truth positions and such generated data. In the last years, several works started investigating the usage of channel state information (CSI), that is, a particular data source related to the WiFi channels, yet not detectable by all WiFi-enabled devices, that showed promising and accurate results in combination with deep learning approaches [83, 280, 287]. CSI is exploited also in [149] where, although focusing on device-free fingerprint positioning, the authors propose a deep learning approach to deal with fingerprint inconsistency issues, formulating the problem as a domain adaptation task (e.g., to adapt to changes that over time affected a given scenario).

Some other methods analyze fingerprints employing recurrent models, such as Recurrent Neural Networks (RNN) and Long Short-Term Memory (LSTM) networks. This allows one to manage sequential information that might arise in positioning, for instance, when dealing with trajectories (i.e., sequences of position/fingerprint) or odometry. In [105], various Recurrent Neural Networks (RNN) models, developed to deal with the positioning problem employing trajectories of full-fledged

fingerprints, are built and confronted. The authors claim that the proposed architectures are capable of addressing problems such as spatial ambiguity, RSS instability, and RSS short collection time per RPs. In [14], a double RNN model for fingerprint-based trajectories is developed. The first network is in charge of providing a rough estimation of the fingerprint trajectory; the second network, starting from the output of the first one, carries out a location filtering, resulting in the final sequence of locations for the input trajectory. To achieve such a result, the two networks are trained together with a common loss function. In [111], two models, one for floor classification and the other for position identification through regression, are proposed. Both of them are implemented by means of RNN and LSTM, showing that the latter has a better performance. Moreover, the authors show that no appreciable gain in the accuracy is obtained by adding layers (i.e., stacked/deep RNN/LSTM), a conclusion also shared by [226].

Our work differs from all the previously-described ones in various respects. We use RNNs in combination with a single fingerprint, not trajectories. Moreover, we do not exploit classical fingerprints with explicit RSSs, but we rely on just a ranking-based representation of them. Finally, we manage the hierarchical structure of the indoor scenario with a single model, instead of combining multiple ones.

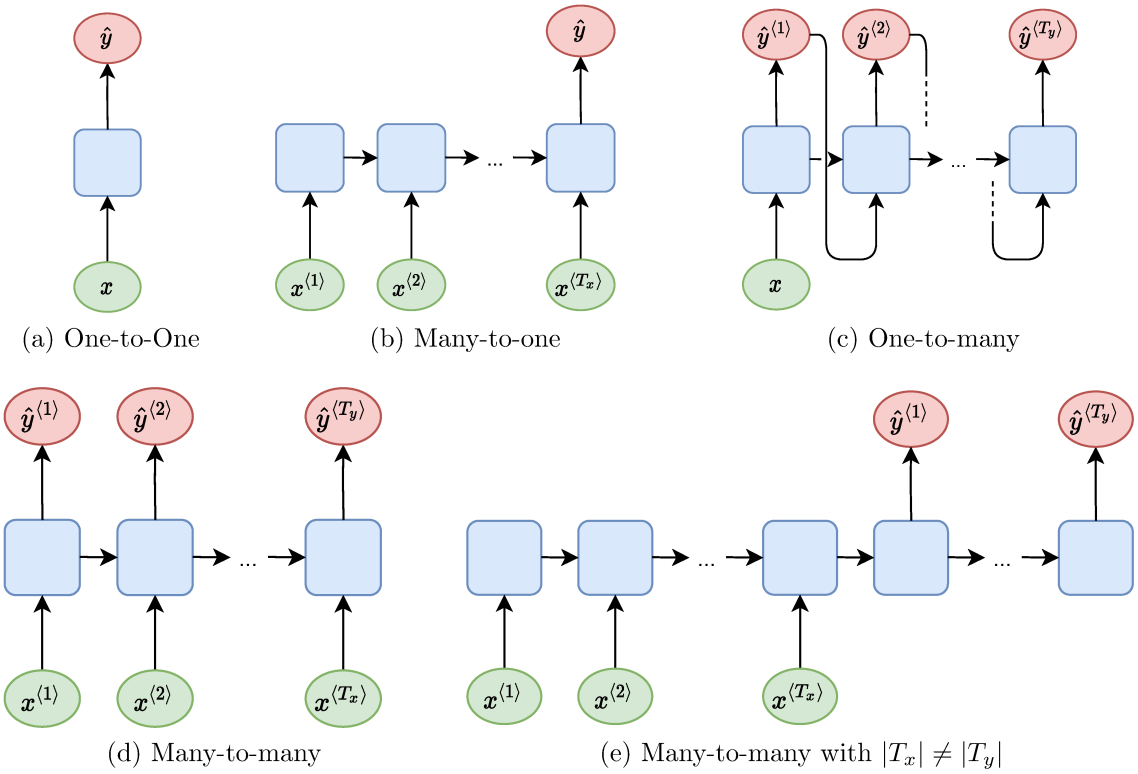
5.2 Recurrent models with ranking-based fingerprinting

We start this section by providing a brief introduction to Recurrent Neural Networks. Then, we formalize the fingerprint representation employed in the work, discussing its main advantages and limitations. Finally, we detail the two models developed in this work.

5.2.1 Preliminaries on RNN

Recurrent Neural Networks (RNN) are a particular class of neural networks often employed to deal with sequential data [222]. The idea behind RNNs is to model each time instant of a sequence with a *unit*. More precisely, given a sequence s with $|s| = T_s$, the i-th unit will be fed with the i-th time instant of s, denoted by $s^{\langle i \rangle}$. Units are linked together to make available at time t a state representation of the elaboration at time $t-1$. Different RNN architectures can be exploited for different tasks. Given x, with $|x| = T_x$, an instance of the training set with target variable y, and a prediction $\hat{y}$ for y (this notation will be adopted for the rest of the chapter), Figure 5.1 depicts different RNN architectures with examples of possible applications.

Despite their many advantages, simple RNNs struggle to handle long-term dependencies among their input elements. To overcome this limitation, two different kinds of unit have been proposed: Gated Recurrent Unit (GRU) [56] and Long



(a) One-to-One (b) Many-to-one (c) One-to-many

(d) Many-to-many (e) Many-to-many with $|T_x| \neq |T_y|$

Figure 5.1: Different types of RNN architecture (initial state omitted). Possible applications: (a) similar to a feed forward neural network; (b) sentiment analysis; (c) sequence generation (e.g., music); (d) name entity recognition; (e) machine translation.

Short-Term Memory (LSTM) [107]. In this work, we will make use of LSTM RNNs (LSTM for short), as they performed better on an early-stage evaluation. LSTM solves the limitations of classical RNNs by combining an internal state with different types of gate. A detailed description of LSTM is out of the scope of this work. For a comprehensive account of RNN and LSTM, we refer the reader to Chapter 10 of [97].

In the formulations of RNN and LSTM given till now, the structure of the model is such that at time t only information about the past is captured. Other architectures exist where the whole input sequence is taken into account. This is the case with bidirectional RNNs, that simultaneously manage the input sequence from the start to the end, and vice versa. The ability to take into consideration, at each time instant, both past and future information results into a network that at time t is more sensitive to the values surrounding t. This class of RNN showed good performance in fields like handwriting, speech recognition, and bioinformatics [97].

The last component worth introducing is the LSTM loss function, which consists of the sum of the losses at each time point $t \in 1, \ldots, T_y$, and is formally defined as

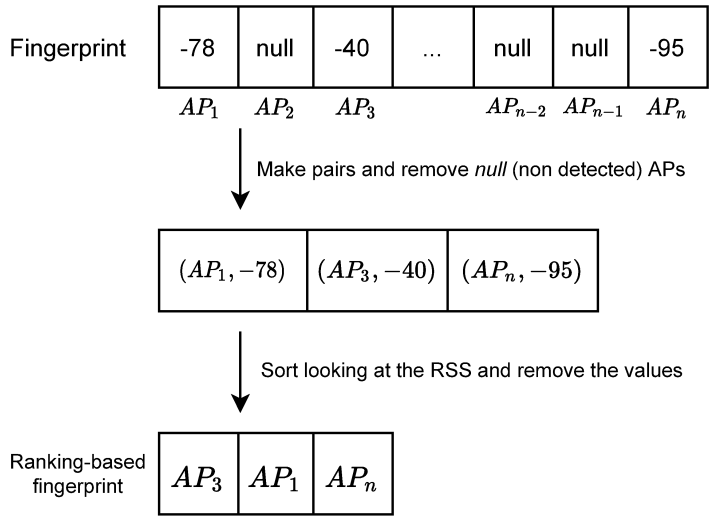


Figure 5.2: Ranking-based fingerprint representation construction process.

follows:

$$L(\hat{y}, y) = \sum_{t=1}^{T_y} L_t(\hat{y}^{\langle t \rangle}, y^{\langle t \rangle}) \ . \tag{5.1}$$

5.2.2 Ranking-based fingerprinting

Ranking-based fingerprinting is an effective approach to indoor positioning that has been first introduced in [54]. Then, in [164] deterministic-like ranking-based fingerprints combined with k-NN is proposed. The authors evaluated multiple metrics, concluding that Spearman's footrule was the most performing one. Then, a further extension [180] of such a work considered a more comprehensive set of distance functions, leading to the superiority of the Lorentzian metric. Another of the few works in this area is [291], where, instead of considering raw signals, authors devise an encoding of fingerprints based on RSS relationships, so as to manage device heterogeneity, with no calibration efforts, in a crowdsourced context. Ranking-based fingerprinting is based on the idea of transforming traditional fingerprints into a different representation. Specifically, each fingerprint is encoded by a vector of AP identifiers such that their position in the vector is determined by the RSS value of the corresponding AP, sorted from the strongest to the weakest signal (see Figure 5.2). APs that are not detected in a fingerprint are not present in its ranking-based representation.

To provide a more mathematical formulation, let's recall that, from the definition given in section 1.3.1, the kth fingerprint associated with the location $\mathbf{l}_i$ is denoted as the vector $\mathbf{f}_{i_k} = [f_{i_k 1}, f_{i_k 2}, ..., f_{i_k q}] \in \mathbb{R}^q$, where q is the total number of APs appearing in the considered scenario (sensed all over the locations in $\mathcal{L}$) and $f_{i_k j}$,

$1 \leq j \leq q$, is the RSS value related to the AP that is given the unique identifier j (or `null` if such AP is not detected). Then, a raking-based fingerprint can now be defined as an (ordered) vector $\mathbf{f^r}_{i_k} = [a_1, ..., a_z]$ s.t. $f_{i_k a_h} \geq f_{i_k a_{h+1}}$, $1 \leq h < z$, where a_h is an AP identifier, and z, $1 \leq z \leq q$ (although generally $z \ll q$), is the number of detected APs (i.e., with RSS different from `null`) in fingerprint $\mathbf{f}_{i_k}$ (equivalently, $\mathbf{f^r}_{i_k}$) at the location $\mathbf{l}_i$.

There are many advantages related to this design. First, the ranked representation is far more compact than the full-fledged fingerprint one, possibly reducing computational, storage and networking costs. In addition, ranked fingerprints are more robust to signal perturbations related to the heterogeneity of the devices: while two different devices might observe different RSSs from the same APs (e.g., due to diversities in their hardware), the corresponding ranked fingerprints will be much similar to each other, e.g., thanks to the fact that rankings are invariant to bias and scaling [155, 164]. Finally, classical fingerprints are sparse. At each location only a small subset of APs is visible with respect to the total q, which is often a large number, possibly leading to algorithmic level issues caused by the curse of dimensionality [4].

On the negative side, the informative content associated with ranked fingerprints is reduced. Indeed, we shift from a rich continuous representation of the RSS values to far simpler sequences of discrete/categorical identifiers. While in the majority of previous work this caused a degradation of the positioning performance [54, 136, 155, 254], we hereby show that an approach based on this technique can be as accurate as those based on classical fingerprints.

5.2.3 The developed models

As we already pointed out, the goal of the present work is to devise a model robust to RSS issues and easily applicable to different fingerprint-based datasets, while providing an accuracy performance at least on par with existing state-of-the-art solutions.

To accomplish the positioning task, we developed two independent deep learning models. The first *flattened model* is based on an architecture similar to the one depicted in Figure 5.1b. More precisely, given a ranking-based fingerprint as input, the model tries to predict the flattened representation of the position, e.g., *"building_A-floor_1-room_3C"* (it can be viewed as a multi-class classification problem, where the number of classes corresponds to the number of considered training locations). The LSTM network thus defines a mapping between the ranking fingerprint representation and the position, without using any information about the hierarchical structure of the building.

Unfortunately, such a representation does not allow one to reason about buildings, floors, and rooms independently. As an example, there is no way to tailor the learning algorithm so as to predict, with good performance, a specific label component. Such a capability can be of interest in applications where, for instance, floors

are particularly relevant. Another shortcoming of this solution is that the output layer consists of as many neurons as the total number of RPs taken into account. Since each RP may correspond to a different room, this number can be quite large, and the situation may further degenerate whenever RPs are defined over a grid, that partitions a building into a set of very small tiles.

A second *hierarchical model* has been developed to overcome these limitations. It produces a dedicated output for each hierarchical level of the considered scenario. As an example, in a multi-floor and multi-room setting, it outputs two labels, respectively predicting the floor and the room. The structure of the representation allows one to reason about the produced output (post-processing), as well as to take into account the prediction for a given level of the hierarchy when generating a prediction for a finer one, e.g., to exploit the building prediction when predicting the floor.

The path usually followed in the literature builds a set of hierarchically structured models, where each model is trained to solve a prediction task for a specific level of the hierarchy, e.g., a model for each building, and then a model for each floor of each building. On the contrary, the proposed LSTM approach accomplishes the goal in a very straightforward manner. The model has a structure very similar to the one depicted in Figure 5.1e, that is, a network with an encoder-decoder structure. The first part of the model is in charge of encoding the representation of the ranking-based fingerprint, which is fed to the model as in the case of the flattened approach. The output of the encoder consists of two components: the actual output and the state representation. Both of them are used by the decoder. The actual output is replicated and provided as input to each unit of the decoder, while the state representation is used as the initial state for the decoder component of the LSTM. The decoder is organized in such a way that the number of LSTM units is the same as the number of hierarchical levels taken into consideration. Each unit is trained to predict only a specific level of the hierarchy. This is accomplished using a multi-loss function defined as follows:

$$L(\hat{\mathbf{l}}_i^c, \mathbf{l}_i^c) = \sum_{j=1}^{|\mathbf{l}_i^c|} w_j \cdot L_j(\hat{\mathbf{l}}_i^c, \mathbf{l}_i^c) \ , \tag{5.2}$$

where $\mathbf{l}_i^c = (building_i, floor_i, room_i) \in \mathcal{B} \times \mathcal{F} \times \mathcal{R}$, with $\mathcal{B}$, $\mathcal{F}$ as defined in Section 1.3.1, is the categorical information about a location $\mathbf{l}_i$, and w_j is the weight of the loss specified for each hierarchical level j, that is, L_j, in terms of its contribution to the total loss.

The architecture for the case of a multi-building, multi-floor, and multi-room environment is sketched in Figure 5.3.

The approach can be easily tailored to richer (resp., poorer) hierarchical representations by simply adding (resp., removing) the LSTM units (and subsequent components) that are in charge of managing the corresponding parts of the output.

It is worth noticing that the usage of LSTM provides indoor positioning with

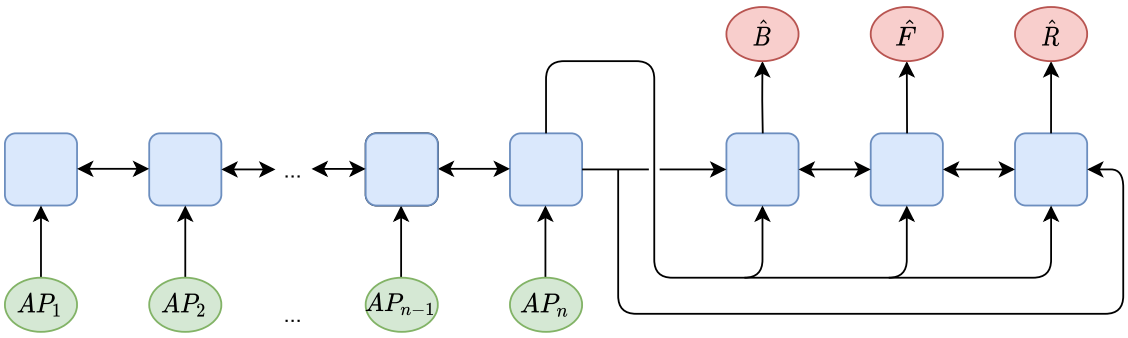


Figure 5.3: Representation of the proposed multi-output bidirectional LSTM-based model for a multi-building, multi-floor, and multi-room environment. $\hat{B}$, $\hat{F}$, and $\hat{R}$, stand for building, floor, and room predictions, respectively.

an additional relevant feature, namely the capability of handling sequences of APs of different lengths. They can be dealt with by introducing a dummy reference AP and concatenating it to the APs composing the ranking-based fingerprint (padding) just before feeding the latter to the LSTM. Another advantage of LSTM is the possibility of managing situations where new APs are identified. This often happens some time after the positioning system has been deployed, when new APs are added to the network infrastructure. These new APs can be dealt with by renaming them to a specific dummy AP introduced at training time. Obviously, such an ability is not unlimited, as if there is a substantial variation of the APs, or their placement is considerably modified, the performance will likely downgrade, possibly requiring a radio-map update and a (partial) retrain of the model. Even though experimental results on one of the considered datasets (see Section 5.3.1) seem to suggest that a quite good performance can be achieved even in these scenarios, a systematic investigation on the ability of the proposed models to resist long-term network changes must be carried out.

Both the described models have been implemented using bidirectional LSTM, according to the idea that a knowledge of the context, that is, a knowledge of both stronger and weaker APs in the ranking, may provide meaningful information for the task at hand. Bidirectionality has also been exploited for the decoder component of the hierarchical model. Knowing the prediction for a certain level of the hierarchy may indeed be relevant for the prediction of all the other (previous and following) ones.

Finally, for both models, an initial spatial dropout layer is used, and dropout and recurrent dropout are employed in all the LSTM units. Since we work in a multi-class classification-like setting, we used categorical cross entropy as the output loss function.

5.3 Experimental setting

The whole project has been developed on a virtual machine hosted on Google Cloud Platform, equipped with 8 virtual CPUs, 30 GB of RAM, and an NVIDIA P100 GPU. We relied on TensorFlow 2.1 with Keras as the development framework.

5.3.1 Datasets description

In this section, the three datasets considered for the evaluation of the proposed models are described. All of them are acknowledged as valid testbeds by the indoor positioning research community, thus enabling for a fair comparison with previous state-of-the-art algorithms. Compared to their general descriptions reported in Section 1.3.3, we applied some preprocessing in order to make the representation homogeneous and compatible with the learning task (categorical prediction).

UJIIndoorLoc [256] is likely the most well-known and exploited dataset for fingerprint-based indoor positioning. It describes a multi-building and multi-floor setting spanning an area of 108703 m^2, with the intent of mimicking the difficulties that could arise in the everyday usage of indoor positioning systems. It includes 19938 training and 1111 test fingerprints. Test fingerprints have been collected four months later than training ones, also using different devices. The temporal aspect is crucial since it allows testing the performance of a model on a scenario affected by a variety of dynamics, such as the introduction or removal of APs compared to training time and other environmental changes. Overall, 520 APs have been detected at multiple locations; around 50 of them are not identified during the training phase, becoming available only after the deployment. Thus, the dataset is a suitable candidate for the scenario outlined in Section 5.2.3. The median value of the number of non-null APs seen among all fingerprints is 16. Each location is identified by four categorical variables: building, floor, room, and relative position (the latter two are typically combined together), as well as by latitude and longitude. Globally, 904 distinct locations (reference positions) have been considered for the fingerprint sampling stage (training). The sampling has been done according to a room-based fashion (i.e., logical tessellation of Section 2.2) and not by grid-partitioning the building structure, thus the distances between the training locations, as well as the fingerprints' density coverages, are rather variable. Regarding the test set, room and relative position identifiers are not provided. This is a reasonable choice since, in the test phase, which aims to mimic the actual usage of the system, users may be located at an arbitrary position that may not match those used for the radio-map construction. To reduce the Out-of-Distribution (OOD) generalization problem [240] caused by a divergence between the scenarios modelled by the training and test sets (both in terms of collecting devices and considered locations), we built a validation set so as to replicate as much as possible such a difference. Specifically, the latter is a subset of the training set instances collected only by the devices with id 7 and 10. This choice allowed us to obtain a (partially) different distribution, without

removing any location from the ones observed in the training set (so as not to limit the spatial knowledge that can be learned by the algorithm).

UTSIndoorLoc [245] is a very recent dataset that describes a single building consisting of 16 floors. The total number of samples is 9494, where 9107 are considered for the training set and the remaining 387 for the test set. Overall, 589 different APs have been detected. The authors report that fingerprints have been collected at 1840 different locations (RPs). Unfortunately, RP information is not directly encoded in the dataset through a categorical variable, since only floor identifiers and latitude and longitude coordinates are available. In order to reconstruct the RPs, which can be seen as the lowest hierarchical level in this dataset, we assigned a label to each entry of the training set according to the results of the execution of agglomerative clustering based on the Euclidean distance between the points. The threshold for the clustering task has been set to $5m$.

Unlike the two previous datasets, the *Tampere crowdsourced* one, proposed in [155], is based on crowdsourcing. Thus, fingerprints are collected by users at random positions, rather than at predefined RPs. The considered scenario is a single building with five floors. The number of sampled fingerprints is 4648: 697 for training and 3951 for test purposes. 991 APs are visible all over the building. Positioning information is encoded as a triple consisting of latitude, longitude, and height. Floor information is given in terms of height and, thus, it has been converted into a categorical variable. In order to reconstruct the RPs, we applied the agglomerative clustering process we exploited for the UTSIndoorLoc dataset with a threshold set, again, to $5m$. Notice that, following this approach, an RP may end up with just one associated fingerprint, due to the randomness inherent in the crowdsourced collection process.

For all the datasets, the ultimate position of (true or reconstructed) RPs is determined by calculating the centroid of all the training instances labelled with such specific RP.

5.3.2 Hyperparameter tuning

Before illustrating the approach adopted for the evaluation of the models and, in particular, the evaluation metrics, we briefly discuss the hyperparameter tuning phase. Due to time and resource constraints, hyperparameters have been fine-tuned using iterative random search only for a specific model (hierarchical) and a single dataset (UJIIndoorLoc). The rationale behind this choice is: (*i*) UJIIndoorLoc is the most widely used dataset and (*ii*) the hierarchical model, due to its multi-loss component, is the hardest one to fine-tune. Nevertheless, as we are going to show in Section 5.4, the obtained hyperparameters produce good results for the other datasets and for the model based on the flattened representation as well. The selected hyperparameters are reported in Table 5.1. It is worth noticing that both the number of LSTM and hidden units depend on the problem, in the sense that they are computed on the basis of information related to each specific dataset.

Table 5.1: Hyperparameters for the proposed LSTM approach.

Parameter	Value
learning rate	0.05
# LSTM units	median(#AP for entry) + std(#AP for entry)
# hidden units	# unique APs in training phase
optimizer	SGDR
recurrent dropout encoder	0.4
dropout encoder	0.4
recurrent dropout decoder *	0.3
dropout decoder *	0.3
spatial dropout	0.2
batch size	128

The number of LSTM units does not take into account those needed for the hierarchical representation in the decoder component (3 of them). Parameters marked with * are not used in the flattened model.

The decision of using Stochastic Gradient Descent with warm Restarts (SGDR) [157] is based on the fact that it provides good performance in finding non-sharp, and thus more stable, local minima, which is an advantage in situations where the test set distribution may significantly differ from the training/validation ones. This is exactly the case with indoor positioning where, at test time, many points do not match the RPs used for training, devices are typically different, and even APs may have changed.

5.3.3 Evaluation metrics

In the following, we illustrate the metrics we are going to use to evaluate the experimental results, and show that they allow us to make an easy and meaningful comparison with existing solutions on the considered datasets. The accuracy of building prediction is evaluated by looking at the number of correct predictions vs. the number of wrong ones. We followed the approach at the base of most other papers, e.g., [155, 261], where additional information, such as floor and position, is considered to be meaningful if and only if the building prediction is correct. In particular, floor accuracy is evaluated in terms of success rate, that is, a value denoting the rate of entries for which both building and floor have been correctly predicted. The 2D positioning error is determined by looking at the Euclidean distance $E(\mathbf{p}, \hat{\mathbf{p}})$ between the predicted $\hat{\mathbf{p}} = (\hat{x}, \hat{y})$ and the ground truth $\mathbf{p} = (x, y)$ coordinates:

$$E(\mathbf{p}, \hat{\mathbf{p}}) = \sqrt{(x - \hat{x})^2 + (y - \hat{y})^2} \, . \tag{5.3}$$

The 2D error is computed only for those entries where building and floor are correctly predicted.

Given the error $E(\mathbf{p}, \hat{\mathbf{p}})$, it is possible to compute the Root Mean Square Error (RMSE) over the set of all estimated positions $\hat{\mathcal{P}}$:

$$RMSE(\hat{\mathcal{P}}) = \sqrt{\frac{1}{|\hat{\mathcal{P}}|} \sum_{\hat{\mathbf{p}} \in \hat{P}} E(\mathbf{p}, \hat{\mathbf{p}})^2} \,. \tag{5.4}$$

Since both the proposed models work in a classification fashion and return the most likely RP, to retrieve the coordinates of a point we proceeded as follows. We can assume the output of the model to be a list of probabilities, each one denoting the likelihood for an entry to be associated with a specific RP. Given the way in which fingerprinting works, the spatial coordinates for each RP are known. Therefore, it is possible to estimate the coordinates for an entry by computing the centroid over the coordinates of the k most likely RPs identified by the model. To make the evaluation more precise, it is also possible to rely on the probabilities to return a weighted estimate. We did that, setting $k = 3$ and using the inverse of the probabilities as a weight function.

Besides the tests aimed at verifying the performance of the models for each considered dataset, two other experiments have been carried out based on the `UJI 1` dataset. The first one consists of an injection of noise to simulate environmental changes possibly affecting the WiFi propagation and, consequently, the positioning accuracy. To this aim, additive white Gaussian noise following the distribution $Z \sim \mathcal{N}(0, \sigma)$ has been applied. We considered intensity values of noise (σ) between 1 and 5 dBm, following the claim of [262] that this is a good enough noise representation. The perturbed fingerprint is then obtained by adding a different random value according to Z to each of its original RSSs. Clearly, all perturbations are applied only to the test set, while the models are still the ones trained on the original, unperturbed data.

The last experiment aims at evaluating how the performance of the algorithm changes varying the maximum admissible length of the sequence of APs (i.e., the ranking) considered by the model. Note that such a choice directly influences the number of LSTM units, since such a number depends on the length of the rankings we consider. The value of this hyperparameter, as reported in Table 5.1, has been defined empirically by looking for a quantity that seemed appropriate for the considered scenarios. With this test, we wanted to understand how considering only a few, RSS-powerful APs, i.e., fewer APs than the default configuration, or, instead, the opposite, i.e., taking into account as many APs as possible, impacted on the accuracy of both the proposed models.

5.4 Results

Table 5.2 and Table 5.3 report the detailed results for both models applied to the considered datasets, while Table 5.4 shows a comparison between our best solutions

Table 5.2: Test set performances of the proposed models.

Dataset	Building (%)		Success rate (%)		Mean Error (m)		Median Error (m)		Min Error (m)		Max Error (m)		RMSE (m)	
	HM	FM	HM	FM	HM	FM	HM	FM	HM	FM	HM	FM	HM	FM
UJIIndoorLoc	100	99.9	94.3	94.1	6.78	6.66	4.82	4.88	0.01	0.04	77.75	77.41	9.65	9.34
UTSIndoorLoc	-	-	94.3	90.5	6.54	7.16	5.73	6.18	0.07	0.38	27.05	25.26	7.82	8.56
Tampere Crowd	-	-	92.3	86.3	7.92	9.79	5.89	7.02	0.15	0.08	94.75	96.06	11.06	13.81

HM = Hierarchical model, FM = Flattened model.

Table 5.3: Performance of both models applied to UJIIndoorLoc with different ranking lengths.

std mul	Len	Building (%)		Success rate (%)		Mean Error (m)		Median Error (m)		Min Error (m)		Max Error (m)		RMSE (m)	
		HM	FM	HM	FM	HM	FM	HM	FM	HM	FM	HM	FM	HM	FM
3	37	100	99.9	94.8	93.1	6.90	6.71	4.83	4.66	0.05	0.07	77.21	77.14	9.92	9.74
2	20	100	99.9	94.3	93.1	7.03	6.68	4.85	4.52	0.04	0.06	84.21	80.65	10.27	9.94
1 (d.)	23	100	99.9	94.3	94.1	6.78	6.66	4.82	4.88	0.01	0.04	77.75	77.41	9.65	9.34
0	16	100	99.7	94.8	94.3	6.57	6.82	4.58	5.04	0.09	0.01	77.37	77.30	9.52	9.72
-1	9	99.9	99.9	95.8	94.8	7.34	6.94	5.09	5.18	0.03	0.08	90.46	77.33	10.86	9.83
-2	2	99.4	99.6	89.6	87.8	11.21	11.72	9.43	9.86	0.09	0.27	85.30	90.13	14.33	14.78

std mul and *Len* represent the multiplication factor of the standard deviation in the formula $median(\#AP) + x \cdot std(\#AP)$ and the result of the formula, respectively. *d.* indicates the default configuration. HM = Hierarchical model, FM = Flattened model.

and the ones found in the literature[1]. The main outcome is that the achieved performance is better or comparable to other state-of-the-art approaches in this field. For what concerns UJIIndoorLoc, results of both models outperform the baseline [256] as well as improve over other contributions [122, 187, 193, 245, 247, 261], and are very close to other state-of-the-art approaches [5, 94]. For the sake of objective evaluation, it is correct to point out that our models perform visibly worse than just one other solution, that is, [259]. Here, the performance is optimised thanks to sector-based search: first, the sector of the user is identified, and only the fingerprints associated with that sector are used for comparison and position estimation. As for [112], where RSS time-series are exploited in combination with multiple convolutional neural networks, authors make their evaluation based on a split of the training set rather than using the official test instances and, as reported in Table 5.4, we achieved a comparable performance in a similar setting. Finally, we would like to mention that, compared to [105] and [106], two contributions where trajectories are exploited, our results may seem to be dramatically worst. However, the considered setup [105, 106] is quite different: they take into account fewer buildings and/or a reduced amount of users. When tested on the same restricted setting, the performance of our hierarchical model is comparable (see, again, Table 5.4).

Since all the previously mentioned contributions exploit available fingerprint data in their full generality, and sometimes even more complex models, e.g., ensembles, or representations, e.g., trajectories, the results that we achieved can be considered as optimal, as they have been obtained relying only on rankings (with no explicit RSS information), and without any kind of data preprocessing.

[1]Observe that some contributions used variants of the metrics we considered, or different splits of the datasets. A note column is provided in Table 5.4 to highlight the differences, reporting the performance of our approach in the same setting for the rows that require it and when possible.

Table 5.4: Comparison with the other state-of-the-art methods for the three considered datasets.

Dat.	Paper	Building (%)	Success rate (%)	Mean Err. (m)	Note
UJI	[256]	-	89.92	7.9	UJIIndoorLoc baseline
UJI	[261]	-	94.78	6.86	WKNN-like approach finetuned (parameter k and metric)
UJI	[193]	-	91.1	-	Only success rate is considered
UJI	[187]	100	94.0	7.73	Best IPIN 2015 competition solution (RTLS@UM, algorithm 1, variant 1). EvAAL metric with penalty on our same split (i.e., their validation). *With the same metric, our approach showed 7.42 mean error*
UJI	[187]	100	91.9	7.45	Best IPIN 2015 competition solution (RTLS@UM, algorithm 1, variant 3). EvAAL metric with penalty on our same split (i.e., their validation). *With the same metric, our approach showed 7.42 mean error*
UJI	[122]	99.82	91.18	9.29	-
UJI	[259]	-	95	3.94	Uses a sector-based search where, first, the sector of the user is identified, and only the FP associated to that sector are used for the comparison
UJI	[112]	100	100	2.77	Authors test on a partition of UJIIndoorLoc training set, not using the official test one. Time-series of fingerprint (i.e., sequences of fingerprint collected at the same location) are used. Specific and multiple models for each hierarchy are employed. *Following the authors' partitioning of the training set, our model achieved 100% building accuracy, 99.94% success rate, and 2.88m mean error*
UJI	[106]	-	-	5	Authors exploit trajectories and focus only on two users (13, 14), without considering floors or building. Moreover, a spatial constraint (20m) between two consecutive points is considered. *Restricting to the same users, our model reported a mean error of 5.4m*
UJI	[5]	-	-	6.46	Buildings are considered separately, flattened representation is used without evaluating floor accuracy. Rooms with few fingerprints are removed
UJI	[105]	-	-	4.2	Authors exploit trajectories and focus only on two users (13, 14), without considering floors or building. Data collected in building 2 are ignored *Restricting to the same users and buildings, our model achieved a mean error of 4.9m*
UJI	[247]	100	93	8.63	-
UJI	[245]	100	96	11.78	-
UJI	[94]	100	95.41	6.78	-
UJI	our	100	94.8	6.57	Hierarchical model with best ranking length
UTS	[245]	-	94.57	7.60	-
UTS	our	-	94.3	6.54	Hierarchical model applied to the UTSIndoorLoc dataset. No ranking length and hyperparameters optimisation are performed
Tamp	[155]	-	92.99	8.65	Best solution on Tampere dataset, employing an approach similar to [261]
Tamp	[155]	-	92.26	8.45	Second best solution on Tampere dataset
Tamp	[155]	-	86.61	9.77	Ranking-based fingerprinting approach proposed in [164] applied to the Tampere dataset
Tamp	our	-	92.3	7.92	Hierarchical model applied to the Tampere dataset. No ranking length and hyperparameters optimisation are performed

The above claim is also supported by the results obtained on the other datasets. As for the UTSIndoorLoc dataset, the performance of the hierarchical model is almost identical to the one achieved by [245]. As for the Tampere crowdsourced dataset and the findings reported in [155], the results achieved by the hierarchical model are better than all but one of the approaches tested by the authors (three of them are reported in Table 5.4), and very close to the performance reached by the best one. It is also extremely encouraging that, according to the results shown in [155], our solutions drastically outperform another well-known ranking fingerprint-based approach [164] (see, again, Table 5.4).

Further interesting insight can be derived from Table 5.2: in all but the UJIIndoorLoc dataset, the hierarchical model achieves a much better performance in building/floor detection than the flattened one. In the case of UJIIndoorLoc, we investigated the similarity of the obtained results by comparing the empirical cumulative distribution functions (ECDFs) of the two models. Three different ways of partitioning the whole UJIIndoorLoc dataset have been taken into account. The first one is the test set described in Section 5.3.1. The other two are subsets of the original training data that have been used as validation sets during the training phase: one is just a classical random split (10% of the training set), while the other (roughly of the same size) corresponds to all the instances belonging to the training set labelled with *phoneid* equal to 7 or 10. The rationale behind this second validation set is to focus on devices which are not in the training set (all of them have been removed to build this validation set), in order to adequately mimic the distribution and the difficulties of the real test set. From an analysis of the ECDFs depicted in Figure 5.4, it emerges that the models have roughly the same performance when applied to the same set.

As a consequence, the difference in performance between the two models on non-UJIIndoorLoc datasets can be attributed to the capability of the hierarchical model to manage scenarios that make the other two datasets structurally different from UJIIndoorLoc, such as the way in which RPs are partitioned, the number of training points that are collected for each RP, and the complexity of the site from a physical point of view. The hierarchical model seems to generalise better over these features, suggesting its appropriateness in many different real-world scenarios.

Let us now analyse the effects of RSS perturbations. As we already mentioned, this is an important contribution of our work. The ECDFs for the hierarchical and the flattened models, trained on the original, unperturbed data, and applied to the RSS-perturbed test dataset, are reported in Figure 5.5 and Figure 5.6, respectively. In the event of sensitivity of the models to RSS variations, we should have observed an increase in the error reported by the ECDF. This is not the case, as the difference between the depicted curves is considerably small. This is confirmed by Table 5.5 as well, which shows only a slight worsening of the considered performance metrics as the intensity of the perturbation increases. The fact that the results are similar for the two models suggests that pairing the ranking-based representation of the fingerprints with RNNs is a possible way to relieve positioning from the effects of

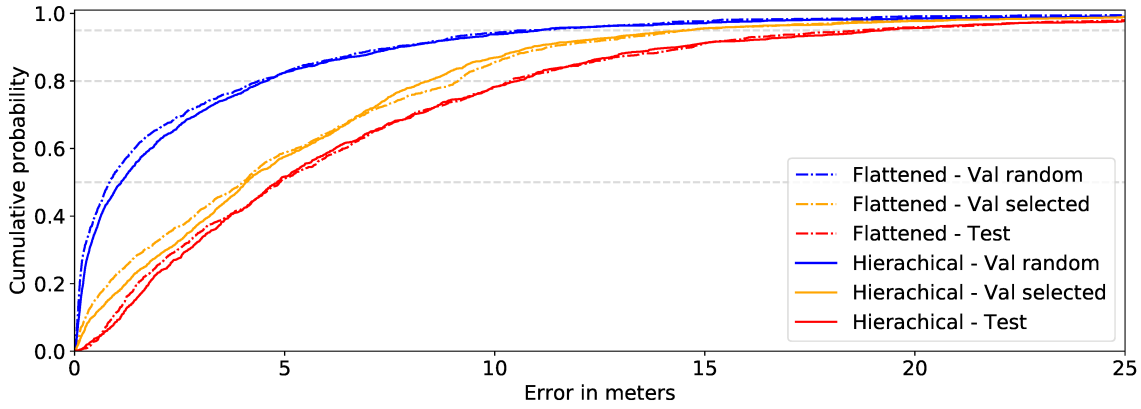


Figure 5.4: ECDFs showing the performances of both models for UJIIndoorLoc dataset.

Table 5.5: Results of RSS perturbations applied to the UJIIndoorLoc test set.

σ	Building (%)		Success rate (%)		Mean Error (m)		Median Error (m)		Min Error (m)		Max Error (m)		RMSE (m)	
	HM	FM	HM	FM	HM	FM	HM	FM	HM	FM	HM	FM	HM	FM
0	100	99.9	94.3	94.1	6.78	6.66	4.82	4.88	0.01	0.04	77.75	77.41	9.65	9.34
1	100	99.9	95.0	93.6	6.80	6.62	4.89	4.87	0.04	0.10	77.89	77.46	9.84	9.49
2	100	100	94.7	94.1	7.00	6.73	5.09	4.86	0.08	0.07	77.51	77.27	10.03	9.57
3	100	99.9	95.0	93.4	7.04	6.78	5.22	4.86	0.03	0.04	77.53	84.16	9.93	9.84
4	99.1	99.9	94.2	92.2	7.13	7.26	5.40	5.22	0.06	0.04	77.64	77.45	9.90	10.29
5	100	99.9	94.1	91.9	7.30	7.23	5.19	5.19	0.08	0.10	77.54	80.40	10.37	10.34

The value σ is the parameter at the base of the applied Additive white Gaussian noise following the distribution $Z \sim \mathcal{N}(0, \sigma)$. HM = Hierarchical model, FM = Flattened model.

RSS perturbations. In order to compare our robustness performance with the results reported in the literature, we tested the resiliency of the approach proposed in [261] (and considered also in [155]). Figure 5.7 reports the variation in performance for what concerns the success rate of our hierarchical model vs. the solution given in the literature. As it can be observed, in a similar setting, as the noise increases, our approach is affected by a lower drop in performance.

The final evaluation that has been carried out aims at understanding the impact on the achieved results of the number of APs considered in the ranking. To produce the results illustrated till now, such a number has been set equal to the median of the number of visible APs of a fingerprint plus the standard deviation of the same quantity (see Table 5.1). Without varying any other hyperparameter, different thresholds have been tested on the UJIIndoorLoc dataset, setting them according to the formula: $median(\#AP) + x \cdot std(\#AP)$, where $x \in \{-2, -1, 0, 1, 2, 3\}$.

The cumulative distribution and the probability density function of the number of visible APs for each fingerprint, with the considered thresholds, are reported in Figure 5.8. From Figure 5.8a, it is worth observing that the distribution is right-skewed, meaning that there is a consistent number of fingerprints that have a larger number of APs compared to the others.

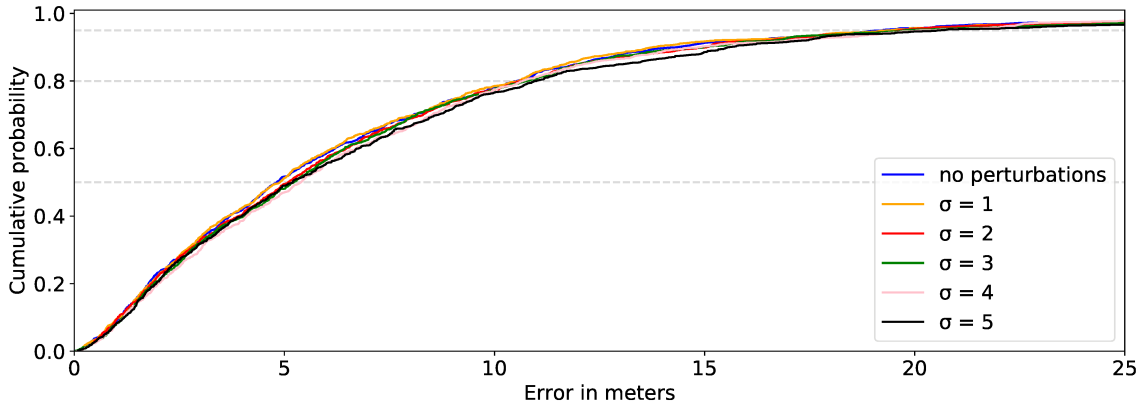


Figure 5.5: ECDFs showing error variations at different rates of perturbations σ for the fully hierarchical model.

