Department of Computer Science
Series of Publications A
Report A-2020-6

Supporting the WLAN Positioning Lifecycle

Teemu Pulkkinen

Doctoral dissertation, to be presented for public examination with the permission of the Faculty of Science of the University of Helsinki, in Porthania, Lecture Hall P674, on the 19th of August, 2020 at 12 o'clock.

University of Helsinki
Finland

Supervisors
Petteri Nurmi, University of Helsinki, Finland
Patrik Floréen, University of Helsinki, Finland

Pre-examiners
Yu Xiao, Aalto University, Finland
Amy L. Murphy, Bruno Kessler Foundation, Italy

Opponent
Mikkel Baun Kjærgaard, University of Southern Denmark, Denmark

Custos
Petteri Nurmi, University of Helsinki, Finland

Contact information

Department of Computer Science
P.O. Box 68 (Pietari Kalmin katu 5)
FI-00014 University of Helsinki
Finland

Email address: info@cs.helsinki.fi
URL: http://cs.helsinki.fi/
Telephone: +358 2941 911

ISSN 1238-8645
ISBN 978-951-51-6351-6 (paperback)
ISBN 978-951-51-6352-3 (PDF)
Helsinki 2020
Unigrafia

Supporting the WLAN Positioning Lifecycle

Teemu Pulkkinen

Department of Computer Science
P.O. Box 68, FI-00014 University of Helsinki, Finland
teemu.pulkkinen@helsinki.fi

PhD Thesis, Series of Publications A, Report A-2020-6
Helsinki, August 2020, 113+73 pages
ISSN 1238-8645
ISBN 978-951-51-6351-6 (paperback)
ISBN 978-951-51-6352-3 (PDF)

Abstract

The advent of GPS positioning at the turn of the millennium provided consumers with worldwide access to outdoor location information. For the purposes of indoor positioning, however, the GPS signal rarely penetrates buildings well enough to maintain the same level of positioning granularity as outdoors.

Arriving around the same time, wireless local area networks (WLAN) have gained widespread support both in terms of infrastructure deployments and client proliferation. A promising approach to bridge the location context then has been positioning based on WLAN signals. In addition to being readily available in most environments needing support for location information, the adoption of a WLAN positioning system is financially low-cost compared to dedicated infrastructure approaches, partly due to operating on an unlicensed frequency band. Furthermore, the accuracy provided by this approach is enough for a wide range of location-based services, such as navigation and location-aware advertisements.

In spite of this attractive proposition and extensive research in both academia and industry, WLAN positioning has yet to become the de facto choice for indoor positioning. This is despite over 20 000 publications and the foundation of several companies. The main reasons for this include: (i) the cost of deployment, and redeployment, which is often significant, if not prohibitive, in terms of work hours; (ii) the complex propagation of the wireless signal, which – through interaction with the environment – renders it inherently stochastic;

(iii) the use of an unlicensed frequency band, which means the wireless medium faces fierce competition by other technologies, and even unintentional radiators, that can impair traffic in unforeseen ways and impact positioning accuracy.

This thesis addresses these issues by developing novel solutions for reducing the effort of deployment, including optimizing the indoor location topology for the use of WLAN positioning, as well as automatically detecting sources of cross-technology interference. These contributions pave the way for WLAN positioning to become as ubiquitous as the underlying technology.

Computing Reviews (2012) Categories and Subject Descriptors:

Networks → Network services → Location based services
Human-centered computing → Ubiquitous and mobile computing
Networks → Network properties → Network reliability
Computing methodologies → Machine learning → Dimensionality reduction and manifold learning
Computing methodologies → Machine learning → Neural networks

General Terms:
Algorithms, Experimentation, Measurement

Additional Key Words and Phrases:
wlan positioning, location based services, semi-supervised learning, neural networks, interference detection

Acknowledgements

I am endlessly grateful for the unwavering support from and encouragement by my thesis supervisors Petteri Nurmi and Patrik Floréen. I sincerely believe I would not have reached this goal without their expert guidance throughout. I would especially like to thank Petteri Nurmi for his support and contributions throughout my academic research career, and in particular for his dedicated encouragement during the year in which I finalized this thesis.

I furthermore would like to thank my opponent, Mikkel Baun Kjærgaard, for agreeing to participate in the defense of this thesis, and for his promptness in responding on what was quite a short notice. His contributions in the field are vast and varied – something which is reflected in the list of references – and it is the utmost honor for me to have him as an opponent. I would also like to thank my pre-examiners Yu Xiao and Amy L. Murphy for not only providing valuable feedback to strengthen the contributions of this thesis, but also for their kind words in the process of doing so.

The warmest gratitude is also extended to Teemu Roos and Petri Myllymäki, whose guidance during my master's thesis helped start my career in research. Their support helped nurture my growing interest for academic research, but also provided me with the first steps into the field with which this thesis is concerned. In that vein, I would furthermore like to thank Petri Myllymäki for introducing me to the Adaptive Computing research group at the Helsinki Institute for Information Technology HIIT. An intellectually stimulating and fun-loving group of researchers without exception, I particularly fondly remember our brainstorming sessions at the local cafeteria, including the requisite "pulla". In addition to my academic trailblazers Samuli Hemminki and Sourav Bhattacharya, I would also like to thank Joel Pyykkö, Andreas Forsblom, Haipeng Guo, Yina Ye, Antti Salovaara, Taneli Vähäkangas, Yiyun Shen, Tony Kovanen, Miika Sirén, Marjo-Anna Hautaviita, and Jara Uitto for making my time at Kumpula something I will always cherish.

I also fondly remember my time at the Innovative Retail Laboratory at DFKI in Saarbrücken, Germany, and the research group lead by Antonio Krüger. I especially want to thank Gerrit Kahl, Lübomira Spassova, and Denise Kahl for being such gracious hosts and teaching me about the refreshing qualities of Apfelschorle during those warm Summer nights on the banks of the Saar.

Very special thanks are also due to Arto Klami and Krista Longi for their enthusiastic collaboration, and for helping me adopt a new research domain during the latter years of my study. I truly feel the contributions of this thesis are stronger and more varied because of their work and guidance.

I gratefully acknowledge the financial support provided to me by the Future Internet Graduate School during my time at Helsinki Institute for Information Technology HIIT, as well as the Department of Computer Science of University of Helsinki, including the Doctoral programme in Computer Science (DoCS). I especially want to thank Pirjo Moen for providing invaluable support in finalizing both my studies and this thesis.

I would also like to extend my gratitude to Ekahau for allowing me to continue my research into the topics of this thesis and for providing me with the time and support for genuine research efforts, which I wholeheartedly recognize is not a given in an industrial setting. I in particular want to highlight the support and guidance provided to me by Johannes Verwijnen, who helped me keep one foot in the academic world, and without whom I likely would not be in this extraordinary position. My colleagues at Ekahau also deserve my unconditional gratitude for fostering an inspiring working environment. From a research perspective, I would also like to thank my colleagues Timo Vanhatupa, Jarno Harno, and Ville Virkkala for the endless intellectual sparring sessions, some of which have supported the contributions of this thesis.

Finally, but most importantly, I would like to thank my parents Lasse and Solveig who have supported me unconditionally throughout my entire life, and encouraged me to pursue my interests without fail. For this I am endlessly thankful. My brother Tomas has similarly stood by me through all these years, and I hope I have made him proud. Last, but most, my undying gratitude belongs to my littlest raccoon Jessica, who has supported me through thick and thin, provided me a shoulder to rest my head on when things got rough, and supported me without question through some of the toughest decisions and tasks I have ever had to face. Thank you.

Espoo, July 5th, 2020
Teemu Pulkkinen

Contents

Original Publications

The thesis is based on the following list of original publications, which are referred to as Articles I - V. The articles are reprinted at the appendices of this thesis.

Article I Petteri Nurmi, Antti Salovaara, Sourav Bhattacharya, Teemu Pulkkinen, Gerrit Kahl.

Influence of Landmark-Based Navigation Instructions on User Attention in Indoor Smart Spaces. In *Proceedings of the 16th International Conference on Intelligent User Interfaces (IUI '11)*. ACM, 2011.

Article II Teemu Pulkkinen, Teemu Roos, Petri Myllymäki.

Semi-supervised Learning for WLAN Positioning. In *Proceedings of the 21st International Conference on Artificial Neural Networks (ICANN '11)*. Springer-Verlag Berlin Heidelberg, 2011.

Article III Teemu Pulkkinen, Petteri Nurmi.

AWESOM: Automatic Discrete Partitioning of Indoor Spaces for WiFi Fingerprinting. In *Proceedings of the 10th International Conference on Pervasive Computing (Pervasive '12)*. Springer-Verlag Berlin Heidelberg, 2012.

Article IV Krista Longi, Teemu Pulkkinen, Arto Klami.

Semi-supervised Convolutional Neural Networks for Identifying Wi-Fi Interference Sources. In *Proceedings of Machine Learning Research*, Volume 77: Asian Conference on Machine Learning, 2017, pages 391–406. JMLR, 2017.

Article V Teemu Pulkkinen, Jukka K. Nurminen, Petteri Nurmi.

Understanding WiFi Cross-Technology Interference Detection in the Real World. In *Proceedings of the 40th International Conference on Distributed Computing Systems (ICDCS '20)*. IEEE, 2020.

Chapter 1

Introduction

Location-based services (LBS) for outdoor scenarios such as navigation, fitness tracking and even augmented reality games have become commonplace ever since selective availability, an intentional degradation of the signal quality, of GPS was abandoned at the turn of the millennium [Phi00]. This popularity has been especially bolstered by the proliferation of the next generation of GPS-enabled devices such as smartphones and wearables.

Despite this success in the outdoor location context, however, a truly ubiquitous solution for location information has yet to emerge. The GPS signal struggles to propagate through buildings, leading to poor quality positioning indoors [KBG+10]. This gap in information has led to fierce competition for the mastery of the indoor location context with technologies ranging from ultrawideband (UWB) [BLB05] to Bluetooth beacons [DCSM16] and even infrastructure-free solutions like pedestrian dead reckoning (PDR) [JSPG09]. Given that human beings reportedly spend up to 87% of their time indoors [KNO+01], the market potential for such a solution is arguably even greater than for the outdoor context. Yet, most systems either require dedicated – additional – infrastructure or struggle with providing a level of accuracy that indoor LBS require [BLM+17].

An early contender for indoor positioning is WLAN. The proliferation of WLAN-capable devices has achieved a level of ubiquity matched only by Bluetooth. Many potentially location-aware applications therefore already exist in spaces where this infrastructure is deployed. The added benefit over Bluetooth – or short-range device-to-device communication technologies in general – is the use of wireless access points as part of the network infrastructure. This allows for relatively painless maintenance in terms of hardware, but also provides a larger coverage for each station, ensuring the signal can be heard from tens of meters away.

The first IEEE 802.11 specification for wireless networks was released in 1997, but the main push for consumer-grade WLAN clients came with the release of 802.11b in late 1999. In many ways, then, the advent of WLAN technology mirrors that of the widespread adoption of GPS. This is also reflected in the publication of the earliest WLAN positioning systems [BP00].

Despite providing promising levels of accuracy at an early stage [BP00], WLAN positioning has yet to become as ubiquitous as the underlying communication protocol. In fact, even 20 years later the research into WLAN positioning has not slowed down to any discernible degree – as can be seen from Figure 1.1. Since the early 2000s when the idea was first introduced, research into WLAN positioning has continued more or less unrelentingly.

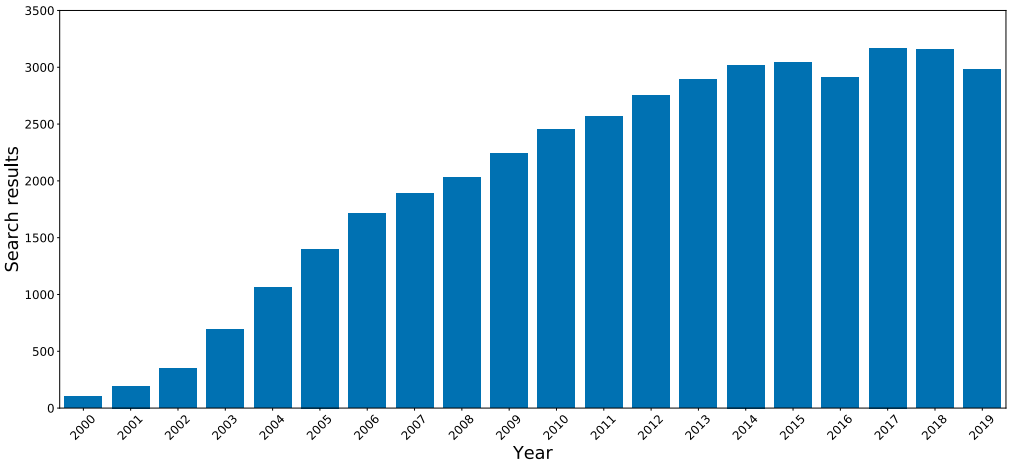

Figure 1.1: Interest in WLAN positioning in academia continues unabated since its inception. Number of search results for the term "wlan positioning" in Google Scholar over the years of research.

The lack of progress in providing a ubiquitous solution likely stems from many of the confounding characteristics of the underlying physical medium. At its core the propagation of the wireless signal indoors is tremendously complex, and often requires detailed modeling to properly account for all possible electromagnetic interactions between obstacles and the propagating signal [UAAL19]. The more common approach has thus been empirical in nature – to teach a machine learning algorithm to associate measured signal strength values with geographical locations (see Section 1.1 for a realization). Though this point of view has achieved

impressive accuracy in many real-world environments [HFL+04, YA05, AER+19], it is not devoid of its own set of constraints. This learning process often requires a significant upfront effort in terms of calibration, and further maintenance whenever the underlying infrastructure (or environment obstacles) changes to a significant degree. For instance, measuring a single office building could take tens of hours [HFL+04] in order to provide the required location granularity. Furthermore, the modeling of the environment for WLAN positioning purposes is not trivial. Whereas the location context for end users often reduces to specific rooms or hallway intersections, there is no guarantee this human interpretation of space is reflected in the measured signal [YA05].

Finally, because WLAN transmitters operate in an unlicensed frequency band, they are by necessity faced with competing technologies and even unintentional transmitters occupying the same set of frequencies. This can have the effect of shortening the range within which signals can be heard or even obscuring certain access points completely. For WLAN positioning systems, relying on signal strength measurements for their location information, this can have a significant impact on the robustness of the system as well as on the accuracy of location estimates, which in turn reduces the consistency and usefulness of location-based services relying on it.

1.1 Thesis Motivation

A machine-learning approach to WLAN positioning is attractive because it rarely requires modeling the physical parameters of the signal, like reflection and multipath, yet is still able to provide a competitive accuracy compared to other indoor positioning systems. In a typical system of this nature, depicted in Figure 1.2, a *calibration survey* is performed in the target environment, in order to associate the signal strength measurements from various access points with geographical locations, or *location labels*. These geographical locations in the simplest case are location coordinates, but more commonly represent a larger location context such as rooms. The machine learning algorithm is then able to use this signal strength database to *learn a location model* by finding the inverse description: an association between signal strength measurements and location. Using this *positioning model* a real-time version of the algorithm can then provide location estimates based on new measurements.