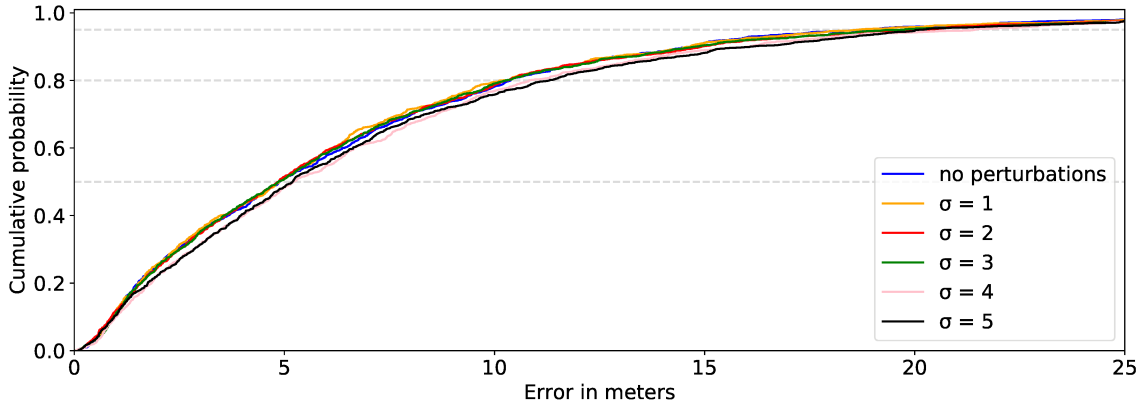


Figure 5.6: ECDFs showing error variations at different rates of perturbations σ for the flattened model.

Figure 5.9 and Figure 5.10 show ECDFs variations at different ranking lengths for the hierarchical and the flattened model, respectively. In general, comparable results are obtained, except for the case when the minimum number of APs is considered, that is, very short ranking lengths. Moreover, looking at Table 5.3, it is possible to see that the best performance of the hierarchical model is achieved not with the default setting, but with a ranking length equal to the median of the number of APs visible by each fingerprint.

The main take-home message from this last analysis is that, while it is possible to achieve good performance even taking into account a relatively small number of APs, such a value is quite relevant for the proposed approach, and its proper selection may lead to a further improvement of the overall model performance.

Of course, a sufficient number of APs must be observable so that different locations reflect on different rankings. This may be an issue in large public-space

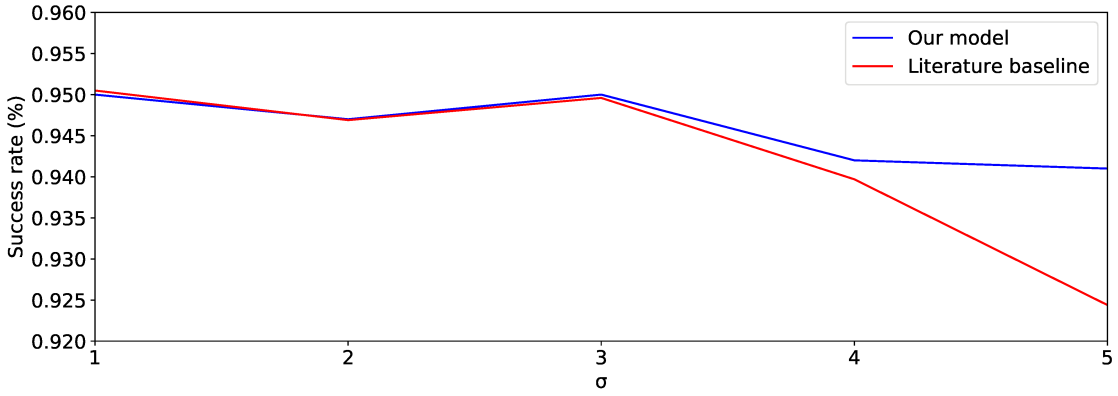
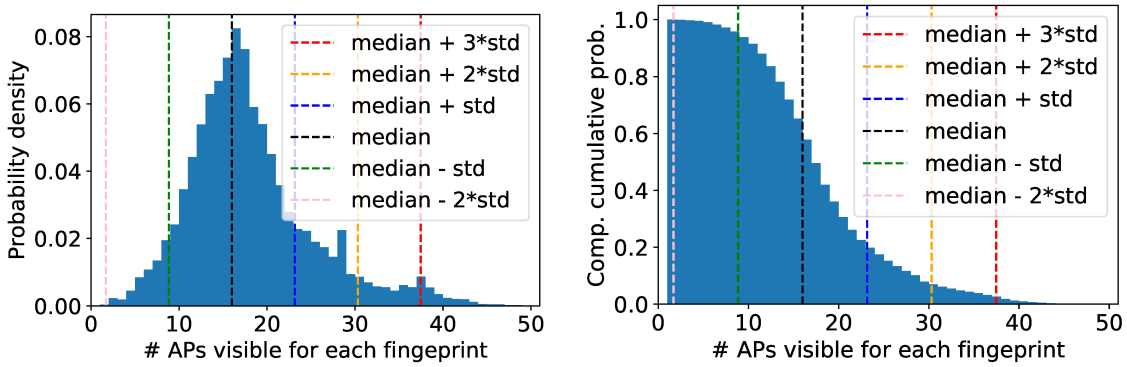


Figure 5.7: Comparison of performance between our hierarchical model and the literature solution proposed in [261] on the perturbed UJIIndoorLoc dataset.



(a) (b)

Figure 5.8: Probability density function 5.8a and complementary cumulative distribution function 5.8b of the number of APs visible for each fingerprint of the UJIIndoorLoc training dataset. Thresholds used for testing are depicted.

scenarios, which are characterised by a low density of APs, as suggested in [305], and where different techniques may be employed (e.g., a combination of multiple sensors).

Last but not least, since the ranking length determines the number of units in the LSTM, its setting is crucial not to produce too big models. Future investigations of the relationships between such a parameter and the other features characterising fingerprinting systems, such as, for instance, the density and arrangement of RPs, will be carried out.

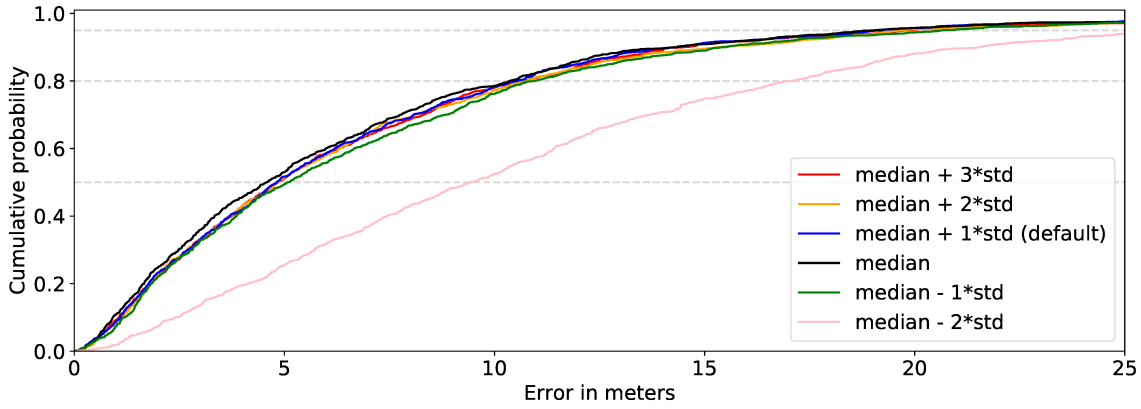


Figure 5.9: ECDFs showing error variations at different ranking lengths for the fully hierarchical model.

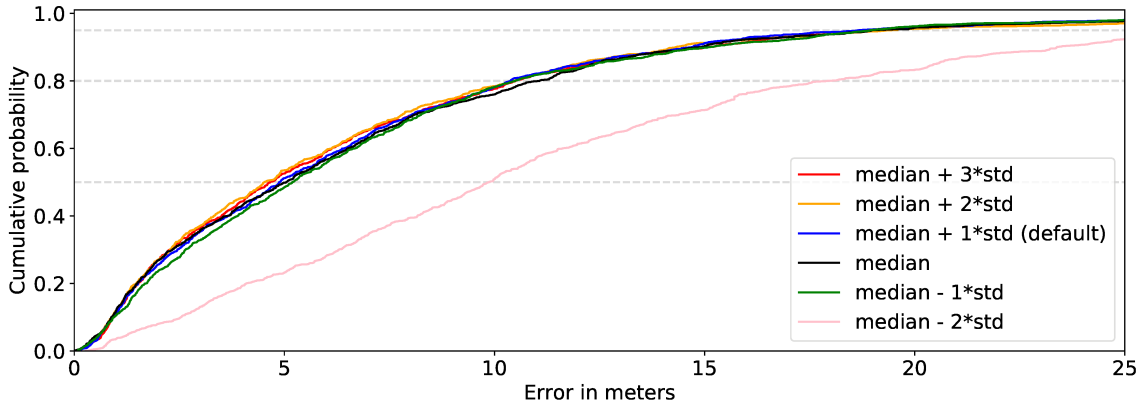


Figure 5.10: ECDFs showing error variations at different ranking lengths for the flattened model.

5.5 Discussion

In this chapter, a new approach to indoor positioning combining LSTM recurrent neural networks with a ranking representation of WiFi fingerprints is presented. The proposed original methodology shows performance comparable to state-of-the-art algorithms, that exploit the full potential of the RSS associated with the access points. At the same time, the models we developed are simpler, as they ignore RSS data and, thanks to this, robustness to RSS perturbations is naturally achieved. In addition, since we rely on a reduced set of information, for which no preprocessing is needed, the developed models can be seamlessly applied to any kind of fingerprint-based dataset. To prove such claims, we performed an extensive evaluation of the two models against multiple datasets, both in normal conditions and in artificially perturbed ones. The model that fully exploits the inherent hierarchical structure of indoor premises achieves the best overall performance, showing the ability to scale

properly across datasets of different sizes and characteristics. It is thus reasonable to conclude that the combination of ranking-based fingerprints and LSTM appears to be a highly promising approach to effectively deal with the indoor positioning task and related issues.

In light of the remarkably positive and encouraging obtained results, we asked ourselves additional questions. Indeed here we showed that raw RSS appears to be not required if we use deep learning models that exploit the correct inductive bias linked with the fingerprint representation we are using. However, could we get additional scientific and operational insights from the usage of deep learning? Can we understand what the model is doing? And, above everything, can we use this potential additional knowledge to support other concrete tasks related to indoor positioning? In the next chapter, we will show how interpretability, in the context of our RNN + ranked fingerprinting framework, can be a tool which appears to give affirmative answers to all these questions.

6

Towards Interpretability in Fingerprint-based Indoor Positioning

The work described in the previous chapter, to the best of our knowledge, is the only attempt in the literature to combine deep learning with ranking-based fingerprinting. The aim of this chapter is to build on the encouraging results presented in such a study by introducing and focusing on the contributions brought by the attention mechanism [11], which we believe is the enabling element for interpretability in fingerprint-based indoor positioning: our ultimate goal. In fact, recently, the lack of interpretability of localization approaches has been questioned [45]. This topic lacks in-depth studies, although it is of great importance considering that indoor positioning is often tackled by means of machine learning [13, 104, 142, 192], where, again, interpretability is a strong yet still largely unfulfilled requirement. Indeed, even if a very good model to address the localization task is available, understanding why it works, to gain scientific and operational insights that go beyond the original task, is compelling. To mention one possible impact, it may help to predict unanticipated failure cases of the system.

Our choice of relying on attention is motivated by domain knowledge rather than mere empirical findings: it is inherently true that some access points are more significant than others when it comes to determining the most probable location of a user, however, we question whether such a role is always played by the most powerful ones. Knowing what the most relevant access points are, irrespective of their RSS, may help in understating the cause behind a model prediction, which adheres to a local definition of interpretability, such as the one provided in [179]. In addition, from a global perspective, access point relevance patterns may characterize different areas of a building and, in turn, this could benefit a variety of tasks, ranging from radio-map maintenance, to the identification of wrong predictions, and to the overall improvement of localization accuracy.

In the work described in this chapter, for the first time, we propose a concept of interpretability for the fingerprint-based indoor positioning domain that links access point relevance with position estimation, and we discuss its practical implications in supporting several positioning-related tasks. In particular, we show that access point relevance patterns obtained through the attention scores of a sequence-to-

sequence deep learning model for ranked-based fingerprints (i.e., extending the idea of Chapter 5) have a strong spatial characterization: the explanations convey specific information of the indoor positioning domain. In order to validate our findings, we perform a series of qualitative and quantitative experiments, also with the aim to quantify the extent to which different attention compatibility functions align with our notion of interpretability, identifying a clear superiority of the *additive* one. Moreover, as a by-product of interpretability, we show how, assuming an optimal strategy to combine attention scores and deep learning model likelihoods, it is possible to improve the overall performance in position estimation.

The rest of the chapter is organized as follows. In Section 6.1, we give an account of interpretability in machine learning and discuss our idea of interpretability for fingerprinting. Then, in Section 6.2 we detail the deep learning model we designed to achieve it. In Section 6.3, the devised experiments and their results are reported. Finally, a thorough discussion of the overall outcomes and their possible practical applications is outlined in Section 6.4.

6.1 Interpretability in fingerprinting

In this section, first, an overview of interpretability in machine learning and its applications is provided. A description of our idea of interpretability for fingerprinting then follows.

6.1.1 An account of interpretability in machine learning

Interpretability in machine and deep learning refers to the ability of an algorithm or a model to provide clear and understandable explanations for its predictions. There exist multiple approaches to achieve interpretability, tied to the type of machine learning model and task at hand [191]. *Inherently interpretable models*, such as decision trees, logistic regression, and linear models, are often easy to interpret since their internal workings are transparent, and can be easily understood in terms of their parameters and decision rules. However, they may not always capture the complexity of the data, thus their performance may be insufficient in some applications. *Semi-inherently interpretable models*, also referred to as *example based methods*, use examples as the basis for their interpretation (e.g., K-NN). *Joint training interpretability* techniques enhance models with interpretability features, such as attention mechanisms or attribute importance scores. The latter include gradient-based methods, like integrated gradients and gradient-weighted class activation mapping (Grad-CAM) [235]. Finally, *post-hoc techniques* focus on explaining the decisions of models without modifying their internal workings. Notable approaches in this class are model-agnostic methods, such as LIME (Local Interpretable Model-Agnostic Explanations) [217] and SHAP (Shapley Additive Explanations) [160].

Examples of applications of interpretability techniques are: in the detection of

heart diseases [275] or fraudulent bank transactions [210], where SHAP can determine the features' contributions to the final output; scenarios such as medical imaging or object recognition, where Grad-CAM can highlight the regions of an image that are most important in making a prediction [197]; and, natural language processing tasks like sentiment analysis or text classification, where LIME can shed light on the decisions of black-box models [169].

Overall, interpretability techniques are critical in ensuring the accountability and trustworthiness of machine learning models, particularly in high-stakes domains such as healthcare and finance. Achieving full interpretability remains a challenging task, and a trade-off between accuracy and interpretability is often required. Therefore, the development of transparent and interpretable machine learning models that can provide reliable and trustworthy explanations is an active area of research.

6.1.2 Our proposal for fingerprinting

As previously mentioned, according to [179], interpretability[1] is the degree to which an observer can understand the cause of a decision. It is not about figuring out everything about a model, but it can be considered as a means to an end, which implies that the form taken by an explanation depends on the needs of the specific application [191]. Our goal is to enhance the positioning process by gaining novel scientific knowledge and operational insights, without worsening the performance concerning the position estimation. In the context of WiFi fingerprinting, we hereby define the concept of local interpretability (i.e., an explanation related to a specific prediction) as the relevance that each access point has to a given position estimate provided by the model. We name this as the *relevance pattern* for a (ranked) fingerprint.

In the past, several works either assumed or pointed out the very prominent role played by the most strongly (in terms of RSS) perceived AP. Indeed, this is one of the motivations that led to the development of the ranking-based fingerprint representation. We believe, instead, the strongest AP(s) not being necessarily the most relevant for the positioning task. For instance, a set of very powerful APs might be detected more or less in the same way at several different places; in that case, the discriminative role could be played by some other less powerful APs, that are only seen at specific locations. Therefore, determining which APs are mostly used by the model to derive a given prediction is a natural way to interpret its behaviour, as well as to highlight some characteristics related to the considered scenario.

In fact, other than local interpretability, it is worth asking ourselves whether a concept of global interpretability can also be defined, i.e., a general insight into the model behaviour based on a set of input data. Let us consider a set of predictions (i.e., location estimates) for which the relevance of the associated access points is known, and let us assume to group them together based on the similarity of

[1]Note that in the remainder we are going to use the terms explainability/interpretabilty, and explanation/interpretation interchangeably.

their relevance patterns. We can derive such a global insight by studying whether the average relevance pattern associated with each different group is capable of (uniquely) characterising a delimited spatial area. In the remainder of the work, we are going to name such a property as *spatial characterization*, and we believe it would be of actual use in several positioning-related tasks (as we will see in Section 6.4.4).

To conclude the section, we now discuss how to obtain, in practice, a measure of relevance for the access points, which considers both the fingerprint representation and the chosen positioning algorithm. A possible approach to obtain interpretability is to train the model and the explanation jointly [191]. In neural networks, the attention mechanism can be used to dynamically highlight relevant features of the input data [92, 185]. At a high level, it works by assigning a weight to each input element, based on its relevance to the current task. These weights are computed using a learned function that takes into account the current state of the model and the input data. The weighted elements are then combined to produce a context vector that represents the most relevant information for the task at hand. Thus, in principle, relying on a deep learning model that can be easily extended with the attention mechanism (such as the one proposed in Chapter 5) should allow us to reach our interpretability goal.

However, whether or not attention and its weights can be considered as a form of model explainability is still an open debate, for instance, in the NLP community [116, 236, 282]. It follows that blindly applying attention is not enough to ensure interpretability. Thus, in the remainder of this work, we show how and why in our case it provides plausible explanations related to the positioning domain. Specifically, *(i)* attention is indeed capable to highlight relevant access points associated with a prediction; *(ii)* attention has a strong spatial characterization; *(iii)* different attention types have different behaviours and capabilities (i.e., not all formulations are equally good); and *(iv)*, attention-based explanations can contribute to downstream positioning tasks.

6.2 Fingerprinting with deep learning and attention

In this section, we provide a description of the considered deep learning model. Then, we detail the attention mechanism and how to integrate it into the model and, thus, in indoor positioning. Compared to the informal description provided in chapter 5, we will be more precise about the mathematical definition of the problem, in such a way as to both show that this family of approaches fits in the probabilistic indoor positioning category as well as the precise contribution brought by the attention mechanism.

A graphical overview of the overall framework and how information flows throughout it is reported in Figure 6.1.

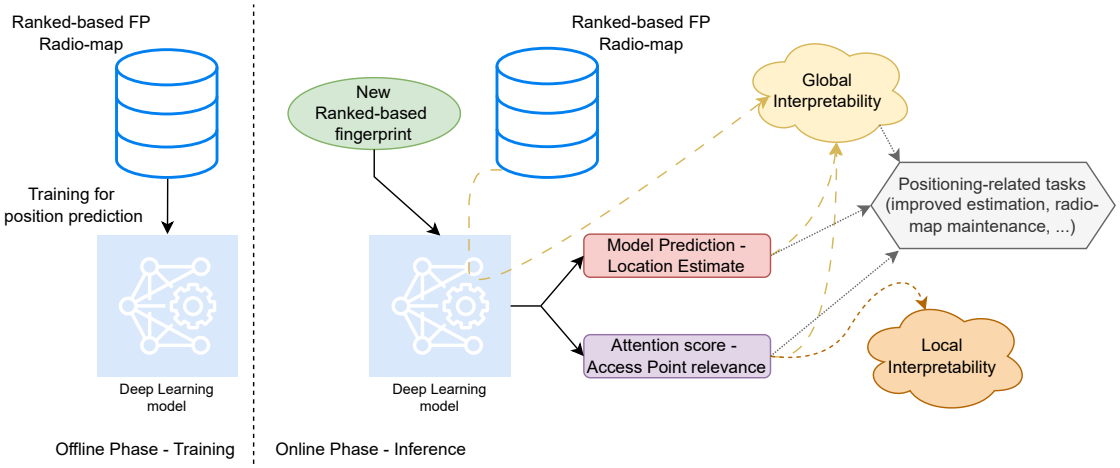


Figure 6.1: Graphical representation of the elements composing the framework, their integration, and how information flows throughout it (i.e., from the radio-map and model training, to the generation of the interpretability outcomes, the positioning estimate, and the combination of such aspects for multiple tasks).

6.2.1 A sequence-to-sequence modeling of probabilistic positioning

Ranking-based fingerprints can be analyzed by means of deep learning approaches that manage sequential data, such as RNN, LSTM, and Transformer architectures. Specifically, we are interested in sequence-to-sequence (seq2seq) models, which turn a sequence into another one. Considering them from a probabilistic perspective, it is possible to observe that (*i*) they perfectly fit in the probabilistic-based indoor positioning paradigm, and (*ii*) the decoder can model and leverage the hierarchical structure of positions (e.g., buildings containing floors containing rooms) through its autoregressive nature.

Given an (array-like) input sequence $\mathbf{x} = [x_1, \ldots, x_n]$ and a (array-like) target sequence $\mathbf{y} = [y_1, \ldots y_m]$ we can define a probabilistic sequential model as follows:

$$P(\mathbf{y}|\mathbf{x}) = P(\mathbf{y}_{1:m}|\mathbf{x}_{1:n}) = \prod_{i=1}^{m} P(y_i|\mathbf{y}_{1:i-1}, \mathbf{x}_{1:n}) \ . \tag{6.1}$$

Considering a position defined as, for instance, the triplet $(building, floor, room)$, and $\mathbf{x}$ as a (ranking-based) fingerprint, it follows from Eq. (6.1) that the model adheres to the indoor positioning probabilistic framework:

$$\begin{aligned}(\widehat{building}, \widehat{floor}, \widehat{room}) = \hat{\mathbf{y}} &= \arg\max_{\mathbf{y}} P(\mathbf{y}|\mathbf{x}) \\ &= \arg\max_{\substack{building \in \mathcal{B}, \\ floor \in \mathcal{F}, \\ room \in \mathcal{R}}} P(building, floor, room|\mathbf{x}) \ ,\end{aligned} \tag{6.2}$$

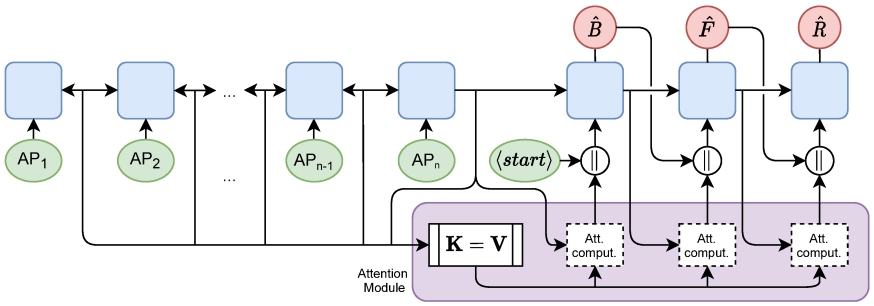


Figure 6.2: Sequence-to-sequence LSTM model with attention for hierarchical position estimation.

where $\mathcal{B}$, $\mathcal{F}$ and $\mathcal{R}$ are respectively the set of all possible categorical labels for the buildings, the floors, and the rooms. Applying the chain rule of probability to Eq. (6.2), we obtain:

$$\begin{aligned} P(building, floor, room|\mathbf{x}) = P(room|floor, building, \mathbf{x}) \\ \cdot P(floor|building, \mathbf{x}) \\ \cdot P(building|\mathbf{x}) \ . \end{aligned} \tag{6.3}$$

Thus, the predictions for the higher hierarchical levels $\mathbf{y}_{1:i-1}$ (e.g., building and floor) are explicitly taken into account by the model to predict the lower levels $\mathbf{y}_{i:m}$ (e.g., room) in a natural and generalizable way. This is in contrast with the majority of previous indoor positioning approaches which used to handle the inherent hierarchical relationships by a set of specialized models arranged in a cascade fashion.

6.2.2 The developed model

In this work, we consider a sequence-to-sequence unidirectional LSTM; the choice of neglecting bidirectionality in the encoder was based on an empirically observed overfitting phenomenon when paired with attention. To allow for the analysis of how attention contributes to the positioning task, the model is equipped with a Bandahu-style attention [11] module. The set of keys/values is generated by the encoder hidden states, while the query employed at each decoder step is given by the output at the previous step. Figure 6.2 depicts the overall architecture. As can be seen, and as done in NLP, to exploit the autoregressive behaviour of the decoder an initial input token $\langle START \rangle$ has to be provided. Teacher forcing [283] has been employed at training time to always provide the correct prior information during the sequential decoding phase.

Without delving too much into details, let us now understand how an attention-enhanced sequence-to-sequence model differs from a vanilla one, like the architecture used in Chapter 5, which can be summarized as in Figure 6.3. To do that, we refer to the probabilistic framework described in Eq. (6.1).

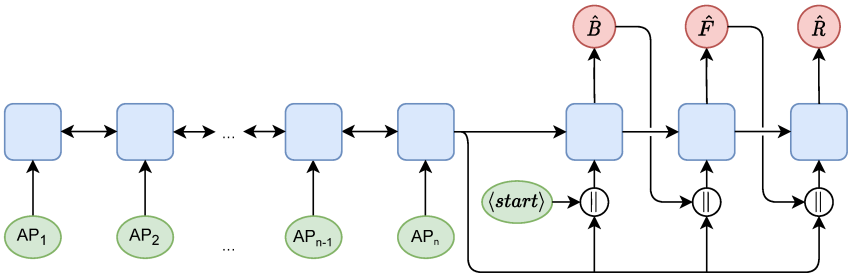


Figure 6.3: Vanilla sequence-to-sequence LSTM model for hierarchical position estimation.

We shall see that the main difference consists of how the context vector $\mathbf{c}$ is constructed and carried on during the computation, allowing the model to interact with the input information in different manners. The encoder component of both models can be represented as a function f_{enc} defined as:

$$f_{enc} : \mathbf{x}_{1:n} \rightarrow [\mathbf{h}_1, ..., \mathbf{h}_n], \text{ with } \mathbf{c}_0 = \mathbf{h}_n \ , \tag{6.4}$$

i.e., a function that produces a sequence of new partial input representations $\mathbf{h}_i$ (referred to as encoder hidden states), the latter of which is a new (squeezed) representation (fixed-size vector) of the overall input sequence, also referred to as the (initial) context vector $\mathbf{c}_0$. It follows that the decoder, in a probabilistic fashion, can be defined as:

$$P_{dec}(\mathbf{y}_{1:m}|\mathbf{c}_0) = \prod_{i=1}^{m} P_{dec}(y_i|\mathbf{y}_{0:i-1}, \mathbf{c}_0) \ , \tag{6.5}$$

where y_0 is the $\langle START \rangle$ token. Each decoder step i, considering both the previous output and the previous context vector, is thus modeled as follows by the recurrent architecture:

$$\begin{aligned} P_{dec}(y_i|\mathbf{y}_{0:i-1}, \mathbf{c}_0) = P_{dec}(y_i|\mathbf{s}_i) = \text{softmax}(\mathbf{s}_i), \\ \text{with } (\mathbf{s}_i, \mathbf{c}_i) = f_{dec}(y_{i-1}, \mathbf{c}_{i-1}) \ . \end{aligned} \tag{6.6}$$

While in the vanilla recurrent model the decoder hidden state $\mathbf{s}_i$ matches the context vector $\mathbf{c}_i$, in the attentive model $\mathbf{c}_i$ is computed by the attention mechanism, by explicitly and jointly considering all $\mathbf{x}_{1:n}$ and $\mathbf{s}_i$. This implies that, while the attention model can directly access and leverage the input information (more details later), the vanilla one only can do that through the condensed vector $\mathbf{c}_0$ obtained as output from the encoder, which is further modified at each step of the decoder[2].

[2]What we have just explained is a *greedy-search* approach, that at each time step considers the element with the highest conditional probability. *Beam search* is a more sophisticated approach, in which at each time step the k elements with the highest conditional probabilities are considered. At each subsequent time step, based on the k candidates at the previous step, it selects other k output elements with the highest conditional probabilities, intuitively generating a tree of possible output sequences [300]. In our work, we relied on beam search with $k = 5$.

Given that the considered positioning problem is framed as a multi-class classification one, and considering L_i as the negative log-likelihood loss applied to each output element, $1 \leq i \leq m$, the overall loss function L is defined as follows:

$$L(\hat{\mathbf{y}}, \mathbf{y}) = \sum_{i=1}^{m} w_i \cdot L_i(\hat{y}_i, y_i) \ , \tag{6.7}$$

where w_i is an optional weight of the loss related to the hierarchical level i.

As a final remark, it is worth pointing out the reasons behind the choice of considering only recurrent models rather than the far more (nowadays) popular Transformer [200] architectures. First, based on preliminary analysis, the performance of LSTM models was observed to be superior in our task with respect to Transformer, the latter perhaps being hindered by the relatively small quantity of training data at its disposal. The second point pertains to the interpretability of the overall approach. Indeed, Transformer makes full use of the attention mechanism, providing different representations at each encoder and decoder layer. Nevertheless, involving multiple layers would have made it more complex to obtain a representation both interpretable and exploitable for our purposes, even considering just the decoder. Conversely, the LSTM model provides just a single attention vector for each sequence.

6.2.3 Attention mechanism

In the case of sequences, like those considered in NLP and our scenario, the core principle behind attention is that of computing a weight distribution on the sequential input, assigning a higher score to those elements that are more relevant for the task at hand.

The attention mechanism can be defined as a weighted average of values that operates on three elements often referred to as keys (K), queries (Q), and values (V). This notation has been first proposed in [264], but it is also possible to think of these elements from a database point of view: we want to determine how much the tuples (K) in a table are relevant (weak definition of matching) for a given input ($q \in Q$), returning as output a value (V).

Formally, let us define $\mathbf{K} \in \mathbb{R}^{d_k \times n_k}, \mathbf{Q} \in \mathbb{R}^{d_k \times n_q}$, and $\mathbf{V} \in \mathbb{R}^{d_k \times n_v}$ as matrices, each composed of a variable number of column vectors (respectively n_k, n_q, and n_v) having the same size (d_k). For our purposes, we will consider the query to be just a single column vector $\mathbf{q} \in \mathbb{R}^{d_k}$ rather than a matrix. The relevance of each column vector $\mathbf{k}_i \in \mathbf{K}$ with respect to $\mathbf{q}$ is evaluated by a compatibility function f, whose output is a vector $\mathbf{e} \in \mathbb{R}^{n_k}$ of energy scores:

$$\mathbf{e} = f(\mathbf{q}, \mathbf{K}) \ . \tag{6.8}$$

Energy scores are then transformed to a vector $\mathbf{a} \in \mathbb{R}^{n_k}$ of attention weights, applying a distribution function. In our case, such a function will be the *softmax*, leading

to the following definition:

$$\mathbf{a} = \text{softmax}(\mathbf{e}) = \text{softmax}([e_1, ..., e_{n_k}]) \ , \tag{6.9}$$

$$\text{softmax}(e_i) = \frac{\exp(e_i)}{\sum_{j=1}^{n_k} \exp(e_j)} \ . \tag{6.10}$$

Attention weights are further combined with matrix $\mathbf{V}$, to obtain a weighted representation of $\mathbf{V}$ itself. Note that for each element $\mathbf{k_i} \in \mathbf{K}$ there is a corresponding element $\mathbf{v_i} \in \mathbf{V}$ (i.e., $n_k = n_v$). The final outcome of the attention submodel is a context vector $\mathbf{c} \in \mathbb{R}^{d_k}$, that is typically employed by other components of the model where the attention architecture is integrated; it is defined as follows:

$$\mathbf{c} = \sum_{i=1}^{n_v} a_i \mathbf{v_i} \ . \tag{6.11}$$

Bringing together the sequence-to-sequence model and the attention mechanism, in order to compute the attention score vector for each output element t, $1 \leq t \leq m$, $\mathbf{q}$ must correspond to the previous hidden state of the decoder $\mathbf{s}_{t-1}$, $\mathbf{K}$ is the column-based matrix obtained combining all the encoder hidden states $\mathbf{h}_i, 1 \leq i \leq n$, and $\mathbf{V} = \mathbf{K}$. Thus, considering our indoor positioning scenario, in which three hierarchical levels (building, floor, and room) are present, $t = 3$. As a result, for each ranking-based fingerprint three attention score vectors are obtained, which can be seen as a matrix $\mathbf{A} \in \mathbb{R}^{3 \times n_k}$, where n_k will be set equal to the median value of the number of non-null APs seen among all (ranked-)fingerprints (those ranked fingerprints with $z > n_k$ are simply truncated).

It is worth noticing that many different attention formulations have been proposed in the literature. Among them, we cannot rely on today's widely used self-attention mechanism (where $\mathbf{Q} = \mathbf{K} = \mathbf{V}$), since for our interpretability purposes we are interested in (cross) attending input and output elements. Nevertheless, many different compatibility functions can be used within the general cross-attention framework (i.e., the one employed by our model). We take into account the following:

$$dot = f(\mathbf{q}, \mathbf{K}) = \frac{\mathbf{q}^T \mathbf{K}}{\sqrt{d_k}} \tag{6.12}$$

$$general = f(\mathbf{q}, \mathbf{K}) = \mathbf{q}^T \mathbf{W} \mathbf{K} \tag{6.13}$$

$$cat = f(\mathbf{q}, \mathbf{K}) = \mathbf{w}^T \tanh(\mathbf{W}[\mathbf{K} \| \mathbf{q}]) \tag{6.14}$$

$$add = f(\mathbf{q}, \mathbf{K}) = \mathbf{w}^T (\mathbf{W}_1 \mathbf{K} + \mathbf{W}_2 \mathbf{q}) \tag{6.15}$$

$$deep = f(\mathbf{q}, \mathbf{K}) = \mathbf{w}^T \mathbf{E}^{(L-1)}, \tag{6.16}$$

$$\text{with} \begin{cases} \mathbf{E}^{(l)} = \text{ReLU}(\mathbf{W}_l \mathbf{E}^{(l-1)}) & \text{if } 1 < l < L \\ \mathbf{E}^{(1)} = \text{ReLU}(\mathbf{W}_1 \mathbf{K} + \mathbf{W}_0 \mathbf{q}) & \text{if } l = 1 \end{cases}$$

where $\mathbf{w}$ is a learnable vector and $\mathbf{W}_{(\mathbf{i})}$ are learnable matrices. The functions follow two main approaches. Attentions *dot* [264] and *general* [161] are based on matching and comparing $\mathbf{K}$ and $\mathbf{q}$, with the main difference that *general*, thanks to the weight matrix $\mathbf{W}$, makes it possible to deal with queries and keys employing different representations. On the other hand, *cat* [264], *add* [11], and *deep* [200] follow a different strategy, and combine the keys and the query in a joint representation that is further weighted by an importance vector $\mathbf{w}$, which conveys the notion of relevance. This makes them a good choice when keys and queries are encoded in significantly different ways. They differ from each other in how they combine $\mathbf{K}$ and $\mathbf{q}$: *cat* concatenates them; *add* computes the contributions separately and then sums them; and, *deep* extends *add* using multiple layers allowing to build richer representations.

In specific works in the deep learning literature [92], attention has been shown to provide for each input sequence element (i.e., $x_i \in \mathbf{x}_{1:n}$ in Eq. (6.1), that will be the keys) a form of relevance to the prediction of each specific output sequence component (i.e., $y_j \in \mathbf{y}_{1:m}$ in Eq. (6.1), each of which will be a query). Nevertheless, there is still no unanimous consensus about its capability to actually fulfil such a task in a general setting, as witnessed for example by the many open debates within the NLP community [116, 236, 282].

Therefore, it is worth studying whether attention, when applied to the previously described ranking-based fingerprint representation, allows one to assess the relevance of the APs in the prediction of a given position (more precisely, of each hierarchical component describing a position), adhering to our concept of interpretability described in Section 6.1.2. Moreover, given that multiple types of attention (compatibility functions) exist, an investigation about their behaviours also becomes necessary. In the following section, thanks to a set of carefully designed qualitative and quantitative experiments, we will shed light on such matters.

6.3 Experimental evaluation

In this section, we first give an account of the considered indoor positioning dataset and the hyperparameter optimization process. Next, we provide a thorough description of the designed experiments and their results, aimed at evaluating the behaviour of the attention mechanism when applied to the indoor positioning context.

The overall experimental workflow is depicted in Figure 6.4. We begin with an analysis of attention, starting with a quantitative assessment carried out by means of clustering, that will allow us to formally determine whether attention is capable of eliciting general spatial relationships in the data. Next, a qualitative analysis is performed, which will reveal some interesting attention patterns and will relate them spatially with the specific real-world premises considered. Finally, results are established regarding the prediction of the position of a given instance, leveraging the

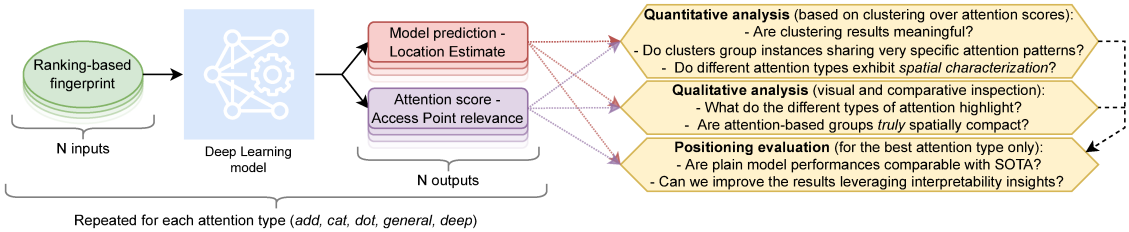


Figure 6.4: Graphical high-level summary of the experimental evaluation pipeline.

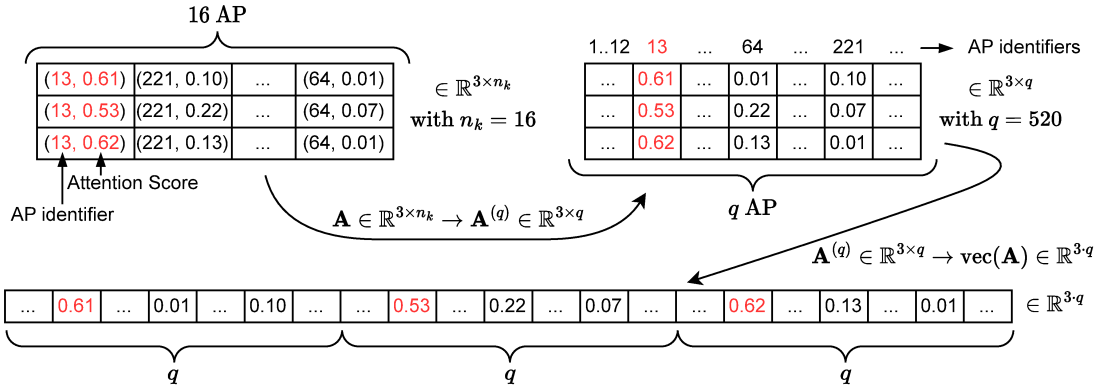


Figure 6.5: Graphical account of the transformation of an attention matrix $\mathbf{A} \in \mathbb{R}^{3\times n_k}$ to its vectorized representation $\mathrm{vec}(\mathbf{A}) \in \mathbb{R}^{3\cdot q}$.

probabilities and the attention values provided by the model when taken separately as well as in conjunction.

In the remainder of the section, we will consider the following representation of the attention matrix $\mathbf{A} \in \mathbb{R}^{3\times n_k}$ described in Section 6.2. First, we transform each row into a vector of size q (i.e., the number of distinct access points), in such a way that the attention value associated with the AP j is mapped into the jth position of the vector. In the resulting matrix (of size $3 \times q$) all elements not associated with an attention value are set to zero. Finally, we consider a vectorized (i.e., flattened) representation of the latter matrix, defined as $\mathrm{vec}(\mathbf{A}) \in \mathbb{R}^{3\cdot q}$. A summary of the process is reported in Figure 6.5. This allows us to compare the attention values associated with different instances irrespective of their AP ranking.

6.3.1 Experimental setting

Dataset

Concerning the overall performance evaluation, for the sake of fair and comprehensive comparison with previous literature, we relied again on *UJIIndoorLoc*, as processed in Section 5.3.1. Table 6.1 summarizes the main features of the considered UJIIndoorLoc dataset splits. Observe that information regarding reference positions

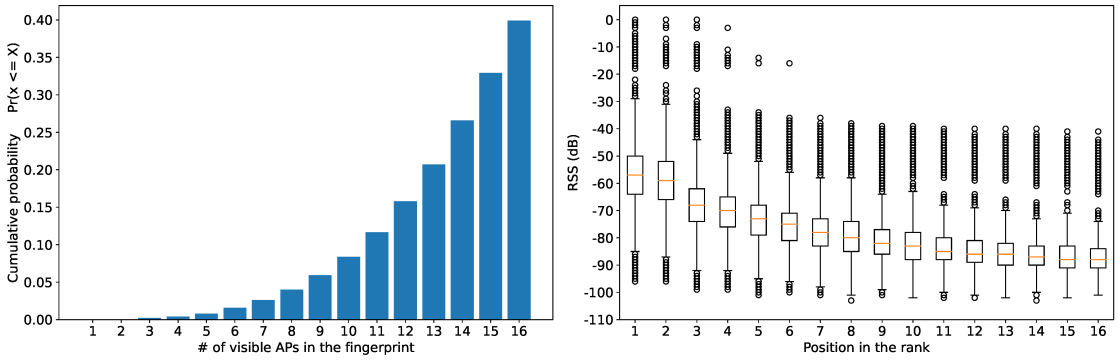


Figure 6.6: Cumulative probability of the number of visible APs in a fingerprint (left) and RSS distribution over the rank positions (right).

Table 6.1: Characteristics of the considered UJIIndoorLoc dataset splits.