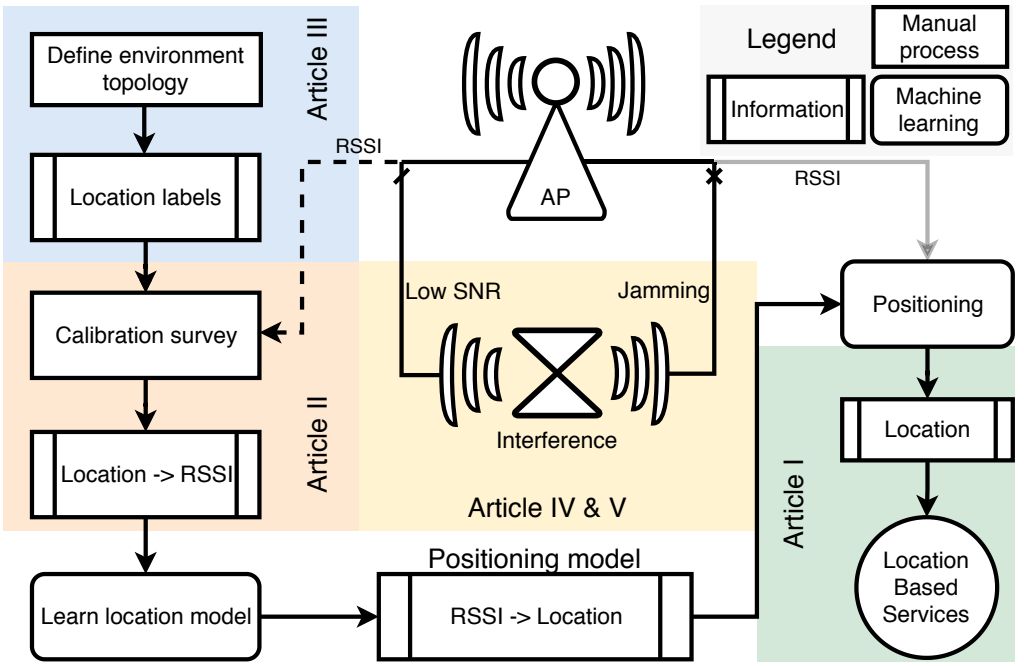

Figure 1.2: Lifecycle of a typical machine-learning based WLAN positioning system. This thesis contributes by supporting the most vulnerable phases.

This lifecycle is vulnerable to the stochasticity of the WLAN signal, especially in the following three phases. First, during the calibration phase, a tremendous manual effort is required to measure all applicable spaces in the environment. For instance, Haeberlen et al. [HFL+04] report a minimum of 14 hours to cover an office building with three floors. In a similar vein, the Horus system was evaluated based on 110 sample locations, in which 100 samples were measured for 300 ms each, for a total of almost an hour per office space [YA05]. In other words, this largely manual effort scales poorly with the increasing size of the deployment. Even if the survey process is augmented by interpolating coordinates between anchor locations, multiple surveys from multiple directions are still required to ensure a usable positioning accuracy [GH16]. The survey process could also be coupled with alternate positioning technologies such as PDR, but this can introduce another source of error due to drift from noisy sensors [BLM+17], in addition to not addressing the underlying issue of manual data collection.

Second, the design of the environment topology can introduce uncertainty into the positioning model. During this phase, the division of the space into location primitives typically follows the needs of the application instead of the limits of the signal heterogeneity. For instance, it has been shown that the signal can vary greatly even within distances shorter than the signal wavelength [YA05]. In [HFL+04] the environment topology for the most part consists of room-level location cells, placed manually. The authors recognize the need for covering large spaces with multiple cells due to the standard deviation of the signal, but provide no automated way of determining this. Approaches like [RMT+02] and [NBK10] opt for a uniform grid, which eases the topology construction greatly, but does not consider the spatial variability of the signal. In [YA05] the need to model so-called "small scale variations" is recognized and accounted for in the inference phase, but no attempt is made to adjust the model topology itself. Even more recent works still use largely heuristic environment partitioning strategies [AER+19]. Forgoing analysis such as this results in less robust solutions, which undermines the usefulness of location-based services.

Third, even the most sophisticated and robust systems can be brought down by disruptions in the physical layers of the wireless channel. In many cases these instances are entirely unforeseen and cannot be accounted for during the design of the positioning model. Depending on which stage of the WLAN positioning lifecycle this disruption occurs, the impact can range from a specific access point not being "visible" during positioning to the positioning model being injected with gaps of information where none are expected. This disruption of the information flow can significantly impact the positioning accuracy, regardless of which wireless technology is used [PLC+17].

The end result of ignoring these issues is a higher degree of uncertainty in the positioning model, which directly translates to a decrease in positioning robustness. In practical terms this means a location-based service, such as the indoor navigation application described in Article I, could end up providing inconsistent or even misleading information. This would serve to frustrate the end user and reduce their trust in the application.

Contributions

This thesis investigates the presented issues and develops novel techniques for mitigating or overcoming them, through the following contributions:

1. An indoor navigation application can be made to work in challenging real-world environments by supporting the positioning algorithm with a graph-

based abstraction of the location information. The manual effort required to initialize the system and design said abstraction provides motivation for our further contributions. This work is presented in Article I [NSB+11] and summarized in Section 2.4.

2. The calibration effort can be significantly reduced by exploiting the inherent interdependency of measurements in the signal space. By modeling this signal space as a multidimensional manifold, the locations of measurements without ground truth labels can be determined through a form of non-linear interpolation from neighboring measurements with known locations. This work is presented in Article II [PRM11] and summarized in Section 3.1.

3. Starting from a traditional discretization of a space, e.g. a uniform grid, the positioning environment can be evaluated in terms of the consistency of measurements in each discrete location. This evaluation is performed automatically through the use of a self-organizing map, which can be used to determine the variability within and between location cells. This same process can be used to provide a more robust, and light-weight, form of discretization that adheres to the underlying signal variation by merging co-located and low-performing regions. Finally, by describing a measure for region fitness, candidate locations for new access point placements can be suggested. This work is presented in Article III [PN12] and summarized in Section 3.2.

4. The presence of competing technologies in the frequency band can be detected through popular neural network constructions previously used for tasks such as image recognition. Even simpler, linear, approaches can be shown to achieve competitive results without the added burden of massive data collection schemes and with arguably more interpretable and applicable results. This work is presented in Article IV [LPK17] and Article V [PNN20], and summarized in Chapter 4.

A central theme of the contributions of this thesis are the complications faced when putting theoretical concepts to practice. Empirical approaches to WLAN positioning require effort and planning but are still faced with a level of uncertainty. To ensure that the contributions provide real-world benefits, the presented techniques have been empirically validated in real-world environments using measurements from off-the-shelf hardware. Many of these contributions have also been validated in complex open spaces such as supermarkets, which are known to be adversarial to WLAN-based indoor positioning.

1.2 Author Contributions

1.2.1 Location-based Service in Supermarket Environment

Article I : Influence of Landmark-Based Navigation Instructions on User Attention in Indoor Smart Spaces

The author helped finalize the positioning system installation in the supermarket environment, performed many of the surveys required for the underlying positioning system and helped construct the connectivity within the store. Furthermore, under the guidance of Petteri Nurmi, the author performed the discretization of the supermarket space, defined the graph structure, the neighborhood abstraction as well as the shortest path solution for the navigation component. Finally, in collaboration with Sourav Bhattacharya, the author helped develop the MONSTRE navigation system and contributed to the writing of the sections of the article relating to the location-dependent aspects of the contribution.

1.2.2 Signal Space Modeling

Article II : Semi-supervised Learning for WLAN Positioning

The initial draft of the article was based on work by the author under the guidance of Teemu Roos and Petri Myllymäki. This included all experiments, related work as well as tuning the Isomap algorithm. The final version of the article was produced in collaboration with the co-authors.

Article III : AWESOM: Automatic Discrete Partitioning of Indoor Spaces for WiFi Fingerprinting

The author implemented the self-organizing map and contributed the various fitness scores, including the use of the rank-based correlation implementation. The refinement of the solution, including the score threshold and the specific clustering technique used, and writing of the article was performed in collaboration with Petteri Nurmi.

1.2.3 Detecting Competing Technologies

Article IV : Semi-supervised Convolutional Neural Networks for Identifying Wi-Fi Interference Sources

In terms of writing, the author provided the related work into interference detection, including the baseline algorithm as well as its implementation and evaluation. The author also performed all measurements for the study, and contributed to the writing related to interference detection as well as the WLAN domain. The author also took part in designing the experimental setup and contributed to the data representation and preprocessing.

Article V : Understanding WiFi Cross-Technology Interference Detection in the Real World

The initial draft, including all experimentation and the problem description was performed by the author. A further draft was prepared under the guidance of Jukka K. Nurminen. The final version was a collaboration between the author and Petteri Nurmi.

Chapter 2

WLAN for Indoor Positioning

When selective availability of GPS was abandoned in 2000 [Phi00], tasks such as navigation, time synchronization, and emergency services greatly benefited from the improved accuracy. These advances were later amplified by (cellular-)assisted GPS providing a faster lock-on, which modern smartphones could utilize. Despite these benefits, GPS positioning carried with it a major caveat that for most intents and purposes has yet to be resolved: the viability of indoor positioning. While the signal itself can be heard almost anywhere, the resulting accuracy is essentially reduced to that of the days of selective availability whenever the positioning device is brought indoors [ZB11]. The problem of indoor location context is further exacerbated by the need for fine-grained positioning. Whereas a 5 meter error outdoors could still provide enough context for automotive navigation to keep track of which road is being traversed – through so-called *map matching* – a similar drift in position indoors could mean the difference of adjacent rooms or even one floor and the next, which is rarely an issue outdoors.

In the following we briefly discuss some of the ways in which indoor positioning has been implemented, before turning our attention to WLAN positioning as a driver for location context.

2.1 Indoor Positioning

Positioning is fundamentally about mapping a measurable, spatially varying but temporally stable, quantity to useful contexts, such as location-dependent information or guidance. Indoors, this has been attempted with a wide array of technologies and modalities; for instance, [BGVGT+17] considers techniques

based on RF signals, light, sound and magnetic fields in their survey. Even measurements of gamma radiation have been successfully used to provide location context [BK08]. However, in many cases these approaches carry additional requirements that hamper their ubiquitous adoption. Though the source of location information can take various forms, a common distinction – as described in [Kjæ07] – is whether or not the approach requires external infrastructure to provide the location information. Whereas *infrastructure-based* techniques can provide fine granularity of location, the added infrastructure often carries a prohibitive cost for larger spaces or requires constant maintenance. *Infrastructure-less* approaches can usually provide location independently, based on sensing the environment without synchronization with an external system. The caveat of these approaches, on the other hand, is the lack of context during initialization or a steady drift from known locations.

Infrastructure-based approaches often need to strike a balance between accuracy and cost. Techniques like ultrawide-band can achieve sub-meter accuracy [AAHA11] but require an added set of infrastructure with precise configuration and do not necessarily handle non-line-of-sight cases very well. Bluetooth beacons, on the other hand, are relatively low-cost and have a less involved setup process. Because these devices are usually battery operated, there is a significant maintenance cost involved, especially for larger spaces [WB15]. Radio-frequency identification tags carry a similar burden, but also have to contend with the added limiting factor of requiring a tag on the receiving device, whereas Bluetooth could be found on most modern smart devices – though rarely enabled continuously in order to conserve energy. Examples of commercial applications utilizing external infrastructure include Quuppa [Quu20], which uses Bluetooth-based Angle-of-Arrival for accurate positioning, and Walkbase [Wal20] which provides a WLAN/Bluetooth hybrid technology for asset tracking.

Technologies such as the inertial sensors (or inertial measurement units, IMU) in modern smartphones do not require external infrastructure to provide location updates. Pedestrian dead reckoning (PDR) allows for continuous tracking of a user based on the estimation of the user's heading – often through a fusion of gyroscope, magnetometer, and accelerometer readings [DP17] – and movement rate (interpreted through step detection or zero-velocity updates [LJW14]). A typical issue in such an approach, however, is that inherent noise in sensors compound over time and will cause drift unless bootstrapped by periodic external location fixes [BLM+17].

Another promising alternative is to use magnetic fields for location information. Due to steel beams in buildings, variations in the magnetic field are

plentiful, and have been shown to provide good accuracy [HK09]. Resolving the three-dimensional pose of a smartphone – especially with varying quality of sensors – and magnetic interference complicate matters greatly [DP17]. Magnetic fingerprinting also requires high granularity, which demands effort [DP17].

The shortcomings of one methodology can sometimes be compensated by another modality, forming hybrid solutions. For instance, the drift of systems like PDR relying on IMU has been successfully curtailed through visual feature tracking with smartphone cameras [LKM13, DNXY19]. A limitation of this particular approach is the need for a live camera feed, which can be energy intensive and cumbersome in many location-based service scenarios. Since drift cannot be fully eliminated even in this methodology, further occasional absolute position information is usually required from an external source [LKM13]. Many applications have also used environmental constraints, such as shapes of hallways [RCPS12], locations of access points [CPIP10] and even the accelerometer pattern caused by elevators [WSE+12] to provide additional sources of location information.

2.2 WLAN as a Positioning Medium

While any positioning approach relying on WLAN is inherently infrastructure-based, using WLAN networks for positioning has garnered great interest mainly due to the ubiquity and low cost of the infrastructure and the clients supporting the protocol. For many potential location aware contexts, a WLAN network has already been set up to provide connectivity for mobile devices. Furthermore, because of the proliferation of WLAN-capable devices, such as smartphones and laptops, WLAN positioning is an attractive proposition because it can largely be performed with software alone. This ensures calculation can be offloaded to the client, which means the location context can be resolved as close to the target as possible. This client side calculation also provides an avenue for privacy preserving applications. This latter aspect is provided by other infrastructure-less approaches like PDR as well, though not without periodic bootstrapping from external systems, such as those listed above.

In the following, we briefly describe the underlying characteristics of this medium, after which we present the fundamental concepts of typical WLAN positioning systems, with a focus on methods that rely on so-called WLAN signal strength *fingerprinting*. Finally, we describe a study which – as part of a project investigating indoor location-based services – used a WLAN positioning system to provide the location information for a navigation application in a supermarket

environment. This environment arguably provides one of the more complicated scenarios within which a WLAN positioning algorithm could be instrumented, given its combination of open spaces [NGL+13] and high pedestrian traffic, which is known to cause fluctuation in signals [dBQAG+17].

2.2.1 WLAN Protocol

The IEEE 802.11 standard describes a protocol for WLAN communication in, among others, the 2.4 GHz and 5 GHz frequency bands. Technically, the concept of WLAN is wider than what the IEEE 802.11 standard specifies. For instance, HiperLAN is an alternative WLAN implementation developed by the European Telecommunications Standards Institute (ETSI) [BC13]. In practice, most WLAN communication today is based upon the IEEE 802.11 specification. Wi-Fi, a registered trademark of the non-profit Wi-Fi Alliance [Wi-20], is often used interchangeably to refer to WLAN devices certified to operate within this standard. For the sake of clarity, in this thesis we will use WLAN to refer to IEEE 802.11 protocol communication, unless otherwise specified.

WLAN access points conforming to the IEEE 802.11 standard broadcast their capabilities to potential clients through so-called *beacon frames*. Through these frames, a client scanning the appropriate wireless channel, knows about the available wireless endpoints in its vicinity. Though the interval between beacon broadcasts is not defined in the specification, a common configuration is 100 time units (1 TU = 1024 μs), or 102.4 ms [SG15]. This interval provides a balance between crowding the airtime of the channel, network responsiveness and energy efficiency.

The specific measure that provides the location context for most wireless clients is the signal strength of the received beacons. According to IEEE specifications, the *received signal strength indicator* (RSSI) is measured by the receiving station during the reception of the beacon preamble [IEE16], i.e. the first fields of the received frame used for synchronizing the transmission. The standard furthermore requires that the measured RSSI *"has an accuracy of +- 5 dB (95% confidence interval) within the specified dynamic range of the receiver"*, meaning the inherent noisiness of the wireless medium is a well understood issue and can directly contribute to errors even during standard operation.

2.2.2 Measurement Characteristics

The signal strength of a beacon frame, i.e. RSSI, is usually expressed through *decibel milliwatts* (dBm). This decibel domain expression has been chosen, among

other things, to allow for the computation of different components of the signal by addition instead of multiplication [Gal08]. This description also has the added benefit of making the relatively small power values measured in the wireless communication framework human readable and comparable. For instance, a measurement of -60 dBm corresponds to a linear power of $1 * 10^{-6}$ mW. The decibel milliwatt is defined as

$$10 * log_{10}\left(\frac{P}{1mW}\right), \tag{2.1}$$

where P is the power, in milliwatts, to be converted. That is, the decibel milliwatt is defined relative to 1 milliwatt, so that 0 dBm corresponds to 1 mW. A doubling of power (in the linear scale) would approximately correspond to adding 3 dB ($10 * log_{10}(2)$) on the decibel scale.

The WLAN signal attenuates, i.e. decreases in power, as it travels. This propagation *path loss* is typically described through the free-space path loss (FSPL) formula, here specifically in its log-distance form, which calculates the loss in decibels directly:

$$L_o + 10\gamma \log_{10} \frac{d}{d_0} + X_g. \tag{2.2}$$

Here the term L_o corresponds to the transmit power at distance 0, the term $10\gamma \log_{10} \frac{d}{d_0}$ is the path loss component, where the exponent γ depends on the medium ($\gamma = 2.0$ in free space) and $\frac{d}{d_0}$ is the distance with respect to the location where L_o was measured, and X_g is a collector term corresponding to all sources of channel fading, usually modelled as Gaussian random noise. An example path loss scenario is presented in Figure 2.1 where, starting from an estimated transmit power of -20 dBm the signal attenuates 40 dB over 80 meters. In practice, however, this attenuation is often greater due to interaction with the environment. This includes attenuation when passing through walls and people moving around, as well as reflection off surfaces. The signal can also be subject to *multipath* [Gol05], meaning it arrives at the receiver multiple times with varying delays and power due to the different paths it took to arrive. These sources of signal attenuation mean the resulting measured power level rarely corresponds to what the FSPL model would predict. Obstacles and materials in the environment need to be modelled precisely to provide accurate measures. In practice, due to all the different sources of (additive) noise in the wireless channel, even measurements made while stationary will tend to a normal distribution, due to the central limit theorem [CT06]. However, research has shown that for WLAN transmissions in

particular the measurements are rarely strictly Gaussian, but are left-skewed to some extent [KK04]. Assuming a Gaussian fit for WLAN measurements in a probabilistic system – something which we explore in Section 2.3.3 – thus carries potential caveats in terms of model uncertainty.

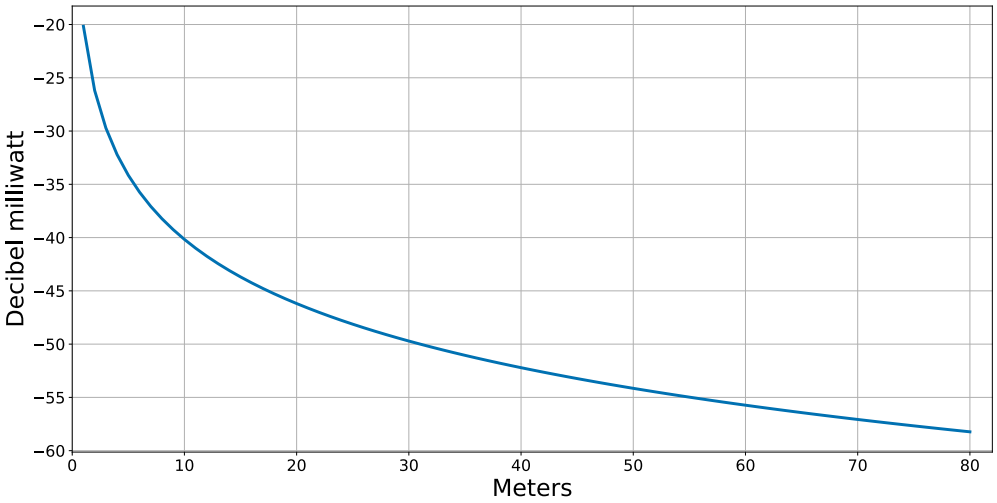

Figure 2.1: Theoretical free-space path loss of WLAN transmission on 2.4 GHz band.

Finally, since signal attenuation is known to increase along with the frequency of the carrier signal [ASVN12], the 2.4 GHz and 5 GHz bands WLAN most commonly uses strike a delicate balance for positioning purposes. In the higher end – e.g. the 60 GHz band – even interaction with oxygen can influence attenuation, with signal loss of up to 20 dB higher than in the 5 GHz band [Cor09]. Lower frequencies, e.g. FM radio in the 100 MHz range suffer less attenuation and better penetration from the environment due to their longer carrier wavelength, but in turn have less spatial variation in the signal. Although ambient FM radio signals, among other lower frequency technologies, have been used for positioning with modest success, in at least one study WLAN signal strength measurements in the same environment provided better accuracy [MD14]. Improved accuracy – comparable to WLAN positioning – can be achieved, but requires instrumenting the environment with further short-range FM transmitters [POM10].

2.3 WLAN Positioning

In WLAN positioning, the location-dependent quantity used for positioning is typically the RSSI from WLAN access points, since its propagation characteristics are known, though in theory any measurement that varies uniquely by location could be used. The way the location dependency of RSSI is exploited varies to a great degree. A common dichotomy, e.g. as defined in the taxonomy described in [Kjæ07], is to determine location either through a so-called *model-based* approach, where location is determined using the known physical propagation characteristics, or an *empirical* approach, where the relationship between signals and the physical space is learned. In this latter approach less information about the physical properties of the signal is required, only that measurements exhibit a spatial dependency and vary smoothly over the physical space.

The model-based approach has been used to great success in outdoor conditions – which the ubiquity of GPS trilateration can attest to – but in indoor spaces the number, type and shape of obstacles quickly becomes intractable to model precisely [OAAJ+18]. In the WLAN positioning field the empirical approach has then gained favor, in particular through the concept of *location fingerprinting*. In the simplest systems only a database of signal strength and location pairs are required for room-level accuracy, which is a sufficient level of abstraction for many location-aware applications. For the sake of completeness, however, we first briefly describe ways in which the model-based approach has been implemented.

2.3.1 Propagation Modeling

Provided that at least three WLAN access points are heard throughout the environment and the specific path loss propagation parameters are known, the location of a WLAN client could be determined through trilateration. In short, given a set of fixed locations and distances to them, trilateration can solve position estimates using the geometry of circles. A propagation model can estimate these distances by interpreting measured signal strength using a path loss formula.

A practical requirement for this approach is that the locations of the access points used for positioning are known, which is not always the case and can require significant effort to determine through a manual survey. Access point locations can be estimated as a separate endeavour [KC11], but any uncertainty in those estimates will transfer into the resulting positioning system as well. In theory, an access point could determine its own location if it is aware of the locations of three other access points in the environment, but because of the

various additive components of the signal arriving at the receiver, trilateration quickly becomes infeasible. Few locations in the environment correspond to the "free-space" that the simplest path loss formula provides, and further refinement requires an intricate understanding of the obstacles and material types in the environment. These environment parameters can to some extent be tuned based on incoming signal strengths and known propagation characteristics. For instance, in [KHLH03] an extended Kalman filter was used to tune the parameters, including the path loss exponent and transmitter and receiver gains. This approach reached a mean accuracy of 3.6 meters. Because these parameters have to be solved for every access point, there is again a risk of introducing errors even before the position estimate itself is resolved.

An alternate scheme is to introduce synchronization into the WLAN network and perform positioning through time-difference-of-arrival (TDOA), as was performed in [YOT+05]. In this work, an accuracy of 2-3 meters was achieved in the 67th percentile, but required an added synchronization component that was not available in the standard specification. In a related work, the multiple antenna arrays supported by the IEEE 802.11n multiple-input and multiple-output (MIMO) protocol was used to achieve an accuracy of 2 meters through angle-of-arrival (AOA) techniques [WKM08]. In addition to being less compatible with off-the-shelf WLAN clients, these results were achieved through simulations, and were not verified within an actual WLAN network or similar signal-to-noise-ratio (SNR) constrained conditions.

2.3.2 Location Fingerprinting

The empirical approach to WLAN positioning entails performing a survey of the environment and labeling measured signal strength vectors with known real-world locations [HPAP09]. These signal strength vectors consist of the signal strengths measured from the access points heard in the environment. Formally, samples of signal strength $S = [s_1, s_2, ..., s_n]$ are labeled with locations L, which traditionally take the form of either coordinates or areas (e.g. rooms). Here n corresponds to the number of access points heard in the target environment. In order to increase robustness, the signal strength vectors typically correspond to the average of multiple measurements.

During the positioning phase, location is then determined by searching the signal space, consisting of one or more measurements S for each location L, for the closest match to a newly measured signal, and estimating a real-world location based on the previously established mapping.

In the first implementation of this technique, RADAR [BP00], the location estimate was performed through k-nearest-neighbors (k-NN) where, as the name implies, the k nearest signal strength neighbors (in terms of Euclidean distance) were considered and the suggested location was the average of the closest matches in the location space. In this work the improvement compared to a model-based technique was first shown; even a simplistic environment with 70 measured locations and 3 access points reached a median accuracy of 2-3 meters.

2.3.3 Probabilistic Modeling

A further sophistication of location fingerprinting is the probabilistic modeling of the signal space. Initially this probabilistic model was described through histograms or Gaussian kernels [RMT+02], but later the Gaussian density function description became a more popular technique [HFL+04, YA05]. In practice this approach consists of storing not only the average signal strength value, but the estimated standard deviation as well. This description allows determining the probability of a newly measured value s_m through

$$P(s_m|L) = \frac{1}{\sqrt{2\pi}\sigma} \exp - \frac{(\mu - s_m)^2}{2\sigma^2}, \tag{2.3}$$

where μ and σ correspond to the average and standard deviation of the signal in location L, respectively. Though this Gaussian description to some extent is in conflict with the study in [KK04] that measurements tend to have non-Gaussian skew, it was found early on that a Gaussian assumption helps smooth out temporal variations and missing values from the measurements [HFL+04]. This description also has the benefit of decreasing the complexity of the probability model [HFL+04], which can have a significant impact on not only storage capacity, but also processing speeds on CPU and energy constrained mobile devices – a typical end client in WLAN positioning systems.

Given this model of the signal for location L, the position estimate is then typically modelled using Bayesian inference [RMT+02]:

$$P(L|s_m) = \frac{P(s_m|L)P(L)}{P(s_m)}. \tag{2.4}$$

Here $P(s_m|L)$ corresponds to the likelihood of the signal given the location – the fundamental component of a probabilistic positioning system – i.e. Equation 2.3 in the Gaussian approach. Though not a strict requirement in approaches satisfied with the location with maximum likelihood, this quantity is usually normalized by the likelihood of the data, in this context the measured signal strength

sample, $P(s_m)$, to provide a probability distribution over locations. A typical measure of this likelihood is $\sum_{i=1}^{m} P(s_m|L_i)$, or the sum of the likelihood of the same signal over all other location candidates.

For the sake of simplicity, the probability $P(L|s_m)$ is often treated (in a naïve Bayes sense) as being independent for each access point, meaning it can be solved for each location through the product of the individual probabilities. Arguably, this independence assumption is violated when the user moves [RMT+02], but in practice this independence assumption has served the task well.

A maximum likelihood approach could at this stage provide a position estimate as the location with the highest likelihood (or probability, if normalized). Usually, however, added context can be ascribed by taking into account the prior probability of the location $P(L)$. A uniform prior would essentially provide no information, but the distribution could also be initialized based on personal behavior [CCKM01]. In other words, the system could be primed to locate users based on their specific day-to-day movement patterns. Another alternative is to use the prior for Bayesian inference by using the previous location probability to inform the next, in a hidden markov model (HMM) sense. This has the effect of improving the tracking accuracy, i.e. updating the location as the user moves.

The increase in accuracy through the use of probabilistic modeling was quickly apparent across contemporary publications. [RMT+02] reached a median accuracy of approximately 1.5 meters, whereas [YA05] – through additional improvements such as an autoregressive model of sequential signal strength samples and an access point clustering module – achieved a median accuracy of around 0.5 meters. In this work – as in many others – validation was performed in an office environment and not in complex everyday environments with less constraints provided by the environment topology. In environments with more open spaces, such as supermarkets or malls, the correspondence between the defined environment topology and the spatial variability of the signal might no longer apply.

To enable a probabilistic model typically involves either using human context labels like rooms or discretizing the environment in a uniform way in order to define the location primitive over which to calculate summary statistics. This is traditionally a manual effort that only scales well for uniform discretization, which in turn has to contend with each location inevitably having different propagation characteristics. Two potential causes of uncertainty are at risk of developing at this stage. First, if a location with multiple modes in signal space is modelled as one cohesive region, the variance of the signal model will increase. In terms of a Gaussian fit, this will directly translate to greater uncertainty about the location even if the measured signal matches any one of the modes in the

distribution. This is illustrated in an example scenario in Figure 2.2. Though the received signal strength measurement (-60 dBm) falls exactly on the mode of the signal model in Location B, Location A will appear more likely simply because its model is less uncertain. Using non-parametric probabilistic models like histograms could alleviate such issues, but to the detriment of model complexity, storage and computational efficiency.

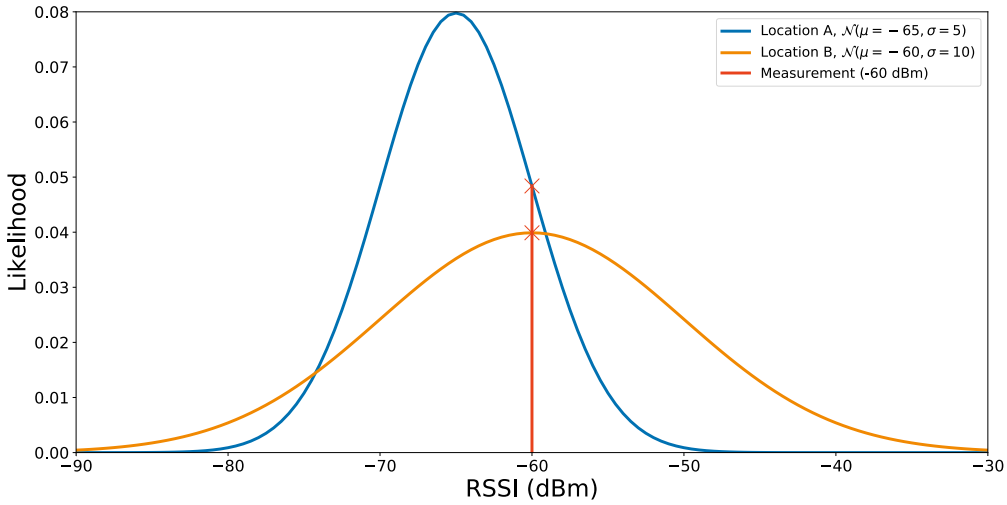

Figure 2.2: Impact of uncertainty in the probabilistic model. A greater variance could make even exact matches appear less likely than competing hypotheses.

Second, if a location with only one mode is partitioned into two or more regions, these locations will appear equally likely in terms of the signal model. Though the position estimate might still be constrained by the union of these regions, the robustness of the estimate will suffer because even a slight change in the measurement might make one region appear more likely than the other.

To provide a serviceable level of accuracy typically tens of measurements per location are required to ensure the probability model describes the signal in a statistically sound way. To maintain such a system further requires that the model is updated whenever significant changes to the environment occurs. These measurements tend to require manual effort to obtain, especially during the labeling process, which can quickly become untenable for large spaces. These and the previously described limitations of the empirical approach to WLAN positioning are the focus of our contributions described in Chapter 3.

2.3.4 Further Advances

Since the previously described initial WLAN positioning approaches, many advances have been made, focusing mainly on improving accuracy and robustness. Among these improvements is accounting for the user's motion during positioning as well as the difference between measurement hardware. In the following we briefly describe notable work in this domain, to the extent that is necessary for motivating our contributions to this field.

Particle Filtering

Many of the previously described approaches indirectly take the mobility of the user into account, but the execution has usually remained on the level of a prior in the Bayesian inference iteration, such as a Hidden Markov Model of the user's motion [KH04]. A more sophisticated and formal description of the same has been presented in the context of particle filtering [HB04]. This allows for a probabilistic modeling of the location as well as the motion of the user. This technique, initially presented as the bootstrap filter [GSS93], models the user location through a set of particles, each of which essentially contain a hypothesis of the user's location. The previously described model of the signal in each location is augmented with a motion model (i.e. noise model in the original bootstrap filter) that gives particles close to the previous location more weight than those that have been sampled further away. Specifically, at each iteration, each particle's hypothesis (e.g. the Gaussian probability density function of signal strength) is tested against the measured signal strength, and reweighted based on the strength of the match. The set of particles is then resampled based on this new distribution, essentially duplicating well-performing particles and eliminating particles with low weight. Before the next measurement phase, the remaining particles are then propagated according to the motion model, which in the simplest case could be a Gaussian kernel over nearby locations. The main benefit of this approach is not in the improvement of local accuracy but rather a smoothing of the motion trajectory over time; using a particle filter can reduce the accumulated motion error significantly [HB04].