Split	# B	# F	FP x RP	# FP	# AP	# $\widetilde{AP}$	# RP	FPρ	Valid APs
Train	3	5	1 to 84	17825	460	-	904	0.65 $\pm$ 0.38	17.7 $\pm$ 7.1
Valid.	2	4	2 to 37	2036	164	5	164	0.35 $\pm$ 0.22	21.7 $\pm$ 7.3
Test	3	5	-	1111	367	55	-	-	16.5 $\pm$ 6.9

B = Number of buildings; #F = Number of floors; FP x RP = fingerprints sampled per reference position; # FP = number of fingerprints; # AP = number of APs seen at least one time; # $\widetilde{AP}$ = number of seen APs not present in the training split; # RP = number of reference positions; FP_ρ = average FP density within a 5 meter radius from each RP; Valid APs = average number of detected APs per fingerprint.

is not available for the test split.

Moreover, recall that the median value of the number of non-null APs seen among all fingerprints is 16. Thus, following the rule of thumb described in Chapter 5, this will be the length of the ranked fingerprints considered. The soundness of such a choice is also confirmed by Figure 6.6, where it can be seen that the number of visible access points and their RSS rapidly decreases with the rank length.

Hyperparameters optimization

Model hyperparameters have been tuned relying on iterative grid search, based on the identified training and validation splits. We did not make use of embeddings for our input data, mapping the APs following a one-hot-encoding fashion. This determined the dimension of the encoder input (here, 520). We also observed that keeping the encoder hidden representation size similar to the dimension of the one-hot encoded input was rather effective. The other hyperparameters are as follows: we chose Stochastic Gradient Descent with warm Restarts (SGDR) [157] as the optimizer, with a learning rate of 0.05; batch size was set to 32; variational dropout was set to 0.5, while attention dropout was 0.2 (when applicable).

6.3.2 Quantitative analysis

In this section, we present a procedure to evaluate which, among the considered attention compatibility functions, is the most capable of capturing spatial information. The overall idea is that instances sharing a similar attention pattern should also have in common some locality properties, e.g., they should be close to each other in the real-world space where they were collected, irrespective of their specific positions.

Experiments

To such an extent, we relied on a clustering analysis performed over training set attention vectors. Specifically, we focused on hierarchical clustering (Scikit-learn's AgglomerativeClustering [202] method), as it allowed us to investigate compositional relationships between clusters through the associated dendrogram representation.

Hierarchical clustering requires the specification of two fundamental parameters, i.e., the metric used to compute the linkage, and the linkage criterion itself. As for the metric, which is calculated among vectors vec($\mathbf{A}$), we relied on *cosine* distance, as it is typically done in embedding-related tasks [148]. Turning to the second parameter, we evaluated the following linkage criteria: *average*, *complete*, *single*, *ward*. Based on a qualitative analysis of the generated dendrograms, we finally chose to rely on methodology *average*, since it showed the advantage of producing well-balanced splits among the clusters.

For each attention compatibility function, we then proceeded as follows. First, hierarchical clustering was run over the vectors vec($\mathbf{A}$) associated with training set instances, obtaining five dendrograms. Then, based on dendrogram analysis, and evaluating silhouette scores at different dendrogram cut points, we determined the most appropriate number of clusters for each function.

At this point, a question arises about whether the identified clusters are genuinely different from each other, i.e., they group together instances sharing a very specific attention pattern. To formally determine that, we devised a statistical procedure based on the Kolmogorov–Smirnov (KS) test [168]. KS is a non-parametric test of the equality of continuous distributions, that can be used to quantify a distance between the empirical distribution functions of two samples.

The overall approach is as follows. Given two clusters, we construct the union set of their AP identifiers according to $\mathbf{A}$[3]. Then, for each AP, within each cluster, we compute its empirical univariate distribution of attention values. Observe that an AP may be detected in just one cluster; in that case, we define a zero-valued dummy distribution for the other one. We now apply the KS test to the distribution pairs of each AP, to determine if they differ. The final cluster similarity is obtained by

[3]This allows us to consider just the APs that are detected by instances belonging to the two clusters, leading to computational savings in the following steps.

combining the single AP results by means of the Benjamini–Hochberg procedure for multiple comparisons [23].

Algorithm 1: Cluster similarity assessment

Input : $dist\_list = [(X_1, Y_1), \dots (X_n, Y_n)]$ list of distributions of pairs related to $cluster_1$ and $cluster_2$
Input : bs_size number of bootstrap samples
Output: `True` if clusters are different, `False` otherwise
Output: The ratio of statistically different distributions

1 $p\_values \leftarrow []$
2 **for** $i \in \{1, \dots, n\}$ **do**
3 $\quad X, Y \leftarrow dist\_list[i]$
4 $\quad KS\_res \leftarrow \text{KolmogorovSmirnovTest}(X, Y)$
5 $\quad combined\_distribution \leftarrow X \| Y$
6 $\quad KS\_boot \leftarrow []$
7 $\quad$ **for** $j \in \{1, \dots, bs\_size\}$ **do**
8 $\quad\quad S_1 \leftarrow \text{DrawSample}(combined\_distribution)$
9 $\quad\quad S_2 \leftarrow \text{DrawSample}(combined\_distribution)$
10 $\quad\quad S_1 \leftarrow S_1 + \mathcal{N}(0, \text{Cov}(S_1, S_2))$
11 $\quad\quad S_2 \leftarrow S_2 + \mathcal{N}(0, \text{Cov}(S_1, S_2))$
12 $\quad\quad KS\_boot[j] \leftarrow \text{KolmogorovSmirnovTest}(S_1, S_2)$
13 $\quad$ **end**
14 $\quad KS\_boot[bsSize + 1] \leftarrow KS\_res$
15 $\quad p\_values[i] \leftarrow \text{Mean}(KS\_boot \geq KS\_res)$
16 **end**
17 $reject\_null\_hypothesis \leftarrow \text{BenjaminiHochberg}(p\_values, 0.05)$
18 $diff\_clusts \leftarrow \text{CountTrue}(reject\_null\_hypothesis) == n$
19 $fract\_diff \leftarrow \text{CountTrue}(reject\_null\_hypothesis)/n$
20 **return** $diff\_clusts, fract\_diff$

The procedure is depicted in Algorithm 1. Given the input distributions pairs, the procedure evaluates each of them independently (Lines 1–16). For a pair X, Y, the KS test is evaluated, obtaining the corresponding KS statistic, i.e., the distance between the empirical distribution functions of two samples (Line 4). Then, a common continuous distribution, which will be used for null hypothesis testing, is generated by concatenating X and Y (Line 5). Observe that, for our purposes, the null hypothesis is that the batches of data are independent simple random samples taken from the common continuous distribution. At this point, as it is typically done when a measure of difference has to be estimated, we apply bootstrapping by resampling from the common distribution (Lines 7–13). For each bootstrap iteration, two random samples are independently extracted from the combined distribution, and random noise is applied to them (Lines 8–11). The KS statistic is then evalu-

ated (Line 12). Then, by comparing bootstrapped and reference KS statistics, the p-value encoding the extent to which the KS distance between X and Y is statistically relevant is determined (Lines 14–15). Note that the p-value corresponds to the average of the boolean vector, interpreted as an integer one, generated by comparing the single KS bootstrap results with the reference one. Intuitively, if many KS bootstrap values are smaller than the reference one, then the two distributions are more likely to be different. At this point (Line 17), we need to combine the p-values related to the single APs in order to determine if the two input clusters are different. To such an extent, we rely on Benjamini–Hochberg procedure for multiple comparisons, computing the adjusted p-values based on a false discovery rate (FDR) level of 0.05, and returning *true* for those hypotheses that can be rejected for the given FDR level. If all null hypotheses can be rejected (Line 18), then the two clusters are deemed to be different. In addition (Line 19), we also compute the fraction of hypotheses that could be rejected by the method, in order to assess, when the clusters are considered to be alike, the ratio of non-similar APs.

Finally, it is also worth evaluating the clustering result from a spatial perspective. The idea is that clusters considered to be similar by the KS test should also be relatively close. Instead, statistically different clusters are intrinsically well-behaved, since they may possibly suggest different APs relevance (and, thus, strategies) followed by the neural network to derive the predictions even within small areas. To determine the distance between two clusters in the spatial domain, we relied on the Hausdorff metric. Informally, two clusters are deemed to be close by such a metric if every point of either cluster is close to some point of the other cluster. The Hausdorff distance is the longest distance one can be forced to travel by an adversary who chooses a point in one of the two clusters, from where he then must travel to the other cluster. In other words, it is the greatest of all the distances from a point in one cluster to the closest point in the other cluster. Formally, we define the Hausdorff distance between two spatial clusters $\mathcal{G}_1, \mathcal{G}_2 \subseteq \mathbb{R}^3$ as:

$$H(\mathcal{G}_1, \mathcal{G}_2) = \max_{\mathbf{g}_1 \in \mathcal{G}_1} \min_{\mathbf{g}_2 \in \mathcal{G}_2} \|\mathbf{g}_1 - \mathbf{g}_2\|_2 \ . \tag{6.17}$$

Nevertheless, considering the maximum within the Hausdorff distance computation can lead to misleading results in the presence of spatial outliers that are likely to be generated during the clustering process. For this reason, as witnessed in literature (see, e.g., [114]), we calculate the 95th percentile of the minimum distances instead.

For each attention compatibility function, we evaluated the Hausdorff distance for all pairs of clusters considered to be similar by the previous KS test, as well as for all pairs of different ones. Intuitively, a well-performing compatibility function should lead to Hausdorff distances among similar clusters that are small, and overall smaller than those among dissimilar clusters (although, as already mentioned, it can be the case that some dissimilar clusters are actually close to each other).

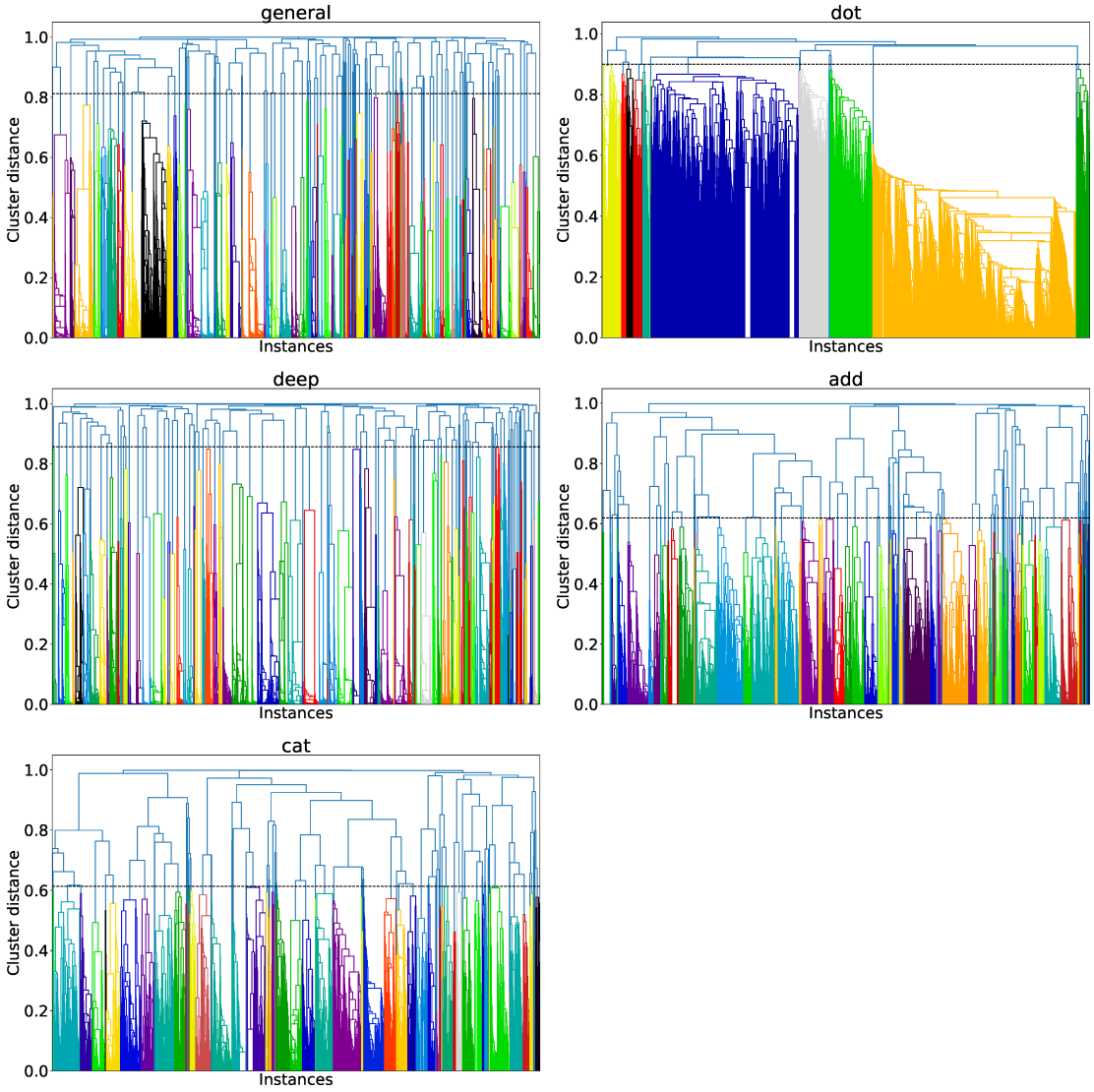


Figure 6.7: Dendrograms obtained by clustering the attention vectors relying on different compatibility functions. The height of the tree branches indicates the distance between the clusters being hierarchically merged. The horizontal axis shows the individual data points being clustered. The black dashed line depicts the chosen cutting point, that leads to the specific sets of clusters identified by the colors. The result provided by of *dot* is unsatisfactory due to poor partitioning; *general* and *deep* produce too many groups; *add* and *cat* generate more balanced partitions where cluster compositionality can be clearly noticed.

Results

Figure 6.7 shows the dendrograms obtained from the hierarchical clustering tasks for each attention compatibility function. As can be seen, despite relying on the same clustering approach, the results are quite heterogeneous[4]. Specifically, *dot* attention generates a very coarse clustering at the chosen cutting point; performing the cut at a lower level would result in a very large number of small clusters, while still not being able to break the largest (orange) group. Such a discrepancy suggests a bad behaviour of *dot* with respect to the clustering task. As for *general* and *deep* attentions, they share similar behaviour. Most of the clusters are aggregated at a very high point on the cluster distance axis, suggesting the presence of (too) many groups, typically quite different between them, and with no clear hierarchical structure. Finally, *add* and *cat* attentions show a more variegate situation as for cluster distances. Here, cluster compositionality can be clearly noticed and choosing different cutting points would always lead to balanced partitions, although of different sizes. Thus, they exhibit the best behaviour when it comes to capturing hierarchical relationships among clusters.

We now turn to study whether the identified clusters are genuinely different from each other, i.e., they group together instances sharing a very specific attention pattern. Figure 6.8 and Figure 6.9 report the result of the KS procedure applied to the previously discussed clusterings. Figure 6.8 shows, for each pair of clusters, whether they are judged to be similar (dark colour) or not (bright colour) by the test. Intuitively, the ideal case is represented by an entirely bright picture, except for the diagonal (that compares each cluster with itself). Thus, the two best attention functions are *add* and *cat*, with a slight preference towards the latter[5]. To determine the extent of similarity between clusters judged to be similar by the procedure, let us consider Figure 6.9. Here, a darker colour corresponds to pairs of clusters whose instances, i.e. attention patterns, highlight a larger number of access points with the same intensity. Intuitively, the optimal case is thus represented by a scenario in which the clusters judged to be similar are still characterized by a low fraction of similar access points. This is precisely what happens with *add* and *cat*, confirming their good behaviour.

Finally, to assess the spatial characterization of the clusters, let us consider Figure 6.10, which reports the distribution of Hausdorff distances among similar and dissimilar clusters (according to KS). Here, the ideal scenario is characterized by distance values for similar clusters typically lower than those for dissimilar ones.

[4]Note that the depicted cutting points have not been cherry-picked, but automatically selected through a silhouette-based approach: given a dendrogram, a set of possible thresholds was considered, each leading to a different clustering. For each clustering, the *silhouette score* [220] was calculated, obtaining a list of values onto which an elbow criterion finally allowed us to determine the best threshold.

[5]Even though *dot* appears to share a similar behaviour, it has poor results because of its severely unbalanced clustering.

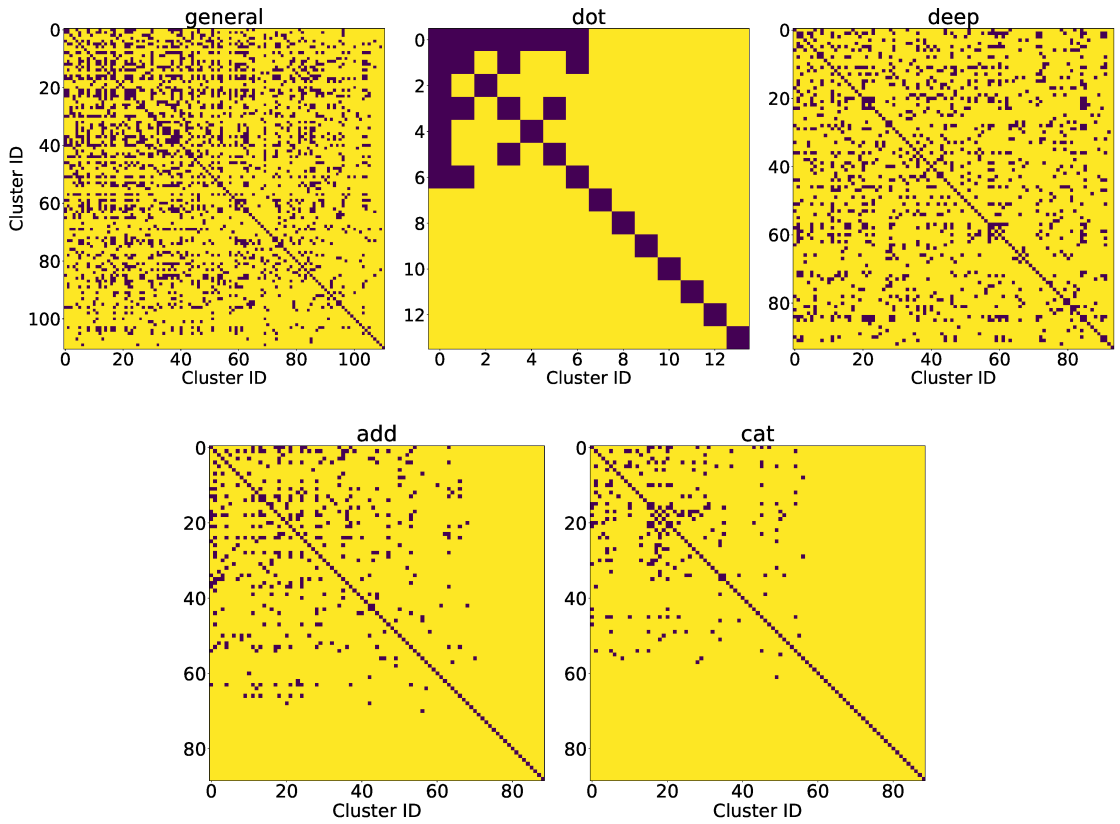


Figure 6.8: KS-based cluster similarity test result. Each cell denotes the comparison between a pair of clusters, which can be judged to be similar (dark colour) or not (bright colour). *add* and *cat* show the best behaviour, since they have a high number of clusters but only few of them are pairwise similar.

Of course, as already mentioned, it can be the case that dissimilar clusters still have a low Hausdorff distance, since two spatially close observations may rightfully consider different access points for the location prediction. Observe that, with the exception of *dot* attention, the median distance among similar clusters is always lower than that between dissimilar ones. Overall, *add* emerges as the best attention compatibility function, given the disjoint interquartile ranges and the low dispersion of distances calculated among similar clusters, leading us to choose it for the experiments that take into account also spatial information.

6.3.3 Qualitative analysis

Other than from a quantitative point of view, it is also worth investigating our findings from a visual perspective. Here, we also relate the attention patterns of the instances with the specific positions in the premises where they were collected.

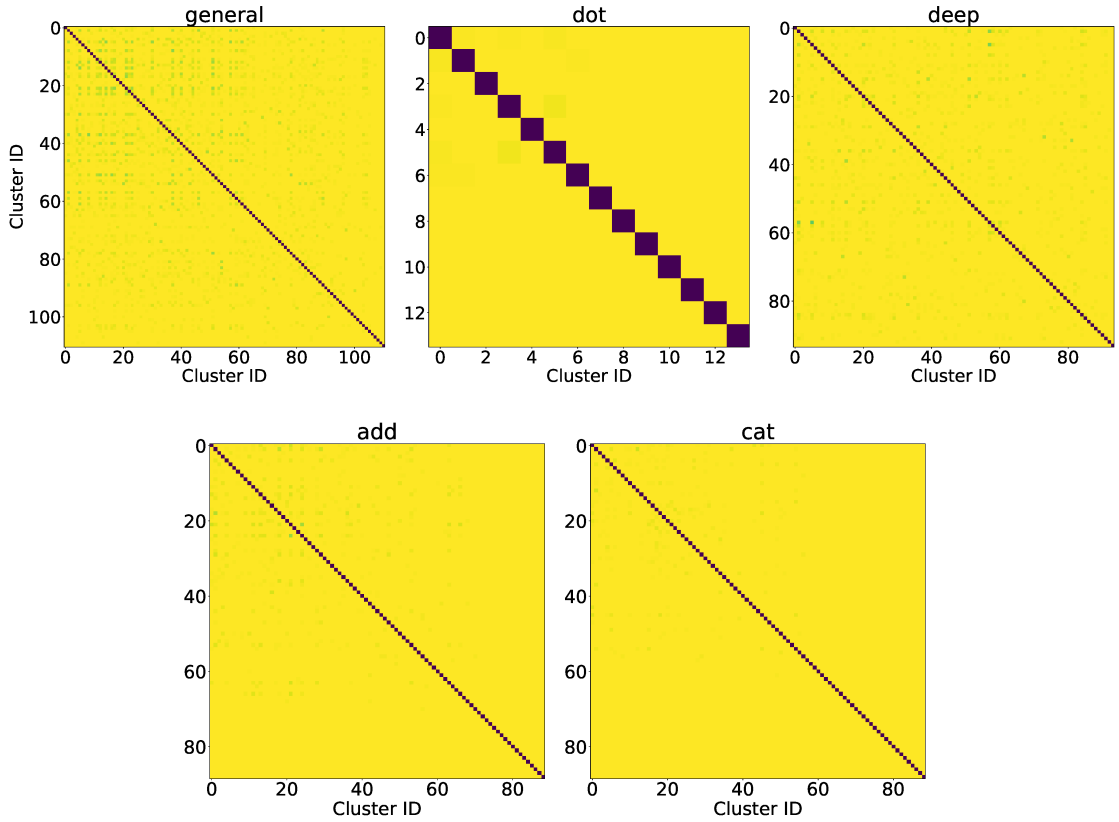


Figure 6.9: KS-based AP similarity ratio test result. Each cell denotes the comparison between a pair of clusters, which can share a large number of equally distributed APs (dark colour) or not (bright colour). *add* and *cat* show the best behaviour, since, in them, the clusters that were considered to be similar have a low number of equally distributed APs, a fact that still justifies their existence.

Experiments

To such an extent, we first graphically compare the attention patterns exhibited by each compatibility function on a set of meaningful samples. Specifically, we want to assess, at different granularity levels (building, floor, and room), the relevance of the access points appearing in a fingerprint. This may allow us to investigate some of the phenomena mentioned in the previous sections, such as: does the deep learning model always consider the strongest access points to derive its output? Are the same access points exploited to perform the building, floor, and room predictions? Do the compatibility functions exhibit different attention patterns?

We finally evaluate the spatial arrangement of the attention-based clusters. Intuitively, if attention values were to characterize specific areas of the indoor scenario, clusters should determine compact and well-separated regions in the considered premises.

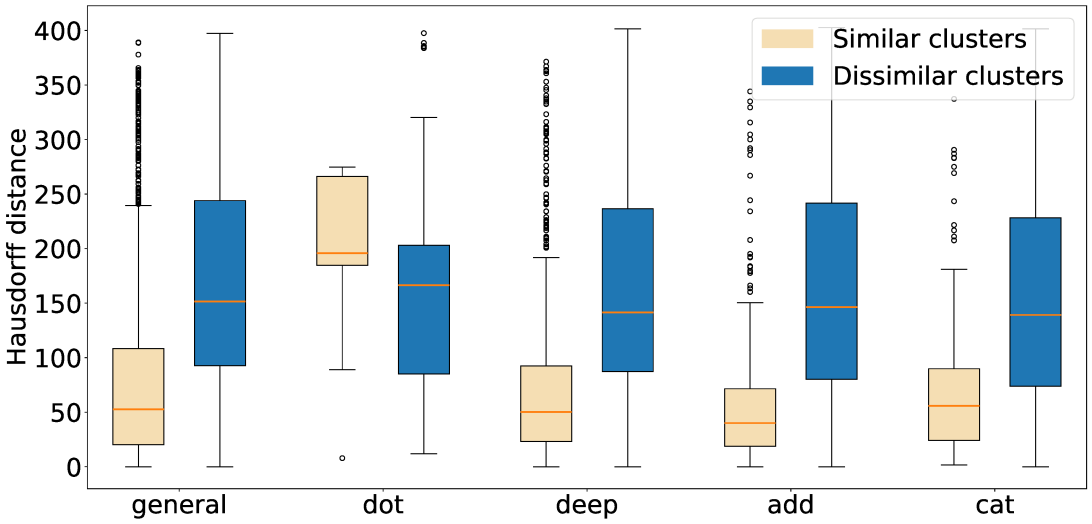


Figure 6.10: Distributions of Hausdorff distances among similar and dissimilar clusters. Each box extends from the first to the third quartile values of the data, with a line at the median. Whiskers extend to the smallest and largest observations which are not outliers (considering 1.5 times the interquartile range). The optimal case is that of disjoint boxes, with distant medians, and distance values for similar clusters lower than those for the dissimilar ones. *dot* exhibits a wrong behaviour, while *add* emerges as the best.

Results

Figure 6.11 shows the matrices $\mathbf{A}$ generated by different attentions (arranged into columns) for some representative instances (one for each row). We can immediately notice that the compatibility functions exhibit radically different behaviours. Interestingly, *general* and *deep* seem to focus just on the strongest access point(s), as opposed to *dot* that considers the last columns of the matrix. The latter is an incorrect behaviour since it also applies to the first case, where the last three columns refer to dummy APs (labelled with zero), that are used to pad the ranked fingerprint when less than n_k APs are detected. While these first attention functions appear to focus on specific columns rather than on the actual APs, this is not the case with *add* and *cat*, that exhibit rather heterogeneous patterns: typically, several access points are considered to be relevant; attention values may be distributed in different manners among them; and, the strongest access point is not always the most important. Finally, notice how, according to all attentions, the model seems to always exploit the same APs to derive the building, floor, and room predictions.

As a final qualitative assessment, let us now take into account also the spatial dimension. To do that, we focus on *add*, which emerged as the best attention compatibility function according to our quantitative analysis. Figure 6.12 shows

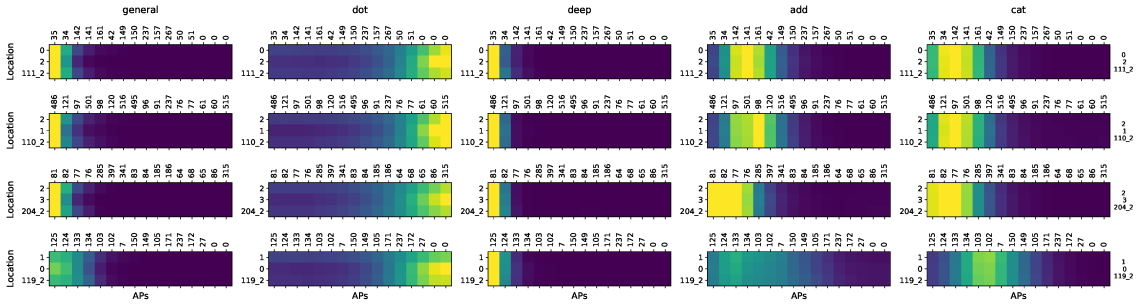


Figure 6.11: Attention matrices $\mathbf{A}$ of four instances (one for each row) generated by different compatibility functions. A brighter colour denotes a higher relevance. The predicted location is reported on the left side of each matrix, while the ground truth is on the extreme right (for each matrix, the first row label is for the building, the second for the floor, and the third for the room hierarchical level). Access points identifiers are shown on the top of the matrices (0 = dummy AP). It is possible to observe that different types of attention focus on different parts of the ranked fingerprints.

the assignments of training set locations to clusters. As can be noticed, the latter display evident spatial relationships, with instances belonging to the same cluster being placed relatively close to one another in the spatial domain. Observe that some dispersion or overlap phenomena can occur. This is the case, for instance, of cluster number 6 that, on floor 2, has its instances spread all over the central part of the map, interleaved with observations belonging to several other clusters. While dispersion may be attributed to an under-splitting of the clusters along the dendrogram, the spatial overlap between clusters may be due to either the presence of several, genuinely different attention patterns in the same areas, or to an over-splitting of the clusters. As a matter of fact, these noise phenomena are intrinsic to cluster-based analysis, and do not hinder our conclusions.

6.3.4 Positioning performance evaluation

Experiments

As witnessed by the recent literature [174], there is still no unanimous consensus on how to evaluate the performance of an indoor positioning system. This stems from the fact that, in a multi-building and multi-floor scenario, the error of being located into a wrong building or floor is far more serious than that of being assigned to a wrong position within the correct floor. Finding a composite evaluation metric is not trivial. In this regard, observe how the 3D distance between the estimated and real position coordinates is not satisfactory. As a consequence, in this work, we rely on two different metrics, which should be jointly considered. First, the so-called *success rate* measures the fraction of instances for which both building and floor

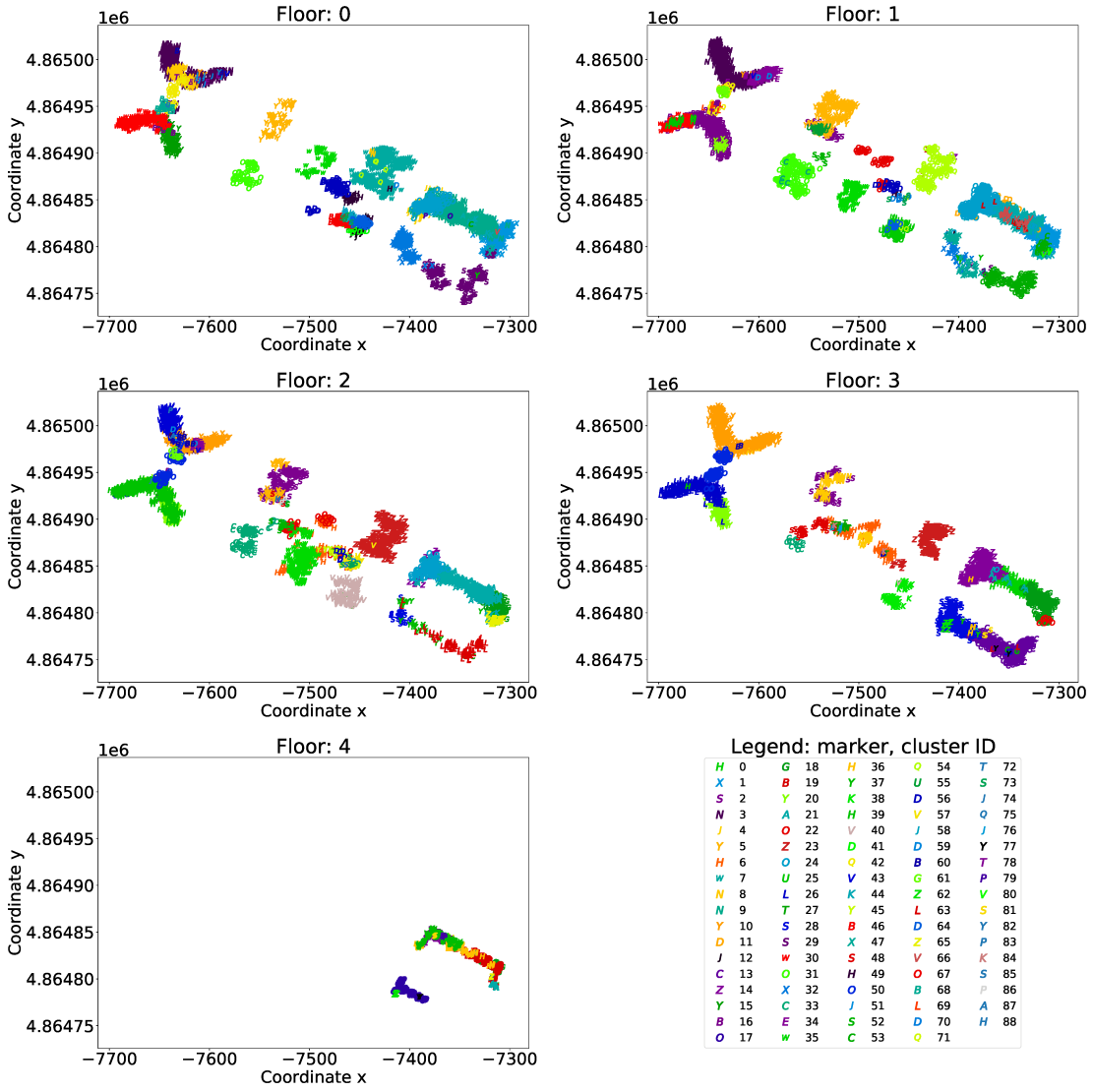


Figure 6.12: Cluster assignments of instances on different floors for *add* attention compatibility function. It is possible to observe that clusters exhibit some spatial locality properties.

have been correctly predicted. Second, among such correctly predicted instances, the 2D positioning error is determined by looking at the Euclidean distance $E(\mathbf{p}, \hat{\mathbf{p}})$ between the predicted $\hat{\mathbf{p}} = (\hat{x}, \hat{y})$ and the ground truth $p = (x, y)$ coordinates:

$$E(\mathbf{p}, \hat{\mathbf{p}}) = \|\mathbf{p} - \hat{\mathbf{p}}\|_2 = \sqrt{(x - \hat{x})^2 + (y - \hat{y})^2} \,. \tag{6.18}$$

Observe how the model developed in Section 6.2 allows us do determine a location triplet $(\widehat{building}, \widehat{floor}, \widehat{room})$ leading to the highest likelihood for a given instance. Then, the probability distribution $P(room|floor = \widehat{floor}, building = \widehat{building})$, can

be also retrieved. In addition, thanks to the attention module, each hierarchical component has an associated attention weight vector which, intuitively, should represent the most relevant access points used by the model to derive the single level prediction.

While the success rate can be estimated comparing $(\widehat{building}, \widehat{floor})$ with the ground truth, for the 2D position estimation we proceed relying on a k-Nearest-Neighbour-like approach. Specifically, considering different similarity approaches to determine the neighbours, we obtain two strategies.

Probability-based. The coordinates for an entry are determined by computing the weighted centroid over the coordinates of the K most likely rooms identified according to the probability distribution, where the weights ($\boldsymbol{\pi} \in \mathbb{R}^K$) correspond to the probabilities. The parameter K can be either considered as fixed, as in Chapter 5, or tuned according to the best performance exhibited on a training+validation split of the training data.

Attention-based. As a preprocessing step, we determine the average attention vector $\text{vec}(\mathbf{A}_\mu) \in \mathbb{R}^{3 \cdot q}$ for every training set location, element-wise aggregating the attention vectors obtained running the model over the training set instances collected at the location. Such a representation is then compared to the attention vector $\text{vec}(\mathbf{A}_{new}) \in \mathbb{R}^{3 \cdot q}$ generated by the model for a *new* instance, employing the *cosine similarity*. The obtained similarity attention scores ($\boldsymbol{\alpha} \in \mathbb{R}^K$) are then used within the k-NN framework following the same procedure as done for the probability-based approach. Again, the parameter K can be fixed or tuned according to an evaluation performed on a training+validation split of the training data. The net result is that training set locations with an attention pattern similar to the given instance are assigned a higher weight during the centroid computation phase.

For comparison purposes, in the next section, we evaluate the performances provided by the two approaches against some recent relevant papers in the area that apply their solution to UJIIndoorLoc. Bear in mind that several works did not use the test set as it is provided, but they restrict to some specific floor or building. This is a conceptually wrong approach that might lead to biased results, since achieving a good positioning performance in certain sub-regions of the data dataset is far harder than in other parts.

Results

We now turn to evaluate the positioning performance provided by the model described in Section 6.2.

Table 6.2 shows the test set results provided by the two k-NN approaches that employ probability- and attention-based similarities. Also, it reports the respective K (the number of neighbours) determined by an optimization procedure based on

Table 6.2: Comparison of the positioning estimation results

Approach	Success rate [%]	Positioning error [m] Mean	Median	75th percentile	95th percentile	RMSE
[256]	89.9	7.9	-	-	-	-
[261]	94.8	6.86	-	-	-	-
[245]	96.0	11.8	-	-	-	-
[223] (Chp. 5)	94.8	6.57	4.58	-	-	9.52
[137]	92.6	9.07	6.32	-	-	-
[276]	93.9	6.95	-	-	-	-
prob (K = 9)	95.23	6.56	4.78	8.97	18.7	9.53
att (K = 1)		9.82	7.07	14.3	28.7	13.9
oracle	95.23	**5.26**	**3.43**	**7.21**	**15.7**	**8.00**
prob $\odot$ att		**6.40**	**4.58**	**8.82**	**18.2**	**9.23**

Note: The results of our best solution and those of the oracle based approach (see Section 6.3.4) are in bold.

the training and validation splits[6]. As can be seen, the achieved success rate is on par with previous works from the literature. Note that, being such a value related to the (categorical) output of the model, it is independent of the specific approach we considered for position estimation. Focusing on the positioning error in meters, probability emerges as the best variant, largely surpassing the attention-based one as well as being on par with one of the best solutions available from the literature [223] (although performing better than the latter from the success rate perspective).

Combining probability and attention

Position estimation can also be performed by considering weights obtained from a suitable combination of probabilities and attention values.

Despite the much worse performance exhibited by the attention-based approach with respect to the probability-based one, it may still be interesting to compare their behaviours. Figure 6.13 shows the distribution of the differences between the errors provided by the two solutions. As can be seen, the histogram is roughly centred on zero, meaning that on a large portion of test set instances the two variants behave similarly, while the superiority of the probability-based one is justified by the larger right tail. Nevertheless, the left tail informs us about the presence of a significant amount of cases in which attention provides the best performance. This suggests

[6]We calculated the mean positioning error on the validation set for every possible value of K and selected the parameter that produced the lowest error as the final choice. This process was repeated independently for both K-NN models, one using probability-based similarities and the other using attention-based similarities.

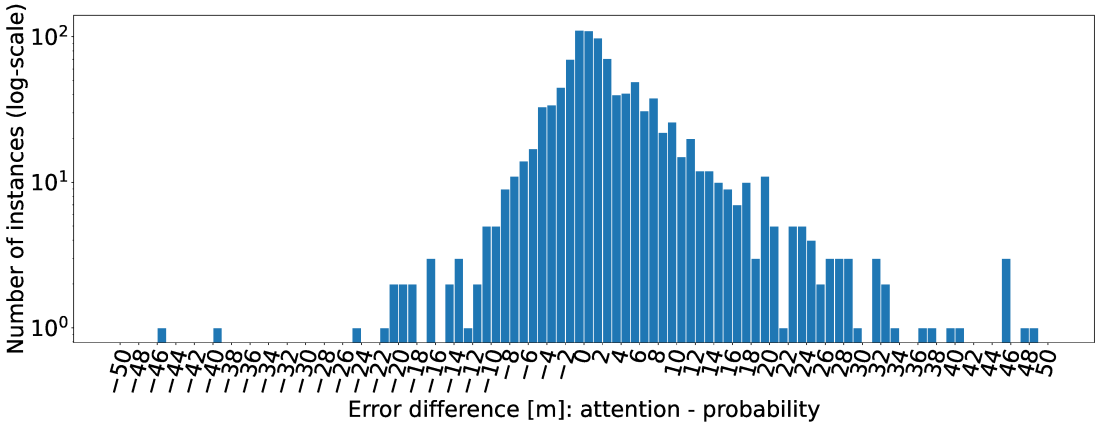


Figure 6.13: Distribution of the differences between the errors of attention-based and probability-based approaches. The larger right tail shows the superiority of the probability-based strategy, although it is possible to observe that for a significant amount of cases the attention-based approach behaves better.

that the two approaches convey different, complementary information, which could be used jointly to devise a better positioning method.

Let us assume now to have at our disposal a strategy capable of correctly suggesting, for each instance, the best variant to rely on for the position estimation, between attention and probability. Such an *oracle-based* approach would achieve a test set average positioning error of 5.26 meters (see Table 6.2). The difference between the oracle and the single variants becomes even more evident in Figure 6.14, which compares the empirical cumulative distribution functions (ECDFs) of the different approaches, confirming the extent of the potentially achievable improvement. Note that the oracle based selection of attention and probability does not necessarily represent the optimal solution: it can be the case that even better performance could be achieved by suitably combining the two basic methodologies, for instance, by means of a mathematical formula. Providing an optimal positioning framework is out of the scope of this work, nevertheless, in the remaining part of the section we show that, even with a straightforward approach, it is possible to exploit the aforementioned insights and improve over the baselines.