Though location smoothing techniques such as this can greatly improve the tracking error in WLAN positioning, especially over time, they are exclusively a post-hoc solution to the fundamental issue of WLAN signal spatial variability. Even a fully instrumented particle filter still has to contend with uncertainties (such as those caused by multiple modes) in the positioning model. The distance between two equally likely locations – and thus successive position estimates –

might be longer than what the motion model is designed for, meaning a fraction of particles might have to be propagated randomly to avoid getting stuck in local optima. In other words, the motion model mostly assumes a smooth transition from one location to the next, which is not always guaranteed especially in noisy environments. This is an issue we will contend with, and to some extent overcome, in our navigation application described in Section 2.4.

Device heterogeneity

An issue that could be missed during the construction of a WLAN-based positioning system is the difference in characteristics of the devices used for training the model and those used for providing the location context to the end user. In addition to potential differences in hardware from the same vendor, especially in the case of low-cost hardware, different implementations of the same hardware concept can cause a misalignment in the positioning model. These differences could include the polarization of the antennas, i.e. the way the receiving antenna is aligned w.r.t. to the transmitting antenna. This is characterized by the so-called *polarization mismatch factor* (PMF), described by [YH10]:

$$PMG = cos^2\alpha, \tag{2.5}$$

where α corresponds to the angle of misalignment. In the extreme case, e.g. a vertically polarized transmitter and a horizontally polarized receiver, no signal could actually be received. Devices also have to contend with different levels of sensitivity as well as antenna transmission patterns.

Some works have taken this imbalance into account. In [HFL+04], device calibration was handled through a linear calibration function

$$c(i) = c_1 * i - c_2, \tag{2.6}$$

where i represents the signal value of the target device and c_1 and c_2 the parameters that are learned through calibration. This adjustment provided an improvement from 70% to 88% correct location estimates after calibration. In another work [KM08], this calibration phase was circumvented by redescribing the fundamental signal strength value as pairwise ratios between access points and performing the traditional probabilistic location modeling as previously described. This description provided nearly identical results to a manual calibration approach. Similar robustness in the face of device heterogeneity has also been accomplished using deep-learning approaches, by forcing a neural network to learn on noisy samples [AER+19].

Hybrid Modeling

Given the inherent effort required by the empirical description of the signal strength variation in the environment, some works have displayed a renewed interest in incorporating aspects of the known signal propagation characteristics into the probabilistic modeling framework. For instance, we found [PVN15] that by simulating signal strength fingerprints using the path-loss equation (see Equation 2.2) given the known locations of access points and simply fixing the path-loss exponent – a factor that could be thought to represent all obstacles in the path – one could reach accuracies close to those of a purely empirical approach. Specifically, we estimated a slightly worse median accuracy but improved worst-case accuracy. The main limitation of this work is the lack of a formal description of the way the path-loss exponent could be determined; in the study this exponent was found by minimizing the positioning error, which inherently requires a baseline empirical measurement and does not as-such alleviate the underlying burden of calibration.

Deep Learning

The success in recent years of deep learning techniques, especially for image recognition [KSH17], has garnered interest in the WLAN positioning community as well. In DeepFi [WGMP15] the spatial variability of another location-dependent WLAN quantity, that of channel state information (CSI), was exploited for positioning purposes. This source of information provides a richer description of the communication channel between the client and the access point, something which has been previously been used to enable decimeter-level accuracy in [KJBK15]. However, CSI measurements require specific WLAN hardware and custom drivers – decreasing the potential for ubiquitous adoption – and the deep-learning aspect requires even more measurements than traditional RSSI fingerprinting. In DeepFi, 500 to 1000 measurements per location – recorded over an interval of 1 to 2 minutes, respectively – was used for training the network.

WiDeep [AER+19] also described a deep-learning approach, but used RSSI for its location information. The work furthermore provided a way to handle device heterogeneity by instrumenting the neural network with a denoising component as well as artificially injecting noise into training samples. Though it was able to improve on the accuracy of DeepFi, it also required up to thousands of measurements for each target location. Even a generous assumption of only one WLAN channel to measure, at a typical beaconing rate of ≈ 100 ms, 1000 measurements would require 100 seconds of measurements per location.

2.4 Application: Indoor Navigation in a Supermarket

The location context provided by a positioning system is given purpose through the location-based services that utilize it. One important subcategory of such services is that of navigation. Navigation is particularly beneficial in complex indoor spaces such as airports, train stations, and supermarkets – where crowding conditions have been found to be one of the main causes of stress [AM98]. In the context of a supermarket, an indoor navigation application could take the form of a shopping assistant, guiding the customer to the products in their shopping list. Such a feature has been determined important in a previous study on shopping assistant feature ratings [BFF+12]. Our first contribution in Article I [NSB+11] provides a study into such a supermarket navigation aid.

The supermarket environment provides one of the most challenging opportunities for indoor positioning systems based on wireless signals due to electromagnetic interference, metallic shelving, great variations in crowd density and multiple open spaces. This is reflected in the positioning system accuracy. In a study [BFF+12] using the same system described in this contribution it was found that whereas the median accuracy is in line with state-of-the-art systems, in the range of 1-2 meters depending on the position update interval and crowding conditions, the 90th percentile accuracy ranges from 3-5 meters – enough, in the worst case, to jump from one end of an aisle to the other. This variability in accuracy is a major liability especially in the supermarket context, where product categories can change multiple times along a shelf, let alone from one aisle to the next. One crucial contribution of the work presented here then is how the uncertainty of the location context in this study was overcome through the abstraction of locations and the concept of location neighborhoods. While the following user study largely focuses on the design of effective navigational aids, it serves as a motivating example for the other contributions of this thesis. A WLAN positioning system was chosen specifically to address the constraints posed by the complex indoor environment, and to take advantage of existing infrastructure that was already in use. Alternative solutions were not deemed feasible due to the need for specialized infrastructure, high cost, or other complicating factors as described previously in Section 2.1. At the same time, implementing a genuine location-based service on top of a WLAN positioning system allowed us to examine the impact of positioning errors on the end-user experience. WLAN positioning already satisfies many of the criteria deemed important for indoor navigation applications, including cost, energy-efficiency, accuracy, and response time [BLM+17]. Problems with robustness, as expressed through low quality

position estimates, has been shown to negatively affect the user experience in outdoor navigation applications [RSK19]. Investigating these issues in the indoor domain – and devising strategies to overcome them – is then an essential component of research into location-based services.

2.4.1 Navigation Instructions and User Attentiveness

Supermarkets are prime candidates for location-based services, such as navigation, as they typically consist of open spaces and have confusing and repetitive layouts. This fact has not gone unnoticed by academia, and there is indeed a rich history of providing intelligent retail services for shoppers. An early application in this domain is the Personal Shopping Assistant[ACK94], which was an early proponent of providing the consumer with a personalized context in their shopping experience instead of relying on static local displays of advertisements, as these could be considered distracting in terms of the shopping experience. In a later study, a more decision-theoretic approach was taken to the task of supermarket navigation [BJJA05]. There a system was implemented under the dual constraint of maximizing the product finding likelihood and minimizing the time spent on navigation. The study described in Article I [NSB+11] considers a further constraint to this system in the trade-off between the customer and the system provider. Whereas traditional indoor location applications, such as offices or hospitals, can help alleviate the cognitive load of visitors trying to navigate an unfamiliar space by leveraging a positioning system, the owners of a commercial space often have conflicting goals. An indoor navigation tool could on its own be used for internal purposes, such as inventory management and shelving, without providing the service to customers. To incentivize the provider to extend the service to customers, the application would likely have to provide benefits beyond an overall sense of customer satisfaction. The aforementioned conflict in goals arises when the customer wants to perform their shopping task as effectively as possible, while the provider – usually the owner of the supermarket – wants to expose the customer to as many products as possible during their visit. That is, customers should ideally be made aware of the advertising and products in order to entice them to do further purchases [NSF+14, GA15]. This means a navigational aid provides an interesting dilemma: decreasing the cognitive load of navigation means visitors need to pay less attention to landmarks, many of which have been designed with great effort and resources to be as enticing as possible. To provide a solution that satisfies both parties, then, requires rethinking the type of landmarks used for navigation.

In the following study, the impact of different landmark strategies was explored by presenting customers with two kinds of navigation instructions, and measuring the cognitive load and recall of landmarks. A set of traditional *sign-based* landmarks focused on salient features of the environment, whereas *product-based* landmarks provided a trade-off between bringing advertisements to the user's attention while still decreasing the overall effort of navigating the space. Though an increase in visual demand of landmarks had the expected effect of increasing the cognitive load of the customers, the navigation aid itself compared favorably to unaided scenarios, which suggests such a landmark strategy can make users pay more attention to branding without sacrificing the usefulness of the location-based service. Though the customers' recall of the products wasn't shown to have increased in a significant way, studies have shown that simply increasing customers' attention can influence purchasing behavior [SKI17].

2.4.2 Designing LBS for Uncertainty

The study described in Article I was carried out as part of a larger project on intelligent services for supermarket environments. In order to provide location-awareness in this project a commercial WLAN positioning system was used for location updates. Specifically, the Ekahau RTLS system was set up in a supermarket in Helsinki, Finland, with 29 access points available for use in positioning. This system is based on concepts originally presented in the work by Roos et al. [RMT+02, Kup05], but also includes later developments such as a particle filter [Mis07]. As part of this deployment, a set of so-called *rails* were drawn on all traversable paths in the environment. This helped the positioning system not only constrain its solution space, but also provided it with a topology that respected environment obstacles as part of its design. This ensured the position could never fall on top of a shelf or other inaccessible environments. The accuracy of this positioning system was evaluated separately to have a baseline median accuracy of 2 meters and 5 meter 90th percentile accuracy [BFF+12].

This worst-case accuracy meant the solution was not stable enough for some of our concurrent LBS research purposes, such as location-based recommendations and product location indexing [BFF+12, NSF+14], which meant a further abstraction of the space was required. This took the form of a partitioning of the environment, as depicted in Figure 2.3. This partitioning was performed based on the granularity of product location information – in the range of 1-1.5 m, i.e. so-called *shelf meters* – and the supported accuracy of the positioning system. This resulted in a discretization where each aisle was divided into three grid cells

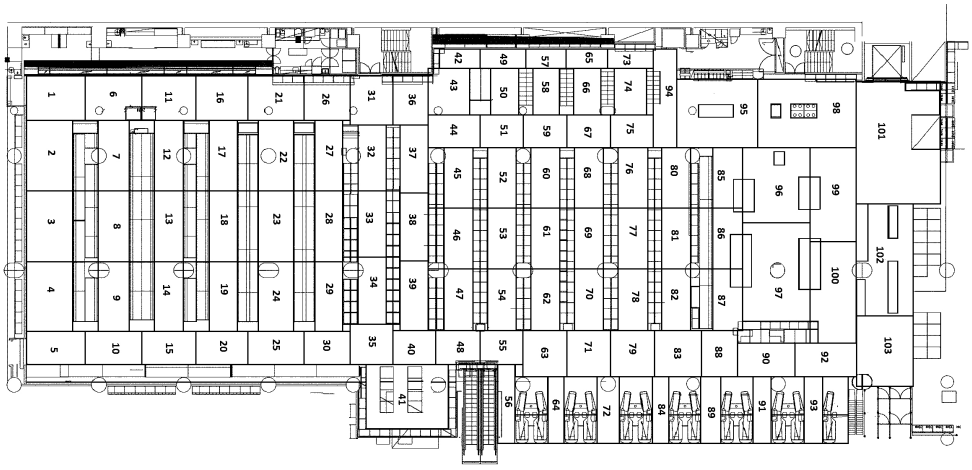

Figure 2.3: Grid-based layout of supermarket positioning model. Modified from previously published version in Article I [NSB+11].

of equal length. Furthermore, the end of each aisle was capped with an additional grid cell that connected the aisle with the ends of neighboring aisles. The produce section (in Figure 2.3 on the right), a largely open space, was divided into larger cells since the product categories were less granular there.

For the purposes of navigation, an additional neighborhood concept was introduced. This allowed navigation instructions to trigger in time – in advance of the target location – as well as ensured that the routing plan remained consistent, which is a key usability consideration [Nie94]. This neighborhood abstraction consisted of the *direct* neighbors of cells, *shelf* neighbors (neighbors across shelves) and *extended neighborhoods* (the union of these). Figure 2.4 depicts two example extended neighborhoods, outlined with red dots. Here, the direct neighbors of cell 23 are 22 and 24, and its shelf neighbors are 18 and 28. The extended neighborhood abstraction was born from empirical observations, where the position estimates occasionally jumped from one end of an aisle to another, or across shelves. This strategy increased the likelihood that the position remained within, at least, an extended neighborhood whenever estimates were uncertain.

This description of the environment – a grid structure with an extended neighborhood relationship – was constructed in order to allow for the navigation to focus on the environment instead of positioning accuracy, and to allow the instructions to be generated in an automatic way. A systematic approach allowed for instructions to be played at a fixed distance before the target, instead of

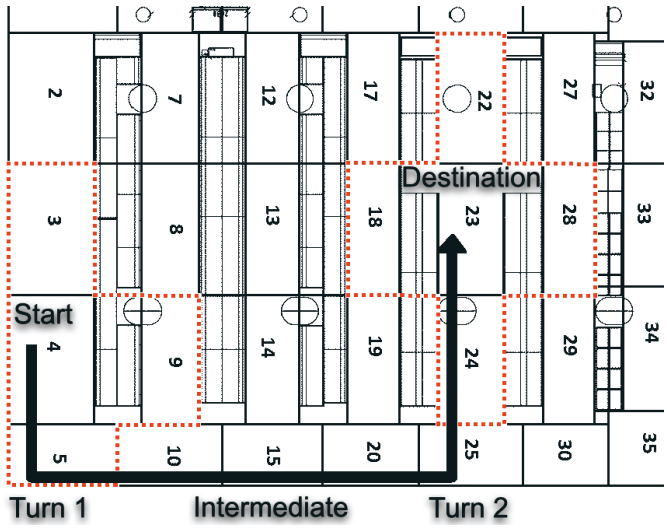

Figure 2.4: Detail view of grid layout with example navigation instructions. Extended neighborhoods highlighted with red dots.

changing depending on what part of the supermarket was entered. This navigation system, Mobile Navigation System for Retail Environments (*MONSTRE*), is described next.

2.4.3 Mobile Navigation System for Retail Environments

By defining a neighborhood for each grid cell, as described before, the environment model could be approximated by a graph structure, where each cell was represented by a vertex and the edge between neighboring cells was weighted by the Euclidean distance between their center coordinates. This meant that the shortest path, along a sequence of non-repeating grid cells, between any two grid cells in the environment could be solved through Dijkstra's algorithm. This environment model and a product retrieval engine described in previous work [NLB+08] formed the foundation of the navigation engine.

In order to guide the customer throughout the navigation task, a set of instructions were generated through a commercial audio synthesizer and presented at opportune moments. Specifically, navigation instructions were presented

- at the starting cell (**Start**)
- at turns (over 45°) (**Turn**)

- in the middle of path sequences longer than 4 cells (**Intermediate**)
- at the ending cell (**Destination**)

An example instruction sequence is depicted in Figure 2.4, containing all possible instruction types (bolded in the above description). Specifically, instructions at the start of each segment (an instruction point in the above list) guided the user to the end point of the segment, i.e. to another instruction point. To accommodate the sometimes uncertain position estimates, instructions were played when the user entered the extended neighborhood of the target cell.

2.4.4 Empirical Study

As mentioned in Section 2.4.1, two types of navigation instructions were studied. The *sign-based* strategy provided the user with instructions of a more generic nature, such as aisle numbers and distinct product category sections, which could be identified by large signs adjacent to the relevant locations. Instructions such as these are likely to cause less cognitive load on the user, since they are designed to aid navigation and product search and are visible from a distance. Instructions of this type included "walk to aisle number five".

The *product-based* strategy, on the other hand, focused on specific product categories in the store. Though the customer might, to some extent, have unconsciously internalized some of the supermarket taxonomy during previous visits – or knows them from similar layouts of other supermarkets – products are nevertheless less salient and can rarely be seen from as far away as signs. This forced the customer to scan the environment for the product described in the navigation instruction. Here, a typical instruction could include "walk past the ice creams".

For both instruction types the specific landmarks were chosen to be as salient, i.e. visually distinctive from their environment, as possible, but the advance visibility was varied in order to increase the visual demand.

To compare the two types of instructions, a set of 5 navigation tasks were created, consisting of two tasks per landmark type, as well as an unaided task to provide a baseline for effectiveness of navigation instructions. The order of landmark types, i.e. either product-sign or sign-product, was randomized for each participant. Participants were tasked with memorizing a small list of common shopping items, and answered questions related to product recall and task workload between each category of landmark instructions.

To ensure that the findings of the study centered on the type of landmark in the instruction, the navigation system was further augmented with a Wizard-

of-Oz interface. In short, during the navigation tasks, a researcher followed the user at a close distance and was prepared to manually enter the location of the user whenever there was a risk that a rogue position estimate would cause confusing navigation instructions. For the purposes of the study 20 participants were recruited at the entrance of the supermarket.

2.4.5 Results

The main finding of the study was that increasing the visual demand of the landmark instructions did not improve the participants' recall of the products they saw during the navigation tasks. An increase in the mean completion time of 80.9 to 106.6 seconds (for sign-based and product-based instructions, respectively) suggests the participants walked at a slower pace and paid more attention to their surroundings, however. Furthermore, in both navigation strategies the participants were always able to find the target product, in contrast to the unaided case, where 4/10 participants failed to find the correct product. In fact, in order to avoid skewing the results of the product recall questionnaire, the unaided navigation task was alleviated for the latter half of the study by directing participants to the correct aisle.

In terms of product recall, the participants were so focused on the navigation task that their level of recall was essentially no better than random guessing. In many cases the participants ended up rationalizing seeing certain products, though they had no opportunity to do so, mainly because they recognized the product category. This effect carried over to the unaided task, where participants had better recall of the products they had seen, suggesting following navigation instructions of any kind causes the customer to pay only minimal attention to their surroundings. This was also observed qualitatively: when following navigation instructions the participants were often observed to follow unconventional pathing when walking through the store. Participants frequently stopped to listen to instructions at the predetermined instruction points, causing confusion among other customers who were unable to anticipate the erratic mode of travel.

Finally, though the Wizard-of-Oz application was close at hand throughout the study, it was only used once, suggesting the layer of abstraction applied on top of the position estimates provided a sufficient level of robustness for location-aware applications. Nevertheless, the positioning errors sometimes meant that instructions were not played in a timely manner, delaying the participant's transition to the next segment of the route. In particular, it was found that these errors sometimes prevented the final message from triggering at the right mo-

ment, meaning participants were not aware that they were close to their target and could proceed with the visual search task.

In conclusion, the navigation aid clearly helped customers traverse the supermarket more efficiently than when they were unaided. Furthermore, though their specific recall of items was not improved directly, the increased time spent navigating using navigation instructions based on products instead of signs suggests more attention was paid to surroundings which could translate to an increase in brand impressions and marketing opportunities.

2.4.6 Discussion

Though the environment modeling applied on top of the positioning system allowed for a study on navigation instructions to focus on psychological metrics instead of the robustness of the positioning system, this abstraction required significant manual effort. The worst-case accuracy in particular necessitated an extended neighborhood concept in order to ensure that instructions arrived in time and did not confuse the end user. The initial effort of constructing the positioning system itself was significant. Multiple surveys at different crowding conditions and along both directions of the aisle were necessary to achieve the necessary positioning accuracy. Such manual tasks do not generalize easily to other spaces, even if the same discretization scheme is used for other supermarket environments. This hierarchy of abstractions highlights the difficulty of enabling a commercially viable indoor location-based system.

Finally, the electromagnetic interference in this environment was significant. This increased the noise level of the wireless channel, which likely affected the robustness of the positioning model. Occasional connectivity issues – critical in a system depending on an external positioning server – could not be decisively narrowed down to any specific source of interference but nevertheless highlights the vulnerability of a real-world system to competing technologies.

For the context of this thesis, this real-world deployment showed that WLAN positioning is generally a viable technology for supporting LBS tasks even in complex environments, as long as the inherent uncertainty is accounted for. Nevertheless, deploying the system required a significant upfront investment in terms of time and manual effort, which could prove prohibitive especially in larger spaces.

In the following chapters we describe techniques that can help alleviate these issues by describing a way to decrease the surveying effort in Section 3.1, constructing an environment topology respecting the signal space in Section 3.2, and detecting potential sources of interference in Chapter 4.

Chapter 3

Signal Space Modeling

A major obstacle to the deployment and maintenance of a WLAN positioning system is the effort – both in terms of labor and processing – required to initialize and calibrate the positioning model. Typically, a surveyor walks around the target environment recording WLAN fingerprints and keeps track of their location. This process – in particular the manual labeling of locations – is time consuming and prone to human error, and is required every time the environment changes enough to warrant recalibration. Examples include a change in network infrastructure or the introduction (or removal) of physical obstacles. The survey method used to initialize the positioning system in Section 2.4 closely resembles the segment-based path survey described in [GH16]. In other words, measurements were collected while walking, and anchor points were labeled at previously chosen landmarks in the environment. As described in [GH16], this can reduce surveying time tremendously compared to a traditional manual survey. However, this approach comes with severe caveats; in order to ensure parity with a completely manual survey, many locations in the environment need to be surveyed multiple times and covering all potential directions. Naturally, it also requires strict adherence to an accurate labeling process and a constant speed while surveying, which increases the cognitive load of the surveyor, and presents opportunities for introducing errors into the positioning model.

Secondly, WLAN positioning systems have often based their location model on an even partitioning of the environment, or used known environmental contexts – such as rooms and corridors – as the basis of the model topology [KH04, HFL+04, DYWY19]. While this discretization makes deployment easier, it does not adhere to the constraints imposed by the variation in signal strength. Depending on the obstacles in the environment, two physically adjacent locations

may be more or less correlated in terms of WLAN fingerprints, but no guarantees of cohesion can be made. If the environment topology is not defined through uniform partitioning or existing contexts, constructing it requires significant manual effort and an understanding of the environment characteristics. An example of such a specialized topology is the one constructed for the supermarket in Section 2.4.2, which needed to consider a combination of open spaces and narrow aisles. WLAN design is also traditionally concerned with maximizing coverage and throughput of the wireless communication, which does not necessarily translate to accuracy in a positioning context. Finally, a complex model increases not only the storage requirements of the model parameters but also the processing demands of the positioning client. Efficient processing improves both latency and energy efficiency [Kjæ12], two critical constraints for location-based services.

In this thesis we contribute novel methods for alleviating these issues. In the following two contributions, we first describe a way to construct the WLAN fingerprint database underpinning most positioning systems in a semi-supervised way in Section 3.1. Specifically, we can dramatically reduce the number of samples that need to be labeled with real-world coordinates by associating unlabeled measurements with real-world locations through their proximity to labeled measurements in a multidimensional signal space manifold. This reduces the manual effort required for the labeling process, and also helps eliminate some of the user errors that this process produces. Second, we describe a way to evaluate a specific environment partitioning, like a uniform grid, in terms of positioning algorithm fitness in Section 3.2. By learning and analyzing a nonlinear mapping of the signal space, we can detect and correct for regions in the environment that are less distinct than their neighbors. Crucially, we can significantly reduce the size of the model by constraining it to the diversity of the signal space instead of the physical space. This has a direct impact on the runtime performance of the application performing positioning.

3.1 Semi-supervised Learning of the Signal Model

A WLAN fingerprint model consisting of signal strength measurements from each access point in the environment will inherently contain as many dimensions as there are physical transmitters. During the calibration survey, however, a latent dimension is indirectly projected onto the measurements because the surveyor – constrained by the physical environment and wireless equipment – will essentially move along a plane with fewer dimensions than the signal manifold. In practice,

we can assume a two- or three-dimensional plane of movement even if the surveyor is an autonomously moving drone.

This intuition drives the hypothesis of our contribution in Article II [PRM11]: the lower dimensional plane will lie on a non-linear multidimensional manifold, which suggests it can be extracted through a manifold learning technique and used to inform a traditional WLAN radio map in an (at least partly) unsupervised way. Specifically, because of the interdependency of measurements in the signal space we need only label a limited amount of samples with real-world coordinates in order to propagate labels to the rest. This can dramatically decrease the surveying effort required to initialize a positioning system.

The non-linear propagation of the WLAN signal (as established in Section 2.2.2) will inherently cause a non-linear dependency between signal strength measurements measured from different locations. To recover a manifold from this space will then require a technique that can take non-linearity into account. A popular manifold learning algorithm of this type is Isomap, introduced by [TSL00]. Leaning on a previously established multi-dimensional scaling (MDS) technique, it is able to handle non-linearity through the use of a geodesic neighborhood map. Whereas MDS can only establish a linear projection, Isomap can learn a non-linear representation by first constructing a neighborhood map in the multidimensional space and then solving the shortest geodesic (manifold respecting) path between points in this manifold. From there, these distances can be provided to the traditional MDS algorithm for further processing. In the following, we go into more detail about this algorithm.

3.1.1 Isomap

As Isomap could be described as a non-linear extension to classical MDS, the intuition of the latter algorithm needs first be described. MDS is a dimensionality reduction technique, and is the most common manifold learning approach in the literature. The basic procedure involves projecting a dataset described by pairwise distances between points onto a lower dimension while minimizing the error in pairwise distances in the new dimension. If the distances used for calculation are Euclidean, MDS is equivalent to PCA [TSL00], which means the minimum incurred loss (also denoted *strain*) can be found by minimizing the equation:

$$L = \|\tau(D_{Y'}) - \tau(D_Y)\|^2 , \tag{3.1}$$

where D_Y and $D_{Y'}$ are the distance matrices for the output and input domains, respectively, and τ is an operator that converts distances to inner prod-

ucts [TSL00]. The solution minimizing this function is found by calculating the eigenvalue decomposition of $\tau(D'_Y)$ and setting the coordinates of the measurement points to the top d eigenvectors, where d is the desired reduced dimension.

Using MDS with Euclidean distances would subject the manifold learning to the possibility of shortcuts between parts of the manifold because the points might appear similar in a Euclidean sense[1]. To resolve this potential drawback, a non-linear manifold learning technique called Isomap is employed. A comparison between these manifold learning algorithms is shown in Figure 3.1, where the so-called Swiss Roll dataset, a common example used for evaluating manifold learning techniques [LV07], is projected onto a two-dimensional plane.

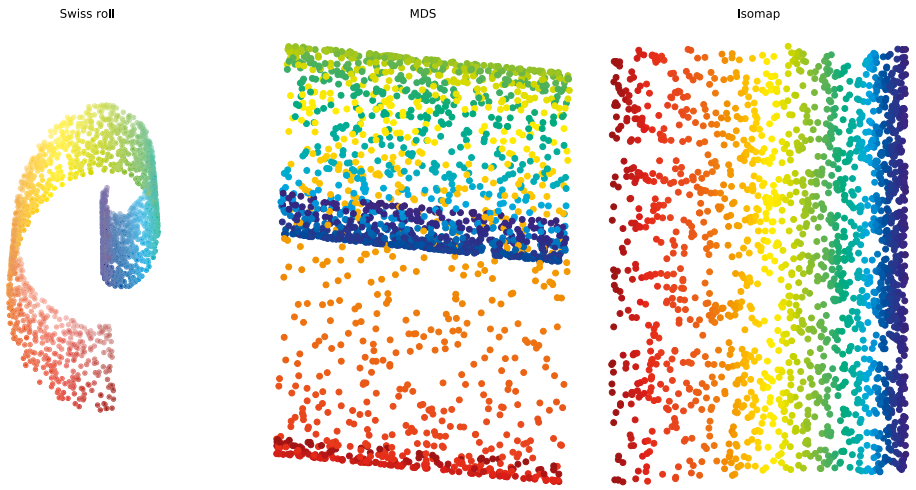

Figure 3.1: Isomap, unlike the linear projection of MDS, is able to extract the intrinsic dimension of the classic Swiss Roll manifold, unfurling it into a two-dimensional plane.

Isomap solves the issue of shortcuts in the manifold by replacing pairwise distances of the distance matrix – estimated between measurements in the original signal domain – by geodesic distances. This entails solving the path between all measurement points in signal space through a shortest path technique, such as Dijkstra's algorithm, whereby the new distance between two measurements is the shortest path – along the manifold, through a sequence of neighbors – between

[1]This notion is further explored in Article III, described in Section 3.2, which uses the real-world Euclidean distance between measurement points to bound the fitting of the signal space.

the two points. This ensures that the distance between two points always respects the shape of the manifold on which the measurements are expected to lie. In our contribution, we discover that measurements assumed to lie on a lower-dimensional manifold embedded in an n-dimensional space, where n corresponds to the number of distinct transmitters, can be extracted using this algorithm.

Isomap requires a single, yet difficult to set [SMR06, SHW07], parameter to function: the size of the neighborhood used for the geodesic mapping. In the extreme case, too few neighbors considered in the projection will fail to capture the geodesic nature of the embedded manifold. Too many neighbors, on the other hand, will subject the algorithm to the same shortcut issues seen with MDS. In the following, we elaborate how Isomap is used in conjunction with a simple linear regression formulation to discover the real-world coordinates of fingerprints along the manifold. We will describe a principled way of discovering a suitable, in the positioning accuracy sense, neighborhood size for the algorithm.

3.1.2 Fitting to Geographical Coordinates

The solution to a manifold learning technique such as Isomap is a point cloud in a space centered around the origin. These points need to be mapped to equivalent geographical locations to serve as the fingerprint model for a positioning algorithm. To provide for this, a small number of *key fingerprints* with known real-world coordinates are embedded into the signal strength dataset before manifold learning is applied. Through this embedding we can both discover the optimal neighborhood size, and anchor the manifold to geographical coordinates.

After running Isomap on the location-enriched fingerprints, we solve a simple linear regression equation to solve the optimal, in the least squares sense, non-linear conversion between coordinates in the origin-centered space and the real world. More formally, we solve this conversion as follows. A *design matrix* X is constructed out of the manifold coordinates whose real-world counterparts are known, the squares of those coordinates, as well as a vector of ones to account for the solution intercept. As an example, in the two-dimensional case, the row i in this matrix would correspond to $[1\ x_i\ y_i\ x_i^2\ y_i^2]$. The addition of squared components ensures that any remaining non-linearity in the mapping can be discovered and accounted for. In other words, X takes the form

$$X = \begin{bmatrix} 1 & m_1 & m_1^2 \\ 1 & m_2 & m_2^2 \\ \dots & \dots & \\ 1 & m_n & m_n^2 \end{bmatrix}, \tag{3.2}$$

where m_i corresponds to the manifold coordinates of key fingerprint i, and n to the amount of key fingerprints in total. Then, given the real world coordinates of the same key points, i.e.

$$y = [r_1, r_2, ..., r_n], \tag{3.3}$$

we can solve the least-squares optimal conversion weights w through the traditional linear equation solution of a matrix left divide

$$w = (X^T X) \backslash (X^T y). \tag{3.4}$$

This w can now be used to convert the rest of the manifold coordinates – provided they are instrumented as described in Equation 3.2 – to real world coordinates through simple matrix multiplication, i.e.

$$y' = X' * w. \tag{3.5}$$

Solving this mapping between domains also provides us with a principled way of discovering the optimal Isomap neighborhood size. Running Isomap with iteratively increasing neighborhood size, we can use the loss incurred from the manifold conversion, i.e.

$$\frac{1}{n} \sum_{i=1}^{n} (r_i - r'_i)^2, \tag{3.6}$$

where r'_i is the solved real world coordinate and r is the known coordinate, to find a neighborhood size that minimizes this loss.

3.1.3 Empirical validation

The described technique was implemented and deployed at the Exactum building in Kumpula, Helsinki. The target environment consisted of hallways and an adjacent open meeting space next to offices and meeting rooms of researchers at the department. Covering an area of 24 m x 7 m, 437 WLAN fingerprints were recorded evenly over the target space. Of these measurements, 38 were labeled with real-world coordinates. An additional 66 measurements were recorded separately as a test set.