Our methodology to combine probabilities with similarity attention scores is rather simple. Given the most likely location $(\widehat{building}, \widehat{floor}, \widehat{room})$ for an instance, we extract the set of all reference positions belonging to the same building and floor, i.e., $\hat{\mathcal{L}} = \{\mathbf{l}_i \,|\, building_i = \widehat{building} \wedge floor_i = \widehat{floor}\}$. Following the probability- and attention-based approaches, we also obtain the vector $\boldsymbol{\pi} \in \mathbb{R}^{|\hat{\mathcal{L}}|}$ of probabilities and the vector $\boldsymbol{\alpha} \in \mathbb{R}^{|\hat{\mathcal{L}}|}$ of cosine similarity attention scores associated to the RPs. We

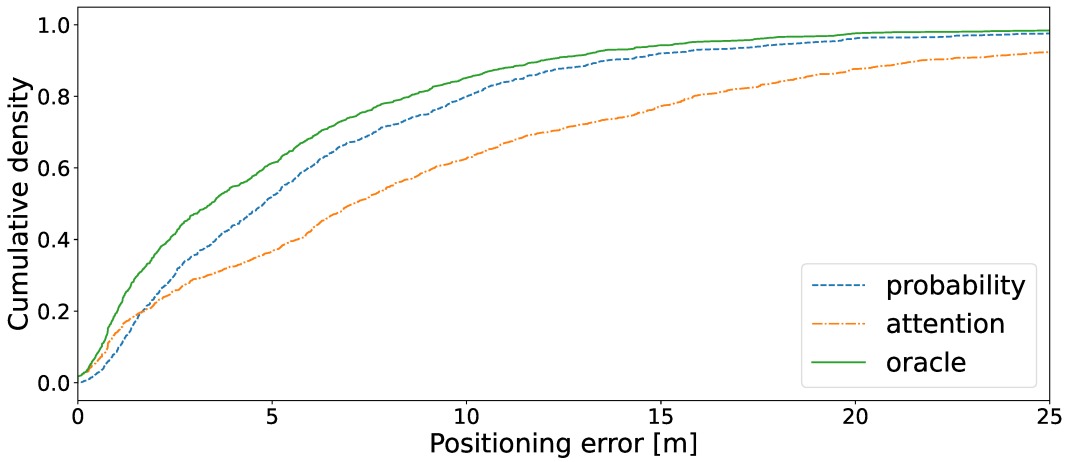


Figure 6.14: ECDF showing the performances of probability- and attention-based approaches in comparison with the oracle. Being able to always choose the optimal approach between the two solutions would lead to a consistent boost of positioning performance.

now compute the vector

$$\boldsymbol{\zeta} = \frac{\boldsymbol{\alpha}^4 \odot \boldsymbol{\pi}}{\sum_{i=1}^{|\hat{\mathcal{L}}|} \alpha_i^4} , \tag{6.19}$$

which is used to calculate the final weighted centroid ($\odot$ is the Hadamard, i.e., element-wise, product).

The overall idea is to employ the attention-based similarity scores as a filter for the probabilities. Intuitively, based on the previously discussed spatial characterization of attention patterns, the filtering aims to reduce the weights associated with regions that are not considered to be relevant by the attention mechanism. Eq. (6.19) indeed achieves this, since $\boldsymbol{\alpha}^4$ maps similarity attention scores to a (double) quadratic scale, and $\sum_{i=1}^{|\hat{\mathcal{L}}|}(\alpha_i)^4$ acts as a normalization factor. A clear advantage of the proposed solution is that it does not require any setting of K, since it considers all the locations belonging to a given floor. A graphical account of a typical scenario in which our method positively affects the result (error reduction from 18.7 to 4.07 meters) is reported in Figure 6.15. Notice how the area around the ground truth is characterized by higher similarity attention scores. This allows to strongly reduce the overall weight assigned to farther RPs that are associated with a relatively high probability, for instance, the one highlighted with a box. Results reported in Table 6.2 confirm the goodness of the approach, that achieves the lowest positioning error.

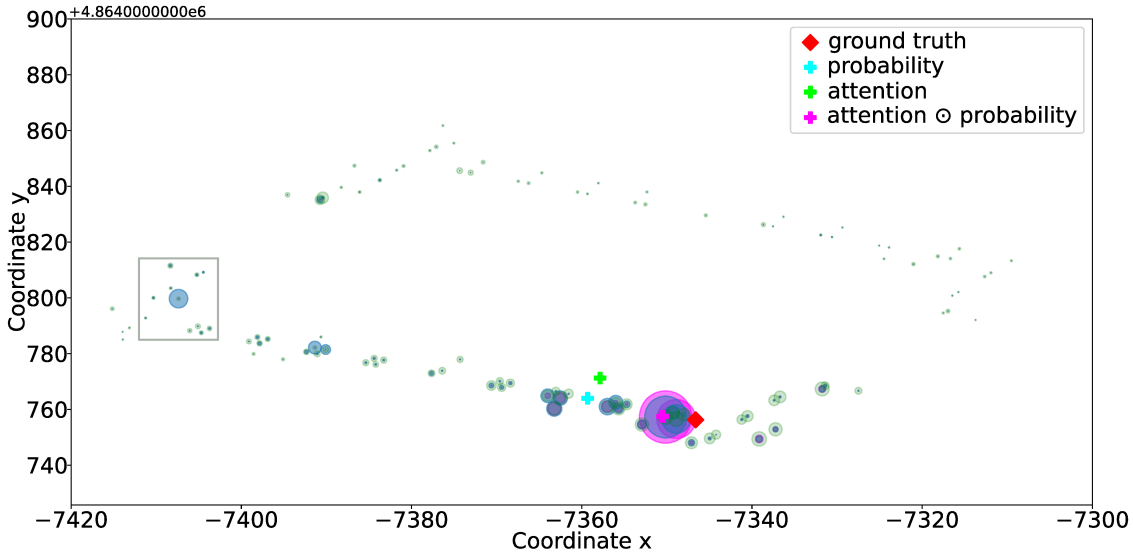


Figure 6.15: A typical positioning scenario. RP weights (larger bubble = higher value/relevance) and position estimation results according to the different approaches are considered. The grey box on the left highlights a situation where probability emphasizes a RP poorly considered by the similarity attention score. As it can be seen, our simple approach to combine probability and attention values leads to a significant improvement.

6.4 Discussion

The main goal of this work was to understand whether a form of interpretability within the fingerprint-based indoor positioning domain could be defined. Specifically, we proposed a local notion of explanation based on determining which access points are the most relevant to a given position estimate. Then, we extended it to a global perspective, conjecturing that access point relevance patterns may characterize specific areas of a building. In practice, such a measure of relevance was obtained by employing a sequence-to-sequence deep learning model equipped with an attention module, designed to operate on a ranking-based representation of fingerprints.

In this section, we assess the work done under several perspectives. We start with a critical analysis of the results reported in Section 4.3. Then, we compare, from the interpretability standpoint, the RSS and ranking-based fingerprinting, highlighting the superiority of the latter. Next, the main differences between our proposed framework and other approaches specifically designed to deal with RSS perturbations and AP selection are discussed. Afterwards, a series of applications that rely on our proposed notion of interpretability is presented.

Table 6.3: Summary of the experiments and their results

Experiment	Description	Method	Results
Quantitative	Are clustering results meaningful?	Analysis of the clustering dendrogram	*dot* is unsatisfactory due to poor partitioning; *general* and *deep* produce too many groups; *add* and *cat* generate balanced partitions that exhibit a clear compositionality
	Do clusters group instances sharing very specific attention patterns?	Statistical procedure based on the Kolmogorov–Smirnov test	*add* and *cat* have the best behaviour: they provide a high number of clusters but only few of them are pairwise similar
	Do different attention types exhibit *spatial characterization*?	Cluster compactness and separation based on the Hausdorff metric	*add* emerges as the best one, since it leads to (spatial) distance values for similar clusters that are typically lower than those for dissimilar ones (according to KS)
Qualitative	What do the different types of attention highlight?	Visual and comparative inspection	*general* and *deep* focus on the strongest access point(s); *dot* considers the last position in the padded rank; *add* and *cat* exhibit rather heterogeneous patterns
	Are attention-based groups truly spatially compact?	Visual inspection	Yes, clusters obtained with *add* exhibit clear spatial locality properties
Positioning evaluation	Are plain model performances comparable with SOTA?	Success rate and positioning error	Yes, they are in par with or better than SOTA solutions
	Can we improve the results leveraging interpretability insights?	Positioning error distribution	Yes, by a large margin, assuming an optimal strategy to combine attention scores and deep learning model likelihoods; regardless, a naive combination method still brings to an improvement

6.4.1 Critical analysis of the results

On the account of a series of thoroughly designed quantitative and qualitative experiments, that are summarized in Table 6.3, we showed that attention-derived patterns indeed characterize well-defined spatial regions, considering a large multi-building multi-floor well-recognized indoor positioning dataset. Nevertheless, our results also pointed out that not all the examined compatibility functions are equally capable of fulfilling such a role (*add* is clearly the best), highlighting the importance of choosing a suitable one. Notably, the spatial characterization of attention was observed even though our model exploits only categorical information about locations, neglecting any kind of proximity relationships among them. During such analyses it also emerged that deep learning models do not always exploit the strongest APs

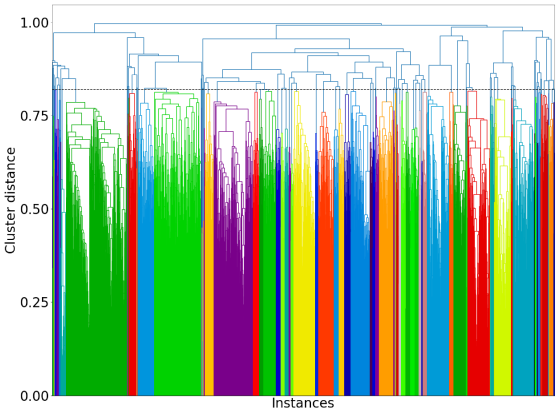
to perform a position estimation. This may suggest that the relevance of the APs depends, at least in part, on the model used. Thus, caution is advised in the design of fingerprint preprocessing and filtering strategies, as their validity could be model-dependent. In essence, attention indeed allowed us to get new scientific and operational insights about WiFi fingerprinting and deep learning.

Although our analysis confirms some very strong interpretability desiderata for our framework, i.e., plausible explanations aligned with the domain (by means of a spatial characterization), one might argue: could the same results have been obtained just by considering the presence of specific access points in the ranked fingerprints, or do the attention values assigned to such access points indeed play a major role? To determine that, we shuffled within each ranked fingerprint the attention values assigned by *add* to the access points, repeating our quantitative and qualitative experiments. The rationale is that if the original attention values were stochastic, the outcomes should be similar to those for the shuffled case, suggesting that our explanations are barely plausible and certainly not faithful nor meaningful. The dendrogram (Figure 6.16a) begins with a coarse partitioning of the instances into three groups, roughly corresponding to the three buildings. This is not surprising, as in different buildings we can expect disjoint sets of access points to be present in the rankings. Nevertheless, at a finer granularity level, the partitioning is more chaotic, as confirmed by the inspection of the cluster assignments on Floor 3, where groups appear highly random (compare Figure 6.16b with Figure 6.12). Indeed, also the KS-based similarity test shows some very different results (compare Figure 6.16c and Figure 6.16d with Figure 6.8 and Figure 6.9 respectively): shuffling the attention values we obtain substantially darker images, meaning that very large number of clusters are now considered to be similar by the KS procedure. Thus, although we remark that attention provides *a possible* explanation and not *the* explanation for the model behaviour, we provided evidence that in our case such an explanation is meaningful and aligned with the domain, as the shuffling experiment led to fundamentally different results, showing, above all, no spatial characterization.

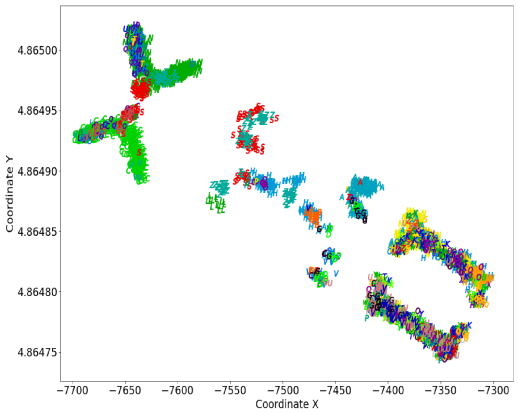
6.4.2 RSS vs ranked fingerprinting for interpretability

In this work, for our interpretability framework, we specifically focused on ranking-based fingerprints and attention. Although in principle interpretability techniques can also be applied to full-fledged fingerprints, when pursuing such an approach there are some inherent difficulties that have to be taken into account.

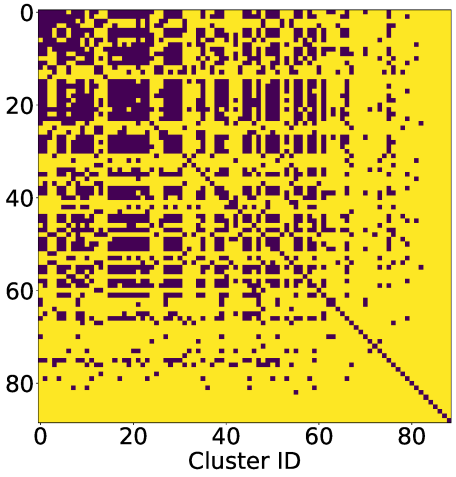
To begin with, recurrent neural networks cannot be applied to RSS fingerprints, since they rely on a sequential inductive bias [20]. Thus, other kinds of deep learning architectures should be exploited, like fully connected neural networks (FCNN) together with gradient-based attribution methods (instead of attention) [8], to highlight the access points that are most relevant to generate a prediction. In any case, RSS fingerprints tend to be very long, having an element for each distinct access point present in the considered scenario (520 APs in the dataset UJIIndoorLoc).



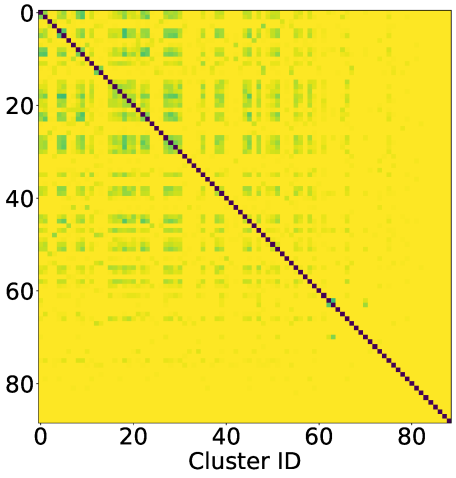
(a) Dendrogram.



(b) Floor 3 instances cluster assignments.



(c) KS-based cluster similarity.



(d) KS-based AP similarity ratio.

Figure 6.16: Results of *add* compatibility function, shuffling the attention values within each fingerprint.

The result is that a large number of relevance values should be taken into account, hindering the *compactness* and *sparsity* requirements of the interpretation, as stated in [191].

In addition, RSS fingerprints pose an issue from a practical point of view, as they are not transparent to the number of access points installed in the considered premises. Indeed, adding a new AP would require changing the architecture of and retraining the whole FCNN (increasing the dimension of the input layer by 1), which is not required with the recurrent approach as shown in Chapter 5.

Finally, note how, in our recurrent solution, the sequence-to-sequence modelling and the autoregressive nature of the decoder allow for a simple and effective extension to different indoor scenarios, both in terms of APs distribution and granularity of the hierarchical structure of positions. By relying on a FCNN, the hierarchi-

cal form of the prediction (building, floor, room), generated using a single model, would be lost, together with the adherence to the probabilistic framework described in Section 6.2.1. To recover it, two alternative directions can be followed. As a first solution, different models could be built for each distinct premise and hierarchical level. In the case of dataset UJIIndoorLoc, this translates to 1 model for the building prediction, 3 models for the floor prediction (one for each building), and around 12–15 models for the final room prediction (one for each floor of the different buildings). Alternatively, 3 models could be considered: 1 model for the building prediction taking in input the RSS fingerprint, 1 model from the floor prediction taking in input the RSS fingerprint and the predicted building, and 1 model for the room prediction taking in input the RSS fingerprint and the building and floor predictions. In both situations, the gradient-based relevance values would have a completely different and more complex interpretation, as they would depend on more than one model and/or rely on heterogeneous features.

6.4.3 Other approaches for access point selection and RSS noise mitigation

Note that our proposed architecture allows us to ignore aspects that are often central to the development of positioning systems, such as those pertaining to the selection of APs (i.e., features), the normalization of their RSS, and the mitigation of RSS perturbation phenomena. Of course, in the literature, other specific approaches are available to deal with such issues.

Considering AP selection, several solutions are available to determine the subset of APs that are more relevant for the prediction of the user location, i.e., those which are more likely to deliver a low localization error. Here, note how the strongest access points may not always provide the best positioning accuracy, as already shown in this work and in [52]. Among the selection approaches, either offline and online techniques can be found. The former selects a static subset of access points for the considered premises based on training data; such access points are then used for all downstream positioning tasks [1, 131, 132]. The latter, given a new fingerprint observed at an unknown location, generates a dynamic subset of APs, based also on the fingerprint information, before performing the actual positioning [53, 129, 306]. Our solution is clearly different with respect to the offline approaches, since attention values are computed for every new prediction. In addition, as opposed to the other online methods, it allows us to seamlessly and jointly perform, within a single model, selection and prediction phases, without the risk of heuristically discarding useful information in the workflow. Specifically, the relevance values generated by the attention mechanism can be considered as fuzzy selectors for the access points. The fact that attention is applied over a limited number of stronger APs does not constitute an issue, as remarked in Section 6.3.1.

As for dealing with RSS perturbation, proposed techniques include hyperbolic

location fingerprints [126], differential fingerprints based on signal strength difference [108], and mean differential fingerprints [134]. Although these works are designed to cope with RSS perturbations caused by device heterogeneity, the same holds also for our framework. Indeed, while ranks are generated starting from RSS values, previous literature [155, 163, 164, 223] showed that ranked fingerprints are more stable than RSS fingerprints, as they are coarser. The downside is that ranks are also less informative; nevertheless, deep learning appears to be an effective manner to cope with this, as showed here and in Chapter 5. In addition, our solution has a much broader scope than the previously mentioned techniques, as it allows us also to develop a domain-related notion of interpretability.

6.4.4 Applications

We believe our proposed *spatial characterization* of interpretability to be of actual use in several positioning-related tasks. To begin with, it could act as a guide for radio-map maintenance tasks. In the event of a known AP being faulty or removed, it becomes necessary to update the radio-map to reflect such changes. Instead of re-collecting fingerprints over a large area (note that an AP could be sensed even on multiple floors), relevance values could be exploited to limit the update operations to just the locations in which such AP was pertained to be useful by the model, leading to savings in terms of human time and computing resources. Of course, the usefulness of this application depends on the frequency of such AP replacement events.

As a second application, the access point relevance patterns could also help to identify unreliable predictions. One big issue in indoor positioning is that the environment is highly dynamic. Besides changes related to people crowding rooms and the intrinsic heterogeneity of devices, a major source of problems is represented by either the removal, the replacement, or the installation of APs. To avoid degradation of the positioning performance, the radio-map and the associated predictive models need to be updated to reflect such changes. Nevertheless, determining when and where such updates are necessary is a difficult task, especially considering that it may not be known which of the APs have undergone changes. A possible solution could be that of determining what is the typical pattern associated with a specific location or region of the premises, and then measuring to which extent a newly given fingerprint, predicted to belong to that area, adheres to the shared pattern. If radical changes occur to the environment, they should be reflected in a divergence between the two relevance patterns, intuitively showing a change to the importance being assigned to the sensed access points. In turn, such a discrepancy could inform the positioning system regarding the possible unreliability of the prediction and, in that case, determine the region affected by the error, consequently suggesting possible corrective actions as suggested earlier in this section.

Finally, consider the general positioning procedure, followed by both the probabilistic and the deterministic approaches; it begins with determining the most prob-

able locations as predicted by the model, leading to the retrieval of a set of points from the radio-map, for which latitude and longitude coordinates are known. Such coordinates are, in turn, exploited to derive the final position estimate. Spatial regions sharing the same access point relevance pattern could act as a constraint to reduce the number of candidates to consider when performing the radio-map lookup, contributing both to an improvement of the positioning performance, as well as to the overall speed of the process. To perform such a screening phase it may be sufficient to compare the relevance values of the given instance with the patterns characterizing the various areas, considering only the locations belonging to the most similar one. This may, for instance, complement or extend the simple position estimation approach that we presented in Section 6.3.4.

6.4.5 Current limitations

The presented work still has some limitations. To begin with, the generalizability of our results is limited by deep learning in itself, given its strong dependence on training data and on the supervised paradigm. Despite that, the developed experimental workflow and its outcomes are based on a rigorous statistical procedure, which should allow to derive similar conclusions in other indoor scenarios, provided that a sufficient amount of training data is available.

Also, note that our improvement in positioning performance, which is not the aim of the work but a way to show how interpretability can contribute in practice in indoor positioning, was based on an intuitive, although rather naive approach. More sophisticated techniques to jointly consider model likelihoods and attention scores should be investigated, however, preliminary analyses point out that such a task is not straightforward. The latter aspect is highly relevant since, as suggested by the oracle, the margin of refinement is considerably large.

Altogether, in this chapter, we confirmed the observation made in [44], namely, that the study of the relationships between attention weights and model interpretability is an active area of investigation that should be carefully considered by the research community.

IV

Towards Indoor-Outdoor Seamless Positioning

7

Crowdsourced Cellular Networks Reconstruction for Outdoor Fingerprinting

A unique challenge in the field of positioning is to develop a system capable of dealing with seamless indoor-outdoor information [166]. Such a goal is hard to accomplish given the peculiarities of the two domains, and the lack of a universal positioning solution. In this dissertation we try to move the very first steps in this direction, focusing on the idea of having a system relying entirely on fingerprints. Indeed, fingerprinting constitutes a valid approach to outdoor positioning due to the GNSS limitations, such as, for instance, its performance reduction in urban canyons, considering both the time required to obtain a position fix and the overall localization accuracy [153, 162, 269], and its high energy consumption [152, 304]. Nevertheless, outdoor fingerprinting often relies on a comprehensive and accurate knowledge of the cellular network configuration, which is not static and, above everything, is not disclosed by network operators.

First introduced in the early 1990s, the cellular radio communication network has evolved over time through successive technological generations (such as GSM, UMTS, LTE, etc.). Today, it is the most widely used communication network globally, providing the transmission of both voice and data signals. The term *cellular radio* refers to the deployment of numerous low-powered cell towers for signal transmission, each having a limited transmission range and a designated radio frequency. Given the global coverage of the cellular network and the common usage of its receivers, cellular signals are suitable for (signal) fingerprinting [113]. In such cases, a fingerprint consists of the collection of the signal strengths of the observed cell towers [22, 49, 195]. It follows that they have a strong spatio-temporal characterization: the gathered fingerprint is associated with a position (ground truth, i.e., observed using GNSS, or estimated); and, two fingerprints sensed in the same place may differ over time due to user equipment, environment, or cellular network changes.

Collecting data related to the cellular network and its configuration is nowadays easy and convenient, thanks to the pervasiveness and ubiquity of mobile devices [290]. Over time, this has resulted in vast amounts of spatio-temporal data being

gathered through crowdsourcing and stored in various datasets, some of which are open-source. The largest community project of this nature is OpenCellID[1], which reports an average of over 1 million measurements added daily. However, the data is not properly arranged, with the repository containing lists of different values stored in tabular formats, such as CSV, lacking a structure that reflects the inherent organization of the cellular network [288], a scenario resembling the one reported for indoor fingerprinting in Chapter 2.

To overcome such a limitation, we propose a general and flexible yet complete database schema for cellular networks, modeled after the information available in signal fingerprints, and capable of fostering the crowdsourced collection of data. Again, this follows the same reasoning we did for indoor fingerprinting: we start to tackle a problem by appropriately modeling the domain and the data. The system supports several operations, ranging from outdoor positioning to advanced spatio-temporal analyses and validation tasks pertaining to the state of the cellular network. Despite being based on standard specifications, each generation of the cellular network presents a challenge due to operators implementing unique organizational structures and potentially making undisclosed modifications. Therefore, obtaining complete and precise knowledge of the cellular network requires the systematic collection, processing, and analysis of new fingerprints. The task is exacerbated due to several characteristics of such a network being time-dependent, i.e., they undergo continuous changes. For instance, a cell may be created or removed, merged with another, or even spatially relocated [266]. This continuous network evolution is taken into account and effectively managed by the proposed system, which overall demonstrates how a deep knowledge of the cellular network arrangement can be achieved and maintained based only on crowdsourced information.

To the best of our knowledge, no other comprehensive attempt to model the considered scenario has been reported in the literature. Previous works concentrated on limited network analyses, performed over specific technologies and with the purpose of solving precise problems, often relying on artificial intelligence techniques [77]. For instance, this is the case with network optimization and planning tasks, such as the one in [130], where the authors propose an approach to determine groups of similarly behaving 3G cells, to support human experts in determining the state of the network; and that of [85], where the collection of a large-scale dataset to foster mobile network planning is presented. Then, a broad set of the literature focuses on anomaly detection and troubleshooting [21, 55, 88, 128, 159, 212, 248, 250]. The only work close in spirit to the one described here is our framework to model fingerprinting and building topology, already discussed in Chapter 2.

Recall that the work at the base of this chapter has been published in [36], but, as reported in Section I.1, in this dissertation we are not going to provide its complete description. Instead, we outline the resulting model and its capabilities, mainly for what concerns the temporal aspect. Such a choice stems from the fact that the

[1]OpenCellID website: `https://opencellid.org/`

publication in [36] builds on previous work by Gubiani et al. [100] (analysis of the context and outline of the framework), Viel, in his PhD dissertation (background, first modeling, and preliminary analysis) [265], and Andreussi, in his master thesis (contribution to the implementation of the revised and spatio-temporal extended database) [9], while our main contributions pertain to the refactoring of the entire model, its extension with the temporal dimension, and the subsequent analyses. Thus, the chapter is organized as follows. In Section 7.1 we provide an account of the overall model, detailing the motivation behind the choice of incorporating spatial and temporal information, with a specific focus on the latter and its usefulness. Then, in Section 7.2, we discuss various capabilities of the system, reporting also some analysis.

7.1 Modeling cellular networks after outdoor fingerprinting

Cellular communications currently operate based on standardized definitions and have global coverage. Numerous cellular technologies have been proposed over time, each possessing unique characteristics. Exhaustive technical information on wireless and cellular networks, including the features of signals and antennas, can be obtained from various academic texts (e.g. [110, 196, 230]).

The approach considered in this work, however, has been different: a comprehensive conceptual model of cellular networks that encompasses all data relevant from a mobile device perspective has been developed. This representation integrates field-gathered elements that may not be present or differ from standard declarations. A model of this type can serve multiple purposes, such as, for instance, aiding in individual positioning requests and analyzing the behaviour of large device groups. Additionally, it can validate the consistency of new observations with previously acquired network information, thereby detecting data anomalies and changes to network arrangements.

For technical terminology clarification, a list of acronyms used throughout the chapter is provided in Table 7.1, while for a detailed account of the cellular network, we refer the reader to the original works [36, 265].

7.1.1 Common issues related to networks management

Irrespective of the kind of network (GSM, UMTS, LTE), there are some underlying issues that have to be kept into account for a successful modeling.

First, as previously remarked, reaching a comprehensive knowledge of the network arrangement is of utmost importance to improve the localization performance of any cellular network-based positioning system. Note how, in turn, this affects the effectiveness of any other task that exploits information regarding the position

Table 7.1: Acronyms related to the specific network technologies used throughout the chapter.

Acronym	Technology	Description	Acronym	Technology	Description
ARFCN	GSM	Absolute Radio Frequency Channel Number	PCI	4G	Physical Cell Identifier
BCC	GSM	Base station Color Code	PCU	GSM	Packet Control Unit
BSC	GSM	Base Station Controller	PLMN	All	Public Land Mobile Network
BSIC	GSM	Base Station Id Code	PSC	UMTS	Primary Scrambling Code
BTS	GSM	Base Transceiver System	RA	GSM, UMTS	Routing Area
CGI	GSM, UMTS	Cell Global Identifier	RAC	GSM	Routing Area Code
CI	GSM, UMTS	Cell Identifier	RAI	GSM	Routing Area Identifier
EARFCN	LTE	Evolved ARFCN	RNC	UMTS	Radio Network Controller
ECI	LTE	E-UTRAN Cell Identifier	RSCP	UMTS	Received Signal Code Power
LA	GSM, UMTS	Location Area	RSRP	LTE	Reference Signal Received Power
LAC	GSM, UMTS	Location Area Code	RXLEV	GSM	Receiving Level
LAI	GSM, UMTS	Local Area Identifier	TA	LTE	Tracking Area
MCC	All	Mobile Country Code	TAC	LTE	Tracking Area Code
MNC	All	Mobile Network Code	TAI	LTE	Tracking Area Identifier
NCC	GSM	Network Control Code	UARCFN	UMTS	UTRA ARFCN

of users, such as behavioural analysis, emergency response, contact tracing, logistics, and so on. Nevertheless, determining the state of the network arrangement is not trivial when relying on just crowdsourced measurements: often the information sensed by the devices provides only a partial view of the environment, for instance, because they just aim to quickly discriminate between local cells. It is therefore essential to make full use of the existing information, enhancing it through the already available network data, and possibly exploiting related spatial knowledge.

Another major challenge is the inherent temporal evolution of the cellular network layout. Indeed, a number of reconfigurations can happen over time. For instance, in [266], the authors considered a cellular fingerprint dataset composed of 785,000 GSM observations collected in 2 years over a wide area of 1,800,000 km^2, finding 198 instances of cells that have changed one or more of their identifiers. The phenomenon is named *cell renaming*, to denote a change, even a partial one, of the cell identifier (e.g., the LAC parameter for a group of cells). Observe that such an alteration has a temporal connotation as it must hold that the first observation of a new cell takes place when the old one is not visible anymore. Moreover, the base station physically remains at the same place, spanning the same coverage, while the parameters of a cell change. Besides cell renamings, base station *relocations* may be performed, where a cell gets assigned the same logical identifiers that were previously ascribed to a different one placed at another location. Such events are complex to model and deal with, as they are characterized by both spatial and temporal changes. Moreover, renaming events concerning administrative areas (i.e., registration or routing areas) or the physical portions of the network (e.g., network controllers and base stations) are also to be hypothesized.

It is clear why these phenomena are a concern for localization if not properly handled. In the case of renaming, it would be impossible to exploit the information of the old cell, as it would not be associated with the currently visible one. This can naturally lead to a substantial degradation of the positioning performance in the affected area, as a considerable amount of data would suddenly no longer be

available. The phenomenon would be accentuated in areas with a low cell density, such as rural regions. In [266], it is shown how accounting for such network changes allows increasing the average number of candidates available for each position estimation by 12.1%. In the case of relocation, the problem would be an inconsistency of the data, which would again result in a positioning error, potentially of high magnitude being it closely related to the spatial displacement affecting the cell. Thus, a system that is able to effectively and efficiently manage the information required to identify these phenomena has great potential in the domain. Finally, observe that the actual identification of the previously mentioned phenomena (which can be delegated to appropriate algorithms) is only one side of the matter; then, the system needs also to be capable of harmonizing and keeping track of the old and the new data, guaranteeing the overall consistency of the information before and after the network modification events.

7.1.2 An account of the overall model

In Figure 7.1, we report the general conceptual model of the domain using a spatio-temporal extension of the classical Entity-Relationship notation, called ChronoGeoGraph (CGG) [101, 102]. As we said, we do not discuss all the choices that led to its development, specifically the various components, which instead can be found in [36].

Briefly, it is possible to observe various parts, each devoted to a specific network component: the PLMN and SUBPLMN, which model information about countries and the networks present in them; the ADMINISTRATIVE AREA, which models the location area, the routing area, and the tracking area; the RADIO ACCESS NETWORK, that deals with the network controller and the base station; the CELL; and, the POSITIONING SYSTEM, which represents the module in charge of dealing with the location of a device.

To model the properties (attributes and relationships) that are specific to each type of network generation, the entities within each part introduce several (total and disjoint) specializations. Notably, since we consider administrative identifiers for the cells, the partial key in the CELL specialization can be either *ci* (in the case of GSM or UMTS technologies) or *eci* (for LTE). In the schema, they have been represented using the single attribute *eci / ci*.

The positioning component, which models fingerprinting, is the one that enables to have spatial information throughout the entire system. Specifically, a DEVICE (identified by its *id*) sends to the system several OBSERVATIONs at different time instants. These observations can be associated with a GNSS position (*GNSSfix*), or not. In the latter case, an estimation (*GNSSestimation*) may be derived using suitable positioning algorithms for fingerprinting [269]. Observations convey information about one or more observed CELLs and the corresponding observed *signal_strength*. The cell can be either the serving one, to which the device is directly connected, or a neighbour one. Note that the observation of a cell allows us to extract all the

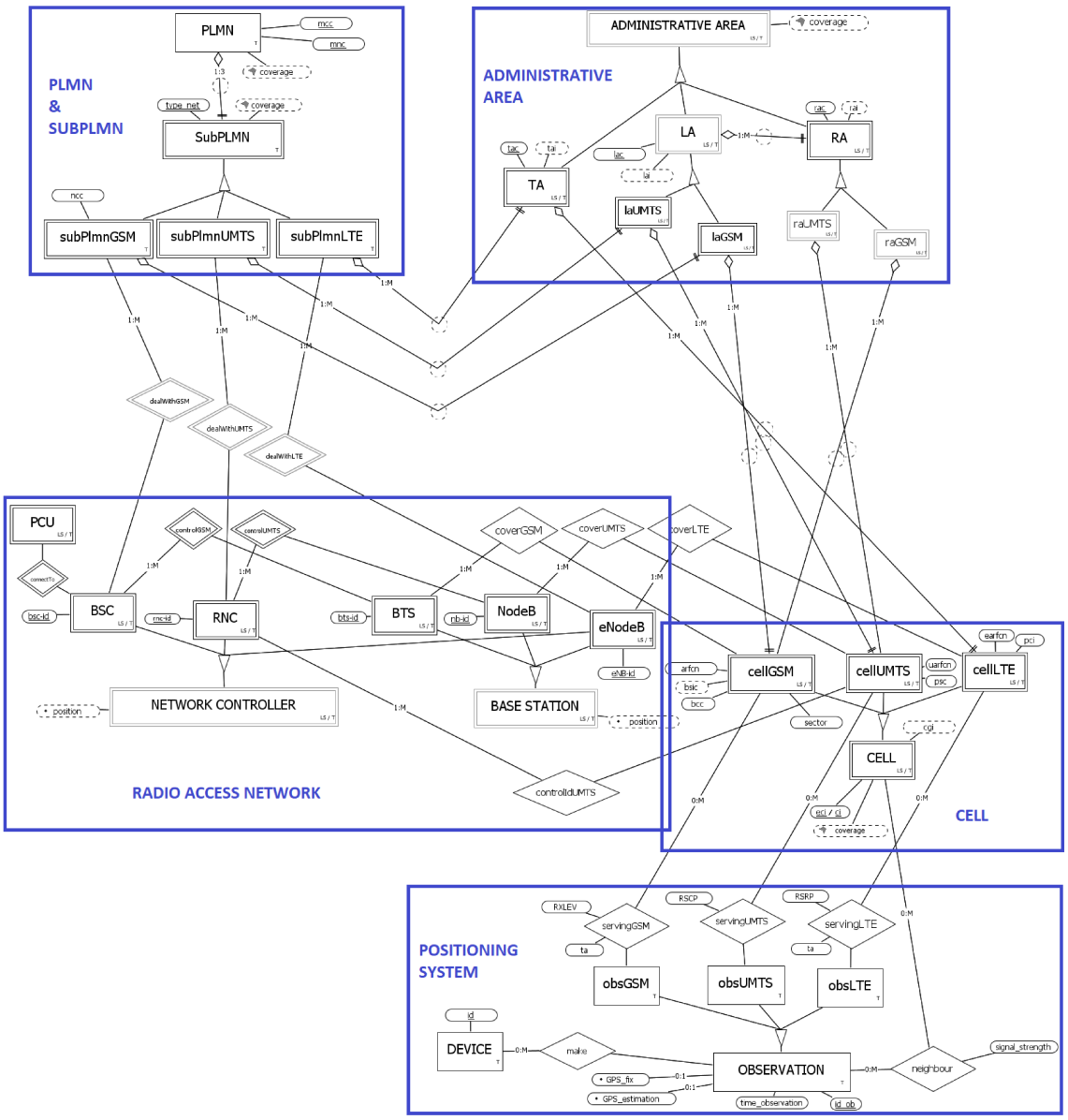


Figure 7.1: Complete cellular network schema, integrated with the positioning system and equipped with spatio-temporal support. Letter T denotes entities that feature transaction time, whereas letters LS denote instances that feature valid time.

other information modeled by the (administrative) entities of the schema, as they are directly detected by devices as parts of the cell id. Of course, observations may in principle be extended with additional attributes, such as the Timing Advance (TA) for GSM and LTE serving cells (attribute *ta*). These sets of attributes may encode information regarding the cell, the device, such as the speed or its direction, and the object that is being tracked (e.g., a person or an animal).

Finally, note that the point-based information provided by the OBSERVATIONs

is not enough to have a precise knowledge of the cellular network from the spatial perspective, as other entities have spatial attributes, some of which represent areas. Specifically, a fundamental part pertains to managing the coverage of the cells as well as other elements in the hierarchy. Thanks to the overall modeling, evaluating the *coverage* of a cell, and in a cascading fashion of the other elements in the hierarchy (i.e., administrative areas and PLMNs components), can be done on the fly aggregating the position (attribute *GNSS_fix*) of the OBSERVATIONs that have perceived the cell, either as a serving or a neighbouring one (e.g., calculating the convex hull). Such on-demand evaluation significantly improves the efficiency of the model, enabling it to store less (redundant) data and to reduce the number of heavy and complex spatial computations, performing them only when strictly necessary. Other spatial features can be reconstructed from the observations. This is the case of the derived attribute *position*, which is part of the NETWORK CONTROLLER and BASE STATION (there exist ad-hoc algorithms, see for instance [265]). Indeed, from the position of the BASE STATIONs one can infer that of the linked NETWORK CONTROLLER.

7.1.3 The temporal aspects

A temporal database is a database that integrates support for handling the evolution of data and knowledge throughout time. The temporal aspects of our interest are the so called *valid time* and *transaction time*: the first represents the time interval during which a fact is true in the real world and it is a user-defined representation of time; the second represents the time interval during which a fact is current in the database and it is a system-defined representation of time, namely managed by the DBMS. A temporal database is physically implemented by augmenting the primary key of each table one wants to temporalize. Specifically, the key is extended with a pair of values for each temporal dimension, representing the extremes of the respective interval. A database that implements both valid and transaction time is referred to as a bi-temporal database.

The schema in Figure 7.1 reports the temporality associated with the entities, by means of the letters that are located at the bottom right corner of the corresponding boxes. Each entity of the cellular network has a transaction time interval (letter T). This allows us to easily retrace the database evolution, showing its content as it was at any previous moment in time. One could argue about the decision to maintain the OBSERVATION's *time_observation* attribute despite the presence of the transaction time; however, such values may be different, for example, due to a delay in the insertion of one instance into the database with respect to its sampling time. Moreover, if updates are carried out on such database records (e.g., to correct errors), they might not have an impact on the *time_observation*, while certainly generating a new version of the record, tracked through the transaction time[2]. In

[2]In temporal databases with transaction time, each operation (insertion, deletion, update)

addition, considering that renaming may affect cell, administrative area and radio access network (as explained in Section 7.1.1), for each of the entities involved we also provide a valid time interval (identified by LS), that allows us to specify the moment of reality in which they existed. For example, this can be the case when we want to keep track of a cell that has been renamed and therefore no longer exists but existed in a certain past time interval.

Finally, to make the distinction between transaction and valid time clearer, as well as to justify the presence of both of them, consider the following two scenarios.

Scenario 1. On 2022-09-01 a new observation is added to the database, which, for the first time, reports the presence of a cell, say $cell_x$, which has never been seen before. Thus, a tuple that describes such a cell is inserted with both transaction and valid time intervals starting from 2022-09-01:[3]

$$\{\, (cell_x, T : [\text{2022-09-01}, \infty], LS : [\text{2022-09-01}, \infty]) \,\}\ .$$

The day after, a delayed observation, originally recorded on 2022-08-25, gets finally processed and inserted into the database. Such an observation also reports the presence of $cell_x$. As a result, the tuples that describe such an event now are:

$$\begin{aligned}\{\, &(cell_x, T : [\text{2022-09-01}, \text{2022-09-02}], LS : [\text{2022-09-01}, \infty]),\\ &(cell_x, T : [\text{2022-09-02}, \infty], LS : [\text{2022-08-25}, \infty]) \,\}\ .\end{aligned}$$

Note how, in this way, the cell's history is kept by means of the logically deleted tuple (i.e., with a closed transaction time), while the current information is encoded by the newly added one.

Scenario 2. Valid and transaction times play a central role also in the management of a cell renaming or relocation. To reliably identify such events, often multiple observations witnessing the change of parameters are needed, collected over an extended time frame (in general, the fewer the measurements related to a given area, the longer the period). Let us assume that a cell, named $cell_x$, first detected and inserted in the database on 2017-07-12, changes part of its global identifier on 2022-09-01, thus being detected, from that moment on, as a new cell, say $cell_y$. The situation is described by the following tuples:

$$\begin{aligned}\{\, &(cell_x, T : [\text{2017-07-12}, \infty], LS : [\text{2017-07-12}, \infty]),\\ &(cell_y, T : [\text{2022-09-01}, \infty], LS : [\text{2022-09-01}, \infty]) \,\}\ .\end{aligned}$$

Furthermore, assume that 10 measurements of $cell_y$ are needed to validate such information, and that we receive a new measurement each day. Thus, only on 2022-09-11 we are able to reliably detect the renaming event and merge $cell_x$ into $cell_y$.

involving a record produce a new record, with updated transaction time interval.

[3]Here and in the following, for the ease of reading, we adopt a simplified version of the tuples, not referring to the actual conceptual schema.

The situation now becomes:

$$\{\,(cell_x, T : [\text{2017-07-12}, \text{2022-09-11}], LS : [\text{2017-07-12}, \infty]),$$
$$(cell_y, T : [\text{2022-09-01}, \infty], LS : [\text{2022-09-01}, \infty]),$$
$$(cell_x, T : [\text{2022-09-11}, \infty], LS : [\text{2017-07-12}, \text{2022-09-01}])\,\}\ .$$

Note that the "old" information pertaining to $cell_x$ has been logically deleted from the database using the transaction time on 2022-09-11 on the first tuple; and, a new tuple reflecting the more recent knowledge about it is added on the same date. Observe as, thanks to the valid time, the newly added tuple encodes the information that, from 2022-09-11, we know that the $cell_x$ is not alive since 2022-09-01. Concerning the other information related to the two cells (e.g., their observations), they are dealt with in a similar way: after 2017-09-01, those of $cell_x$ become associated with $cell_y$ by means of changes to their valid and transaction times. This is particularly relevant from the point of view of positioning: when a new measurement detects $cell_y$, it can now leverage all the available information observations irrespective of the renaming operation.