Following the procedure described in the previous section, the optimal Isomap neighborhood size was found to be 15, which left an average error of 1.9 m in the training set. Figure 3.2 depicts the results of the manifold calibration process. For illustrative purposes, a rectangular bounding box was converted along with

the manifold coordinates. The non-linearity of the conversion can clearly be seen in its transformed shape, justifying the use of squared components in the design matrix in Equation 3.2. By overlaying the discovered manifold onto the known floorplan of the environment, in Figure 3.3, it is apparent that the fitting has successfully extracted the environment shape, especially along hallways. The open space at the end of the corridor, on the right in Figure 3.3, shows less conformity with the physical restrictions of the environment, however, indicating a degree of non-linearity still present in the embedding.

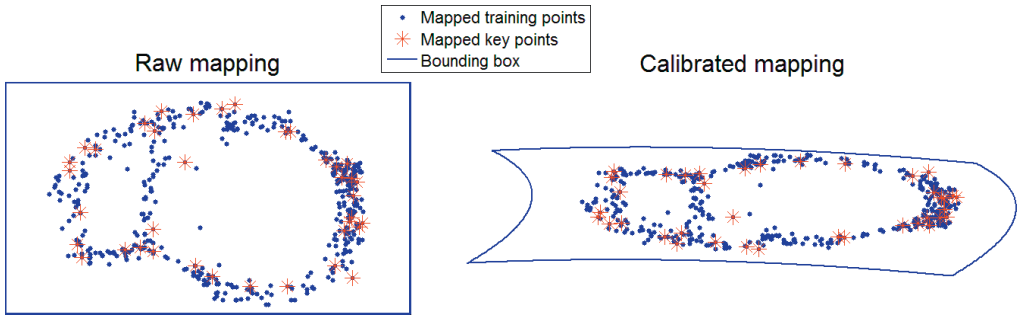

Figure 3.2: Mapping between manifold coordinates and real world coordinates. Previously published in Article II [PRM11].

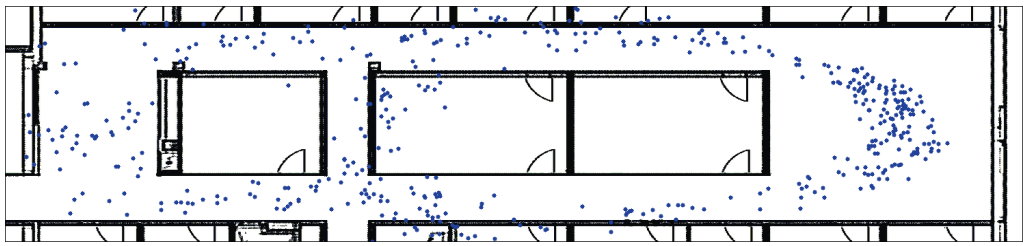

Figure 3.3: Discovered and calibrated 2D fingerprint manifold overlaid onto actual floorplan. Previously published in Article II [PRM11].

Given this reconstructed radio map with newly discovered labels, standard k-NN positioning was used to position test measurements to an average accuracy of 2.0 m and a median accuracy of 1.5 m. This accuracy is comparable with other k-NN approaches, such as [BP00], while requiring only 10% of the signal fingerprints measured to be labeled. This reduction in effort is comparable to more advanced techniques, including deep-learning [LMH18], while using a simpler architecture.

In conclusion, the crucial benefit of this approach is that the vast majority of measurements required for training the signal model could be extracted in a wholly unsupervised way. Though a subset of measurements need to be anchored to real-world coordinates, the effort required for this task is significantly reduced from a complete environment survey and could be performed by the network administrator or enthusiast users in an organic way [PCC+10]. Some of these anchors could even be provided automatically through opportune GPS fixes, as in [CPIP10], or by leveraging PDR and the organic landmarks in [WSE+12].

3.1.4 Discussion

The key points in this study were placed in a uniform way, and the number of points was chosen largely heuristically. As displayed in the open space area, a more sophisticated approach for selecting key points could improve this approach further. For instance, key points could be placed in the center points of clusters determined based on a k-means clustering of the floorplan pixels. This would be equivalent to sampling according to a hexagonal lattice, common in spatial sample design [BOB07]. A similar notion has been explored in [KP03].

This work only considered Isomap for learning the non-linear manifold. T-distributed stochastic neighborhood embedding (t-SNE) [vdMH08], another popular dimensionality reduction technique, could provide interesting future work. Handling residual non-linearity in the embedding was also only investigated with respect to squared components in the least squares formulation. Other candidates for future research could include logarithmic components, or a formulation using aspects of Gaussian kernels.

3.2 Automatic Environment Partitioning

As described in Section 2.3.3, in any practical scenario, most techniques depending on a probabilistic modeling of WLAN fingerprints require a discretization of the target environment. This is performed in order to reduce the effort required for calibration, but also to alleviate the storage and handling of fingerprint information. In practice, this discretization can only be avoided in relatively small environments where the number of required measurements is constrained. Using a single WLAN fingerprint attached to a specific location will quickly become not only prohibitively cumbersome in any environment larger than a single office, but is inherently less robust than using summary statistics to capture the variable nature of the signal. This discretization has taken many forms over the years;

Krumm and Horvitz reduce the environment into a graph with edges between location nodes which contain the probabilistic model [KH04], Haeberlen et al. base their topology on real-world constraints like rooms and obstacles [HFL+04], and Nurmi et al. use a uniform grid in an outdoor scenario [NBK10]. Even in more modern applications the environment topology is often not strictly defined and sampling points are largely heuristic [AER+19]. This can cause uncertainty in the positioning model, as highlighted in Section 2.3.3. This problem of environment discretization is also alluded to in [YA05], which suggests that the granularity of the 2.4 GHz signal could be even less than the wavelength of the carrier frequency, i.e. 12.5 cm. In their study on small scale variations of the signal they noted a difference of up to 10 dB within a 7.6 cm range.

It is then in the best interest to fit the environment discretization not (only) to the physical space, but to the variance of the target signal. Inspired by our previously described work on the spatial dependency of the WLAN signal, Article III [PN12] presents an approach where the non-linearity of the WLAN fingerprint space is modeled, but regularized with real-world constraints. Though the semi-supervised manifold learning approach, described in the previous section and Article II, showed that there is a dependency that is exploitable, it also suffered from heterogeneity of the signal space and there was a clear need to model at least some parts of the environment in a way that more closely represented the signal characteristics. Here, instead of purely projecting the fingerprint space to (what is assumed) the underlying two-dimensional manifold, we instead specifically construct a two-dimensional lattice and *learn* the relationships between signals through the use of a self-organizing map (SOM).

This work provides two main contributions, with several additional insights into the usefulness of this description of the wireless domain. First, we show that using the learned mapping, the compatibility of an existing partitioning scheme and the signal environment can be evaluated and scored. This score can then be used to suggest an improved form of discretization. Second, a fitness measure is described that can evaluate the existing access point deployment in terms of suitability for positioning. This measure can then be used to inform the placement of additional access points, or even potentially suggest an initial access point layout that results in a positioning model with good accuracy. Finally, once the model is constructed, it could allow for a principled way of associating new measurements with the corresponding partition, reducing re-calibration efforts in a semi-automatic way. In the following we briefly introduce the structure and iterative learning cycle of a SOM, after which we describe how it was used for our contribution in two real-world locations.

3.2.1 Self-organizing Maps

A self-organizing map is an early form of neural network, initially described by Kohonen [Koh82]. In our work, we follow the formulation described in [Hay98], specifically. In its simplest form, a SOM consists of a two-dimensional lattice of nodes or "neurons". These neurons are related to neighboring neurons through a distance function – often a Gaussian kernel – and are represented by a weight vector that connects to the input signal. In the following we describe how data is fed into the network and propagated through neighboring connections.

Let k correspond to the number of features in the weight vector, in our case the number of measured access points in the network. We can then describe the input vector (a WLAN fingerprint) in the form

$$\boldsymbol{x} =< x_1, ..., x_k > . \tag{3.7}$$

Correspondingly, the weight vector of neuron j takes the form

$$\boldsymbol{w_j} =< w_{j1}, ..., w_{jk} > . \tag{3.8}$$

Importantly, the weight vector is the same size as our input vector. In this way, each neuron represents a distinct area in the network (environment) through a weighted average of the input signal (fingerprints).

The network is trained by an update cycle where each new input signal is matched to a so-called winning neuron and the residual weight from the mismatch is propagated through the winning neuron's neighborhood, to more closely resemble the input signal over time. Specifically, the winning neuron is selected by the inner product of the input vector and the weight vector of each neuron in the network, i.e. $argmax_j(\boldsymbol{w_j x})$. The weight vectors of the network are then updated by the difference between the input vector and the winning weight vector, and the distance of each neuron from the winning neuron, i^*. Formally:

$$\boldsymbol{w_j}(t+1) = \boldsymbol{w_j}(t) + \eta(t) H_{i^*,j}(t)(\boldsymbol{x} - \boldsymbol{w_j}(t), \tag{3.9}$$

where t is the index of the iteration, $\eta(t)$ is a decay function that decreases the impact of measurements over time and $H_{i^*,j}$ is a function that describes how weights are propagated in the network. This takes the form of a neighborhood distance function around the winning neuron, often specifically a Gaussian kernel:

$$H_{i^*,j} = exp\left(-\frac{d_{i^*,j}}{2\eta(t)^2}\right). \tag{3.10}$$

The choice of a Gaussian kernel ensures that the winning neuron receives the largest weight update and the weight then decreases monotonically towards zero as the distance goes towards infinity [Hay98]. Here $d_{i^*,j}$ corresponds to distance, within the network, between the winning neuron and neuron j. The decay factor, $\eta(t)$, is typically an exponential function, given as

$$\eta(t) = \eta(0)exp\left(-\frac{t}{\beta}\right). \tag{3.11}$$

This factor ensures that most importance is given to the initial iterations of the input signal, but also that the width of the kernel shrinks over time. In our implementation, the number of iterations was set to 100 times the number of neurons meaning the same input vector was provided for learning multiple times over the course of training. The constants $\eta(0)$ and β are learning rate parameters which in our implementation were given default initialization values of 0.1 and 1000, as suggested in [Hay98], respectively.

This iterative learning process is illustrated through an example weight update in Figure 3.4. Neurons in this example lattice are uniformly spaced, meaning the distance within the network reduces to the Manhattan distance between neurons. Because we are interested in the real-world dynamics involved, however, our neurons were placed in a network where the distances correspond to the Euclidean distances between measurement sites in our deployments. This ensured that weight propagation was based on real-world distances.

The weight vectors were initialized using the eigenvectors corresponding to the largest eigenvalues to span the two-dimensional space, i.e. PCA, in order to speed up convergence [Koh97]. Though we have previously shown that PCA (in the form of MDS) might not be ideal for projecting WLAN fingerprints, we will partly overcome this issue through the non-linearity of the weight propagation step of the update cycle. The fact that PCA speeds up convergence is an indicator that the underlying variance-explaining intuition is sound, but a non-linear component is needed to make the domain-specific implementation robust. Though we have shown that Isomap recovers this non-linearity to a great extent as well, it is also distinctly a batch algorithm that needs the entire dataset as input. A SOM is used specifically because of its iterative nature, which provides a principled approach to analyze further fingerprints once the map has been learned. It also allows for a way to describe the environment constraints as part of the model.

The number of neurons in the network could be any $M \leq N$, where N is the number of measurement locations. We use $M = N$ to evaluate an existing

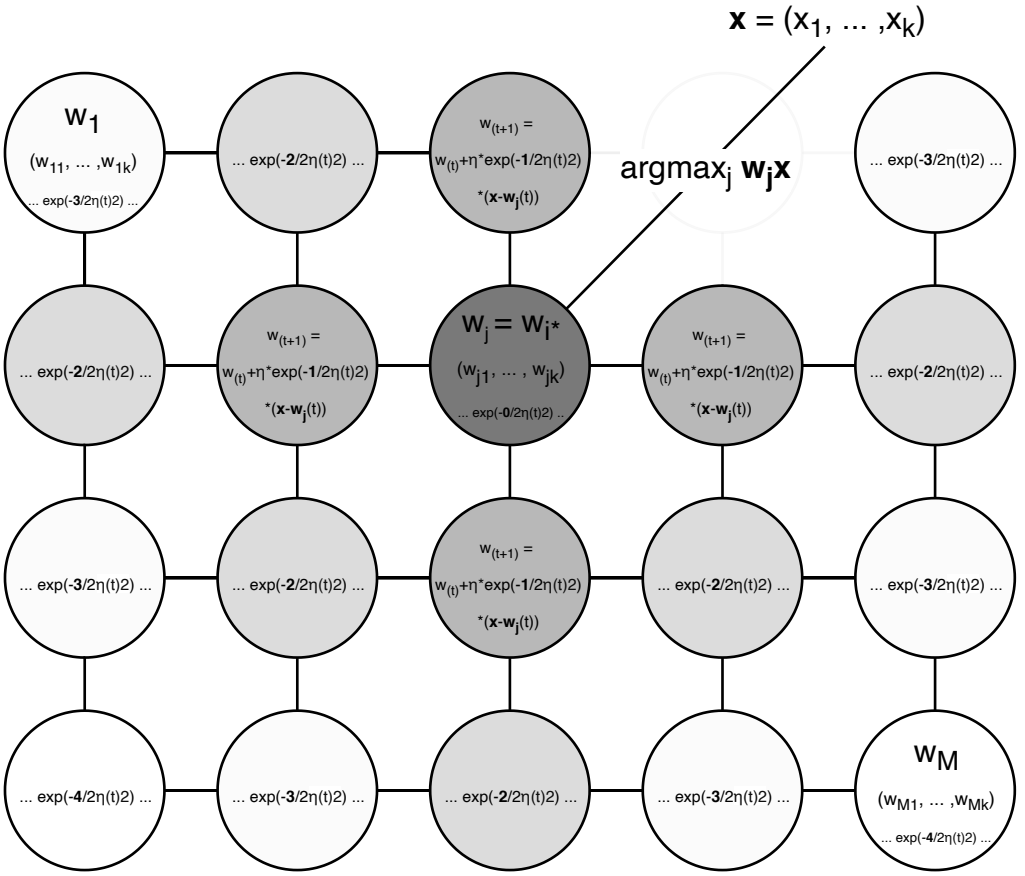

Figure 3.4: Example iteration of SOM algorithm. Weights from the activation are distributed according to distance from the winning neuron to its neighbors.

partitioning of the environment and because we are interested in analyzing the relationship between the learned signal topology and the real-world space.

3.2.2 Dynamic Signal-aware Partitioning

Our first application of the learned signal map is a measure of similarity between regions in the signal space. This measure is then supplied to a density-based clustering technique to provide a new partitioning scheme that adheres more closely to the inherent topology in signal space.

Our intuition for this technique is that related WLAN fingerprints tend to cause a similar set of *activations* in the learned SOM, in terms of the activation function $\boldsymbol{w_j x}$. This same function was used for finding the winning neuron during the training phase. Here, instead of focusing on just the winning neuron, we use the *ranking* of activation values for neurons as a feature for comparison. Specifically, neurons are sorted in a descending order of activation score and compared to each other using Kendall's τ correlation. This correlation measure compares the similarity between two rankings. Formally, τ is defined as

$$\tau = \frac{\sum_{i=1}^{|P|} \mathbb{1}(P_{g1}(i) = P_{g2}(i)) - \sum_{i=1}^{|P|} \mathbb{1}(P_{g1}(i) \neq P_{g2}(i))}{|P|}, \tag{3.12}$$

where P_{g1} and P_{g2} correspond to the possible pairs of the ranks of activations for neurons $g1$ and g_2, and i to the index of a specific pair of neurons in the ranked list. The indicator function $\mathbb{1}$ returns 1 if the corresponding clause is true and 0 if it is false, and $|P|$ is the cardinality, or size, of the set of pairs, defined as the binomial function $\binom{M}{2}$, i.e. all possible ways to choose a pair from the set of M neurons. The clause $(P_{g1}(i) = P_{g2}(i))$ is considered true if the pairs agree, i.e. both sets rank the pair (g_a, g_b) in the same order; either $g_a > g_b$ or $g_a < g_b$. Similarly, $(P_{g1}(i) \neq P_{g2}(i))$ is true if there is a disagreement of the ranking order.

As with the manifold learning approach presented in Section 3.1 and Article II, we have empirically observed multiple modes in WLAN fingerprint similarity, where two neurons, or fingerprints, resemble each other even though they are not spatially co-located. Whereas this could cause short-circuit issues in the linear projection techniques earlier, in the case of creating a signal map this could cause two distant regions to be merged if we rely exclusively on the correlation measure during the clustering phase. We have thus chosen a regularization scheme where an added constraint of physical proximity is added to the similarity measure. Specifically, we use the Euclidean *path distance* between the known regions, i.e. the length of the shortest path between target regions, as a constraint.

Using this physical distance measure for region similarity, we perform a density-based clustering of the existing partitioning, which was used as the framework for our SOM lattice. Specifically, we use DJCluster [ZFL+04], a technique closely related to DBSCAN [EKSX96] that attempts to alleviate some of the performance issues of the latter approach. The clustering algorithm works by finding, for each region g_i all regions g_* within distance d_ϵ in the physical space. Once these regions have been found, we further prune the set of points by removing regions that are not proximate in the signal space; i.e. our correlation measure described earlier. Our criterion for pruning is decided using a significance test

of the rank correlation coefficient. Two regions are considered co-located in signal space if the correlation value exceeds $\tau^* = 0.64$, which was determined by a significance level of 0.01 in a one-tailed significance test, assuming a sample size of 10. We considered this a suitable number of measurements to ensure significant correlation. Once two regions are concordant in terms of these two measures, they can be merged and any two partitions with common points can be progressively merged to form larger partitions. For the final partitioning, the geographical shape of partitions is determined by the bounding box of the regions contained within.

3.2.3 Region Fitness

The constructed non-linear signal map can also be used for other forms of analysis. Next, we provide a way to score individual regions of a partition through a measure of region *fitness*. This measure can be used to not only evaluate poorly performing regions – in terms of suitability for positioning – but also as a criterion for access point placement.

Whereas previously we compared the activation rankings of two regions, here we analyze the rankings of individual regions and examine the spatial distribution of the regions in the ranking. The intuition lies in the observation that poorly performing regions tend to have measurements that correlate with spatially distant regions. In this case, the signal distribution differs distinctly from nearby regions, which increases uncertainty for positioning algorithms that use the region's measurements in their model of the environment. In practical terms, this reduces the robustness of location-based services, which is critical for the consistency of the end-user experience.

Specifically, for each region g_i we find all regions g_* with significant correlation. The same criterion for similarity is used here as before, i.e. a threshold of $\tau^* = 0.64$. Defining a set of regions with significant correlation as G, our fitness score is determined by the average distance to these regions. Formally, the fitness score is given by

$$F(g_i) = \sum_{j=1}^{|G|} \frac{d(g_i, g_j)}{|G|}, \tag{3.13}$$

where $d(g_i, g_j)$ corresponds to the physical (shortest path) distance between region g_i and region g_j and $|G|$, as before, denotes the cardinality of the set of compatible regions. This measure is illustrated in Figure 3.5. Note that this formulation means a large score indicates low fitness.

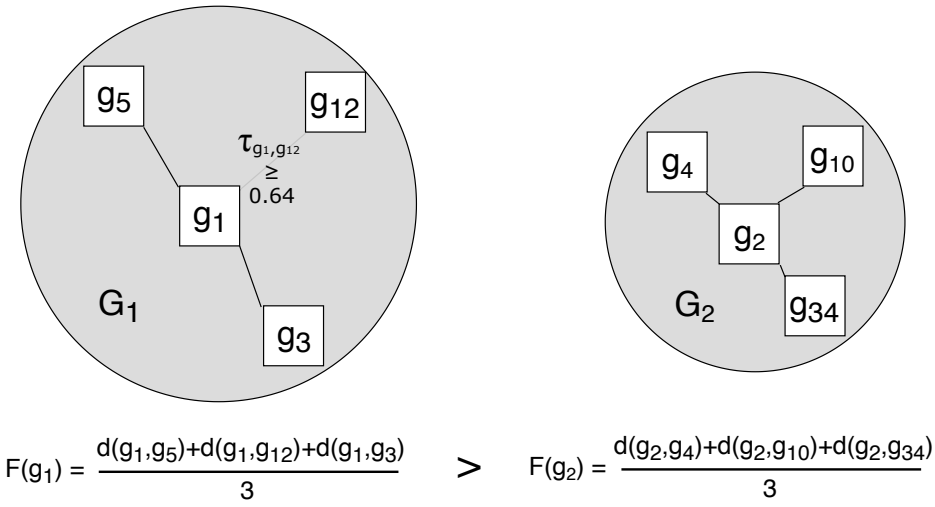

Figure 3.5: Fitness comparison between two example regions indicates that region g_1 has a worse fit than region g_2 because the average distance to other regions with similar activation patterns is greater.

One practical use of this score is to inform the placement of new access points for improved positioning accuracy. The intuition behind this improvement relates to the way the WLAN signal propagates, as described in Section 2.2.2. Because of the logarithmic nature (in the decibel domain) of the path loss propagation, the most informative – in terms of variance – section of the propagating signal is close to the access point. Thus, the optimal placement for a new access point is close to an area that is known to be the least concordant with its neighbors. Placing an access point in, or close to, a partition with low fitness ensures that fingerprints in the vicinity will be enriched with location-dependent information, while lessening their similarity with parts of the environment that are not relevant for the partition in question. In Figure 3.5 this would be graphically equivalent to decreasing the size of the circle corresponding to the set of regions G.

3.2.4 Empirical Validation

To validate our partitioning scheme and fitness measure, SOMs were constructed and analyzed based on measurements from two large-scale supermarket environments, one in Helsinki, Finland and one in Saarbrücken, Germany. Next, we briefly summarize the main results; see Article III for more details.

The initial partitioning and access point placement of the Helsinki environment is displayed at the top part of Figure 3.6. This construction was previously used for the grocery store navigation application discussed in Section 2.4.

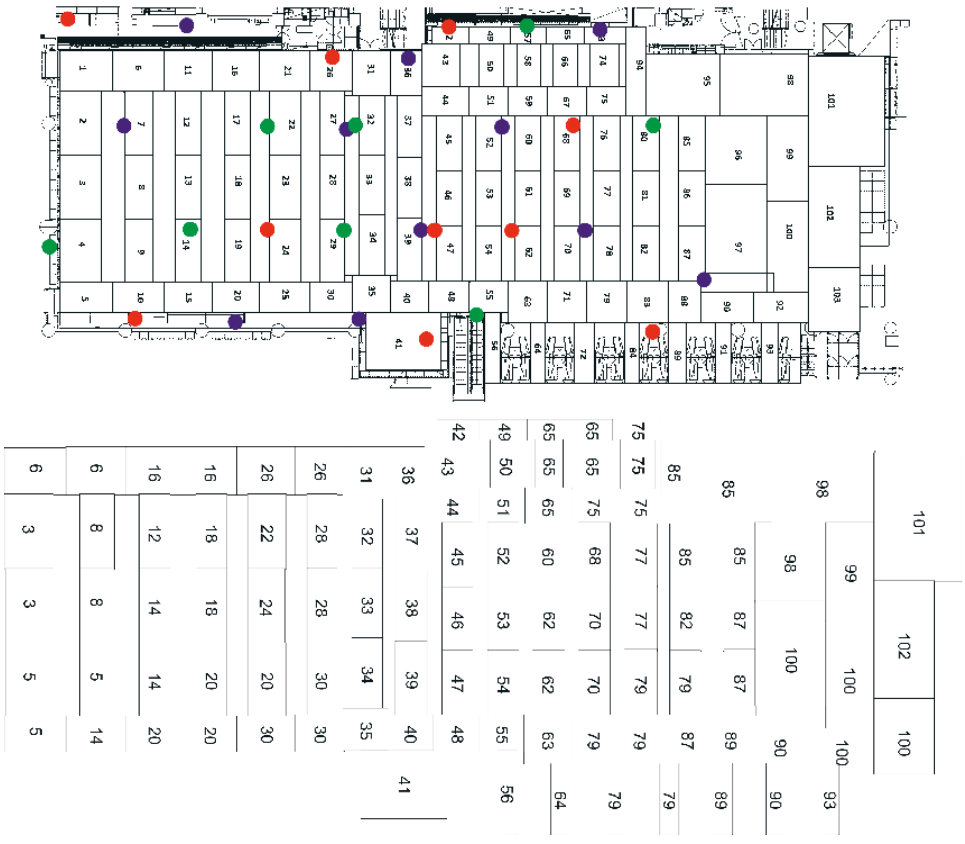

Figure 3.6: Initial partitioning, with locations of APs and their channels represented by colors, in the supermarket in Helsinki (top) and the suggested reconfigured partitioning (bottom). Previously published in Article III [PN12].

For each region in this environment (103 regions in total), 30 WLAN fingerprints were measured, and a SOM was trained as described in Section 3.2.1. As a proof-of-concept of our similarity measure, we first evaluated the dependency of the rank correlation between region measurement activations and the distance between regions measured in region transitions, which correspond to real-world

distance at an approximate rate of 3-4 regions to 10 meters. The results of this evaluation, performed for both test environments, are displayed in Figure 3.7. It is clear from these graphs that our correlation coefficient shows a distinct decrease as the distance increases, motivating its use as our clustering metric.

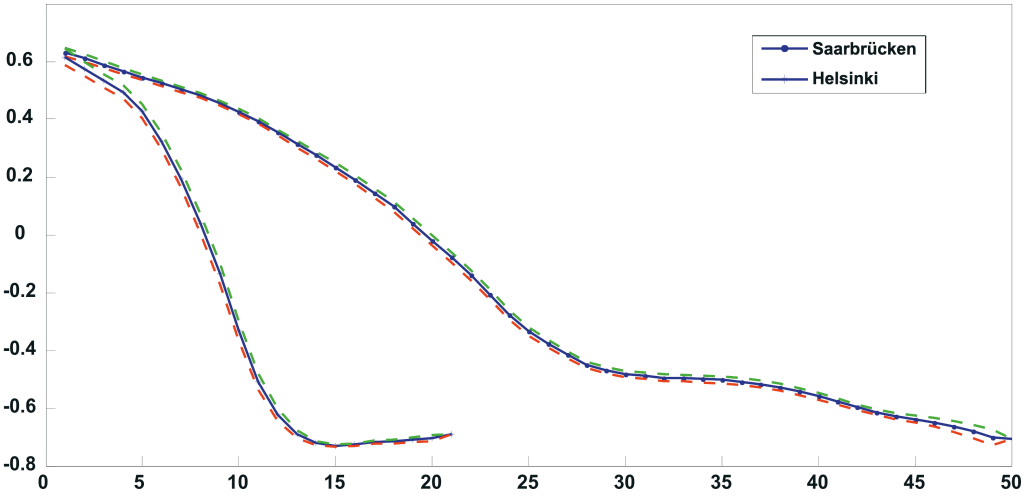

Figure 3.7: Dependency between activation ranking correlation and spatial distance is clearly apparent in the two experimental environments. The Y-axis corresponds to our rank correlation coefficient, and the X-axis is the number of "hops" between target regions. Previously published in Article III [PN12].

Next, we performed a re-partitioning of the Helsinki environment using the technique described previously, using a distance threshold of $d_\epsilon = 10\ m$, determined based on our earlier estimation of the signal's dependence on distance. The result of this new partitioning is displayed at the bottom of Figure 3.6. Smaller adjacent regions have been merged to form larger continuous regions, and the Euclidean regularization has prevented any disjoint clusters from forming. To validate this new partitioning, a dataset measured with a separate device was used to test two instances of the positioning system initialized with each environment model. This experiment resulted in a slight increase in median accuracy, but a slight decrease in 95th percentile (worst case) accuracy. More importantly, the runtime of the positioning algorithm was decreased by about 60%, a distinct improvement in model complexity. This suggests the initial environment partitioning was more complex than what the signal space could support, and regions could be merged without impacting the positioning accuracy significantly.

Finally, the use of the region fitness score for access point deployment was evaluated by instantiating the positioning system with datasets with increasing density of access points. Specifically, the original training dataset was reduced to measurements from two access points at opposing ends of the target environment, and measurements from other access points were added iteratively until the dataset contained measurements from 10 access points. We compared a random placement approach to one where the access point was chosen based on proximity to a region with low fitness (corresponding to a high fitness score, as described in Equation 3.13). The results are displayed in Figure 3.8.

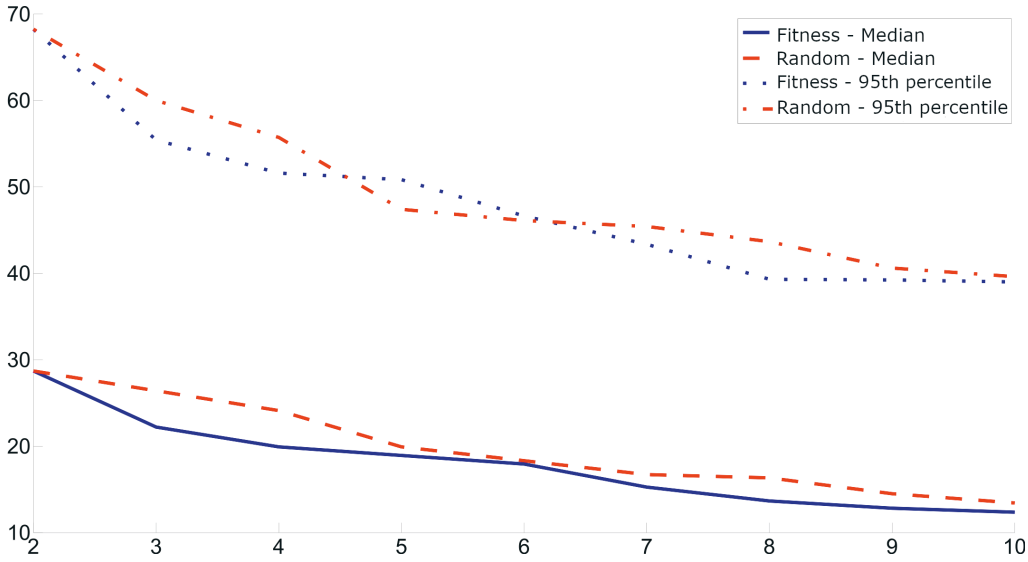

Figure 3.8: Evolution of positioning accuracy in meters as more access points are placed in the environment. Strategy based on region fitness compared to a random approach. Modified from previously published version in Article III [PN12].

The tendency of both approaches is to increase positioning accuracy, as expected. The fitness score clearly helps to choose better access point candidates, consistently outperforming the random approach in terms of median accuracy and in all but one case in worst case accuracy. This could potentially be used to deploy a WLAN positioning infrastructure from the ground up, but also to enhance an existing deployment. The traditional incentive for WLAN positioning is the existing WLAN network deployed for communication purposes. In some environments, however, this initial access point density might not suffice

to provide a positioning system with high accuracy [WTD+13]. Our approach could then be used to evaluate the existing signal topology and to suggest the necessary additions in a cost-effective way.

In conclusion, through this approach we have discovered a way to evaluate the WLAN fingerprint consistency in specific locations as well as a technique for remediating a partitioning that is not serving its intended purpose. This has the effect of decreasing correlation between regions of the environment that are not spatially co-located as well as reducing the complexity of the environment model, which can have a significant impact on latency, especially on performance constrained devices. Furthermore, the region fitness score could be used not only to inform the addition of access points in low-performing areas, but also to initialize construction of a positioning system infrastructure.

3.2.5 Discussion

One limitation of this work is the lack of a thorough comparison to previously established clustering techniques, like k-means. However, generic approaches such as k-means are not as easily encoded with the environment structure embedded in the lattice of the self-organizing map, and do not provide an intuitive way of assigning cluster membership once the clustering has been performed. The choice of the activation function was also directly based on literature, and does not directly correspond to the Euclidean distance, which is the more typical fingerprint distance measure in this domain. However, we found that this activation function served the purpose well, which is reflected in the presented results.