As a matter of fact, to deal in practice with the temporal dimensions, we cannot directly model the entities with just their natural keys and the temporal intervals, but we need to introduce a set of surrogate keys. The reason for such a change is twofold. First, consider a possible renaming operation. Such procedure can affect attributes that are part of the (natural) primary keys of a table[4]. Let us assume, for instance, a renaming operation affecting a given cell $cell_{ci1}$ (where $ci1$ denotes the value of its attribute ci), turning it into $cell_{ci2}$. Since changing part of the primary key means having a new, independent entry, the aforementioned event would result in not being able to recognize that $cell_{ci1}$ evolved in $cell_{ci2}$, as no link would be present among the corresponding (temporal) records, both still present in the database. Introducing a surrogate primary key *id_cell* solves the issue, as it is going to be immutable since the first appearance of a cell. In the considered renaming case, only the attribute ci would change, thus it would still be possible to recognize that the cell with $id\_cell = 1, ci = 1$, evolved in $id\_cell = 1, ci = 2$, linking the information between older and newer versions of the same cell. Note that such a procedure applies to any scenario where we merge multiple cells together: ci can be used to retrieve all the records in CELL composing the actual cell (within a given administrative area), while *id_cell* discriminates between the single (merged) entries. Recalling the renaming example we described above, we now have the

[4]Recall that, in a bi-temporal database, the primary key of a temporalized entity is augmented with the attributes representing the transaction and valid time intervals, allowing the system to keep a history of each instance composed of all its subsequent modifications.

following scenario:

$$\begin{aligned}
&\{\,(id_cell = 1, ci = 1, T : [\text{2017-07-12}, \infty], LS : [\text{2017-07-12}, \infty]),\\
&\quad (id_cell = 2, ci = 2, T : [\text{2022-09-01}, \infty], LS : [\text{2022-09-01}, \infty])\,\}\\
&\qquad\qquad\qquad\qquad \Big\downarrow \text{rename } ci = 1 \text{ in } ci = 2 \text{ on 2022-09-11}\\
&\{\,(id_cell = 1, ci = 1, T : [\text{2017-07-12}, \text{2022-09-11}], LS : [\text{2017-07-12}, \infty]),\\
&\quad (id_cell = 2, ci = 2, T : [\text{2022-09-01}, \infty], LS : [\text{2022-09-01}, \infty]),\\
&\quad (id_cell = 1, ci = 2, T : [\text{2022-09-11}, \infty], LS : [\text{2017-07-12}, \text{2022-09-01}])\,\}\,.
\end{aligned}$$

Compared to the case where ci is part of the (temporal) primary key and the surrogate key does not exist, here it is possible to retain the fact that $(id\_cell = 1, ci = 1)$ evolved in $(id\_cell = 1, ci = 2)$. Note that, after the renaming, the cell with $ci = 2$ is logically composed of two records, and that cell $ci = 1$ is not alive by itself anymore. Of course, the procedure can be repeated indefinitely, always preserving the entire knowledge about network's reconfigurations:

$$\begin{aligned}
&\{\,(id_cell = 1, ci = 1, T : [\text{2017-07-12}, \text{2022-09-11}], LS : [\text{2017-07-12}, \infty]),\\
&\quad (id_cell = 2, ci = 2, T : [\text{2022-09-01}, \infty], LS : [\text{2022-09-01}, \infty]),\\
&\quad (id_cell = 1, ci = 2, T : [\text{2022-09-11}, \infty], LS : [\text{2017-07-12}, \text{2022-09-01}]),\\
&\quad (id_cell = 3, ci = 3, T : [\text{2022-09-15}, \infty], LS : [\text{2022-09-15}, \infty])\,\}\,.\\
&\qquad\qquad\qquad\qquad \Big\downarrow \text{rename } ci = 2 \text{ in } ci = 3 \text{ on 2022-09-20}\\
&\{\,(id_cell = 1, ci = 1, T : [\text{2017-07-12}, \text{2022-09-11}], LS : [\text{2017-07-12}, \infty]),\\
&\quad (id_cell = 2, ci = 2, T : [\text{2022-09-01}, \text{2022-09-20}], LS : [\text{2022-09-01}, \infty]),\\
&\quad (id_cell = 1, ci = 2, T : [\text{2022-09-11}, \text{2022-09-20}], LS : [\text{2017-07-12}, \text{2022-09-01}]),\\
&\quad (id_cell = 3, ci = 3, T : [\text{2022-09-15}, \infty], LS : [\text{2022-09-15}, \infty]),\\
&\quad (id_cell = 2, ci = 3, T : [\text{2022-09-20}, \infty], LS : [\text{2022-09-01}, \text{2022-09-15}]),\\
&\quad (id_cell = 1, ci = 3, T : [\text{2022-09-20}, \infty], LS : [\text{2017-07-12}, \text{2022-09-01}])\,\}\,.
\end{aligned}$$

The second reason motivating the introduction of surrogate keys pertains to the propagation of updates in the database. For instance, let's assume a natural key attribute of a given registration area, say $ra1$, changes. Then, according to the schema of Figure 7.1 which does not make use of surrogate keys, this would cause a cascading update of all the foreign keys of cells connected to $ra1$. Such an event happens because the hierarchy defines a chain of weak entities, starting from the PLMN till the CELL. Again, this unnecessary computational burden is avoided with the introduction of a (immutable) surrogate key.

As a final note, besides introducing surrogate keys to effectively support the temporal capabilities of the system, other restructuring steps are needed in order to have a conceptual schema that can be easily translated into a logical one. In such

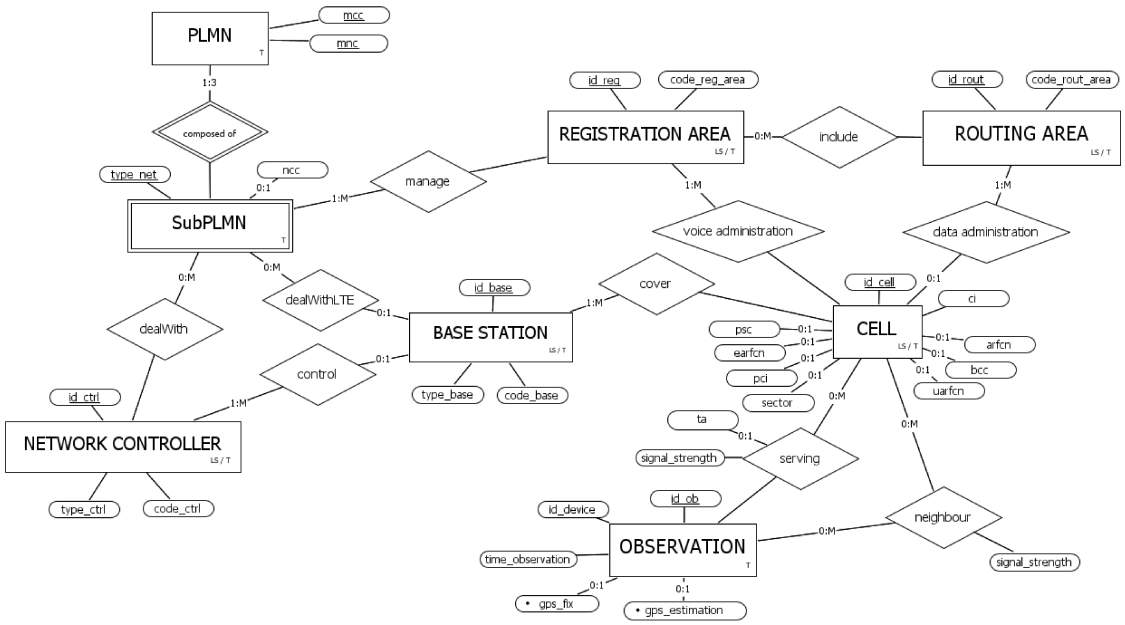


Figure 7.2: Restructured cellular network schema.

steps, we remove/combine unnecessary attributes and simplify entities, specifically concerning the specializations related to the network generations. The resulting restructured schema is shown in Figure 7.2, while for the logical schema (obtained with standard mapping rules) and its implementation, we refer the reader to [9, 36].

7.2 An overview of the system capabilities

In order to verify the actual capabilities of the system, we relied on two different datasets incorporating information about the cellular networks and where those observations have been made. Such a scenario exactly matches our crowdsourced fingerprinting assumption.

OpenCellID is a collaborative community project that collects measurements and cell towers' data by means of an API and a ready-to-use mobile phone application. The dataset we considered ranged from 2014-01-01 (02:02:44) to 2017-03-17 (06:34:24), spanning the entire world and including 42,952,377 measurements based on three different cellular technologies: GSM (26,896,809), UMTS (6,195,903), and LTE (9,859,665). OpenCellID dataset makes no distinction between serving and neighbour cells, thus all measurements are considered to be distinct and all entries are serving cells. For further testing the generality of the developed system, we extended the OpenCellID data with a proprietary dataset gathered by the company *u-blox*, ranging from 2016-06-23 (23:21:20) to 2016-06-30 (23:00:57). For privacy reasons, we have obtained only information about the cellular network and not any

details about devices and users. Overall, the dataset, obtained by parsing raw logs, includes 12,492,545 measurements, partitioned into GSM (11,998,811) and UMTS (493,734), distinguishing between serving and neighbours cells . Compared to the OpenCellID dataset, u-blox data have lower worldwide coverage, with some areas more densely represented (e.g., South Africa and Europe). However, those datasets do overlap in time (we recognized 63,839 cells appearing in both).

7.2.1 Continuous and periodic validation

Of course, it is not possible to populate the database with raw data directly, especially given that the main goal is to deal with heterogeneous data sources. Thus, some "continuous validity checks"should be performed in order to determine the correctness of each new measurement. Such checks include verifying that the observed values belong to appropriate ranges, that there are no `null` values in mandatory fields, and that there are no inconsistent values (e.g., using GNSS to verify that a measurement is actually in the state identified by the observed *mcc*).

Other types of controls made available by the systems are the "periodic validity checks", which are instead run only at regular intervals, due to their semantic nature and computational complexity. An example of a periodic check is as follows: if a cell has enough associated observations, it can be determined whether the (possibly estimated) locations of its latest observations (that are, those entered after the last launch of this periodic check) are consistent with the coverage of the cell, i.e., the geometric distances of the locations with respect to the previously known cell extension are plausible (e.g., not too large). Finally, the detection of cell renaming phenomena has also been implemented as a periodic routine: briefly, on the basis of [266], it verifies whether several (spatio-temporal) conditions characterizing the renaming phenomenon are satisfied. Ideally, periodic checks should complement and be run in parallel with continuous ones.

In order to assess the effectiveness of both types of checks, the database has been populated simulating the continual arrival of data, pertaining to both the considered datasets, respecting their natural and real evolution over time. The outcomes, which details may be found in [36, 265], are that such checks and the modeling are fundamental to storing in an organized fashion data coming from multiple, heterogeneous sources (i.e., around 7% of measurements are filtered out, and the remaining information suitably integrates together).

7.2.2 Basic, spatial and temporal analysis

Here, we present some relevant use cases that are made possible by the developed system. Clearly, most of the described analyses can also be performed starting from raw data, but in that case the procedures are much more complex and computationally intensive than using the structured model.

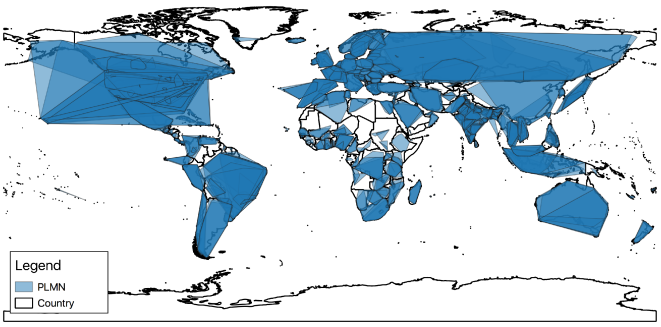


Figure 7.3: A general overview of PLMNs' coverage.

A number of straightforward analyses to extrapolate statistics concerning the cellular network and its configuration can be carried out. For instance, it is easy to answer questions like determining the number of PLMNs available in each country or finding the countries with the highest number of PLMN operators. The same logic works for any other kind of components of the network modeled by the system.

Indeed, such a set of basic analyses can be extended in order to incorporate spatial information. For instance, as shown in Figure 7.3, we can derive the coverage area of each PLMN. The reported polygons are the convex hulls made from all the validated observations belonging to each specific PLMN. For this reason, geographical areas that lack polygons do not necessarily correspond to zones with no radio coverage at all, but rather are probably characterised by too few or too closely collected observations.

Thanks to the structuring of our model, it is possible to obtain other types of information at each level of the administrative organisation of the mobile network, also with respect to different technologies. For instance, let us consider Germany, which is the country with the largest number of observations (21,713,580) and the second in terms of observations density (after Singapore) in the database; it is also the second country in terms of the number of cells (560,803, after the USA) and the twelfth in terms of cells density. Below we show how different levels of the network hierarchy provide different types of information about the selected area.

Let us consider the case where you would like to inspect the coverage for a given area with respect to a single PLMN and/or a specific technology. A possible usage of such an analysis is to point out which PLMN has the best coverage with respect to a specific cellular technology. For instance, Figure 7.4 (top part) shows the coverages associated with the three different technologies considered in this work for a German PLMN (`mcc=262`, `mnc=1`). Although their areas may look very similar, if we calculate the bounding boxes at the cell level (still grouping them by subPLMN), we obtain a very different picture, as shown in Figure 7.4 (bottom part). Such observations can be noticed even if we consider the numerical information derived from such spatial queries: around 50% of PLMNs (combination of `mcc`, `mnc`) are detected for all three considered technologies (457 over 996), 302 for two technologies, and only 237 for

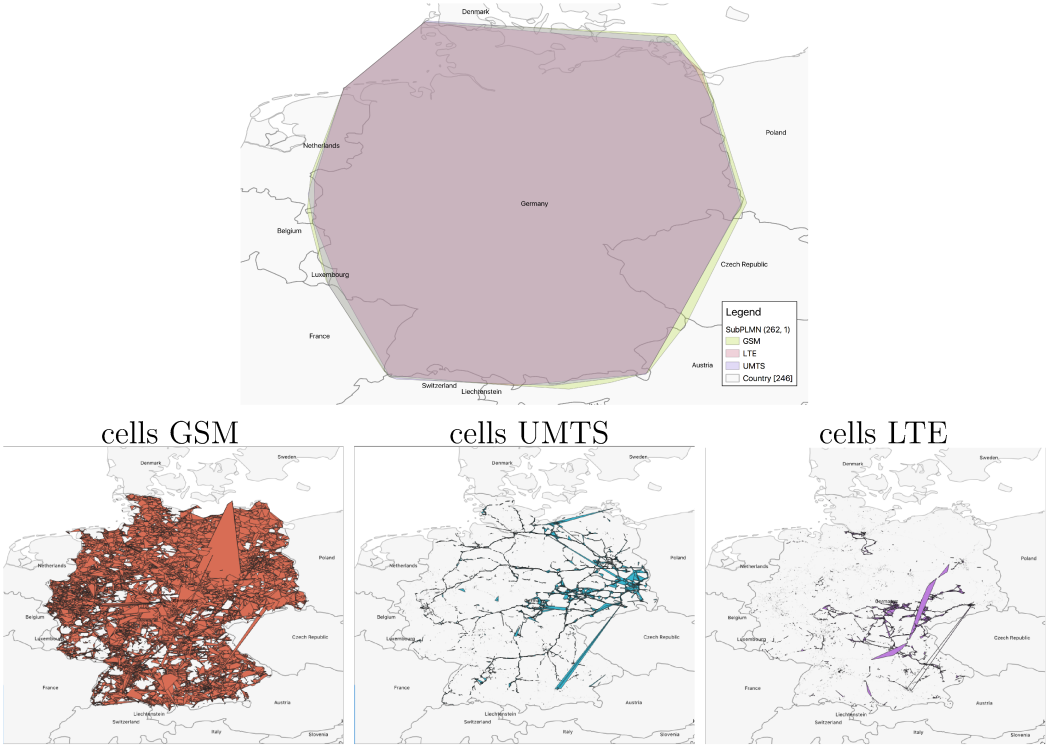


Figure 7.4: Coverage of different technologies of the same PLMN.

only one. This remarks the usefulness of structuring the information at different granularity levels, modeled within a flexible hierarchy.

We can get new information also by logically grouping cells into their administrative areas, which are useful to provide localization at a coarse granularity [73, 133, 237]. As a matter of fact, the density of the cells within an area, and, thus, that of administrative areas, is likely to be useful to distinguish between urban and rural environments [79, 237, 302]. For instance, the administrative partitioning considering the GSM network with $mcc = 262$ and $mnc = 3$ is reported in Figure 7.5. It is immediately seen that the cluster on the left (low density) represents a rural area, while the two on the right (high density) pertain to urban contexts.

Moving to the temporal modeling, notice that, in general, a set of observations represents a given state of the network (at a precise moment in time). Moreover, neither spatial nor temporal regularities are guaranteed for observations. Thus, no assumption can be made about the state of the network between two measurements involving the same cell, especially if they are not close in time (i.e., we can not conclude that, if we did not see the cell, no changes have affected it). Hence, for instance, it is not possible to establish whether a cell was not visible because it was not operating, or simply because no device had made observations in its coverage area. In the following, under the simplifying assumption that the latter hypothesis holds, we outline some useful temporal analyses.

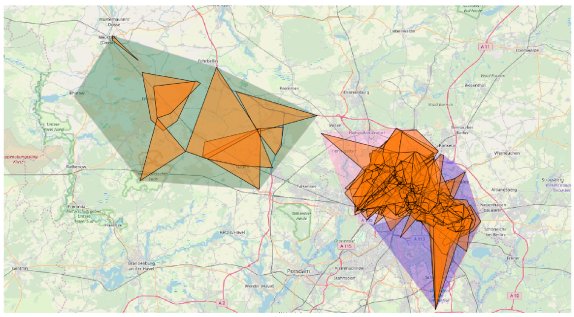


Figure 7.5: Administrative areas in the Berlin area: urban (violet and pink polygons) and rural (green polygon) area. Orange polygons represent cell coverages.

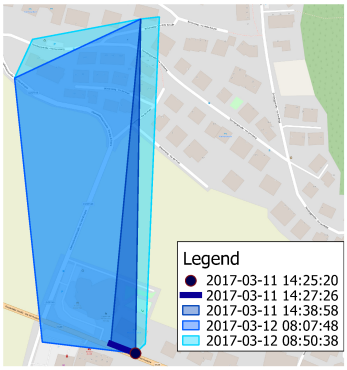


Figure 7.6: Evolution of the coverage of a cell as new measurements are added to the database over time. A brighter color denotes a more recent state.

The presence of both transaction time and valid time dimensions allows us to easily track the evolution of the coverage of a cell over time. The idea is to exploit the transaction time of the involved instances to easily go back in time progressively considering the changes affecting the state of a cell. Figure 7.6 illustrates the coverage of a cell in the province of Bolzano (Italy) as it changes over time. Specifically, we report the resulting shape as new observations for such a cell are made. In the beginning, the coverage is just a single point, as only one observation detected such a cell. Then, the area progressively grows till it reaches the extension of the bright cyan polygon on 2017-03-12 at 08:50:38, i.e., when the last observation of the cell is added to the database. It is worth pointing out how the overall knowledge about the cell considerably changed over a very short time interval.

Let us now turn to another practical scenario where the temporal modeling of the data shows its usefulness: based on the valid time recorded for each instance, we can inspect the evolution of the coverage of the UMTS network in Germany at two different time points (2016-03-17 and 2017-03-17), as shown in Figure 7.7. Indeed, the coverage increases over time. Similar analyses can be performed to monitor the coverage evolution of several mobile operators, considering different technologies. In turn, those data may support the detection of deficiencies in an operator's network,

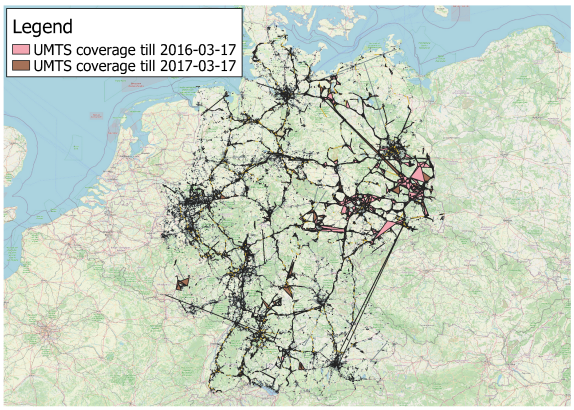


Figure 7.7: Temporal evolution of the UMTS coverage in Germany, measurements obtained from 2015-03-17 up to 2016-03-17 (pink color) or 2017-03-17 (brown color). Only serving cells are considered.

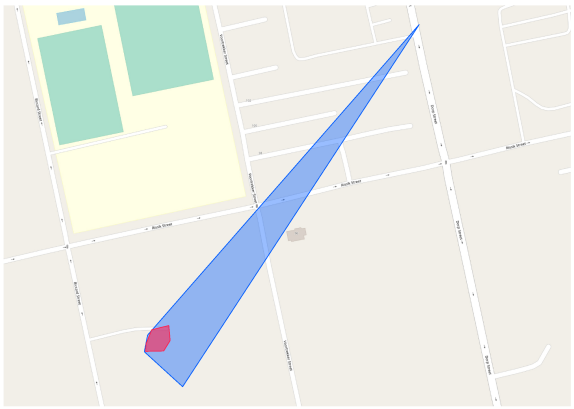


Figure 7.8: Backtracking a cell renaming operation. The two original cells are shown in red and blue. The cell after the rename matches the blue one since the red cell is fully contained in the other.

the comparison of competitors' coverages, and the development of temporal-based machine learning models able to fully exploit historical data.

Finally, we present the case of an actual cell that has been involved in a renaming operation, detected thanks to the developed system. In such a case, it may still be useful to investigate how the network arrangement was before such an operation. Figure 7.8 depicts the information pertaining to a cell in the city of Polokwane (South Africa). Thanks again to the support of valid and transaction times offered by the system, we can go back in time within the database prior to the renaming operation, allowing us to investigate the situation before the merging (i.e., the last step in the renaming process), which consist of two distinct cells (red and blue polygons).

7.3 Discussion

In this chapter, we reported an outline of how it is possible to reconstruct and maintain information about the infrastructure of cellular networks by making use of crowdsourced data sensed by mobile devices, in order to support outdoor fingerprinting. The gained knowledge was formally encoded by means of a conceptual database schema encompassing spatio-temporal information, flexible enough to deal with several kinds of network technologies (and easily extendable to others) and able of accommodating crowdsourced measurements. We then showed how the system supports several network-related tasks, considering some real datasets. Most importantly, it allows one to maintain an accurate and up-to-date representation of the network infrastructure, through the detection of inconsistent measurements coming from mobile devices, e.g., due to the violation of spatio-temporal constraints on their collection, and cell renaming phenomena. In addition, a selection of exemplary analyses has been presented, ranging from basic ones to more complex spatial and temporal use cases. Overall, the proposed system poses as a solid basis to foster all kinds of tasks based on outdoor positioning and cellular network analysis.

Towards our goal of developing a system for seamless indoor-outdoor localization, this chapter covers a fundamental step. The underlying principle follows that of Chapter 2, where we modeled indoor positioning, and developed a system accordingly. The main difference lies in the fact that for the outdoor case, we did not have the counterpart of the building topology yet. While from one perspective this role can indeed be performed by the hierarchy derived from the cellular network structure, according to the usage of the system, the granularity offered by it could be too coarse: for instance, the area covered by a cell might be large by nature, or small due to a partial lack of knowledge, leading to too much uncertainty. A possible workaround, which does not require us to define ad-hoc modeling as done for the indoor case, would be of integrating into the database information about the road network. Such data is already openly available, also in a format aligned with our spatio-temporal modeling. In this way, the outdoor topological model could be composed of the road network data and, possibly, the cellular network hierarchy. Then, such a model could be easily linked with its indoor counterpart, leading us towards having a holistic modeling of fingerprint-based positioning. However, the topological information would not be fully exploited till the point we introduce the concept of trajectory, i.e., sequence of spatio-temporal points. Trajectories are not a primitive concept, but they convey a large amount of information, which, beyond contributing towards developing more precise algorithms, we believe may play a crucial role precisely when a user/device moves from outdoor to indoor and vice versa. As we showed in Chapter 2, introducing support for this type of information in the model is not challenging (the indoor DB already does this). Nevertheless, to the best of our knowledge, there is a lack of work in the area of trajectories applied to (outdoor) fingerprinting, a topic which we, therefore, address in the next chapter.

8

Represent and Compare Outdoor Fingerprint-based Trajectories

A trajectory can be defined as a sequence of timestamped locations of a user or a device. Although it is not a primitive concept, trajectories and their comparison are fundamental in many important tasks like clustering, classification, and mining of user behaviour, which are common operations in transportation, urban planning, environmental studies, business, and public security [172]. Besides these applications, we believe that, concerning our goal of developing a seamless indoor-outdoor positioning system, trajectories can play a crucial role in modeling transitions between different contexts. Indeed, these sequences of points can be characterised by heterogeneous observation with respect to the sampling frequency, devices, and type of observed data. Thus, it is not difficult to imagine how they can play a pivotal role in our context, specifically supporting positioning tasks regarding people and devices moving from an outdoor to an indoor environment and vice versa. Nevertheless, to fulfill the role, a measure is needed to estimate how much two distinct trajectories are similar/different (similarity measure). The similarity evaluation problem has been extensively studied in the case of trajectories reconstructed from Global Navigation Satellite System (GNSS) points, like the Global Positioning System (GPS), under different perspectives [6, 48, 144, 151, 165, 274, 297]. Here, we focus on fingerprinting relying on the cellular network, i.e., exploiting the unique cell tower identifier (cell-ID) and the related Received Signal Strength Indicator (RSSI). Among the cells observed by a device and composing a fingerprint, we recall that it is possible to identify two different types: the serving cell, which is always present and it is the one to which the device is connected; and, the neighbours' cells, i.e., all the others. Concerning the spatial information, it will be represented by means of latitude and longitude only. Observe that such data could be either retrieved by GNSS-like technologies, for the few devices equipped with the proper module, or could be estimated by position estimation (PE) fingerprinting algorithms.

The overall goal of this chapter is to understand which is the best methodology to measure similarities in the fingerprinting realm, with a specific focus on determining the differences with respect to the GNSS case. To achieve this, we first make a careful evaluation of the applicability of similarity measures that have been origi-

nally developed in the GNSS setting to cellular fingerprint trajectories, and then we propose an original approach tailored to the considered context. Compared to the GNSS case, cellular fingerprinting is characterised by significant differences in the frequency of observations (heterogeneity), which reflect differences in the behaviours of the devices/users. In many cases, such a frequency turns out to be quite low. In the evaluation of existing similarity measures, we will indeed pay attention to the distinctive features of low-sampling heterogeneous trajectories. Moreover, given that in fingerprinting we can rely on both spatial and cellular information, we will investigate which features are best suited for evaluating the similarity among this kind of trajectories.

On the basis of such an investigation, we propose an approach that exploits multiple sources of information, including both spatial identifiers and cell identifiers with RSSI. A careful experimental evaluation of the various similarity measures based on a large dataset of fingerprint trajectories from real users concludes the work.

The rest of the chapter is organised as follows. In Section 8.1, we review some existing measures of similarity, focusing on those that can be more easily adapted to cellular fingerprinting, providing a precise mathematical formulation of them. The application of suitable variants of these measures to fingerprint trajectories is illustrated in Section 8.2, where we also introduce an approach to compare two (signal-based) fingerprints, and a similarity measure specifically tailored to fingerprint trajectories. In Section 8.3, we outline the framework for the experimental evaluation of the considered measures. The outcomes of the experimentation are reported in Section 8.4.

8.1 Comparison of GNSS and non-GNSS trajectories

In the first part of the section, we briefly survey similarity measures for (GNSS) trajectories, focusing on those that can be adapted to fingerprint ones. Then, the three standard measures EDR, ERP, and LCSS are formally defined.

8.1.1 An overview

A trajectory T can be represented as a sequence of positions $\langle p_1, ..., p_n \rangle$. Each position p_k is, in fact, a data structure encoding the coordinates of the spatial position, the acquisition timestamp, and other available data. In the case of fingerprinting systems, we have one or more cells with the associated signal strength.

Different (dis)similarity measures have been proposed in the literature. Two basic approaches are the Closest-Pair Distance (CPD), which defines the distance between two trajectories as the minimal distance between their points, and the Sum-of-Pair Distance (SPD), which expresses (dis)similarity as the sum of the distances of

aligned points, under the assumption that trajectories have the same length/number of points [301]. Even if intuitive, these measures suffer from some shortcomings: the CPD approach is too coarse, and the assumption on the length of trajectories by SPD is hard to fulfil in real scenarios.

To overcome their limitations, more advanced solutions have been proposed. Dynamic Time Warping (DTW) looks for similar patterns among trajectories [28]. To this end, trajectories are non-linearly warped in the temporal dimension, to obtain a similarity measure independent of certain variations on that axis. Measures based on Edit Distance [147], such as Edit Distance for Real sequences (EDR) [47] and Edit Distance with real Penalty (ERP) [46], determine the minimum number of operations (insertions, deletions, and substitutions) that have to be performed to transform one trajectory into the other. Longest Common Subsequence (LCSS) [270] finds the longest subsequence that a set of sequences have in common. A trajectory similarity measure based on it is $LCSS_{\delta,\epsilon}$ [268]. The latter two groups of measures turn out to be easily transferable to the fingerprinting setting.

To the best of our knowledge, there is not a widely adopted similarity measure specifically developed to handle fingerprint trajectories. However, some attempts to compare trajectories consisting of locations which are not directly expressed as coordinates can be found in the literature. In the following, we briefly present some related work.

In [298], a solution based on Call Detailed Records (CDR) is proposed, where each point of a trajectory is identified by a spatial point and a cell-ID. The evaluation of trajectory similarity makes use of a measure close to Edit Distance, that takes into account both spatial and temporal dimensions. The novelty of the proposal is the usage of centroid displacements resulting from the application of edit operations as the cost metric. Its main limitation is that CDR data are, in general, not publicly available. Another solution, that borrows its main features from research on sequence comparison, has been proposed in [195]. It is based on the Smith-Waterman algorithm [243] and views trajectories simply as sequences of cell-IDs. It introduces two changes to the classic Smith-Waterman algorithm. First, the last seen cell appearing in the trajectory must produce a matching edit operation when compared to trajectories stored in the database; second, the costs of matches and mismatches are respectively 1 and -0.5. In [162], a method using cell-IDs and handoff patterns is proposed. It combines string alignment on cell-IDs with information about cells similarity. The handoff pattern between two cells corresponds to the number of mutual switches between them. Since these patterns can be proved to be stable over the same route, two trajectories are similar if they have a long connection time to similar cell towers that appear in similar orders.

A different representation of trajectories is given when semantic locations are considered: points of trajectories are tags (strings), that encode knowledge about the locations they represent, e.g., points of interest [173, 204]. A semantic-based method is proposed in [293, 294]. The similarity is assessed by determining how many parts the two trajectories have in common. To this end, the notion of Maximal

Semantic Trajectory Pattern Similarity, which is based on LCSS, is exploited. Latent Semantic Analysis (LSA) comes into play in the method described in [198]. It views trajectories as documents and defines a mapping from the spatial/cellular space to a grid one, which can be represented by a matrix. Similarity of trajectories is then reduced to similarity of grid cells, which is determined by applying a suitable similarity function. A more semantic evaluation of trajectory similarity can be obtained by applying LSA to the matrices, thus generating reduced ones, and then measuring their similarity.

8.1.2 EDR, ERP, and LCSS

Let $\mathbf{r}$ and $\mathbf{s}$ be two GNSS positions, and let $R = \langle \mathbf{r}_1, .., \mathbf{r}_m \rangle$ and $S = \langle \mathbf{s}_1, .., \mathbf{s}_n \rangle$, with $|R| = m$ and $|S| = n$, be two trajectories.

The *Edit Distance for Real sequences* (EDR) is based on Edit Distance, and it allows one to compare sequences which are not simple strings of characters [47]. The rationale followed by this measure (as well as from the others considered) is that the overall dissimilarity score is given by the sum of the contribution brought by each pair of points belonging to the two considered trajectories. Such contribution is determined by means of a *cost* function, which is characteristic for each (dis)similarity measure. The cost function $d_{EDR}(\mathbf{r}, \mathbf{s})$ for EDR is defined as follows:

$$d_{EDR}(\mathbf{r}, \mathbf{s}) = \begin{cases} 0 & \text{if } d_{point}(\mathbf{r}, \mathbf{s}) \leq \epsilon \ ; \\ 1 & \text{if } d_{point}(\mathbf{r}, \mathbf{s}) > \epsilon \ ; \\ 1 & \text{if } \mathbf{r} = gap \text{ or } \mathbf{s} = gap \ , \end{cases} \tag{8.1}$$

where d_{point} returns the (spatial) distance between two points[1], ϵ is a suitable threshold, whose value depends on the considered domain, and *gap* represents a missing point (it can be viewed as a symbolic value to be used for distance evaluation when an insertion or a deletion has been applied). Intuitively, d_{EDR} assigns a cost equal to 0 if the distance between $\mathbf{r}$ and $\mathbf{s}$ is lower than ϵ (i.e., $\mathbf{r}$ and $\mathbf{s}$ are considered as matching points), and a cost equal to 1 if it is greater than ϵ (i.e., to cope with

[1]It is worth pointing out that, in the following, whenever we refer to the (spatial) distance between two points, we assume that an appropriate measurement is performed (e.g., Haversine distance), depending on whether points are locally projected or not.

insertion, deletion, and substitution). $EDR(R,S)$ is defined as follows:

$$EDR(R,S) = \begin{cases} n & \text{if } m = 0 \ ; \\ m & \text{if } n = 0 \ ; \\ EDR(Rest(R), Rest(S)) & \text{if } d_{EDR}(\mathbf{r}_1, \mathbf{s}_1) = 0 \ ; \\ min \begin{cases} EDR(Rest(R), Rest(S)) + d_{EDR}(\mathbf{r}_1, \mathbf{s}_1) \\ EDR(Rest(R), S) + d_{EDR}(\mathbf{r}_1, gap) \\ EDR(R, Rest(S)) + d_{EDR}(gap, \mathbf{s}_1) \end{cases} & \text{otherwise} \ , \end{cases} \tag{8.2}$$

where $Rest(T)$ is the sub-trajectory extracted from T by removing its first element.

Similarly to EDR, the *Edit distance with Real Penalty* (ERP) is based on Edit Distance. The main difference between ERP and EDR lies in the cost function. In EDR, each edit operation adds a cost equal to a fixed value, while in ERP the cost is proportional to the actual relation (i.e., distance) between the considered elements. The cost function $d_{ERP}(\mathbf{r}, \mathbf{s})$ which characterise ERP is defined as follows:

$$d_{ERP}(\mathbf{r}, \mathbf{s}) = \begin{cases} d_{point}(\mathbf{r}, \mathbf{s}) & \text{if } \mathbf{r}, \mathbf{s} \neq gap \ ; \\ d_{point}(\mathbf{r}, \mathbf{g}) & \text{if } \mathbf{s} = gap \ ; \\ d_{point}(\mathbf{g}, \mathbf{s}) & \text{if } \mathbf{r} = gap \ , \end{cases} \tag{8.3}$$

where d_{point} and gap are defined as for $d_{EDR}(\mathbf{r}, \mathbf{s})$, and $\mathbf{g} = (0, 0)$ is a reference point. Assuming that R and S have been normalised with the procedure given in [46], $ERP(R,S)$ is defined as follows:

$$ERP(R,S) = \begin{cases} \sum_{i=1}^{n} d_{ERP}(gap, \mathbf{s}_i) & \text{if } m = 0 \ ; \\ \sum_{i=1}^{m} d_{ERP}(\mathbf{r}_i, gap) & \text{if } n = 0 \ ; \\ min \begin{cases} ERP(Rest(R), Rest(S)) + d_{ERP}(\mathbf{r}_1, \mathbf{s}_1) \\ ERP(Rest(R), S) + d_{ERP}(\mathbf{r}_1, gap) \\ ERP(R, Rest(S)) + d_{ERP}(gap, \mathbf{s}_1) \end{cases} & \text{otherwise} \ , \end{cases} \tag{8.4}$$

where $Rest(\cdot)$ is the same as for $EDR(\cdot, \cdot)$. The reference point $\mathbf{g}$ makes it possible to extend the cost function, which is proportional to the distance between the two considered points, to insertions and deletions, where only one point is available.

Unlike EDR and ERP, $\text{LCSS}_{\delta,\epsilon}$ is a trajectory similarity measure based on the *Longest Common SubSequence* criterion [268]. In general, it takes into consideration both the spatial and temporal dimensions. The temporal parameter δ specifies the maximum number of mismatching points that can exist between two matching ones.

The spatial parameter ϵ defines how far away two points can be to be considered as matching. Such a similarity measure is used, for instance, in [225, 267].

Here we consider a simple version of LCSS, proposed in [47, 165], that makes use of the spatial parameter ϵ only. Its cost function $d_{LCSS}(\mathbf{r}, \mathbf{s})$ is defined as follows:

$$d_{LCSS}(\mathbf{r}, \mathbf{s}) = \begin{cases} 0 & \text{if } d_{point}(\mathbf{r}, \mathbf{s}) \leq \epsilon \ ; \\ 1 & \text{if } d_{point}(\mathbf{r}, \mathbf{s}) > \epsilon \ , \end{cases} \tag{8.5}$$

where d_{point} and ϵ are defined as for $d_{EDR}(r, s)$. $LCSS_{sim}(R, S)$ is defined as follows:

$$LCSS_{sim}(R, S) = \begin{cases} 0 & \text{if } m = 0 \text{ or } n = 0 \ ; \\ LCSS_{sim}(Rest(R), Rest(S)) + 1 & \text{if } d_{LCSS}(\mathbf{r}_1, \mathbf{s}_1) = 0 \ ; \\ max \begin{cases} LCSS_{sim}(Rest(R), S) \\ LCSS_{sim}(R, Rest(S))) \end{cases} & \text{otherwise} \ , \end{cases} \tag{8.6}$$

where, again, $Rest(\cdot)$ is the same as for $EDR(\cdot, \cdot)$. The normalised dissimilarity version of LCSS is computed as follows:

$$LCSS = 1 - \frac{LCSS_{sim}(R, S)}{\min\{|R|, |S|\}} \ . \tag{8.7}$$

8.2 A proposal for fingerprint trajectories

In this section, we focus on fingerprint trajectories, where the information about location, rather than being the precise GNSS one, is often either represented implicitly by means of cellular data indeed observed at a certain location or encoded explicitly as a result of a position estimation (PE). However, all this information can coexist within a trajectory point, enabling trajectories to be represented in multiple ways, according to the set of data that is considered. Since in the remainder of the work, we might need to refer to a specific representation, given a generic trajectory point $\mathbf{p}$, we will use $\mathbf{p}^s$ and $\mathbf{p}^c$ to refer to its spatial or cellular one respectively. As a meta-observation, note that when PE is expressed as a spatial point within a common reference system, e.g., latitude and longitude coordinates, as indeed we do in this work, all the similarity measures for GNSS-based trajectories can be directly applied, as the two representation are structurally identical. Nevertheless, whether or not they provide the same results in terms of quality, it is a matter of discussion for Section 8.4.

In the following, we first show how to adapt EDR, ERP, and LCSS to the case of fingerprints concerning their cellular representation (Section 8.2.1), and then we propose a similarity measure tailored to a hybrid scenario (Section 8.2.2).

8.2.1 EDR, ERP, and LCSS revisited

From the cellular perspective, a trajectory point $\mathbf{p}$, which is a fingerprint, can be represented as an arbitrarily long vector of the form:

$$\mathbf{p}^c = [(\textit{cell-}ID_1, \mathbf{data}_1), ..., (\textit{cell-}ID_m, \mathbf{data}_m)] \,, \tag{8.8}$$

where $\mathbf{data}_i$ is a vector of cellular data associated with the cell tower identified by $\textit{cell-}ID_i$. In the following, among those data, we consider RSSI only.

Given that cell-IDs are unique and basically stable, a possible representation is that of describing a trajectory as a sequence of (timestamped) locations, each labelled with the cell-ID of the serving cells of the corresponding fingerprint. Thus, it is possible to revise the EDR, ERP, and LCSS similarity measures to fit the new scenario, essentially changing their cost function.

Both EDR and LCSS come from string matching and assign a unitary cost, equal to 0 or 1, to a mismatch. To allow cellular-based comparison, we use the same principle in redefining their cost function: we test whether two cell-IDs are equivalent and determine the unitary cost accordingly. Indeed, it is easy to see that trajectories based on serving cell-IDs strongly resemble full-fledged strings.

Let $\mathbf{r}^c$ and $\mathbf{s}^c$ be two fingerprints, and let r^c_{cid} and s^c_{cid} be the corresponding serving cell-IDs.

EDR for cellular data is obtained from Definition (8.2) by redefining the cost function $d_{EDR}(\cdot, \cdot)$ as follows:

$$d^c_{EDR}(\mathbf{r}^c, \mathbf{s}^c) = \begin{cases} 0 & \text{if } r^c_{cid} = s^c_{cid} \,; \\ 1 & \text{if } r^c_{cid} \neq s^c_{cid} \,; \\ 1 & \text{if } r = \textit{gap} \text{ or } s = \textit{gap} \,, \end{cases} \tag{8.9}$$

where *gap* is defined as in the basic case.

The same rewriting applies to LCSS:

$$d^c_{LCSS}(\mathbf{r}^c, \mathbf{s}^c) = \begin{cases} 0 & \text{if } r^c_{cid} = s^c_{cid} \,; \\ 1 & \text{if } r^c_{cid} \neq s^c_{cid} \,. \end{cases} \tag{8.10}$$

The above changes cannot be applied to ERP, urging us to propose a new cost function in order to maintain the ERP key advantage of being based on penalties proportional to a given distance between two elements (e.g., the spatial distance in the case of GNSS points). To such an extent, with the aim of better discriminating among fingerprints, we use all available cellular data, not only the cell-ID of the serving cells. Such an approach has the positive effect of mitigating the impact on the similarity evaluation of some cellular networks issues, such as unexpected cell reselection [231].

As a preliminary step, let us introduce some notations. First of all, let $\mathit{Cells}(\mathbf{p}^c)$ be a function that, given a fingerprint $\mathbf{p}^c$ defined as in (8.8), returns the set of cell-IDs observed in $\mathbf{p}^c$ (i.e., the cell-IDs of the serving cell and those of the neighbouring

cells). Then, given two generic fingerprints $\mathbf{p}^c$ and $\mathbf{q}^c$, let us define:

$$ListCellID(\mathbf{p}^c, \mathbf{q}^c) = Cells(\mathbf{p}^c) \cup Cells(\mathbf{q}^c) \; ; \tag{8.11}$$

$$Match(\mathbf{p}^c, \mathbf{q}^c) = Cells(\mathbf{p}^c) \cap Cells(\mathbf{q}^c) \; ; \tag{8.12}$$

$$Mismatch(\mathbf{p}^c, \mathbf{q}^c) = ListCellID(\mathbf{p}^c, \mathbf{q}^c) \setminus Match(\mathbf{p}^c, \mathbf{q}^c) \; . \tag{8.13}$$

Given a fingerprint $\mathbf{p}^c$, the number of cell-IDs associated with it (cardinality of $Cells(\mathbf{p}^c)$) is highly correlated to its geographic area, as different regions are characterised by different cell densities. In particular, in urban areas there are more cells than in rural ones [100], possibly leading to comparing two instances with a very different number of cells. In such an eventuality, the cost computed by the similarity evaluation procedure could be largely influenced by the number of cells. To avoid this, we introduced the *Normalised Cellular Distance* (NCD), a measure that returns continuous values between 0 and 1, which is defined as follows:

$$NCD(\mathbf{p}^c, \mathbf{q}^c) = \frac{\sqrt{\sum_{k \in Match(\mathbf{p}^c, \mathbf{q}^c)} (\mathbf{p}^c[k_{RSSI}] - \mathbf{q}^c[k_{RSSI}])^2} + |Mismatch(\mathbf{p}^c, \mathbf{q}^c)| \cdot 110}{|ListCellID(\mathbf{p}^c, \mathbf{q}^c)| \cdot 110} \; , \tag{8.14}$$

where $\mathbf{p}^c[k_{RSSI}]$ denotes the RSSI of the cell observed in $\mathbf{p}^c$ and identified by the cell-ID k.[2]

According to Definition (8.14), for any two fingerprints r and s, we redefine the cost function $d_{ERP}(\cdot, \cdot)$ as follows:

$$d^c_{ERP}(\mathbf{r}^c, \mathbf{s}^c) = \begin{cases} NCD(\mathbf{r}^c, \mathbf{s}^c) & \text{if } \mathbf{r}^c, \mathbf{s}^c \neq gap \; ; \\ NCD(\mathbf{r}^c - 1, \mathbf{s}^c) & \text{if } \mathbf{r}^c = gap \wedge \mathbf{r}^c - 1 \in R \; ; \\ NCD(\mathbf{r}^c, \mathbf{s}^c - 1) & \text{if } \mathbf{s}^c = gap \wedge \mathbf{s}^c - 1 \in S \; ; \\ 1 & \text{otherwise} \; , \end{cases} \tag{8.15}$$

where $\mathbf{r}^c - 1$ and $\mathbf{s}^c - 1$ respectively denote the points before $\mathbf{r}^c$ and $\mathbf{s}^c$ in the corresponding (full) trajectories. Such a formulation is very close to that of d_{ERP} given in Definition (8.3), but for the fact that Definition (8.15) deals with gaps in a way similar to that of the DTW technique [28]. In classical d_{ERP} (Definition (8.3)), when a *gap* is encountered a comparison is done between the point which is not a *gap* and the reference point; in the cellular setting (Definition (8.15)), the available point is compared with the point preceding the *gap* in the *gap*-affected trajectory (as it partially happens with DTW), if available.

[2]The number 110, which occurs in the definition of NCD, represents what we consider to be the minimum signal strength detectable by a device (precisely, $-110dbm$). Whenever the detected RSSI is lower than that, it is a reasonable assumption to round it to $-110dbm$.

8.2.2 Spatial Edit Distance for Fingerprints (SEDF)

In Section 8.2.1, we showed how to adapt GNSS point-based similarity measures to deal with data available in fingerprinting, considering cellular data only. In this section, we propose a new similarity measure, specifically designed for fingerprint trajectories, that incorporates spatial information. It is inspired by the measure proposed in [298], which is based on Edit Distance.

The key idea is to exploit the centroid of the trajectories. As in the case of GNSS-based trajectories, the centroid is computed by averaging the coordinates of the points of the trajectory. Thus, it acts like a balancing point for the trajectory, minimising the square of the Euclidean distances between itself and the other points. Notice that the removal of a point close to the centroid causes a small variation of it, while the removal of a distant one may have a more significant impact.

The novelty of the proposed measure, called *Spatial Edit Distance for Fingerprints* (SEDF), lies in the way it deals with the different types of available data. We introduce a new coefficient, named *mitigation factor*, which is in charge of adjusting the cost of the three edit operations (substitution, deletion, and insertion). In our setting, in its base formulation, the cost is purely spatial, as it only depends on the centroid displacement introduced by a single edit operation. On the other hand, the mitigation factor takes into account both spatial and cellular information, and it has the effect of reducing or increasing the cost of an edit operation.

Let R and S, with $|R| = m$ and $|S| = n$, be two trajectories. $SEDF(R, S, i, j)$ is defined as follows:

$$SEDF(R,S,i,j) = \begin{cases} 0 & \text{if } m = i \wedge n = j; \\ SEDF(R,S,i,j+1) + MitigatedIns(R,S,j) & \text{if } m = i\ ; \\ SEDF(R,S,i+1,j) + MitigatedDel(R,i) & \text{if } n = j\ ; \\ min \begin{cases} SEDF(R,S,i+1,j+1) + MitigatedSub(R,S,i,j) \\ SEDF(R,S,i+1,j) + MitigatedDel(R,i) \\ SEDF(R,S,i,j+1) + MitigatedIns(R,S,j) \end{cases} & \text{otherwise}\ , \end{cases} \tag{8.16}$$

where i and j are two indices used to scan the trajectories. The mitigation factors are defined as follows:

$$MitigatedSub(R,S,i,j) = NCD(\mathbf{r}_i^c, \mathbf{s}_j^c) \cdot CostSubstitution(R,S,i,j)\ ; \tag{8.17}$$

$$MitigatedDel(R,i) = (Mitigation(R,R,i) + 0.5) \cdot CostDelete(R,i)\ ; \tag{8.18}$$

$$MitigatedIns(R,S,j) = (Mitigation(R,S,j) + 0.5) \cdot CostInsert(R,S,j)\ , \tag{8.19}$$

where, following [298] but discarding the temporal components, the base cost functions are defined as:

$$CostSubstitution(R,S,i,j) = \left\| \left(\frac{\sum_{\mathbf{r}_k \in R} \mathbf{r}_k^s}{|R|} \right) - \left(\frac{\mathbf{s}_j^s + \sum_{\mathbf{r}_k \in R, \mathbf{r}_k \neq \mathbf{r}_i} \mathbf{r}_k^s}{|R|} \right) \right\| ; \quad (8.20)$$

$$CostDelete(R,i) = \left\| \left(\frac{\sum_{\mathbf{r}_k \in R} \mathbf{r}_k^s}{|R|} \right) - \left(\frac{\sum_{\mathbf{r}_k \in R, \mathbf{r}_k \neq \mathbf{r}_i} \mathbf{r}_k^s}{|R|-1} \right) \right\| ; \quad (8.21)$$

$$CostInsert(R,S,j) = \left\| \left(\frac{\sum_{\mathbf{r}_k \in R} \mathbf{r}_k^s}{|R|} \right) - \left(\frac{\mathbf{s}_j^s + \sum_{\mathbf{r}_k \in R} \mathbf{r}_k^s}{|R|+1} \right) \right\| . \quad (8.22)$$

As it is clear from Definitions (8.17), (8.19), and (8.18), the mitigation component is computed in different ways depending on the edit operation.

Mitigated substitution. As for substitutions, given two trajectory points $\mathbf{r}_i$ and $\mathbf{s}_j$, their cellular data (resp. $\mathbf{r}_i^c, \mathbf{s}_j^c$), are compared by means of the NCD function of Definition (8.14). If the two points involved in the swap produce a high centroid displacement, the NCD value may decrease the cost by evaluating how similar those points are from the cellular perspective. As an example, if they turn out to be very similar, NCD returns a value close to 0, cancelling the cost introduced by the substitution operation.

Mitigated deletion and insertion. Dealing with the other two edit operations turns out to be more difficult, as deletions and insertions act on a single point of R (resp., S). The idea is to suitably exploit aggregated data to compute the mitigation factor. Let us assume that we want to remove a point $\mathbf{t}_i$ from a trajectory T. Each point includes both spatial ($\mathbf{t}_i^s$) information and cellular data ($\mathbf{t}_i^c$). First, we group all the points by looking at the cell-ID of their serving cell, i.e., two points will belong to the same cluster if they have the same servant. Then, for each cluster, we compute its representing point, i.e., a centroid, by looking at the spatial component of cluster members. In this way, for each $\mathbf{t}_i$, besides the PE provided by the fingerprint algorithm, we have new spatial information, as we can associate to it the point representing the cluster to which the serving cell of $\mathbf{t}_i$ belongs. We refer to this latter point as $C_{serv}(\mathbf{t}_i)$.

Now, thanks to the structure of fingerprints, each trajectory point is also paired with a set of neighbouring cells. Since these cells have a cell-ID, we can retrieve the spatial point of the corresponding cluster (it will exist for the majority of cells), obtaining a set of spatial points representing the set of neighbouring cells. Let us denote the centroid of such a latter set for the neighbouring cells of $\mathbf{t}_i$ by $C_{neigh}(\mathbf{t}_i)$. Naming $C(T)$ the centroid of the trajectory T calculated on the basis of the PE, the

mitigation factor is defined as follows:

$$Mitigation(T', T'', i) = \min \left\{ \frac{\|C(T') - C_{neigh}(\mathbf{t}_i'')\|}{\|C(T') - C_{serv}(\mathbf{t}_i'')\|} \cdot \frac{1}{2}, 1 \right\} . \tag{8.23}$$

To compute the mitigation factor, the above function determines the spatial relation among $C(T)$, $C_{neigh}(\mathbf{t}_i)$, and $C_{serv}(\mathbf{t}_i)$. The value that $Mitigation(\cdot, \cdot, \cdot)$ returns is greater than or equal to 0 and less than or equal to 1. Basically, if $C_{neigh}(\mathbf{t}_i)$ is closer to $C(T)$ than to $C_{serv}(\mathbf{t}_i)$, it means that, from the cellular aggregate perspective, the fingerprint should be closer to the trajectory centroid as well. As a consequence, a mitigation factor that causes a reduction of the cost is returned. Conversely, if the point $C_{neigh}(\mathbf{t}_i)$ is further away from $C(T)$ than from $C_{serv}(\mathbf{t}_i)$, the mitigation function increases the cost introduced by the edit operation.

8.3 Experimental setting

8.3.1 Dataset

In this section, we introduce the data and the methodology employed for the experimentation. In order to evaluate the behaviour of the considered measures over different types of data, we use a dataset pertaining to a proprietary fingerprint localisation system. Data is distributed over a single day tracking devices across the entire world.

Starting from the complete dataset, we only considered fingerprint trajectories for which the GNSS ground truth, the position estimated by the fingerprinting algorithm (PE), and the cellular data are all known.

Moreover, we narrowed the dataset keeping trajectories with at least 4 points and with a sampling time (i.e., time elapsed between consecutive samples) greater than 1 minute only. Since in a high-frequency sampling scenario collection times typically range from 10 to 30 seconds [158], and we did not select such trajectories, the low-frequency nature of the sampling is granted.

Given that the goal of the experimental evaluation is to compare trajectories in order to measure their similarity, we first identified sets of trajectories, that we call areas, whose components are reasonably comparable. To generate the set of candidate areas, we grouped trajectories by means of the density-based clustering algorithm DBSCAN [78] applied to the trajectory centroid. We set the minimum distance between points to $\epsilon = 25$km and the minimum number of points in a cluster to 20. Such values turned out to be the most suitable ones for area partitioning, and were identified after several tests over a range of possible values.

Among the identified areas, we selected the three reported in Table 8.1, which are reasonably similar. As a matter of fact, the resulting areas/clusters contain more or less the same number of trajectories and, on average, a similar number of points. The sampling times also suggest that the selected areas are comparable. The main

Table 8.1: Description of the selected areas.

	# trajectories	# points in trajectories	Distance between points in trajectories (m)	Sampling time (h:m:s)
		Avg (Min / Max)	Avg (Min / Max)	Avg (Min / Max)
Area 1	45	9,51 (4 / 26)	28.020,72 (1.373,62 / 193.318,87)	01:35:49 (00:27:05 / 03:17:46)
Area 2	46	12,24 (4 / 36)	51.113,54 (18.152,84 / 130.017,76)	01:35:00 (00:24:08 / 03:12:26)
Area 3	49	8,16 (4 / 30)	5.776,25 (896,40 / 42.566,44)	01:48:23 (00:18:53 / 05:17:15)

difference lies in the distance between pairs of consecutive points, which makes it evident that the areas refer to different contexts. More specifically, trajectories in *Area 3* refer to devices that are either moving more slowly or facing more stops than those in *Area 1* and *Area 2*. A possible explanation for such a phenomenon is that, most probably, *Area 3* is an urban environment.

8.3.2 Analysis methodology

The experimental methodology is close to the one proposed by Wang et al. in [274]: we apply some transformations to the set of trajectories to measure the similarity between the original trajectories and the transformed ones. However, compared to [274], where the effectiveness of some similarity measures for the GNSS case is evaluated, our experimental analysis is different in various respects that we summarize below.

Comparison among different descriptions of trajectories. We consider trajectories at multiple levels: GNSS-based trajectories, PE-based trajectories, cellular-based trajectories, and trajectories based on mixed data.

Different kinds of data. We focus on low-sampling and heterogeneous domains. Thus, beside a variety of representations from the data perspective, the proposed framework encompasses trajectories whose semantic features differ from those dealt with in common scenarios from the literature.

Different measures and perturbations. The considered similarity measures have been illustrated in Section 8.1.2 and Section 8.2. As for perturbations, we apply a set of them to each trajectory with different intensities and, for each similarity measure, we assess how the similarity between the original trajectory and the disrupted one is affected. We considered two perturbations already used in [274], namely, (i) *point shift*, which spatially shifts a point of a given amount, and (ii) *decrease of the sampling rate*, which removes a certain amount of points from the trajectory. In addition, we devised three new perturbations tailored to the cellular data domain. The first one is the *servant swap* (perturbation (iii)) which consists in switching the cell-ID of the serving cell associated with a point with the cell-ID of a randomly

Table 8.2: Overview of parameters setting for perturbations.

Perturbation	Parameters		Affected points
Point Shift	1000m [$< \epsilon$] 2000m [$> \epsilon$]		Random 10% to 60% of T points
Decrease sampling rate	-		"
Servant swap	Servant + Random cell		"
Signal noise	Random RSSI value		"
Point Shift + Signal noise	1000m [$< \epsilon$] 2000m [$> \epsilon$]	Random RSSI value	" Independence between Point Shift and Signal noise

chosen neighbouring cell belonging to the fingerprint (notice that just the cell-IDs are swapped, not the corresponding signals). The second one is the *signal noise* (perturbation (iv)), which changes the RSSIs values of a certain number of cells appearing in a fingerprint by a certain, randomly determined, magnitude. The third one (perturbation (v)) is a combination of perturbations (i) and (iv): first, a *point shift* is executed; then, a *signal noise* is applied to the same trajectory. The settings of the parameters of the perturbation functions are summarised in Table 8.2. In the following, given the distances and the lengths involved (see Table 8.1), we are going to assume 1500m as a reasonable value for the parameter ϵ in both (spatial) EDR and LCSS.

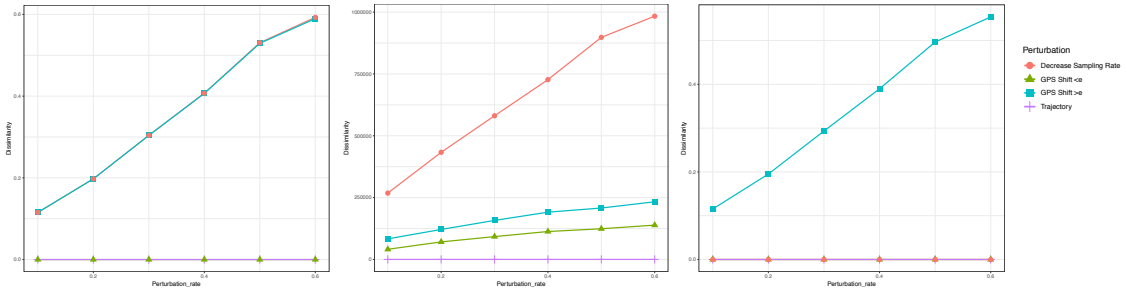
8.4 Results

As explained in Section 8.3.2, we tested the various measures by applying the perturbation functions with different intensities. In order to evaluate the behaviour of the various measures against the perturbations, we make use of a suitable graphical representation of the results.

8.4.1 Spatial, cellular, and mixed data

GNSS. As for the GNSS scenario, the point shift and decrease in the sampling rate perturbations have been applied.

As shown in Figure 8.1a (EDR measure), if the point shift magnitude is less than ϵ, there is not an increase in dissimilarity. On the contrary, if it is greater than ϵ, the comparison causes an increase in the cost by 1. In fact, the plot shows that dissimilarity increases as the number of points affected by the shift increases. The same behaviour is observable for the decrease in the sampling rate perturbation. This allows us to conclude that EDR is highly and equally sensitive to both these



(a) EDR (b) ERP (c) LCSS

Figure 8.1: Experimental results: GNSS data.

perturbations. This result partially differs from those reported in [274], where these two perturbations do not cause the same effect.

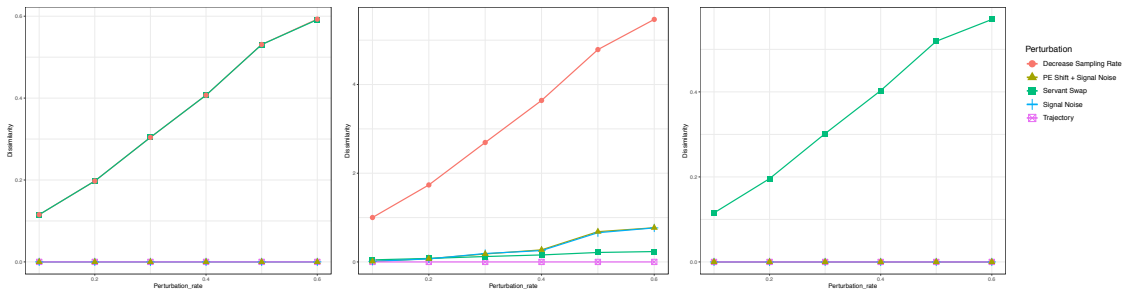
In the case of ERP (see Figure 8.1b), the measure is sensitive to the decrease in the sampling rate like EDR. As for the point shift, the growth in dissimilarity marginally depends on the number of points to which the perturbation is applied, but it is highly dependent on the amount of displacement. This behaviour is consistent with the cost definition for ERP, which is based on the real (spatial) distance between the compared points: if a point is moved by δ, the dissimilarity with respect to the original trajectory increases by δ. To get into a situation equivalent to the one we have for EDR, where the two considered perturbations have the same effect on dissimilarity, it would be necessary to set a much higher value for the point shift than the one we used, that is, considerably greater than 2000 meters.

As for LCSS (Figure 8.1c), the only perturbation affecting it is the point shift with values greater than ϵ. While such behaviour is easy to interpret, as LCSS is based on a threshold, the absence of effects due to a decrease in the sampling rate is less obvious. We need to keep in mind that LCSS looks for the longest common subsequence. By definition, such a subsequence may also consist of non-contiguous points. If the two trajectories are identical, except for some points which are missing in one of them, it follows that the LCSS of the two trajectories is determined by the length of the shortest one. Since this trajectory is the one with the fewest points, by applying the normalisation, we have that the dissimilarity is always equal to 0.

Point Estimation. The experimentation showed that exploiting PE leads to the same behaviour observed in the GNSS case (remember that PE representation is the same as GNSS one). Since PE and GNSS graphs are identical, we do not report them.

Cellular data. The perturbations applied to the trajectories based on cellular data (Figure 8.2) include the decrease in the sampling rate, the servant swap, the signal noise, and the combination of the signal noise with the point shift.

As for EDR, the only two perturbations increasing dissimilarity are the servant swap and the lowering in the sampling rate. These perturbations produce two curves



(a) EDR (b) ERP (c) LCSS

Figure 8.2: Experimental results: cellular data.

similar to the ones generated by the point shift (greater than ϵ) and the decrease in the sampling rate perturbations when applied to GNSS data (Figure 8.1a). Such a similarity suggests the existence of a semantic link between the considered GNSS and cellular perturbations. In addition, it leads us to conclude that if the compared trajectories are similar to one another, EDR, regardless of the type of data used, captures that similarity. Thus, the issue becomes whether available data are enough in order to effectively discriminate between different trajectories with respect to the application domain.

As for ERP, we observe that there are several detectable perturbations. The decrease in the sampling rate is responsible for the highest gain in dissimilarity. With a far lower impact, we can identify two perturbations whose effect is very similar. They mimic signal noise to measure input data. While one of them consists of signal noise only, the other one is a combination of signal noise and point shift. Clearly, point shift does not cause any effect in this case and the slight difference between the two curves in Figure 8.2b depends barely on the random component from the signal noise perturbation. The perturbation with the lowest effect is the servant swap. Its influence strongly depends on the usage of NCD to compute ERP costs. Since NCD does not discriminate between serving and neighbouring cells, the dissimilarity gain caused by the servant swap is determined solely by the fact that the signal strengths are now associated with different cell-IDs (NCD matches cell-IDs and looks at their RSSI differences).

As for LCSS, we observe that the effect produced by the servant swap corresponds to a point shift greater than ϵ in the case of GNSS-based LCSS. It follows that the behaviour of the two LCSSs is roughly the same and, therefore, the measure is sensitive only to the servant swap like the GNSS case is sensitive only to a point shift that exceeds ϵ value.

Mixed data. Figures 8.3 and 8.4 report the results of the experimentation with SEDF, the only measure exploiting all the available data (of different types). In this case, all the proposed perturbations are meaningful and thus have been applied. Figure 8.3 shows that the decrease in the sampling rate is the only perturbation

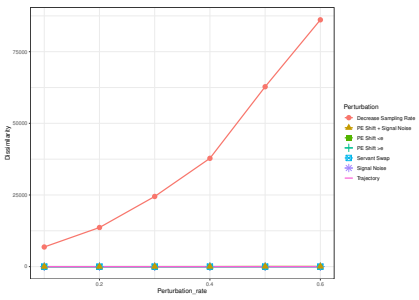


Figure 8.3: Experimental results: mixed data - SEDF.

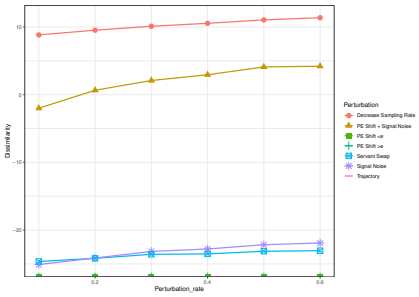


Figure 8.4: Experimental results: mixed data - log(SEDF).

causing an increase in dissimilarity. This is because, considering both cellular and spatial information, the measure can mitigate the perturbations acting only on a specific data type. As an example, if point shift is applied, its contribution to dissimilarity is attenuated by the fact that, at the level of cellular data, the two trajectories are identical. Nevertheless, the perturbation that combines point shift and signal noise should be identifiable in Figure 8.3. As this is not the case, the logarithmic version of SEDF has been investigated (Figure 8.4). It allows us to appreciate other variations in dissimilarity, even if all are lower than the one linked to a decrease in the sampling rate. In particular, while servant swap and signal noise are negligible, the combination of point shift and signal noise becomes visible. This result highlights the capability of SEDF to recognize such variations and, therefore, to detect small differences between trajectories from the considered data.

8.4.2 Robust, fair, and sensitive

In the following, we analyse the outcomes of the experimentation by establishing how robust a similarity measure is with respect to the considered perturbations. For each perturbation, we classify the behaviour of any similarity measure as robust, fair, or sensitive (as a matter of fact, these categories are close to those used in [274]). We say that a measure is robust with respect to a perturbation if the application of the latter does not cause a sharp increase in dissimilarity. We say that it is fair if the application of the perturbation causes an increase in dissimilarity, but lower than

Table 8.3: Outcomes of the experimentation in terms of robustness, fairness, and sensitiveness for GNSS and PE data.

	EDR_{GNSS}	ERP_{GNSS}	$LCSS_{GNSS}$	EDR_{PE}	ERP_{PE}	$LCSS_{PE}$
Point Shift$_{<\epsilon}$	Robust	Fair	Robust	Robust	Fair	Robust
Point Shift$_{>\epsilon}$	Sensitive	Fair	Sensitive	Sensitive	Fair	Sensitive
Decrease sampling rate	Sensitive	Sensitive	Robust	Sensitive	Sensitive	Robust
Noise	Robust	Sensitive	Robust	Robust	Sensitive	Robust

Table 8.4: Outcomes of the experimentation in terms of robustness, fairness, and sensitiveness for cellular and mixed data.

	EDR_{cell}	ERP_{cell}	$LCSS_{cell}$	$SEDF$
PE Shift$_{<\epsilon}$	Robust	Robust	Robust	Robust
PE Shift$_{>\epsilon}$	Robust	Robust	Robust	Robust
Servant swap	Sensitive	Fair	Sensitive	Fair
Signal Noise	Robust	Fair	Robust	Fair
PE Shift + Signal Noise	Robust	Fair	Robust	Fair
Decrease sampling rate	Sensitive	Sensitive	Robust	Sensitive

the one produced by other perturbations, or if such an increase, given the nature of the considered measure, is reasonable/acceptable. Finally, we say that a measure is sensitive to a perturbation if the application of the latter causes a very large increase in dissimilarity.

Results are reported in Tables 8.3 and 8.4. In Table 8.3 (GNSS and PE), we added a row for *noise*. It is related to point shift, and it says whether or not a given measure is sensitive to small variations (noise) in the spatial data.

Establishing what is good and what is evil is not immediate. Clearly, a measure which is sensitive to all the perturbations is not suitable for any domain or usage. In a low-sampling and heterogeneous scenario, however, we are not looking for the most robust measure, but rather a balanced one. In [274], robustness to a perturbation is interpreted as the ability of a measure to recognise as similar trajectories that have been perturbed with that precise function. According to such an interpretation, the best possible measure is the one that always recognises as similar trajectories that have been perturbed (all-robust measure). On the contrary, we interpret robustness as insensitivity to (certain) variations in the data. In order to assess the similarity of trajectories, the considered measures look for these variations, and thus if a measure is blind to any variation in the data, it may classify as similar trajectories which basically are not.

In view of the above considerations and the experimental results, we came to

the conclusion that in the considered framework (heterogeneous and low-sampling) similarity is better assessed by measures whose cost is not unitary. This is because ERP and SEDF, both exploiting penalties proportional to spatial and/or cellular distance, are the most balanced measures in terms of robustness, fairness, and sensitiveness.

Both ERP and SEDF are sensitive to the decrease in the sampling rate. This means that the comparison of trajectories consisting of a quite different number of points remains a difficult task. As for ERP, with GNSS and PE data, it suffers from noise sensitivity. While in other domains this can be a critical issue (given the typical scale of error that may affect both GNSS and PE observations), the impact of noise sensitivity in the low-sampling domain is limited. Having said that, when very similar trajectories are compared, sensitivity to noise may lead to an erroneous similarity evaluation. Mixing available data can be a possible way out. Combining spatial and cellular information, as SEDF does, may allow one to reduce the noise without losing the ability to correctly identify trajectories dissimilarity. The experimental outcomes seem to confirm such an expectation.

8.4.3 Correlation analysis

We performed a correlation analysis to compare the results given by the similarity measures applied to real-world trajectories and to understand whether they behave the same or not.

Given a trajectory, we evaluated its similarity with all the other trajectories, considering all the similarity measures. Then, for every possible pair of similarity measures, we estimated the rank correlation score, which assesses how much the rank provided by a measure is similar to the rank resulting from another measure (the same ranking leads to a score equal to 1). Finally, we repeated the process for every single trajectory, averaging the ranking scores measure-wise. The ultimate average correlation matrix is reported in Table 8.5.

The rank correlation matrix shows some relevant features:

- a strong correlation between the same measure applied to GNSS and PE points. This suggests that fingerprinting position estimations are suitable information for similarity evaluation, although the uncertainty introduced by the algorithm has an impact, as the correlation with GNSS is not perfect;

- for all the different kinds of data taken into consideration, there is a strong correlation between EDR and LCSS;

- as for the mixed data, SEDF is highly correlated with all ERPs; moreover, looking at the correlation between SEDF and LCSSs, and between SEDF and EDRs, we observe that, for GNSS and PE, those correlations are higher than the ones with the corresponding ERPs. This, as expected, suggests that SEDF

Table 8.5: Analysis of rank correlation.

		GNSS			PE			Cell			Mix
		EDR	ERP	LCSS	EDR	ERP	LCSS	EDR	ERP	LCSS	SEDF
GNSS	EDR	1	.284	.815	.893	.285	.737	.664	.362	.579	.466
	ERP	.284	1	.124	.270	.980	.119	.198	.668	.069	.723
	LCSS	.815	.124	1	.747	.125	.888	.580	.215	.676	.302
PE	EDR	.893	.270	.747	1	.271	.809	.652	.347	.570	.449
	ERP	.285	.980	.125	.271	1	.118	.200	.666	.071	.723
	LCSS	.737	.119	.888	.809	.118	1	.570	.205	.665	.292
Cell	EDR	.664	.198	.580	.652	.200	.570	1	.340	.814	.262
	ERP	.362	.668	.215	.347	.666	.205	.340	1	.226	.653
	LCSS	.579	.069	.676	.570	.071	.665	.814	.226	1	.158
Mix	SEDF	.466	.723	.302	.449	.723	.292	.262	.653	.158	1

has a different behaviour than the other metrics, but still partly aligned and not unreasonable;

- low or absence of a significant correlation between ERP and both EDR and LCSS; this suggests that ERP captures variations different from those detected by the other two measures (notice that this is completely independent of the type of considered data);
- there is a fairly good correlation between EDR, ERP, and LCSS with cellular data and the corresponding measures with both GNSS and PE data, confirming the overall value of cellular information.

Since the above observations have been drawn on the basis of the average rank correlation matrix, to check whether they can be generalised and eventually observe new phenomena, we extended the analysis to every single area. As a matter of fact, the outcomes of the analysis of the average correlation matrix are basically the same outcomes we obtained by restricting attention to single areas. Two new elements emerged from the area-level analysis.

First, in the case of GNSS and PE points, the correlation of SEDF with ERPs turns out to be lower than the one suggested by the values spotted in the average case (Area 3). Moreover, when this happens, an analogous reduction in the correlation with the other measures does not occur; on the contrary, correlation with EDRs and LCSSs tends to increase. This suggests that the measure, or, more generally, the mixed data approach, has a flexible behaviour. This observation lets us emphasise that a preferable approach to the development of similarity measures is to make them dynamic with respect to the context of the application, possibly considering mixtures of data, and exploiting the best type as needed.

The second remark is about the relationships among different areas. While *Area 1* and *Area 2* roughly have the same magnitude of correlations among the measures,

Area 3 significantly diverges. This is most probably related to the population density of the involved area, suggesting, again, that such a feature should be taken into account by a similarity measure.

8.5 Discussion

Trajectories and their similarity evaluation are important notions widely exploited in positioning. Their management involves several tasks ranging from prediction-making to mining user behaviour. The problem of evaluating the similarity of trajectories has been largely studied for sequences of GNSS points, while few works have been done on sequences of fingerprint observations.

In this chapter, we focused on the similarity evaluation of fingerprint-based trajectories, with the main objective of determining which is the most suitable representation to evaluate the similarity in our setting, and what pros and cons are brought by each of them. To such an extent, we conducted an analysis on real data, investigating GNSS, point estimation, cellular, and mixed representation of the trajectories' points. Starting with the classical metrics used for the pure spatial case, i.e., EDR, ERP, and LCSS, and given the lack of measures capable to deal with fingerprint data (cellular and mixed), we devised a similarity measure (SEDF) and a cost/distance function (NCD), both tailored to our domain. We discovered that similarity measures that apply a cost proportional to the estimated spatial and/or cellular distance, such as ERP and SEDF, are more suitable for the fingerprinting context. Moreover, we also observed that PE is appropriate for similarity evaluation, although the performance may largely depend on the accuracy of the positioning algorithms, which is influenced by multiple, unpredictable factors. This remark suggests how devising similarity measures based on mixed data is crucial for fingerprinting and domains alike, as they are less affected by errors occurring on only one of the data types they use. Indeed, SEDF is a first proof of concept in this direction.

As a final remark, note that outdoor fingerprint trajectories are not a primitive concept, and, as we saw, they are quite different from the GNSS case, as they often describe a low-sampling and heterogeneous context. Even if we focused on outdoor trajectories, similar characteristics and similar problems appear in the indoor scenario. For instance, depending on the application, WiFi data on an Android smartphone could be collected each time every few minutes (i.e., low sampling). As a matter of fact, we analysed trajectories for fingerprinting as they may play a crucial role in developing seamless indoor-outdoor localization, especially concerning modeling and supporting the transition of a device from an indoor to an outdoor environment and vice versa. Given the finding of this chapter, we have now a better view of how to approach the problem, such as, for instance, considering multiple (spatial, cellular, WiFi) types of data jointly.

Conclusions

In this dissertation, we presented several studies about fingerprint-based localization, devoting particular attention to the indoor context.

We began our journey tackling the problem of modelling indoor positioning. To such an extent, we presented a comprehensive, yet general and extensible framework to support indoor positioning tasks, whose practical implementation has been a relational database. Its main characteristic is the ability to represent topological information of indoor premises, which can be seamlessly combined with fingerprint positioning data. The flexibility of the system makes it capable of accommodating heterogeneous indoor scenarios and supporting a large number of tasks, both concerning its industrial deployment, as well as considering its usage within the research community.

Then, we turned to investigate whether classical metrics used to measure distances among fingerprints incorporated information about the spatial domain, specifically considering the distances among the locations associated with the fingerprints. As we pointed out during the discussion, such a problem is different and more fundamental than indoor positioning. Notably, having a metric that allows one to reason in spatial terms just by looking at fingerprints would significantly help in multiple tasks, ranging from improving accuracy to reducing the effort to build and maintain the radio map. We carried out an extensive analysis involving several datasets, normalization functions, metrics, and granularity levels. The outcome was that the metrics exhibit very different behaviours and that some of them, e.g., cosine, show better properties than others, although all are far from the optimal performance concerning our main goal. Moved by such findings, we developed a novel fingerprint distance function based on genetic programming, maximizing its capability of preserving real-world spatial information, and addressing the criticalities shown by classical approaches. The overall idea has been that of relying on machine learning to combine existing fingerprint distance functions into a meta-distance, solving a symbolic regression task. Through an extensive evaluation, we demonstrated an improvement over some other popular machine learning techniques, as well as over the single measures taken in isolation, concerning both spatial information preservation and accuracy of the positioning task. In addition, the genetically evolved metric clearly exhibited generalization capabilities: the improvement over the baselines has been achieved by training only a well-known benchmark dataset and testing, without retraining, on 15 other independent ones. The take-home messages from this line of research are two. First, learning a meta-metric seems to be a promising direction towards developing a universal metric, which can be applied to different settings just as it is while offering better properties than classical distance functions.

Second, although the function has been learned and optimized according to a proxy correlation metric, it also demonstrated a high positioning accuracy. This suggests a clear link between the two tasks, which was confirmed by subsequent experiments. Based on this latter finding, we started investigating an original usage of deep metric learning for continuous similarity, with the goal of learning an effective metric starting directly from the fingerprints. There is still a long and winding road ahead but, as witnessed by the preliminary results, this new approach is quite promising.

Next, our work focused on exploiting deep learning to reduce the impact of some issues affecting fingerprinting. The cornerstone idea has been employing a ranking-based representation of the fingerprints paired with recurrent neural networks. Rankings have properties, such as invariance to scale and bias, associated with a higher capability to deal with signal perturbations. However, when they are used in positioning tasks, their performances have always proved to be worse than full-fledged fingerprints, mainly due to their reduced information content. By leveraging deep learning, we showed how this downside can be considerably reduced, resulting in an approach that is robust to perturbations while being in line with, and in some cases more accurate than, several state-of-the-art solutions. On top of that, given the architecture of our framework, we investigated whether the latter could be used to extract new knowledge and operational insights about indoor positioning and related tasks. To achieve that, we relied on the attention mechanism, as a means to obtain interpretability. After defining domain-related local and global notions of interpretability, we showed that indeed our model produced interpretations aligned with them and that they were conveying specific information about the indoor positioning tasks. Overall, by showing that deep learning can be employed to extract new domain knowledge, we believe we encouraged some broader investigation, going beyond its mere use to improve positioning accuracy.

In the last part of the dissertation, we moved the first steps towards developing seamless indoor-outdoor positioning systems based on fingerprinting. Indeed, this is a long-term research goal, and here we focused on just two (among many) aspects that we deemed to be fundamental to its achievement. As we had done for indoor positioning, we started with modelling the outdoor scenario. Outdoor fingerprinting heavily relies on a comprehensive and accurate knowledge of the cellular network configuration, and, in this respect, we showed how positioning data as well as the cellular network and its evolution can be modelled through a spatio-temporal database. In doing so, the main difficulty was that the necessary network information was not publicly available, demanding its reconstruction from crowdsourced observations, a unique capability of our system. Finally, we turned our focus to the topic of trajectories, specifically their similarity evaluation, studying which was the most suitable representation for them in the fingerprinting case. Through our investigation, we showed that, despite the challenges characterizing the domain (e.g., low-rate and heterogeneous sampling), some peculiarities of fingerprinting, such as the availability of mixed types of data (e.g., spatial and cellular), make it feasible to fully exploit this non-primitive concept. We believe that this result fosters

the possibility of exploiting trajectories to model transitions between different contexts, notably regarding people and devices moving from an outdoor to an indoor environment and vice versa.

Overall, in this thesis, we looked at fingerprint-based positioning from a much broader perspective than merely proposing some new algorithms for position estimation. As a matter of fact, we took on multiple heterogeneous challenges relying on methods typical of data science, such as data modelling, analysis, and machine learning. We know that, in doing so, we have opened and left open several doors; but this journey, after all, has only just begun.

A

Entity-Relationship diagram notation

Here we describe, by means of a series of examples, a surrogate of notation employed in the Entity-Relationship diagram of Chapter 2 and Chapter 7. Figure A.1 depicts a strong entity set named *Person*, that has a primary key composed of the attributes *Name* and *Surname*. Each entity of *Person* may have at most one *Email* address, and one or more *Phone* numbers. In addition, it always has a *Birthdate*, based on which the value for the derived attribute *Age* is established.

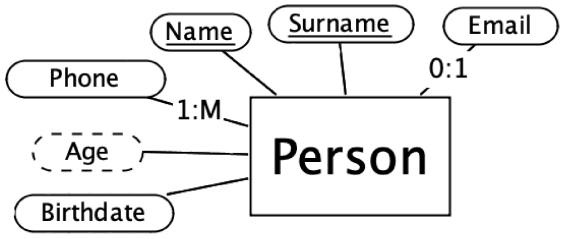


Figure A.1: Strong entity set notation.

Figure A.2 reports the case of a weak entity set, named *Song*, that has the attribute *Title* as its partial identifier. Its identifying relationship is *Belongs to*, thus, the title of a song is unique within a given album. The entity set *Album* has *Name* as its primary key. Each album contains one or more songs, and a song belongs to one and only one album (the constraint $1 : 1$ is assumed by default by our notation, and thus has been omitted on the *Song* side of the relationship).

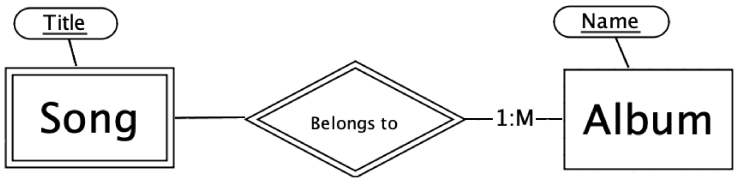


Figure A.2: Weak entity set notation.

Figure A.3 shows the notation for a total and disjoint specialization. Each entity of entity set *Professor* is uniquely identified by its *SSN*, and it corresponds to either a *Full* or an *Associate* professor.

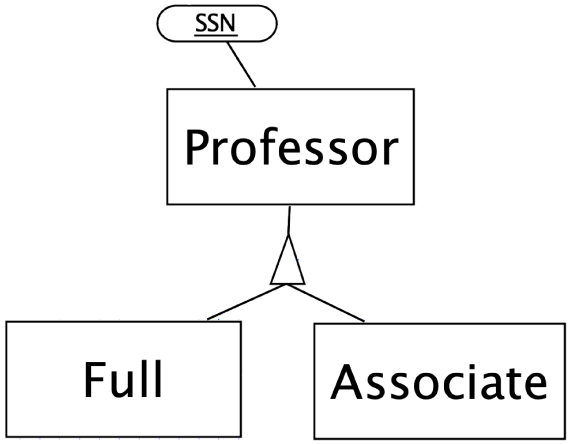


Figure A.3: Total specialization notation.

Finally, Figure A.4 represents the case of a partial specialization. Here, an entity of entity set *Employee*, uniquely identified by its *SSN*, can also be an entity of entity set *Supervisor*. This is quite natural, since supervisors are themselves employees, but not all employees are supervisors.

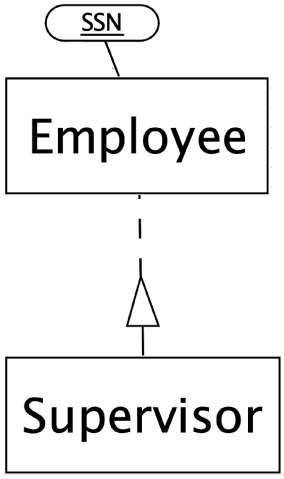


Figure A.4: Partial specialization notation.

B

Notes on the Indoor Database usage

In the following, for illustrative purposes, we report some simple SQL queries that can be used to extract relevant information from the relational database for indoor positioning, described in Chapter 2. For each query, we also report the number of returned rows and the average running time in milliseconds when executed against the working demo of the system [38]. The server running the demo is hosted on a virtual machine equipped with 4 dedicated cores (Intel(R) Xeon(R) CPU X5550 running at 2.67 GHz) and 20 GB main memory.

Query 1

It extracts topological information regarding the dataset UJIIndoorLoc [256]. Specifically, for each building, it retrieves its structuring into floors and sites [905 rows, 180 msec].

```
SELECT
    building_place.name AS building_name,
    floor_place.name AS floor_name,
    site_place.name AS site_name
FROM place AS building_place
        JOIN place_data_source ON
        place_data_source.place_id = building_place.id
        JOIN data_source ON
        data_source.id = place_data_source.data_source_id
        JOIN building ON building_place.id = building.place_id
        JOIN contains AS cts_floor ON
        cts_floor.container_place_id = building_place.id
        JOIN contains AS cts_site ON
        cts_site.container_place_id = cts_floor.contained_place_id
        JOIN place AS floor_place ON
        floor_place.id = cts_floor.contained_place_id
        JOIN place AS site_place ON
        site_place.id = cts_site.contained_place_id
WHERE data_source.name = 'UJI1'
ORDER BY building_place.name, floor_place.name, site_place.name;
```

Query 2

It extracts the *id* of all the tiles that are (directly or indirectly) reachable from the tile with *id = 524465*. Note that, in order to perform such an "unlimited" visit of the graph, we need to rely on a recursive strategy [110 rows, 140 msec].

```
WITH RECURSIVE reachable AS (
    SELECT
        adjacent_to_tile.tile_2_place_id
    FROM adjacent_to_tile
    WHERE walkable AND tile_1_place_id = 524465
  UNION
    SELECT
        succ.tile_2_place_id
    FROM reachable AS prev
            JOIN adjacent_to_tile AS succ ON
            succ.tile_1_place_id = prev.tile_2_place_id AND
            succ.walkable
)
SELECT tile_2_place_id AS reachable_tile_id
FROM reachable;
```

Query 3

It extracts the WiFi portion of the fingerprint with *id = 520857*. Observe that, since in the database only information pertaining to the detected access points is stored, in order to recover the full WiFi fingerprint (with respect to all access points in a *data_source*) outer join operations are necessary [544 rows, 245 msec].

```
SELECT
    ap.id,
    COALESCE(ap_detection.rss, -110) AS rss
FROM fingerprint
        JOIN observation_wifi ON
        observation_wifi.fingerprint_id = fingerprint.id AND
        fingerprint.id = 520857
        JOIN ap_detection ON
            ap_detection.observation_wifi_fingerprint_id =
                observation_wifi.fingerprint_id
        RIGHT OUTER JOIN ap ON ap_detection.ap_id = ap.id
        RIGHT OUTER JOIN ap_data_source ON ap.id = ap_data_source.ap_id
        RIGHT OUTER JOIN data_source ON
        ap_data_source.data_source_id = data_source.id AND
        data_source.name = 'UJI1'
ORDER BY ap.id;
```

Query 4

Given the (WiFi) fingerprint with *id = 533530*, the code extracts all sites containing fingerprints that have at least one access point in common with it, together with

the number of such fingerprints (a *zone* or *logical* tessellation is assumed) [545 rows, 10 sec].

```
SELECT
    contains.container_place_id AS site_id,
    COUNT(DISTINCT fp_2.id) AS num_fingerprints
FROM fingerprint AS fp_1
        JOIN ap_detection AS ap_detection_1 ON
        ap_detection_1.observation_wifi_fingerprint_id = fp_1.id
        JOIN ap_detection AS ap_detection_2 ON
        ap_detection_1.ap_id = ap_detection_2.ap_id
        JOIN fingerprint AS fp_2 ON
        fp_2.id = ap_detection_2.observation_wifi_fingerprint_id
        JOIN contains ON
        contains.contained_place_id = fp_2.acquired_at_tile_place_id
        JOIN site ON contains.container_place_id = site.place_id
WHERE
    fp_1.id = 533530 AND fp_2.is_radio_map AND
    fp_1.id != fp_2.id
GROUP BY contains.container_place_id
ORDER BY num_fingerprints DESC;
```

B.1 Usage of the online demo of the system

At the address `http://158.110.145.70:5050/` a demo of the system can be accessed. Upon connection, users will find a *pgAdmin* web server interface, asking for the login data. A read-only user, that has the privileges to perform `SELECT` operations over the *public* and *evaluation_support* schemas of the database named *Open_Fingerprinting* has been provided, with the following credentials:

$$\begin{aligned} &username = \text{tester@indoor.uniud.it} \ , \\ &password = \text{tSUD22\$Indo0r} \ . \end{aligned}$$

The database comes already populated with information originating from several datasets[1]. Moreover, some user-defined functions aimed at easing the interaction with the system have been implemented[2] as well as various examples of queries on the database[3].

[1] `https://github.com/dslab-uniud/Database-indoor/tree/main/Datasets`

[2] `https://github.com/dslab-uniud/Database-indoor/tree/main/Database\#implemented-user-defined-functions`

[3] `https://github.com/dslab-uniud/Database-indoor/blob/main/Database/exemplary_SQL.sql`

Bibliography

[1] Ahmed K Abed and Ikhlas Abdel-Qader. Access point selection using particle swarm optimization in indoor positioning systems. In *Proceedings of the 2018 IEEE National Conference on Aerospace and Electronics (NAECON)*, pages 403–410. IEEE, 2018.

[2] Imad Afyouni, Cyril Ray, and Christophe Claramunt. Spatial models for context-aware indoor navigation systems: A survey. *Journal of Spatial Information Science*, 1(4):85–123, June 2012.

[3] Milad Afzalan and Farrokh Jazizadeh. Indoor Positioning Based on Visible Light Communication: A Performance-based Survey of Real-world Prototypes. *ACM Computing Surveys*, 52(2):1–36, March 2020.

[4] Charu C. Aggarwal, Alexander Hinneburg, and Daniel A. Keim. On the Surprising Behavior of Distance Metrics in High Dimensional Space. In *Proceedings of the 8th International Conference International Conference on Database Theory (ICDT)*, volume 1973, pages 420–434. Springer, 2001.

[5] Beenish A. Akram, Ali H. Akbar, and Omair Shafiq. HybLoc: Hybrid Indoor Wi-Fi Localization Using Soft Clustering-Based Random Decision Forest Ensembles. *IEEE Access*, 6:38251–38272, 2018.

[6] Douglas Alves Peixoto, Han Su, Hung Nguyen Quoc Viet, Bela Stantic, Bolong Zheng, and Xiaofang Zhou. Concept for Evaluation of Techniques for Trajectory Distance Measures. In *Proceedings of the 19th IEEE International Conference on Mobile Data Management (MDM)*, pages 276–277. IEEE, June 2018.

[7] Md Sayedul Aman, Haowen Jiang, Cuyler Quint, Kumar Yelamarthi, and Ahmed Abdelgawad. Reliability evaluation of iBeacon for micro-localization. In *Proceedings of the 7th IEEE Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON)*, pages 1–5. IEEE, October 2016.

[8] Marco Ancona, Enea Ceolini, Cengiz Öztireli, and Markus Gross. Gradient-based attribution methods. *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*, 11700:169–191, 2019.

[9] Angelo Andreussi. Temporal extension of a database for the support of a fingerprint positioning system, 2022. Master thesis, Università degli Studi di Udine.

[10] Mohammad O. A. Aqel, Mohammad H. Marhaban, M. Iqbal Saripan, and Napsiah Bt. Ismail. Review of visual odometry: types, approaches, challenges, and applications. *SpringerPlus*, 5(1):1897, December 2016.

[11] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural Machine Translation by Jointly Learning to Align and Translate. *Proceedings of the 3rd International Conference on Learning Representations (ICLR)*, abs/1409.0473:15, 2014.

[12] Paramvir Bahl and Venkata N. Padmanabhan. RADAR: an in-building RF-based user location and tracking system. In *Proceedings of the 19th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, volume 2, pages 775–784. IEEE Comput. Soc., 2000.

[13] Siqi Bai, Yong-Jie Luo, Mingjiang Yan, and Qun Wan. Distance metric learning for radio fingerprinting localization. *Expert Systems With Applications*, 163:113747, 2021.

[14] Siqi Bai, Mingjiang Yan, Qun Wan, Long He, Xinrui Wang, and Junlin Li. DL-RNN: An accurate indoor localization method via double RNNs. *IEEE Sensors Journal*, 20(1):286–295, 2019.

[15] Karolina Baras, Adriano J. C. Moreira, and Filipe Meneses. Navigation based on symbolic space models. In Rainer Mautz, Melanie Kunz, and Hilmar Ingensand, editors, *Proceedings of the 2010 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pages 1–5. IEEE, 2010.

[16] Paolo Barsocchi, Stefano Chessa, Francesco Furfari, and Francesco Potorti. Evaluating Ambient Assisted Living Solutions: The Localization Competition. *IEEE Pervasive Computing*, 12(4):72–79, October 2013.

[17] Paolo Barsocchi, Stefano Lenzi, Stefano Chessa, and Gaetano Giunta. A novel approach to indoor RSSI localization by automatic calibration of the wireless propagation model. In *Proceedings of the 69th IEEE Vehicular Technology Conference (VTC)*. IEEE, 2009.

[18] Peter Bartlett, Yoav Freund, Wee Sun Lee, and Robert E. Schapire. Boosting the margin: a new explanation for the effectiveness of voting methods. *The Annals of Statistics*, 26(5):1651–1686, October 1998.

[19] Anahid Basiri, Elena Simona Lohan, Terry Moore, Adam Winstanley, Pekka Peltola, Chris Hill, Pouria Amirian, and Pedro Figueiredo e Silva. Indoor location based services challenges, requirements and usability of current solutions. *Computer Science Review*, 24:1–12, May 2017.

[20] Peter W. Battaglia, Jessica B. Hamrick, Victor Bapst, Alvaro Sanchez-Gonzalez, Vinícius Flores Zambaldi, Mateusz Malinowski, Andrea Tacchetti, David Raposo, Adam Santoro, Ryan Faulkner, Çaglar Gülçehre, H. Francis Song, Andrew J. Ballard, Justin Gilmer, George E. Dahl, Ashish Vaswani, Kelsey R. Allen, Charles Nash, Victoria Langston, Chris Dyer, Nicolas Heess, Daan Wierstra, Pushmeet Kohli, Matthew M. Botvinick, Oriol Vinyals, Yujia Li, and Razvan Pascanu. Relational inductive biases, deep learning, and graph networks. *CoRR*, abs/1806.01261, 2018.

[21] Yosra Ben Slimen, Sylvain Allio, and Julien Jacques. Anomaly Prevision in Radio Access Networks Using Functional Data Analysis. In *Proceedings of the 2017 IEEE Global Communications Conference (GLOBECOM)*, pages 1–6, December 2017.

[22] Jozef Benikovsky, Peter Brida, and Juraj Machaj. Localization in Real GSM Network with Fingerprinting Utilization. In *Proceedings of the 2nd international conference on mobile lightweight wireless systems (MOBILIGHT)*, volume 45, pages 699–709. Springer, 2010.

[23] Yoav Benjamini and Yosef Hochberg. Controlling the False Discovery Rate: A Practical and Powerful Approach to Multiple Testing. *Journal of the Royal Statistical Society: Series B (Methodological)*, 57(1):289–300, January 1995.

[24] James Bergstra, Daniel Yamins, and David Cox. Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures. In *Proceedings of the 30th International Conference on Machine Learning (ICML)*, pages 115–123, 2013.

[25] Joseph Berkson. Application to the Logistic Function to Bio-Assay. *Journal of the American Statistical Association*, 39(227):357, September 1944.

[26] Andrea Bernardini, Andrea Brunello, Gian Luigi Gigli, Angelo Montanari, and Nicola Saccomanno. AIOSA: an approach to the automatic identification of obstructive sleep apnea events based on deep learning. *Artificial Intelligence in Medicine*, 118:102133, 2021.

[27] Andrea Bernardini, Andrea Brunello, Gian Luigi Gigli, Angelo Montanari, and Nicola Saccomanno. OSASUD: A dataset of stroke unit recordings for the detection of obstructive sleep apnea syndrome. *Scientific Data*, 9(1):177, 2022.

[28] Donald J. Berndt and James Clifford. Using dynamic time warping to find patterns in time series. In *Proceedings of the 3rd International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 359–370. AAAI Press, 1994.

[29] Jean-Charles de Borda. Mémoire sur les élections au scrutin. *Histoire de l'Academie Royale des Sciences pour 1781*, 1784.

[30] Vadim Borisov, Tobias Leemann, Kathrin Sessler, Johannes Haug, Martin Pawelczyk, and Gjergji Kasneci. Deep Neural Networks and Tabular Data: A Survey. *IEEE Transactions on Neural Networks and Learning Systems*, pages 1–21, 2022.

[31] Kevin Bouchard, Ramin Ramezani, and Arash Naeim. Features based proximity localization with Bluetooth emitters. In *Proceedings of the 7th IEEE Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON)*, pages 1–5. IEEE, October 2016.

[32] Leo Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, August 1996.

[33] Leo Breiman. Random Forests. *Machine Learning*, 45(1):5–32, 2001.

[34] Ramon F. Brena, Juan Pablo García-Vázquez, Carlos E. Galván-Tejada, David Muñoz-Rodriguez, Cesar Vargas-Rosales, and James Fangmeyer. Evolution of Indoor Positioning Technologies: A Survey. *Journal of Sensors*, 2017:1–21, 2017.

[35] Andrea Brunello, Marcello Civilini, Stefano De Martin, Antonella Felice, Marinella Franchi, Lucilla Iacumin, Nicola Saccomanno, and Nicola Vitacolonna. Machine learning-assisted environmental surveillance of legionella: a retrospective observational study in Friuli-Venezia Giulia region of Italy in the period 2002–2019. *Informatics in Medicine Unlocked*, 28:100803, 2022.

[36] Andrea Brunello, Andrea Dalla Torre, Paolo Gallo, Donatella Gubiani, Angelo Montanari, and Nicola Saccomanno. Crowdsourced reconstruction of cellular networks to serve outdoor positioning: Modeling, validation and analysis. *Sensors*, 23(1):352, 2023.

[37] Andrea Brunello, Angelo Montanari, and Nicola Saccomanno. GitHub page of the relational database for indoor positioning project. `https://github.com/dslab-uniud/Database-indoor`. Accessed: May 2022.

[38] Andrea Brunello, Angelo Montanari, and Nicola Saccomanno. Live implementation of the relational database for indoor positioning. `http://158.110.145.70:5050/`. Accessed: May 2022.

[39] Andrea Brunello, Angelo Montanari, and Nicola Saccomanno. A framework for indoor positioning including building topology. *IEEE Access*, 10:114959–114974, 2022.

[40] Andrea Brunello, Angelo Montanari, and Nicola Saccomanno. A genetic programming approach to wifi fingerprint meta-distance learning. *Pervasive and Mobile Computing*, 85:101681, 2022.

[41] buildingSMART. Industry foundation classes standard. `https://www.buildingsmart.org/standards/bsi-standards/industry-foundation-classes/`. Accessed: May 2022.

[42] Vicente Cantón Paterna, Anna Calveras Augé, Josep Paradells Aspas, and María Pérez Bullones. A Bluetooth Low Energy Indoor Positioning System with Channel Diversity, Weighted Trilateration and Kalman Filtering. *Sensors*, 17(12):2927, December 2017.

[43] Yongtao Cao, Byran J. Smucker, and Timothy J. Robinson. On using the hypervolume indicator to compare Pareto fronts: Applications to multi-criteria optimal experimental design. *Journal of Statistical Planning and Inference*, 160:60–74, May 2015.

[44] Sneha Chaudhari, Varun Mithal, Gungor Polatkan, and Rohan Ramanath. An Attentive Survey of Attention Models. *ACM Transactions on Intelligent Systems and Technology*, 12(5):1–32, October 2021.

[45] Changhao Chen, Bing Wang, Chris Xiaoxuan Lu, Niki Trigoni, and Andrew Markham. A Survey on Deep Learning for Localization and Mapping: Towards the Age of Spatial Machine Intelligence. *CoRR*, abs/2006.12567:26, 2020.

[46] Lei Chen and Raymond T. Ng. On the marriage of lp-norms and edit distance. In *Proceedings of the 30th International Conference on Very Large Data Bases (VLDB)*, pages 792–803. Morgan Kaufmann, 2004.

[47] Lei Chen, M. Tamer Özsu, and Vincent Oria. Robust and fast similarity search for moving object trajectories. In *Proceedings of the 2005 ACM international conference on Management of data (SIGMOD)*, pages 491–502. ACM, June 2005.

[48] Ling Chen, Mingqi Lv, and Gencai Chen. A system for destination and future route prediction based on trajectory mining. *Pervasive and Mobile Computing*, 6(6):657–676, December 2010.

[49] Mike Y. Chen, Timothy Sohn, Dmitri Chmelev, Dirk Haehnel, Jeffrey Hightower, Jeff Hughes, Anthony LaMarca, Fred Potter, Ian Smith, and Alex Varshavsky. Practical Metropolitan-Scale Positioning for GSM Phones. In *Proceedings of the 8th International Conference on Ubiquitous Computing (UbiComp)*, volume 4206, pages 225–242. Springer, 2006.

[50] Peter Pin-Shan Chen. The entity-relationship model—toward a unified view of data. *ACM Transactions on Database Systems*, 1(1):9–36, March 1976.

[51] Tianqi Chen and Carlos Guestrin. XGBoost: A Scalable Tree Boosting System. In *Proceedings of the 22nd ACM International Conference on Knowledge Discovery and Data Mining(SIGKDD)*, pages 785–794, August 2016.

[52] Yiqiang Chen, Qiang Yang, Jie Yin, and Xiaoyong Chai. Power-efficient access-point selection for indoor location estimation. *IEEE Transactions on Knowledge and Data Engineering*, 18(7):877–888, 2006.

[53] Yen-Kai Cheng, Hsin-Jui Chou, and Ronald Y Chang. Machine-learning indoor localization with access point selection and signal strength reconstruction. In *Proceedings of the 83rd IEEE Vehicular Technology Conference (VTC)*, pages 1–5. IEEE, 2016.

[54] Yu-Chung Cheng, Yatin Chawathe, Anthony LaMarca, and John Krumm. Accuracy characterization for metropolitan-scale Wi-Fi localization. In Kang G. Shin, David Kotz, and Brian D. Noble, editors, *Proceedings of the 3rd ACM International Conference on Mobile Systems, Applications, and Services (MobiSys)*, pages 233–245. ACM, 2005.

[55] Fedor Chernogorov, Sergey Chernov, Kimmo Brigatti, and Tapani Ristaniemi. Sequence-based detection of sleeping cell failures in mobile networks. *Wireless Networks*, 22(6):2029–2048, August 2016.

[56] Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, volume abs/1406.1078, pages 1724–1734, 2014.

[57] Sumit Chopra, Raia Hadsell, and Yann LeCun. Learning a similarity metric discriminatively, with application to face verification. In *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 539–546. IEEE Computer Society, 2005.

[58] Open Geospatial Consortium. CityGML encoding standard. `https://www.ogc.org/standards/citygml`. Accessed: May 2022.

[59] Open Geospatial Consortium. IndoorGML - OGC standard for indoor spatial information. `http://www.indoorgml.net/`. Accessed: May 2022.

[60] Oracle Corporation. The java database connectivity (JDBC). `https://www.oracle.com/java/technologies/javase/javase-tech-database.html`. Accessed: May 2022.

[61] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, September 1995.

[62] David R. Cox. The Regression Analysis of Binary Sequences. *Journal of the Royal Statistical Society: Series B (Methodological)*, 20(2):215–232, July 1958.

[63] Vipul K. Dabhi and Sanjay Chaudhary. Empirical modeling using genetic programming: a survey of issues and approaches. *Natural Computing*, 14(2):303–330, June 2015.

[64] Pavel Davidson and Robert Piche. A Survey of Selected Indoor Positioning Methods for Smartphones. *IEEE Communications Surveys & Tutorials*, 19(2):1347–1370, 2017.

[65] Dimas de la Fuente, Miguel A. Vega-Rodríguez, and Carlos J. Pérez. Automatic selection of a single solution from the Pareto front to identify key players in social networks. *Knowledge-Based Systems*, 160:228–236, November 2018.

[66] Kalyanmoy Deb and Himanshu Jain. An Evolutionary Many-Objective Optimization Algorithm Using Reference-Point-Based Nondominated Sorting Approach, Part I: Solving Problems With Box Constraints. *IEEE Transactions on Evolutionary Computation*, 18(4):577–601, August 2014.

[67] Andrew G. Dempster, Binghao Li, and Ishrat Quader. Errors in deterministic wireless fingerprinting systems for localisation. In *Proceedings of the 3rd International Symposium on Wireless Pervasive Computing*, pages 111–115. IEEE, May 2008.

[68] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 4171–4186. Association for Computational Linguistics, 2019.

[69] A. A. Diakité, S. Zlatanova, A. F. M. Alattas, and K. J. Li. TOWARDS INDOORGML 2.0: UPDATES AND CASE STUDY ILLUSTRATIONS. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XLIII-B4-2020:337–344, August 2020.

[70] Indoor Air Division, Office of Atmospheric and Indoor Air Programs, Office of Air Radiation, , and Office of Research and Development prepared by US Environmental Protection Agency. Report to congress on indoor air quality: Volume 2. Assessment and control of indoor air pollution., 1989.

[71] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *Proceedings of the 9th International Conference on Learning Representations (ICLR)*. OpenReview.net, 2021.

[72] Harris Drucker. Improving regressors using boosting techniques. In *Proceedings of the 14th International Conference on Machine Learning (ICML)*, pages 107–115. Morgan Kaufmann, 1997.

[73] Marek Dvorsky, Libor Michalek, Pavel Moravec, and Roman Sebesta. Improved GSM-Based Localization by Incorporating Secondary Network Characteristics. In *Proceedings of the 2012 International Conference on Research in Networking workshops (NETWORKING)*, volume 7291, pages 139–144. Springer, 2012.

[74] Agoston E. Eiben and J. E. Smith. *Introduction to evolutionary computing*. SpringerVerlag, 2003.

[75] Samih Eisa and Adriano J. C. Moreira. Requirements and metrics for location and tracking for ambient assisted living. In *Proceedings of the 2012 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pages 1–7. IEEE, 2012.

[76] Hassan Eldeeb, M. Arafa, and Mohamed T. Faheem Saidahmed. Optimal placement of access points for indoor positioning using a genetic algorithm. In *Proceedings of the 12th International Conference on Computer Engineering and Systems (ICCES)*, pages 306–313, December 2017.

[77] Ngozi Clara Eli-Chukwu, JM Aloh, and Christopher Ogwugwuam Ezeagwu. A Systematic Review of Artificial Intelligence Applications in Cellular Networks. *Engineering, Technology & Applied Science Research*, 9(4):4504–4510, August 2019.

[78] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei 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)*, pages 226–231. AAAI Press, 1996.

[79] Massimo Craglia Fabio Ricciato, Peter Widhalm and Francesco Pantisano. *Estimating population density distribution from network-based mobile phone data*. Luxembourg: Publications Office of the European Union, 2015.

[80] Ramsey Faragher and Robert Harle. Location Fingerprinting With Bluetooth Low Energy Beacons. *IEEE Journal on Selected Areas in Communications*, 33(11):2418–2428, November 2015.

[81] Xu Feng, Khuong An Nguyen, and Zhiyuan Luo. A survey of deep learning approaches for WiFi-based indoor positioning. *Journal of Information and Telecommunication*, 6(2):163–216, April 2022.

[82] Evelyn Fix and Joseph Lawson Hodges. Discriminatory analysis. Nonparametric discrimination: Consistency properties. *International Statistical Review/Revue Internationale de Statistique*, 57(3):238–247, 1989.

[83] Anastasios Foliadis, Mario H. Castaneda Garcia, Richard A. Stirling-Gallacher, and Reiner S. Thoma. CSI-Based Localization with CNNs Exploiting Phase Information. In *Proceedings of the 2021 IEEE Wireless Communications and Networking Conference (WCNC)*, volume abs/2101.08983, pages 1–6, March 2021.

[84] Félix-Antoine Fortin, François-Michel De Rainville, Marc-André Gardner, Marc Parizeau, and Christian Gagné. DEAP: Evolutionary algorithms made easy. *Journal of Machine Learning Research*, 13:2171–2175, 2012.

[85] Paolo Di Francesco, Francesco Malandrino, and Luiz A. DaSilva. Assembling and Using a Cellular Dataset for Mobile Network Analysis and Planning. *IEEE Transactions on Big Data*, 4(4):614–620, December 2018.

[86] Jerome H. Friedman. Greedy function approximation: A gradient boosting machine. *The Annals of Statistics*, 29(5):1189–1232, October 2001.

[87] Jerome H. Friedman. Stochastic gradient boosting. *Computational Statistics & Data Analysis*, 38(4):367–378, February 2002.

[88] Rewbenio A. Frota, Guilherme A. Barreto, and Joao Mota. Anomaly detection in mobile communication networks using the self-organizing map. *Journal of Intelligent & Fuzzy Systems*, 18(5):493–500, 2007.

[89] Zheren Fu, Yan Li, Zhendong Mao, Quan Wang, and Yongdong Zhang. Deep metric learning with self-supervised ranking. In *Proceedings of the 35th AAAI Conference on Artificial Intelligence, (AAAI), 33rd Conference on Innovative Applications of Artificial Intelligence (IAAI), 11th Symposium on Educational Advances in Artificial Intelligence (EAAI)*, pages 1370–1378. AAAI Press, 2021.

[90] Francesco Furfari, Antonino Crivello, Paolo Baronti, Paolo Barsocchi, Michele Girolami, Filippo Palumbo, Darwin Quezada-Gaibor, Germán M. Mendoza Silva, and Joaquín Torres-Sospedra. Discovering location based services:

A unified approach for heterogeneous indoor localization systems. *Internet of Things*, 13:100334, March 2021.

[91] Francesco Furfari, Antonino Crivello, Paolo Barsocchi, Filippo Palumbo, and Francesco Potortì. What is next for indoor localisation? taxonomy, protocols, and patterns for advanced location based services. In Francesco Potortì, Valérie Renaudin, Kyle O'Keefe, and Filippo Palumbo, editors, *Proceedings of the 2019 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pages 1–8. IEEE, 2019.

[92] Andrea Galassi, Marco Lippi, and Paolo Torroni. Attention in Natural Language Processing. *IEEE Transactions on Neural Networks and Learning Systems*, 32(10):4291–4308, October 2021.

[93] Paolo Gallo, Donatella Gubiani, Angelo Montanari, and Nicola Saccomanno. A new similarity measure for low-sampling cellular fingerprint trajectories. In *Proceedings of the 21st IEEE International Conference on Mobile Data Management (MDM)*, pages 9–18, June 2020.

[94] Hengyi Gan, Mohd Haris Bin Md Khir, Gunawan Witjaksono Bin Djaswadi, and Nordin Ramli. A Hybrid Model Based on Constraint OSELM, Adaptive Weighted SRC and KNN for Large-Scale Indoor Localization. *IEEE Access*, 7:6971–6989, 2019.

[95] Marcus Goetz and Alexander Zipf. Extending OpenStreetMap to indoor environments: Bringing volunteered geographic information to the next level. *Urban and regional data management: UDMS annual*, 2011:47–58, 2011.

[96] Ivo Gonçalves and Sara Silva. Balancing Learning and Overfitting in Genetic Programming with Interleaved Sampling of Training Data. In *Proceedings of the 16th European Conference on Genetic Programming*, volume 7831, pages 73–84, 2013.

[97] Ian J. Goodfellow, Yoshua Bengio, and Aaron C. Courville. *Deep learning*. Adaptive computation and machine learning. MIT Press, Cambridge, USA, 2016.

[98] PostgreSQL Global Development Group. PostgreSQL. `https://www.postgresql.org/`. Accessed: May 2022.

[99] David Gualda, Jesus Urena, Juan C. Garcia, Enrique Garcia, and Daniel Ruiz. RSSI distance estimation based on Genetic Programming. In *Proceedings of the 2013 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pages 1–8. IEEE, October 2013.

[100] Donatella Gubiani, Paolo Gallo, Andrea Viel, Andrea Dalla Torre, and Angelo Montanari. A Cellular Network Database for Fingerprint Positioning Systems. In *Proceedings of the 2019 European Conference on Advances in Databases and Information Systems (ADBIS)*, volume 1064 of *Communications in computer and information science*, pages 111–119. Springer, 2019.

[101] Donatella Gubiani and Angelo Montanari. ChronoGeoGraph: An expressive spatio-temporal conceptual model. In *Proceedings of the 15th Italian Symposium on Advanced Database Systems (SEBD)*, pages 160–171, 2007.

[102] Donatella Gubiani and Angelo Montanari. A tool for the visual synthesis and the logical translation of spatio-temporal conceptual schemas. In *Proceedings of the 15th italian symposium on advanced database systems (SEBD)*, pages 495–498, 2007.

[103] Suining He and S.-H. Gary Chan. Wi-Fi Fingerprint-Based Indoor Positioning: Recent Advances and Comparisons. *IEEE Communications Surveys & Tutorials*, 18(1):466–490, 2016.

[104] Noelia Hernández, Ignacio Parra, Héctor Corrales, Rubén Izquierdo, Augusto Luis Ballardini, Carlota Salinas, and Iván García. WiFiNet: WiFi-based indoor localisation using CNNs. *Expert Systems with Applications*, 177:114906, September 2021.

[105] Minh Tu Hoang, Brosnan Yuen, Xiaodai Dong, Tao Lu, Robert Westendorp, and Kishore Reddy. Recurrent Neural Networks for Accurate RSSI Indoor Localization. *IEEE Internet of Things Journal*, 6(6):10639–10651, December 2019.

[106] Minh Tu Hoang, Yizhou Zhu, Brosnan Yuen, Tyler Reese, Xiaodai Dong, Tao Lu, Robert Westendorp, and Michael Xie. A Soft Range Limited K-Nearest Neighbors Algorithm for Indoor Localization Enhancement. *IEEE Sensors Journal*, 18(24):10208–10216, December 2018.

[107] Sepp Hochreiter and Jürgen Schmidhuber. Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780, November 1997.

[108] AKM Mahtab Hossain, Yunye Jin, Wee-Seng Soh, and Hien Nguyen Van. Ssd: A robust rf location fingerprint addressing mobile devices' heterogeneity. *IEEE Transactions on Mobile Computing*, 12(1):65–77, 2013.

[109] Harold Hotelling. Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*, 24(6):417–441, September 1933.

[110] Joseph Hoy. *Forensic radio survey for cell site analysis*. Wiley, 2013.

[111] He-Yen Hsieh, Setya Widyawan Prakosa, and Jenq-Shiou Leu. Towards the Implementation of Recurrent Neural Network Schemes for WiFi Fingerprint-Based Indoor Positioning. In *Proceedings of the 88th IEEE Vehicular Technology Conference (VTC-Fall)*, pages 1–5. IEEE, August 2018.

[112] Mai Ibrahim, Marwan Torki, and Mustafa ElNainay. CNN based Indoor Localization using RSS Time-Series. In *Proceedings of the 2018 IEEE Symposium on Computers and Communications (ISCC)*, pages 01044–01049. IEEE, June 2018.

[113] Mohamed Ibrahim and Moustafa Youssef. CellSense: A Probabilistic RSSI-Based GSM Positioning System. In *Proceedings of the 2010 IEEE Global Telecommunications Conference (GLOBECOM)*, pages 1–5. IEEE, December 2010.

[114] Fabian Isensee, Marianne Schell, Irada Pflueger, Gianluca Brugnara, David Bonekamp, Ulf Neuberger, Antje Wick, Heinz-Peter Schlemmer, Sabine Heiland, Wolfgang Wick, Martin Bendszus, Klaus H. Maier-Hein, and Philipp Kickingereder. Automated brain extraction of multisequence MRI using artificial neural networks. *Human Brain Mapping*, 40(17):4952–4964, December 2019.

[115] Anil K. Jain, M. Narasimha Murty, and Patrick J. Flynn. Data clustering: a review. *ACM Computing Surveys*, 31(3):264–323, September 1999.

[116] Sarthak Jain and Byron C. Wallace. Attention is not Explanation. In *Proceedings of the 2019 Conference of the North*, pages 3543–3556. ACL, 2019.

[117] K. Kaemarungsi and P. Krishnamurthy. Properties of indoor received signal strength for WLAN location fingerprinting. In *Proceedings of the 1st Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services (MOBIQUITOUS)*, pages 14–23. IEEE Computer Society, 2004.

[118] Kamol Kaemarungsi and Prashant Krishnamurthy. Analysis of WLAN's received signal strength indication for indoor location fingerprinting. *Pervasive and Mobile Computing*, 8(2):292–316, April 2012.

[119] Hae-Kyong Kang and Ki-Joune Li. A Standard Indoor Spatial Data Model—OGC IndoorGML and Implementation Approaches. *ISPRS International Journal of Geo-Information*, 6(4):116, April 2017.

[120] Mahmut Kaya and Hasan Sakir Bilge. Deep metric learning: A survey. *Symmetry*, 11(9):1066, 2019.

[121] Ali Khalajmehrabadi, Nikolaos Gatsis, and David Akopian. Modern WLAN Fingerprinting Indoor Positioning Methods and Deployment Challenges. *IEEE Communications Surveys & Tutorials*, 19(3):1974–2002, 2017.

[122] Kyeong Soo Kim, Sanghyuk Lee, and Kaizhu Huang. A scalable deep neural network architecture for multi-building and multi-floor indoor localization based on Wi-Fi fingerprinting. *Big Data Analytics*, 3(1):4, December 2018.

[123] Sungyeon Kim, Dongwon Kim, Minsu Cho, and Suha Kwak. Proxy anchor loss for deep metric learning. In *Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3235–3244. Computer Vision Foundation / IEEE, 2020.

[124] Sungyeon Kim, Minkyo Seo, Ivan Laptev, Minsu Cho, and Suha Kwak. Deep metric learning beyond binary supervision. In *Proceedings of the 2019 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2288–2297. Computer Vision Foundation / IEEE, 2019.

[125] Thomas King, Stephan Kopf, Thomas Haenselmann, and others. CRAWDAD dataset mannheim/compass (v. 2008-04-11). `http://crawdad.org/mannheim/compass/20080411/`, 2008.

[126] Mikkel Baun Kjærgaard. Indoor location fingerprinting with heterogeneous clients. *Pervasive and Mobile Computing*, 7(1):31–43, 2011.

[127] JohnR. Koza. Genetic programming as a means for programming computers by natural selection. *Statistics and Computing*, 4(2):87–112, June 1994.

[128] Pekka Kumpulainen and Kimmo Hätönen. Local anomaly detection for mobile network monitoring. *Information Sciences*, 178(20):3840–3859, October 2008.

[129] Azadeh Kushki, Konstantinos N Plataniotis, and Anastasios N Venetsanopoulos. Kernel-based positioning in wireless local area networks. *IEEE Transactions on Mobile Computing*, 6(6):689–705, 2007.

[130] Jaana Laiho, Kimmo Raivio, Pasi Lehtimäki, K. Hatonen, and Olli Simula. Advanced analysis methods for 3G cellular networks. *IEEE Transactions on Wireless Communications*, 4(3):930–942, May 2005.

[131] Elina Laitinen and Elena-Simona Lohan. Are all the access points necessary in wlan-based indoor positioning? In *Proceedings of the 2015 International Conference on Location and GNSS (ICL-GNSS)*, pages 1–6. IEEE, 2015.

[132] Elina Laitinen and Elena Simona Lohan. On the choice of access point selection criterion and other position estimation characteristics for wlan-based indoor positioning. *Sensors*, 16(5):737, 2016.

[133] Heikki Laitinen, Jaakko Lahteenmaki, and Tero Nordstrom. Database correlation method for GSM location. In *Proceedings of the 53rd IEEE Vehicular Technology Conference (VTS)*, volume 4, pages 2504–2508, 2001.

[134] Christos Laoudias, Panayiotis Kolios, and Christos Panayiotou. Differential signal strength fingerprinting revisited. In *Proceedings of the 2014 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pages 30–37. IEEE, 2014.

[135] Christos Laoudias, Adriano Moreira, Sunwoo Kim, Sangwoo Lee, Lauri Wirola, and Carlo Fischione. A Survey of Enabling Technologies for Network Localization, Tracking, and Navigation. *IEEE Communications Surveys & Tutorials*, 20(4):3607–3644, 2018.

[136] Christos Laoudias, Robert Piché, and Christos G. Panayiotou. Device self-calibration in location systems using signal strength histograms. *Journal of Location Based Services*, 7(3):165–181, 2013.

[137] Marius Laska and Jörg Blankenbach. DeepLocBox: Reliable Fingerprinting-Based Indoor Area Localization. *Sensors*, 21(6):2000, March 2021.

[138] Wenhua Le, Zhanbin Wang, Jingcheng Wang, Guanglei Zhao, and Haoxuan Miao. A novel WIFI indoor positioning method based on Genetic Algorithm and Twin Support Vector Regression. In *Proceedings of the 26th Chinese Control and Decision Conference (CCDC)*, pages 4859–4862, May 2014.

[139] Yann LeCun, Bernhard E. Boser, John S. Denker, Donnie Henderson, Richard E. Howard, Wayne E. Hubbard, and Lawrence D. Jackel. Backpropagation applied to handwritten zip code recognition. *Neural Computation*, 1(4):541–551, 1989.

[140] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[141] Jiyeong Lee. A Spatial Access-Oriented Implementation of a 3-D GIS Topological Data Model for Urban Entities. *GeoInformatica*, 8(3):237–264, September 2004.

[142] Soo-Hwan Lee, Won-Yeol Kim, and Dong-Hoan Seo. Automatic self-reconstruction model for radio map in Wi-Fi fingerprinting. *Expert Systems With Applications*, 192:116455, 2022.

[143] Jeff Leek. The key word in "Data Science" is not Data, it is Science. `https://simplystatistics.org/posts/2013-12-12-the-key-word-in-data-science-is-not-data-it-is-science/`. Accessed: Dec 2022.

[144] Andre L. Lehmann, Luis Otavio Alvares, and Vania Bogorny. SMSM: a similarity measure for trajectory stops and moves. *International Journal of Geographical Information Science*, 33(9):1847–1872, September 2019.

[145] Sergio Lembo, Seppo Horsmanheimo, and Petri Honkamaa. Indoor Positioning Based on RSS Fingerprinting in a LTE Network: Method Based on Genetic Algorithms. In *Proceedings of the 2019 IEEE International Conference on Communications Workshops (ICC Workshops)*, pages 1–6, May 2019.

[146] Filip Lemic, Vlado Handziski, Niklas Wirstrom, Tom Van Haute, Eli De Poorter, Thiemo Voigt, and Adam Wolisz. Web-based platform for evaluation of RF-based indoor localization algorithms. In *Proceedings of the 2015 IEEE International Conference on Communication Workshop (ICCW)*, pages 834–840. IEEE, June 2015.

[147] Vladimir I. Levenshtein. Binary codes capable of correcting deletions, insertions and reversals. *Soviet Physics Doklady*, 10:707, 1966.

[148] Bohan Li, Hao Zhou, Junxian He, Mingxuan Wang, Yiming Yang, and Lei Li. On the Sentence Embeddings from Pre-trained Language Models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, volume 2011.05864, pages 9119–9130, 2020.

[149] Hang Li, Xi Chen, Ju Wang, Di Wu, and Xue Liu. DAFI: WiFi-based Device-free Indoor Localization via Domain Adaptation. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 5(4):1–21, December 2021.

[150] Ki-Joune Li, Sisi Zlatanova, J. Torres-Sospedra, A. Perez-Navarro, Christos Laoudias, and Adriano Moreira. Survey on Indoor Map Standards and Formats. In *Proceedings of the 2019 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pages 1–8. IEEE, September 2019.

[151] Quannan Li, Yu Zheng, Xing Xie, Yukun Chen, Wenyu Liu, and Wei-Ying Ma. Mining user similarity based on location history. In *Proceedings of the 16th ACM international conference on Advances in geographic information systems (SIGSPATIAL)*, pages 1–10. ACM, November 2008.

[152] Xiangyu Li, Xiao Zhang, Kongyang Chen, and Shengzhong Feng. Measurement and analysis of energy consumption on Android smartphones. In *Proceedings of the 4th IEEE International Conference on Information Science and Technology*, pages 242–245. IEEE, April 2014.

[153] Kai Lin, Min Chen, Jing Deng, Mohammad Mehedi Hassan, and Giancarlo Fortino. Enhanced Fingerprinting and Trajectory Prediction for IoT Localization in Smart Buildings. *IEEE Transactions on Automation Science and Engineering*, 13(3):1294–1307, July 2016.

[154] Nick Littlestone and Manfred K. Warmuth. The Weighted Majority Algorithm. *Information and Computation*, 108(2):212–261, February 1994.

[155] Elena Lohan, Joaquín Torres-Sospedra, Helena Leppäkoski, Philipp Richter, Zhe Peng, and Joaquín Huerta. Wi-Fi Crowdsourced Fingerprinting Dataset for Indoor Positioning. *Data*, 2(4):32, October 2017.

[156] Elena Simona Lohan. Additional TAU datasets for Wi-Fi fingerprinting- based positioning. `https://doi.org/10.5281/zenodo.3819917`, May 2020.

[157] Ilya Loshchilov and Frank Hutter. SGDR: Stochastic Gradient Descent with Warm Restarts. In *Proceedings of the 5th International Conference on Learning Representations (ICLR)*, page 16. OpenReview.net, 2016.

[158] Yin Lou, Chengyang Zhang, Yu Zheng, Xing Xie, Wei Wang, and Yan Huang. Map-matching for low-sampling-rate GPS trajectories. In *Proceedings of the 17th ACM International Conference on Advances in Geographic Information Systems (SIGSPATIAL)*, pages 352–361. ACM, November 2009.

[159] Pablo Muñoz Luengo, Raquel Barco, E. Cruz, Ana Gómez-Andrades, Emil J. Khatib, and N. Faour. A method for identifying faulty cells using a classification tree-based UE diagnosis in LTE. *EURASIP Journal on Wireless Communications and Networking*, 2017(1):130, December 2017.

[160] Scott M. Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, editors, *Proceedings of the 31st International Conference on Neural Information Processing Systems (NeurIPS)*, pages 4765–4774, 2017.

[161] Thang Luong, Hieu Pham, and Christopher D. Manning. Effective Approaches to Attention-based Neural Machine Translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1412–1421. ACL, 2015.

[162] Mingqi Lv, Ling Chen, Yanbin Shen, and Gencai Chen. Measuring cell-id trajectory similarity for mobile phone route classification. *Knowledge-Based Systems*, 89:181–191, November 2015.

[163] Zixiang Ma, Bang Wu, and Stefan Poslad. A WiFi RSSI ranking fingerprint positioning system and its application to indoor activities of daily living recognition. *International Journal of Distributed Sensor Networks*, 15(4), 2019.

[164] Juraj Machaj, Peter Brida, and Robert Piché. Rank based fingerprinting algorithm for indoor positioning. In *Proceedings of the 2011 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pages 1–6. IEEE, September 2011.

[165] Nehal Magdy, Mahmoud A. Sakr, Tamer Mostafa, and Khaled El-Bahnasy. Review on trajectory similarity measures. In *Proceedings of the 7th IEEE International Conference on Intelligent Computing and Information Systems (ICICIS)*, pages 613–619, December 2015.

[166] Halgurd S. Maghdid, Ihsan Alshahib Lami, Kayhan Zrar Ghafoor, and Jaime Lloret Mauri. Seamless outdoors-indoors localization solutions on smartphones: Implementation and challenges. *ACM Computing Surveys*, 48(4):53:1–53:34, 2016.

[167] Nelson Marques, Filipe Meneses, and Adriano J. C. Moreira. Combining similarity functions and majority rules for multi-building, multi-floor, wifi positioning. In *Proceedings of the 2012 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pages 1–9. IEEE, 2012.

[168] Frank J. Massey. The Kolmogorov-Smirnov Test for Goodness of Fit. *Journal of the American Statistical Association*, 46(253):68–78, March 1951.

[169] Sherin Mary Mathews. Explainable artificial intelligence applications in NLP, biomedical, and malware classification: a literature review. In *Proceedings of the 2019 Computing Conference (CompCom)*, pages 1269–1292. Springer, 2019.

[170] Rainer Mautz. *Indoor positioning technologies*. PhD thesis, ETH Zurich, Department of Civil, Environmental and Geomatic Engineering, 2012.

[171] Fazeelat Mazhar, Muhammad Gufran Khan, and Benny Sällberg. Precise Indoor Positioning Using UWB: A Review of Methods, Algorithms and Implementations. *Wireless Personal Communications*, 97(3):4467–4491, December 2017.

[172] Jean Damascène Mazimpaka and Sabine Timpf. Trajectory data mining: A review of methods and applications. *Journal of Spatial Information Science*, 2016(13):61–99, December 2016.

[173] Ronaldo dos Santos Mello, Vania Bogorny, Luis Otavio Alvares, Luiz Henrique Zambom Santana, Carlos Andres Ferrero, Angelo Augusto Frozza, Geomar Andre Schreiner, and Chiara Renso. MASTER: A multiple aspect view on trajectories. *Transactions in GIS*, 23(4):tgis.12526, May 2019.

[174] German Martin Mendoza-Silva, Joaquin Torres-Sospedra, Francesco Potorti, Adriano Moreira, Stefan Knauth, Rafael Berkvens, and Joaquin Huerta. Beyond Euclidean Distance for Error Measurement in Pedestrian Indoor Location. *IEEE Transactions on Instrumentation and Measurement*, 70:1–11, 2021.

[175] Germán Mendoza-Silva, Miguel Matey-Sanz, Joaquín Torres-Sospedra, and Joaquín Huerta. BLE RSS Measurements Dataset for Research on Accurate Indoor Positioning. *Data*, 4(1):12, January 2019.

[176] Germán Mendoza-Silva, Philipp Richter, Joaquín Torres-Sospedra, Elena Lohan, and Joaquín Huerta. Long-Term WiFi Fingerprinting Dataset for Research on Robust Indoor Positioning. *Data*, 3(1):3, January 2018.

[177] Germán Martín Mendoza-Silva, Joaquín Torres-Sospedra, and Joaquín Huerta. A Meta-Review of Indoor Positioning Systems. *Sensors*, 19(20):4507, October 2019.

[178] Filipe Meneses and Adriano J. C. Moreira. Using GSM cellid positioning for place discovering. In Juan Carlos Chachques and Upkar Varshney, editors, *Proceedings of the 1st International ICST Conference on Pervasive Computing Technologies for Healthcare (PervasiveHealth)*, pages 1–8. IEEE, 2006.

[179] Tim Miller. Explanation in artificial intelligence: Insights from the social sciences. *Artificial Intelligence*, 267:1–38, February 2019.

[180] Georgy Minaev, Ari Visa, and Robert Piche. Comprehensive survey of similarity measures for ranked based location fingerprinting algorithm. In *Proceedings of the 2017 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pages 1–4. IEEE, September 2017.

[181] Tom M. Mitchell. *Machine learning*. McGraw-Hill international editions. McGraw-Hill, 1997.

[182] Vahideh Moghtadaiee and Andrew G. Dempster. WiFi fingerprinting signal strength error modeling for short distances. In *Proceedings of the 2012 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pages 1–6. IEEE, November 2012.

[183] Vahideh Moghtadaiee, Andrew G. Dempster, and Binghao Li. Accuracy indicator for fingerprinting localization systems. In *Proceedings of the 2012 IEEE/ION Position, Location and Navigation Symposium*, pages 1204–1208, April 2012.

[184] Ferenc Mogyorósi, Péter Revisnyei, Azra Pašić, Zsófia Papp, István Törös, Pál Varga, and Alija Pašić. Positioning in 5G and 6G Networks—A Survey. *Sensors*, 22(13):4757, June 2022.

[185] Akash Kumar Mohankumar, Preksha Nema, Sharan Narasimhan, Mitesh M. Khapra, Balaji Vasan Srinivasan, and Balaraman Ravindran. Towards Transparent and Explainable Attention Models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4206–4216. ACL, 2020.

[186] Raúl Montoliu, Emilio Sansano-Sansano, Joaquín Torres-Sospedra, and Oscar Belmonte. IndoorLoc platform: A public repository for comparing and evaluating indoor positioning systems. In *Proceedings of the 2017 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pages 1–8. IEEE, September 2017.

[187] Adriano Moreira, Maria Joao Nicolau, Filipe Meneses, and Antonio Costa. Wi-Fi fingerprinting in the real world - RTLS@UM at the EvAAL competition. In *Proceedings of the 2015 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pages 1–10. IEEE, October 2015.

[188] Adriano Moreira, Ivo Silva, and Joaquin Torres-Sospedra. The DSI dataset for Wi-Fi fingerprinting using mobile devices. `https://doi.org/10.5281/zenodo.3778646`, April 2020.

[189] Adriano J. C. Moreira, Ivo Silva, Filipe Meneses, Maria João Nicolau, Cristiano G. Pendão, and Joaquín Torres-Sospedra. Multiple simultaneous wi-fi measurements in fingerprinting indoor positioning. In *Proceedings of the 2017 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pages 1–8. IEEE, 2017.

[190] Filippo Mortari, Eliseo Clementini, Sisi Zlatanova, and Liu Liu. An indoor navigation model and its network extraction. *Applied Geomatics*, 11(4):413–427, December 2019.

[191] Kevin P. Murphy. *Probabilistic machine learning: Advanced topics*. MIT Press, 2023.

[192] Mohammad Nabati and Seyed Ali Ghorashi. A real-time fingerprint-based indoor positioning using deep learning and preceding states. *Expert Systems with Applications*, 213:118889, March 2023.

[193] Michał Nowicki and Jan Wietrzykowski. Low-Effort Place Recognition with WiFi Fingerprints Using Deep Learning. In *Proceedings of the 2017 international conference AUTOMATION*, volume 550 of *Advances in intelligent systems and computing*, pages 575–584. Springer, 2017.

[194] Patryk Orzechowski, William La Cava, and Jason H. Moore. Where are we now?: a large benchmark study of recent symbolic regression methods. In *Proceedings of the 2018 Genetic and Evolutionary Computation Conference (GECCO)*, pages 1183–1190, July 2018.

[195] Jeongyeup Paek, Kyu-Han Kim, Jatinder P. Singh, and Ramesh Govindan. Energy-efficient positioning for smartphones using Cell-ID sequence matching. In *Proceedings of the 9th international conference on Mobile systems, applications, and services*, pages 293–306. ACM, June 2011.

[196] Kaveh Pahlavan and Prashant Krishnamurthy. *Principles of wireless access and localization*. Wiley, 2013.

[197] Harsh Panwar, PK Gupta, Mohammad Khubeb Siddiqui, Ruben Morales-Menendez, Prakhar Bhardwaj, and Vaishnavi Singh. A deep learning and grad-CAM based color visualization approach for fast detection of COVID-19 cases using chest X-ray and CT-Scan images. *Chaos, Solitons & Fractals*, 140:110190, 2020.

[198] Apostolos N. Papadopoulos. Trajectory retrieval with latent semantic analysis. In *Proceedings of the 2008 ACM symposium on Applied computing*, pages 1089–1094. ACM, March 2008.

[199] Valter Pasku, Alessio De Angelis, Guido De Angelis, Darmindra D. Arumugam, Marco Dionigi, Paolo Carbone, Antonio Moschitta, and David S. Ricketts. Magnetic Field-Based Positioning Systems. *IEEE Communications Surveys & Tutorials*, 19(3):2003–2017, 2017.

[200] John Pavlopoulos, Prodromos Malakasiotis, and Ion Androutsopoulos. Deeper Attention to Abusive User Content Moderation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1125–1135. ACL, 2017.

[201] Karl Pearson. LIII. *On lines and planes of closest fit to systems of points in space. The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2(11):559–572, November 1901.

[202] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake VanderPlas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Edouard Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[203] Zhe Peng, Philipp Richter, Helena Leppakoski, and Elena Simona Lohan. Analysis of crowdsensed WiFi fingerprints for indoor localization. In *Proceedings of the 21st Conference of Open Innovations Association (FRUCT)*, pages 268–277. IEEE, November 2017.

[204] Lucas May Petry, Carlos Andres Ferrero, Luis Otavio Alvares, Chiara Renso, and Vania Bogorny. Towards semantic-aware multiple-aspect trajectory similarity measuring. *Transactions in GIS*, 23(5):960–975, October 2019.

[205] Riccardo Poli, William Langdon, and Nicholas Mcphee. *A field guide to genetic programming*. Lulu Enterprises, UK Ltd, January 2008.

[206] Francesco Potorti, Antonino Crivello, Filippo Palumbo, Michele Girolami, and Paolo Barsocchi. Trends in smartphone-based indoor localisation. In *Proceedings of the 2021 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pages 1–7. IEEE, November 2021.

[207] Francesco Potorti, Sangjoon Park, Antonino Crivello, Filippo Palumbo, Michele Girolami, Paolo Barsocchi, Soyeon Lee, Joaquin Torres-Sospedra, Antonio Ramon Jimenez Ruiz, Antoni Perez-Navarro, German Martin Mendoza-Silva, Fernando Seco, Miguel Ortiz, Johan Perul, Valerie Renaudin, Hyunwoong Kang, Soyoung Park, Jae Hong Lee, Chan Gook Park, Jisu Ha, Jaeseung Han, Changjun Park, Keunhye Kim, Yonghyun Lee, Seunghun Gye, Keumryeol Lee, Eunjee Kim, Jeong-Sik Choi, Yang-Seok Choi, Shilpa Talwar, Seong Yun Cho, Boaz Ben-Moshe, Alex Scherbakov, Leonid Antsfeld, Emilio Sansano-Sansano, Boris Chidlovskii, Nikolai Kronenwett, Silvia Prophet, Yael Landay, Revital Marbel, Lingxiang Zheng, Ao Peng, Zhichao Lin, Bang Wu, Chengqi Ma, Stefan Poslad, David R. Selviah, Wei Wu, Zixiang Ma, Wenchao Zhang, Dongyan Wei, Hong Yuan, Jun-Bang Jiang, Shao-Yung Huang, Jing-Wen Liu, Kuan-Wu Su, Jenq-Shiou Leu, Kazuki Nishiguchi, Walid Bousselham, Hideaki Uchiyama, Diego Thomas, Atsushi Shimada, Rin-Ichiro Taniguchi, Vicente Cortes Puschel, Tomas Lungenstrass Poulsen, Imran Ashraf, Chanseok Lee, Muhammad Usman Ali, Yeongjun Im, Gunzung Kim, Jeongsook Eom, Soojung Hur, Yongwan Park, Miroslav Opiela, Adriano Moreira, Maria Joao Nicolau, Cristiano Pendao, Ivo Silva, Filipe Meneses, Antonio Costa, Jens Trogh, David Plets, Ying-Ren Chien, Tzu-Yu Chang, Shih-Hau Fang, and Yu Tsao. The IPIN 2019 Indoor Localisation Competition—Description and Results. *IEEE Access*, 8:206674–206718, 2020.

[208] Francesco Potorti, Joaquin Torres-Sospedra, Darwin Quezada-Gaibor, Antonio Ramon Jimenez, Fernando Seco, Antoni Perez-Navarro, Miguel Ortiz, Ni Zhu, Valerie Renaudin, Ryosuke Ichikari, Ryo Shimomura, Nozomu Ohta, Satsuki Nagae, Takeshi Kurata, Dongyan Wei, Xinchun Ji, Wenchao Zhang, Sebastian Kram, Maximilian Stahlke, Christopher Mutschler, Antonino Crivello, Paolo Barsocchi, Michele Girolami, Filippo Palumbo, Ruizhi Chen, Yuan Wu, Wei Li, Yue Yu, Shihao Xu, Lixiong Huang, Tao Liu, Jian Kuang, Xiaoji Niu, Takuto Yoshida, Yoshiteru Nagata, Yuto Fukushima, Nobuya Fukatani, Nozomi Hayashida, Yusuke Asai, Kenta Urano, Wenfei Ge, Nien-Ting Lee, Shih-Hau Fang, You-Cheng Jie, Shawn-Rong Young, Ying-Ren Chien, Chih-Chieh Yu, Chengqi Ma, Bang Wu, Wei Zhang, Yankun Wang, Yonglei Fan, Stefan Poslad, David R. Selviah, Weixi Wang, Hong Yuan, Yoshitomo Yonamoto, Masahiro Yamaguchi, Tomoya Kaichi, Baoding Zhou, Xu Liu, Zhining Gu, Chengjing Yang, Zhiqian Wu, Doudou Xie, Can Huang, Lingxiang Zheng, Ao Peng, Ge Jin, Qu Wang, Haiyong Luo, Hao Xiong, Linfeng Bao, Pushuo Zhang, Fang Zhao, Chia-An Yu, Chun-Hao Hung, Leonid Antsfeld,

Boris Chidlovskii, Haitao Jiang, Ming Xia, Dayu Yan, Yuhang Li, Yitong Dong, Ivo Silva, Cristiano Pendao, Filipe Meneses, Maria Joao Nicolau, Antonio Costa, Adriano Moreira, Cedric De Cock, David Plets, Miroslav Opiela, Jakub Dzama, Liqiang Zhang, Hu Li, Boxuan Chen, Yu Liu, Seanglidet Yean, Bo Zhi Lim, Wei Jie Teo, Bu Sung Lee, and Hong Lye Oh. Off-Line Evaluation of Indoor Positioning Systems in Different Scenarios: The Experiences From IPIN 2020 Competition. *IEEE Sensors Journal*, 22(6):5011–5054, March 2022.

[209] Francesco Potortì, Sangjoon Park, Antonio Jiménez Ruiz, Paolo Barsocchi, Michele Girolami, Antonino Crivello, So Lee, Jae Lim, Joaquín Torres-Sospedra, Fernando Seco, Raul Montoliu, Germán Mendoza-Silva, Maria Pérez Rubio, Cristina Losada-Gutiérrez, Felipe Espinosa, and Javier Macias-Guarasa. Comparing the Performance of Indoor Localization Systems through the EvAAL Framework. *Sensors*, 17(10):2327, October 2017.

[210] Ismini Psychoula, Andreas Gutmann, Pradip Mainali, Sharon H Lee, Paul Dunphy, and Fabien Petitcolas. Explainable machine learning for fraud detection. *Computer*, 54(10):49–59, 2021.

[211] Antoni Pérez-Navarro, Joaquín Torres-Sospedra, Raul Montoliu, Jordi Conesa, Rafael Berkvens, Giuseppe Caso, Constantinos Costa, Nicola Dorigatti, Noelia Hernández, Stefan Knauth, and others. Challenges of fingerprinting in indoor positioning and navigation. In *Geographical and fingerprinting data to create systems for indoor positioning and Indoor/Outdoor navigation*, Intelligent data-centric systems, pages 1–20. Academic Press, San Diego, USA, 2019.

[212] Xiaowei Qin, Shuang Tang, Xiaohui Chen, Dandan Miao, and Guo Wei. SQoE KQIs anomaly detection in cellular networks: Fast online detection framework with Hourglass clustering. *China Communications*, 15(10):25–37, October 2018.

[213] J. Ross Quinlan. Induction of Decision Trees. *Machine Learning*, 1(1):81–106, 1986.

[214] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. 2018.

[215] Ion Emilian Radoi, Dumitru Cirimpei, and Valentin Radu. Localization Systems Repository: A Platform for Open-source Localization Systems and Datasets. In *Proceedings of the 2019 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pages 1–8. IEEE, September 2019.

[216] Valerie Renaudin, Miguel Ortiz, Johan Perul, Joaquin Torres-Sospedra, Antonio Ramon Jimenez, Antoni Perez-Navarro, German Martin Mendoza-Silva, Fernando Seco, Yael Landau, Revital Marbel, Boaz Ben-Moshe, Xingyu

Zheng, Feng Ye, Jian Kuang, Yu Li, Xiaoji Niu, Vlad Landa, Shlomi Hacohen, Nir Shvalb, Chuanhua Lu, Hideaki Uchiyama, Diego Thomas, Atsushi Shimada, Rin-Ichiro Taniguchi, Zhenxing Ding, Feng Xu, Nikolai Kronenwett, Blagovest Vladimirov, Soyeon Lee, Eunyoung Cho, Sungwoo Jun, Changeun Lee, Sangjoon Park, Yonghyun Lee, Jehyeok Rew, Changjun Park, Hyeongyo Jeong, Jaeseung Han, Keumryeol Lee, Wenchao Zhang, Xianghong Li, Dongyan Wei, Ying Zhang, So Young Park, Chan Gook Park, Stefan Knauth, Georgios Pipelidis, Nikolaos Tsiamitros, Tomas Lungenstrass, Juan Pablo Morales, Jens Trogh, David Plets, Miroslav Opiela, Shih-Hau Fang, Yu Tsao, Ying-Ren Chien, Shi-Shen Yang, Shih-Jyun Ye, Muhammad Usman Ali, Soojung Hur, and Yongwan Park. Evaluating Indoor Positioning Systems in a Shopping Mall: The Lessons Learned From the IPIN 2018 Competition. *IEEE Access*, 7:148594–148628, 2019.

[217] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "why should i trust you?" Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD)*, pages 1135–1144. ACM, 2016.

[218] Philipp Richter, Elena Simona Lohan, and Jukka Talvitie. WLAN (WiFi) RSS database for fingerprinting positioning. `https://doi.org/10.5281/zenodo.1161525`, January 2018.

[219] Karsten Roth, Timo Milbich, Samarth Sinha, Prateek Gupta, Björn Ommer, and Joseph Paul Cohen. Revisiting training strategies and generalization performance in deep metric learning. In *Proceedings of the 37th International Conference on Machine Learning (ICML)*, volume 119 of *Proceedings of Machine Learning Research*, pages 8242–8252. PMLR, 2020.

[220] Peter J. Rousseeuw. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20:53–65, November 1987.

[221] Priya Roy and Chandreyee Chowdhury. A Survey of Machine Learning Techniques for Indoor Localization and Navigation Systems. *Journal of Intelligent & Robotic Systems*, 101(3):63, March 2021.

[222] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. Learning representations by back-propagating errors. *Nature*, 323(6088):533–536, October 1986.

[223] Nicola Saccomanno, Andrea Brunello, and Angelo Montanari. Let's Forget About Exact Signal Strength: Indoor Positioning based on Access Point Ranking and Recurrent Neural Networks. In *Proceedings of the 17th EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services (MobiQuitous)*, pages 215–224. ACM, December 2020.

[224] Nicola Saccomanno, Andrea Brunello, and Angelo Montanari. What You Sense is Not Where You are: on the Relationships Between Fingerprints and Spatial Knowledge in Indoor Positioning. *IEEE Sensors Journal*, 22(6):4951–4961, March 2022.

[225] Susheel Sagar, Lars Lundberg, Lars Skold, and Julia Sidorova. Trajectory Segmentation for a Recommendation Module of a Customer Relationship Management System. In *Proceedings of the 2017 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*, pages 1150–1155. IEEE, June 2017.

[226] Ayesha Sahar and Dongsoo Han. An LSTM-based Indoor Positioning Method Using Wi-Fi Signals. In *Proceedings of the 2nd International Conference on Vision, Image and Signal Processing*, pages 1–5. ACM, August 2018.

[227] Wilson Sakpere, Michael Adeyeye Oshin, and Nhlanhla BW Mlitwa. A State-of-the-Art Survey of Indoor Positioning and Navigation Systems and Technologies. *South African Computer Journal*, 29(3), December 2017.

[228] Arthur L. Samuel. Some Studies in Machine Learning Using the Game of Checkers. *IBM Journal of Research and Development*, 3(3):210–229, July 1959.

[229] Mahbubeh Sattarian, Javad Rezazadeh, Reza Farahbakhsh, and Alireza Bagheri. Indoor navigation systems based on data mining techniques in internet of things: a survey. *Wireless Networks*, 25(3):1385–1402, April 2019.

[230] Martin Sauter. *From GSM to LTE: An introduction to mobile networks and mobile broadband*. Wiley, 2011.

[231] Stefano Savazzi, Sanaz Kianoush, Vittorio Rampa, and Umberto Spagnolini. Cellular Data Analytics for Detection and Discrimination of Body Movements. *IEEE Access*, 6:51484–51499, 2018.

[232] Robert E. Schapire. The strength of weak learnability. *Machine Learning*, 5(2):197–227, June 1990.

[233] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 815–823. IEEE Computer Society, 2015.

[234] Bernhard Schölkopf, Alexander Smola, and Klaus-Robert Müller. Nonlinear Component Analysis as a Kernel Eigenvalue Problem. *Neural Computation*, 10(5):1299–1319, July 1998.

[235] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-CAM: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV)*, pages 618–626. IEEE Computer Society, 2017.

[236] Sofia Serrano and Noah A. Smith. Is Attention Interpretable? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2931–2951. ACL, 2019.

[237] Shafqat Ali Shad and Enhong Chen. Precise Location Acquisition of Mobility Data Using Cell-id. *CoRR*, abs/1206.6099, 2012.

[238] Wenhua Shao, Haiyong Luo, Fang Zhao, Yan Ma, Zhongliang Zhao, and Antonino Crivello. Indoor Positioning Based on Fingerprint-Image and Deep Learning. *IEEE Access*, 6:74699–74712, 2018.

[239] Jian Shen, Chen Jin, and Dengzhi Liu. A Survey on the Research of Indoor RFID Positioning System. In *Proceedings of the 2nd International Conference on Cloud Computing and Security (ICCCS)*, volume 10040 of *Lecture notes in computer science*, pages 264–274, 2016.

[240] Zheyan Shen, Jiashuo Liu, Yue He, Xingxuan Zhang, Renzhe Xu, Han Yu, and Peng Cui. Towards Out-Of-Distribution Generalization: A Survey. *CoRR*, abs/2108.13624:22, 2021.

[241] Shweta Shrestha, Jukka Talvitie, and Elena Simona Lohan. Deconvolution-based indoor localization with WLAN signals and unknown access point locations. In *Proceedings of the 2013 International Conference on Localization and GNSS (ICL-GNSS)*, pages 1–6. IEEE, June 2013.

[242] Abraham Silberschatz, Henry F. Korth, and S. Sudarshan. *Database system concepts*. McGraw-Hill, New York, USA, 7 edition, 2019.

[243] Temple F. Smith and Michael S. Waterman. Identification of common molecular subsequences. *Journal of Molecular Biology*, 147(1):195–197, March 1981.

[244] Hyun Oh Song, Yu Xiang, Stefanie Jegelka, and Silvio Savarese. Deep metric learning via lifted structured feature embedding. In *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4004–4012. IEEE Computer Society, 2016.

[245] Xudong Song, Xiaochen Fan, Chaocan Xiang, Qianwen Ye, Leyu Liu, Zumin Wang, Xiangjian He, Ning Yang, and Gengfa Fang. A Novel Convolutional Neural Network Based Indoor Localization Framework With WiFi Fingerprinting. *IEEE Access*, 7:110698–110709, 2019.

[246] Zeqi Song, Hongwei Du, Hejiao Huang, and Chuang Liu. Indoor Localization via Candidate Fingerprints and Genetic Algorithm. In *Proceedings of the 9th international conference on combinatorial optimization and applications*, volume 9486, pages 319–333. Springer-Verlag, 2015.

[247] Bedionita Soro and Chaewoo Lee. Performance Comparison of Indoor Fingerprinting Techniques Based on Artificial Neural Network. In *Proceedings of the 2018 IEEE Region 10 Conference (TENCON)*, pages 0056–0061. IEEE, October 2018.

[248] Pichanun Sukkhawatchani and Wipawee Usaha. Performance evaluation of anomaly detection in cellular core networks using self-organizing map. In *Proceedings of the 5th International Conference on Electrical Engineering/-Electronics, Computer, Telecommunications and Information Technology*, volume 1, pages 361–364, May 2008.

[249] Richard S. Sutton and Andrew G. Barto. *Reinforcement learning - an introduction*. Adaptive computation and machine learning. MIT Press, 1998.

[250] Peter Szilagyi and Szabolcs Novaczki. An Automatic Detection and Diagnosis Framework for Mobile Communication Systems. *IEEE Transactions on Network and Service Management*, 9(2):184–197, June 2012.

[251] Zain Bin Tariq, Dost Muhammad Cheema, Muhammad Zahir Kamran, and Ijaz Haider Naqvi. Non-GPS Positioning Systems: A Survey. *ACM Computing Surveys*, 50(4):1–34, July 2018.

[252] pgAdmin Development Team. pgAdmin. `https://www.pgadmin.org/`. Accessed: May 2022.

[253] Peter JG Teunissen and Oliver Montenbruck. *Springer handbook of global navigation satellite systems*, volume 10. Springer, 2017.

[254] Saideep Tiku and Sudeep Pasricha. Portloc: A portable data-driven indoor localization framework for smartphones. *IEEE Design & Test*, 36(5):18–26, 2019.

[255] Joaquin Torres-Sospedra, German M. Mendoza-Silva, Raul Montoliu, Oscar Belmonte, Fernando Benitez, and Joaquin Huerta. Ensembles of indoor positioning systems based on fingerprinting: Simplifying parameter selection and obtaining robust systems. In *Proceedings of the 2016 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pages 1–8. IEEE, October 2016.

[256] Joaquin Torres-Sospedra, Raul Montoliu, Adolfo Martinez-Uso, Joan P. Avariento, Tomas J. Arnau, Mauri Benedito-Bordonau, and Joaquin Huerta.

UJIIndoorLoc: A new multi-building and multi-floor database for WLAN fingerprint-based indoor localization problems. In *Proceedings of the 2014 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pages 261–270. IEEE, October 2014.

[257] Joaquin Torres-Sospedra, Darwin Quezada-Gaibor, German M. Mendoza-Silva, Jari Nurmi, Yevgeni Koucheryavy, and Joaquin Huerta. New Cluster Selection and Fine-grained Search for k-Means Clustering and Wi-Fi Fingerprinting. In *Proceedings of the 2020 International Conference on Localization and GNSS (ICL-GNSS)*, pages 1–6. IEEE, June 2020.

[258] Joaquin Torres-Sospedra, Ivo Silva, Lucie Klus, Darwin Quezada-Gaibor, Antonino Crivello, Paolo Barsocchi, Cristiano Pendao, Elena Simona Lohan, Jari Nurmi, and Adriano Moreira. Towards Ubiquitous Indoor Positioning: Comparing Systems across Heterogeneous Datasets. In *Proceedings of the 2021 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pages 1–8. IEEE, November 2021.

[259] Joaquín Torres-Sospedra, Joan Avariento, David Rambla, Raúl Montoliu, Sven Casteleyn, Mauri Benedito-Bordonau, Michael Gould, and Joaquín Huerta. Enhancing integrated indoor/outdoor mobility in a smart campus. *International Journal of Geographical Information Science*, 29(11):1955–1968, November 2015.

[260] Joaquín Torres-Sospedra, Antonio Jiménez, Adriano Moreira, Tomás Lungenstrass, Wei-Chung Lu, Stefan Knauth, Germán Mendoza-Silva, Fernando Seco, Antoni Pérez-Navarro, Maria Nicolau, António Costa, Filipe Meneses, Joaquín Farina, Juan Morales, Wen-Chen Lu, Ho-Ti Cheng, Shi-Shen Yang, Shih-Hau Fang, Ying-Ren Chien, and Yu Tsao. Off-Line Evaluation of Mobile-Centric Indoor Positioning Systems: The Experiences from the 2017 IPIN Competition. *Sensors*, 18(2):487, February 2018.

[261] Joaquín Torres-Sospedra, Raúl Montoliu, Sergio Trilles, Óscar Belmonte, and Joaquín Huerta. Comprehensive analysis of distance and similarity measures for Wi-Fi fingerprinting indoor positioning systems. *Expert Systems with Applications*, 42(23):9263–9278, December 2015.

[262] Joaquín Torres-Sospedra and Adriano Moreira. Analysis of Sources of Large Positioning Errors in Deterministic Fingerprinting. *Sensors*, 17(12):2736, November 2017.

[263] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of machine learning research*, 9(11), 2008.

[264] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention Is All You

Need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems (NeurIPS)*, pages 5998–6008. Curran Associates, Inc., 2017.

[265] Andrea Viel. *Methods, techniques, and algorithms for the management of cellular fingerprints in positioning systems*. PhD thesis, Università degli Studi di Udine, 2018.

[266] Andrea Viel, Paolo Gallo, Angelo Montanari, Donatella Gubiani, Andrea Dalla Torre, Federico Pittino, and Chris Marshall. Dealing with network changes in cellular fingerprint positioning systems. In *Proceedings of the 2017 International Conference on Localization and GNSS (ICL-GNSS)*, pages 1–6. IEEE, June 2017.

[267] Michail Vlachos, Marios Hadjieleftheriou, Dimitrios Gunopulos, and Eamonn Keogh. Indexing multi-dimensional time-series with support for multiple distance measures. In *Proceedings of the 9th ACM international conference on Knowledge discovery and data mining (SIGKDD)*, pages 216–225. ACM, August 2003.

[268] Michail Vlachos, George Kollios, and Dimitrios Gunopulos. Discovering similar multidimensional trajectories. In *Proceedings 18th International Conference on Data Engineering (ICDE)*, pages 673–684. IEEE, 2002.

[269] Quoc Duy Vo and Pradipta De. A Survey of Fingerprint-Based Outdoor Localization. *IEEE Communications Surveys & Tutorials*, 18(1):491–506, 2016.

[270] Robert A. Wagner and Michael J. Fischer. The String-to-String Correction Problem. *Journal of the ACM*, 21(1):168–173, January 1974.

[271] Boyuan Wang, Xuelin Liu, Baoguo Yu, Ruicai Jia, and Xingli Gan. An Improved WiFi Positioning Method Based on Fingerprint Clustering and Signal Weighted Euclidean Distance. *Sensors*, 19(10):2300, May 2019.

[272] Feng Wang and Huaping Liu. Understanding the behaviour of contrastive loss. In *Proceedings of the 2021 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2495–2504. Computer Vision Foundation / IEEE, 2021.

[273] Hai Wang, Baoshen Guo, Shuai Wang, Tian He, and Desheng Zhang. CSMC: Cellular Signal Map Construction via Mobile Crowdsensing. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies (IMWUT)*, 5(4):1–22, December 2021.

[274] Haozhou Wang, Han Su, Kai Zheng, Shazia Wasim Sadiq, and Xiaofang Zhou. An effectiveness study on trajectory similarity measures. In *Proceedings of the*

24th Australasian Database Conference (ADC), volume 137 of *CRPIT*, pages 13–22. Australian Computer Society, 2013.

[275] Ke Wang, Jing Tian, Chu Zheng, Hong Yang, Jia Ren, Yanling Liu, Qinghua Han, and Yanbo Zhang. Interpretable prediction of 3-year all-cause mortality in patients with heart failure caused by coronary heart disease based on machine learning and SHAP. *Computers in Biology and Medicine*, 137:104813, 2021.

[276] Liping Wang, Saideep Tiku, and Sudeep Pasricha. CHISEL: Compression-Aware High-Accuracy Embedded Indoor Localization with Deep Learning. *CoRR*, abs/2107.01192:4, 2021.

[277] Xinshao Wang, Yang Hua, Elyor Kodirov, Guosheng Hu, and Neil Martin Robertson. Deep metric learning by online soft mining and class-aware attention. In *Proceedings of the 33rd AAAI Conference on Artificial Intelligence (AAAI) the 31st Innovative Applications of Artificial Intelligence Conference (IAAI), the 9th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI)*, pages 5361–5368. AAAI Press, 2019.

[278] Xinshao Wang, Yang Hua, Elyor Kodirov, and Neil M. Robertson. Ranked list loss for deep metric learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(9):5414–5429, 2022.

[279] Xun Wang, Xintong Han, Weilin Huang, Dengke Dong, and Matthew R. Scott. Multi-similarity loss with general pair weighting for deep metric learning. In *Proceedings of the 2019 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5022–5030. Computer Vision Foundation / IEEE, 2019.

[280] Xuyu Wang, Lingjun Gao, Shiwen Mao, and Santosh Pandey. CSI-based Fingerprinting for Indoor Localization: A Deep Learning Approach. *IEEE Transactions on Vehicular Technology*, 66(1):1–1, 2016.

[281] Joe H. Ward. Hierarchical Grouping to Optimize an Objective Function. *Journal of the American Statistical Association*, 58(301):236–244, March 1963.

[282] Sarah Wiegreffe and Yuval Pinter. Attention is not not Explanation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 11–20. ACL, 2019.

[283] Ronald J. Williams and David Zipser. A Learning Algorithm for Continually Running Fully Recurrent Neural Networks. *Neural Computation*, 1(2):270–280, June 1989.

[284] Chenshu Wu, Jingao Xu, Zheng Yang, Nicholas D. Lane, and Zuwei Yin. Gain Without Pain: Accurate WiFi-based Localization using Fingerprint Spatial Gradient. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies (IMWUT)*, 1(2):1–19, June 2017.

[285] Shixiong Xia, Yi Liu, Guan Yuan, Mingjun Zhu, and Zhaohui Wang. Indoor Fingerprint Positioning Based on Wi-Fi: An Overview. *ISPRS International Journal of Geo-Information*, 6(5):135, April 2017.

[286] Jiang Xiao, Zimu Zhou, Youwen Yi, and Lionel M. Ni. A Survey on Wireless Indoor Localization from the Device Perspective. *ACM Computing Surveys*, 49(2):1–31, June 2017.

[287] Xuyu Wang, Lingjun Gao, Shiwen Mao, and Santosh Pandey. DeepFi: Deep learning for indoor fingerprinting using channel state information. In *Proceedings of the 2015 IEEE Wireless Communications and Networking Conference (WCNC)*, pages 1666–1671. IEEE, March 2015.

[288] Kuldeep Yadav, Vinayak Naik, Amarjeet Singh, Pushpendra Singh, and Umesh Chandra. Low Energy and Sufficiently Accurate Localization for Non-smartphones. In *Proceedings of the 13th IEEE International Conference on Mobile Data Management (MDM)*, pages 212–221, July 2012.

[289] Jinjin Yan, Sisi Zlatanova, and Abdoulaye A. Diakité. A unified 3D space-based navigation model for seamless navigation in indoor and outdoor. *International Journal of Digital Earth*, 14(8):985–1003, 2021.

[290] Jie Yang, Alexander Varshavsky, Hongbo Liu, Yingying Chen, and Marco Gruteser. Accuracy characterization of cell tower localization. In *Proceedings of the 12th ACM international conference on Ubiquitous computing (UbiComp)*, ACM international conference proceeding series, pages 223–226. ACM, September 2010.

[291] Sungwon Yang, Pralav Dessai, Mansi Verma, and Mario Gerla. FreeLoc: Calibration-free crowdsourced indoor localization. In *Proceedings of the 2013 IEEE International Conference on Computer Communications (INFOCOM)*, pages 2481–2489. IEEE, April 2013.

[292] Ali Yassin, Youssef Nasser, Mariette Awad, Ahmed Al-Dubai, Ran Liu, Chau Yuen, Ronald Raulefs, and Elias Aboutanios. Recent Advances in Indoor Localization: A Survey on Theoretical Approaches and Applications. *IEEE Communications Surveys & Tutorials*, 19(2):1327–1346, 2017.

[293] Josh Jia-Ching Ying, Wang-Chien Lee, Tz-Chiao Weng, and Vincent S. Tseng. Semantic trajectory mining for location prediction. In *Proceedings of the 19th*

ACM International Conference on Advances in Geographic Information Systems (SIGSPATIAL), pages 34–43. ACM, November 2011.

[294] Josh Jia-Ching Ying, Eric Hsueh-Chan Lu, Wang-Chien Lee, Tz-Chiao Weng, and Vincent S. Tseng. Mining user similarity from semantic trajectories. In *Proceedings of the 2nd ACM International Workshop on Location Based Social Networks*, pages 19–26. ACM, November 2010.

[295] Moustafa A Youssef, Ashok Agrawala, and A Udaya Shankar. WLAN location determination via clustering and probability distributions. In *Proceedings of the 1st IEEE International Conference on Pervasive Computing and Communications (PerCom)*, pages 143–150. IEEE Computer Society, 2003.

[296] Baosheng Yu and Dacheng Tao. Deep metric learning with tuplet margin loss. In *Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6489–6498. IEEE, 2019.

[297] Jing Yuan, Yu Zheng, Chengyang Zhang, Wenlei Xie, Xing Xie, Guangzhong Sun, and Yan Huang. T-drive: driving directions based on taxi trajectories. In *Proceedings of the 18th International Conference on Advances in Geographic Information Systems (SIGSPATIAL)*, pages 99–108. ACM, November 2010.

[298] Yihong Yuan and Martin Raubal. Measuring similarity of mobile phone user trajectories– a Spatio-temporal Edit Distance method. *International Journal of Geographical Information Science*, 28(3):496–520, March 2014.

[299] Faheem Zafari, Athanasios Gkelias, and Kin K. Leung. A Survey of Indoor Localization Systems and Technologies. *IEEE Communications Surveys & Tutorials*, 21(3):2568–2599, 2019.

[300] Aston Zhang, Zachary C. Lipton, Mu Li, and Alexander J. Smola. Dive into deep learning. *arXiv preprint arXiv:2106.11342*, 2106.11342:839, 2021.

[301] Yu Zheng. Trajectory Data Mining: An Overview. *ACM Transactions on Intelligent Systems and Technology*, 6(3):1–41, May 2015.

[302] Yifan Zhou, Zhifeng Zhao, Yves Louet, Qianlan Ying, Rongpeng Li, Xuan Zhou, Xianfu Chen, and Honggang Zhang. Large-Scale Spatial Distribution Identification of Base Stations in Cellular Networks. *IEEE Access*, 3:2987–2999, 2015.

[303] Yuan Zhuang, Luchi Hua, Longning Qi, Jun Yang, Pan Cao, Yue Cao, Yongpeng Wu, John Thompson, and Harald Haas. A Survey of Positioning Systems Using Visible LED Lights. *IEEE Communications Surveys & Tutorials*, 20(3):1963–1988, 2018.

[304] Zhenyun Zhuang, Kyu-Han Kim, and Jatinder Pal Singh. Improving energy efficiency of location sensing on smartphones. In *Proceedings of the 8th international conference on Mobile systems, applications, and services (MobiSys)*, pages 315–330, June 2010.

[305] Han Zou, Zhenghua Chen, Hao Jiang, Lihua Xie, and Costas Spanos. Accurate indoor localization and tracking using mobile phone inertial sensors, WiFi and iBeacon. In *Proceedings of the 2017 IEEE International Symposium on Inertial Sensors and Systems (INERTIAL)*, pages 1–4. IEEE, March 2017.

[306] Han Zou, Yiwen Luo, Xiaoxuan Lu, Hao Jiang, and Lihua Xie. A mutual information based online access point selection strategy for wifi indoor localization. In *Proceedings of the 2015 IEEE International Conference on Automation Science and Engineering (CASE)*, pages 180–185. IEEE, 2015.