Through the training of a nonlinear signal mapping the presented approach has indirectly created a positioning model, where positioning could be performed by finding the region with the strongest activation to new signal strength measurements. More training data could be automatically associated with the proper region, which could benefit positioning algorithm robustness. This latter aspect was touched upon, but not explored in this work and could provide an interesting avenue for future research.

Later on, we will discuss the complex interplay between the number of access points in the environment, sources of interference and the impact these factors have on the resulting positioning accuracy. Placing an access point without also considering the WLAN channel layout of neighboring access points runs the risk of increasing the congestion of channels, which has been shown to decrease positioning accuracy. However, providing suggestions for alternate locations of access points could also help mitigate the effects of interference that is otherwise

unavoidable. In this sense, this contribution allows for interesting future work, but also provides potential novel uses in noisy environments.

3.3 Related Work

The interest in calibration-free or zero-effort WLAN positioning systems extends beyond this thesis, as other publications have explored similar trends. For instance, rather than reduce the number of *labeled* fingerprints, [LKE08] instead reduce the number of fingerprints needed per label. While this can shorten surveys per location considerably, the manual effort in traversing the environment and labeling locations is still extensive.

In many cases, the contributors to calibration-free approaches were originators of positioning systems like RADAR [CPIP10, RCPS12] and Horus [WSE+12]. Techniques like Gaussian processes that have initially been used for positioning [FHF06], have later been harnessed to perform SLAM (*simultaneous localization and mapping*) without known ground truth coordinates [FFL07]. A related topic has been the so-called "organic" positioning system, where calibrations for the positioning system are crowdsourced [PCC+10].

Though these efforts have shown great promise in reducing the calibration effort, and even eliminating it to some extent, the contributions of this thesis explore the nature of WLAN positioning measurements in more detail, and could potentially be used in a complementary way to previous approaches. For instance, the manifold learning of the signal space described in Section 3.1 could act as the foundation for a more organic approach to crowdsourced calibration. The main contribution of that work is that most WLAN fingerprints in the environment could be collected in a purely unsupervised way, leaving the reduced labeling effort either to dedicated experts or to enthusiast users. In this sense, the contribution is complementary rather than orthogonal.

In [FFL07] a Gaussian Process Latent Variable Model is initialized using the Isomap algorithm, also presented in Section 3.1, to avoid local minima and improve performance before the gradient descent step is applied. Since the resulting manifold does not have a well-defined mapping to real-world coordinates, the system is evaluated by how accurately it can detect the user returning to a previously visited location. One of the main contributions of our work is the semi-supervised mapping of the signal space to real world coordinates, which means the technique in [FFL07] could likely be augmented in a similar way in order to provide a more generic solution.

Both EZ [CPIP10] and Zee [RCPS12] are based on solving the location based on inherent environmental constraints. EZ uses the path-loss model (see Equation 2.2 in Section 2.2.2) to deduce the distance to a set of access points to narrow down the potential locations within which the user could reside, and connects the mapping to real-world coordinates through opportunistic GPS fixes. Zee, on the other hand, uses inertial sensors in smartphones along with an augmented particle filter to use the floorplan itself as a constraint. In other words, given a motion trace, the number of specific locations in the environment that matches its shape is limited and in the limit the location is uniquely defined. In contrast to the approaches presented in this thesis, additional sensors are required for a location fix. Again, the approaches here could act in a complementary way to our work. The GPS fixes of EZ or floorplan constraints of Zee could provide anchor points for the learned manifold to resolve real-world coordinates. Environment-specific landmarks, such as entrance gates in a transit station [EYU+16], have also previously been extracted with high accuracy, providing yet another opportunity for manifold anchoring.

UnLoc [WSE+12] provides a more generic scheme for determining so-called "organic landmarks" in terms of smartphone sensor readings. By discovering and locating specific anomalous sensor readings in the environment – e.g. accelerometer traces of elevator rides or areas with great flux in magnetometer readings – PDR can be (re-)bootstrapped to reset the drift of the sensors. Even in this work, however, at least one real-world fix is required for initialization, and the underlying positioning technique is based on PDR. The added requirement of multiple sensors, with unknown differences from one vendor to the other, also adds a level of uncertainty. Nevertheless, when such landmarks are available, the manual labeling effort could be reduced even further.

In recent years, many contributions in the field of automated model construction have employed deep-learning architectures to reduce deployment effort. For instance, in [LMH18] it is shown that linear learners such as PCA are not able to learn from unsupervised data as thoroughly as deep neural network architectures. These discoveries align with our previous established sentiment that the inherent non-linearity of WLAN fingerprints can benefit from techniques that can take this into account. In [LMH18], deep-learning is shown to reduce the need for labeled data to as low as 10%, which is comparable to our results in Section 3.1.3 despite our work employing a much simpler architecture. In [CA19] as few as 5% of labels are utilized for competitive levels of accuracy by employing Variational Autoencoders. In order to provide such robust results, however, separate encoder-decoder pairs are learned for each geographical dimension (lati-

tude, longitude, and elevation) and prediction results are projected to the convex hulls of the closest labeled neighbors. As with any deep-learning approach, these advances also require a careful tuning of parameters for each use case, might not necessarily generalize well, and are inherently imbued with a level of stochasticity that could prove difficult to mitigate in real-world use cases. We will explore these themes further from the perspective of interference detection in Section 4.3.

The closest to our work in Article III, described in Section 3.2, are [LSE10] and [LKHK09], which present a similar region-merging strategy based on a novel similarity measure. The proposed similarity measure determines the intersection between Gaussian distributions constructed from the measurements of each access point and then normalizes based on the number of total access points heard within the region. Similar to our approach, clustering is then performed by merging nearby regions until no two regions are similar enough to reach a predefined threshold. The work in Section 3.2 differs in key ways, however. Not only is the signal mapping learned with physical constraints in mind, an additional environment regularization limits the physical range of regions to merge. This merging is further based on a statistical, rather than empirical, threshold. Furthermore, the same signal model is used to determine a measure of fitness for a region, which could be used for access point placement. The concept of region fitness is also explored in [LKHK09], but no access point placement validation is performed. Finally, our work was evaluated in complex, open environments which are more prone to violate physical constraints in signal space.

A region similarity measure based on Gaussian distributions has recently also been described by [HTKM17]. Here, reference points in the environment are modeled as vertices in a graph, with edges between them weighted by the Cauchy-Schwarz pdf divergence between Gaussian mixture models constructed from signal strength time series in the reference points. Clustering itself is performed using the NCut algorithm, which utilizes an enhanced version of the k-means algorithm. Finally, the optimal number of clusters is found using the Akaike Information Criterion. This environment partitioning mainly serves as a criterion for choosing informative access points subsets for each region. This filtering of access points results in improved positioning accuracy, though the fitness of the discovered regions is not evaluated as such.

Chapter 4

Detecting Competing Technologies

The previous sections have described aspects of a WLAN positioning system that concern the model creation and initialization phases.Yet, another challenge remains, which could affect all phases of the WLAN positioning lifecycle. Many competing technologies occupy the same frequency band, but may not consider existing WLAN networks when they operate. In extreme interference cases even the wireless beacons that most WLAN fingerprinting systems depend on will not be transmitted, let alone received. In one study on the impact of concurrently transmitting Bluetooth beacons, the accuracy of a WLAN positioning system was reduced from 2.5 meters to 5-6 meters in the proximity of strong Bluetooth transmitters [PLC+17], mainly due to measurements not being received – particularly in the weaker range of signal strengths. Besides this direct impact on positioning, the initial channel design of the network might need to be designed around a persistent source of interference, which in itself has been shown to affect positioning accuracy.

In the following we will discuss the overall impact of interference, both in terms of communication and from the point of view of WLAN positioning, as well as two approaches to detecting some of the potential interfering technologies. In this, we summarize the contributions of Article IV and Article V.

4.1 Non-WLAN Interference

Though the WLAN protocol is based on a public standard, the underlying frequency bands (e.g. 2.4 GHz and 5 GHz) do not require a license to use. This has resulted in competition of frequency use between technologies, especially for 2.4

GHz, since its use for ISM (industrial, scientific, and medical) applications dates back to 1947 [FCC47]. In the internet-of-things (IoT) domain, some candidate frequency bands exist which do not overlap with WLAN use [WS19]. Technologies like Bluetooth, which to some extent is designed for co-existence with WLAN [GWGS07], and ZigBee (specified in IEEE 802.15.4), however operate on the same ISM band. Especially in the industrial IoT domain, the use of these technologies could prove difficult to reconcile with existing WLAN networks.

The unlicensed nature of the band has also attracted devices without a standardized protocol. Though many modern devices use either WLAN or Bluetooth (in particular Bluetooth Low Energy, BLE), some devices still use proprietary and non-standardized communication schemes. Such devices include wireless video cameras, frequency hopping baby monitors as well as headsets and computer peripherals. Further exacerbating the issue are devices that use the band for purposes other than transmitting, like microwave ovens or sulfur lamps [SNSK05] where energy is emitted into the 2.4 GHz spectrum due to poor shielding. In benign cases, interference could be generated unintentionally as part of standard operation, as is the case with some USB 3 devices [Int20]. For adversarial purposes a wide range of options exist, including the transmission of random bits with no defined protocol [GLY14].

For conformant devices, the WLAN protocol specifies a contention scheme where a client can hold off from transmitting if another WLAN transmission is sensed on the channel at a specific power level. Since this behavior is part of the specification it is a less severe issue, but will still reduce the overall capacity of the channel. For non-WLAN devices, a WLAN client is not able to demodulate and decode the transmission. An alternate power threshold is then defined in the specification [IEE16], at which the WLAN client has to cease transmitting [MBMS17]. In this scheme, the client waits an integer number of time slots before sampling the channel again [GWGS07]. A particularly strong source of interference can thus dominate the channel and block the transmission of any WLAN frames and even interference at lower power levels can severely impact the throughput of the wireless channel [RPB11].

4.1.1 Impact on Positioning

Any non-WLAN transmission stronger than the received power from a beacon frame will effectively obscure the transmitting access point from the client's view. This has the effect of shortening the range within which the access point can be heard, decreasing the amount of information available for positioning. Missing

beacon frames from specific access points has different effects depending on which phase of the WLAN positioning lifecycle the source of interference is introduced.

Interference during training

Though most positioning systems are able to handle a single access point missing from measurements, this masking decreases the available information in the environment, especially if the access point has been deployed specifically for positioning purposes (as described in Section 3.2).

Whereas an access point not being heard around the limits of its transmission range is expected, a sudden gap in connectivity in a signal range that otherwise would be considered strong will cause a discontinuity in the signal space that is difficult to model. This could have a direct impact on the techniques described in Chapter 3, since the spatial dependency or similarity of RSSI fingerprints might be less distinct, or even no longer apply. On a similar note, any system relying on propagation modeling will also have to re-evaluate the assumption that an access point can be heard in a specific location based on physical models.

Finally, since strong interference effectively masks beacon transmissions, a mismatch between the distribution of signal strengths measured during training and testing could appear. This would have the effect of further skewing the signal distribution away from the Gaussian assumption, the implications of which we discussed in Section 2.3.3.

Interference during use

As time passes since the calibration of the positioning system, opportunities arise for the introduction of non-WLAN sources of interference. Given that the positioning model is tightly integrated with the signal strength of access points in specific geographical locations, a new source of interference could have a significant impact on the resulting positioning accuracy. In [YA05] an important component of the light-weight positioning is the ability to cluster access points based on a sorted list of signal strength measured during the testing phase. A source of interference could at this point impact large sections of the model that depend on the measurements from a specific access point.

Intermittent interference

Many wireless devices depend on a frequency hopping or periodic transmission scheme. This can cause issues both during the training and deployment phases.

An access point that is intermittently available will cause uncertainty in the positioning model, especially for ones relying on a specific number of samples. In many cases [BP00, YA05] whether or not an access point is heard can be a significant source of information, and positioning models that do not account for this to happen in an intermittent way will likely suffer from significant uncertainty where the impacted access point is concerned.

Practical implications

Only a small number of works have considered the impact of interference on positioning, with most works focusing on the data rate of the wireless channel instead [RPB11]. Besides the previously described study on co-existence with Bluetooth [PLC+17], inter-technology interference (also known as *co-channel* or *adjacent channel interference*) has been shown to impact positioning performance [CBM10]. In the 2.4 GHz frequency band only 3-4 channels (depending on the regulatory domain) can be used concurrently without overlapping frequency ranges. This limitation means that in addition to access point placement, the channel assignment also needs consideration in a functioning network. For instance, this requirement informed the access point configuration in Article I and Article III; see Figure 3.6 for a description of the channel assignment. For the purposes of positioning, choosing an appropriate scheme (minimizing co-channel interference) has been shown to improve positioning accuracy up to 10% [CBM10]. Though the contributions of this thesis do not address this kind of interference specifically, the work in [CBM10] indicates that any kind of channel congestion has a direct impact on positioning performance. External interference could constrict the available set of channels even further, impacting a wider range of the environment than the original source of interference.

In commercial deployments, a typical minimum requirement for a functioning positioning system is that the client device can hear three access points concurrently throughout the environment [SBT+06], especially at a high enough signal strength, [Cis20] suggests a minimum of -75 dBm. Such a precarious deployment would obviously be vulnerable to even one missing access point, as it would likely be impossible to unambiguously determine a position estimate based on two access point measurements alone. In academia, reducing the number of available access points is a staple of the performance evaluation for WLAN positioning algorithms. For instance, a set of 3 to 6 access points was found in [AER+19] to suffice for sub-3-meter accuracy, but performance continued to improve until 20 to 40 access points were included in the experimentation. This indicates that al-

though a minimal deployment can provide sufficient performance, accuracy tends to increase the more access points that are available.

As discussed previously, any decrease in positioning accuracy or robustness will directly impact the granularity with which the location-based service can be designed. In the navigational aid described in Article I, the direct consequence is an inconsistent navigation experience, which would serve to undermine much of the progress that our other contributions hope to achieve. In addition, because the failure modes of positioning under (cross-technology) interference are poorly understood it is especially important to recognize it.

4.1.2 Interference Classification

Whereas intelligent systems built around WLAN measurements can refer to a predefined specification and certified devices, many competing technologies constitute interference from the perspective of a WLAN client. If the target device follows a known standard or other specification, detection and identification in the simplest case requires demodulating the signal and recognizing standard syncwords – a known sequence of data at the beginning of the transmission – of known communication technologies. Even in this case, a separate identification scheme is required for each individual technology, which can quickly become cumbersome. Ideally, then, an interference detection scheme could learn to classify sources of interference based on received traces of transmissions. This would have the added benefit of detecting devices without specifications or intended transmitting capabilities to begin with.

Interference detection has previously been performed through cyclostationary signal analysis [HK11], packet analysis [KNS14] or technology specific features, like the 60 Hz (i.e. AC power) periodicity of microwave ovens [WOK14]. In many such cases, detection typically requires dedicated hardware for analysis and performs classification based on device characteristics, or is based on proprietary algorithms. A more widely applicable strategy is then to perform this same process through machine learning on top of measurements from off-the-shelf WLAN clients. This greatly increases the number of devices and scenarios that can be supported, and decreases the amount of time spent on adding new devices. One of the earliest approaches is AirShark [RPB11], which performs classification using decision trees for 8 different devices and a set of device-specific features. Among the features described are the spectral signature, i.e. the frequency domain representation of the device, duty cycle, so-called pulse statistics and sweep detection (specifically in use for microwave ovens).

Nevertheless, even approaches such as AirShark for the most part require hand-engineered features and require separate classifiers for each target device category. Ideally, interference detection should work on measurements made with off-the-shelf hardware and require minimal investment to add new devices to the detection model. A promising technique for this is to treat a set of WLAN band spectrum sweeps, or a *spectrogram*, as an image and use a convolutional neural network (CNN) to perform detection similarly to image recognition.

Other works have also considered such networks for interference (or technology) detection. In many cases, however, these works have either focused on simulated signals of transmitters [RSIC18, DWWZ18] or a constrained set of technologies with known protocols [BMR17, MRSN18, KKMD18]. Our contributions focus specifically on household items, including transmitters with unknown (or no) protocols, as well as measurements in a real environment.

4.2 Deep Learning for Interference Detection

The work in Article IV [LPK17] develops techniques for detecting sources of interference from WLAN spectrum sweeps, collectively termed spectrograms. Detection is performed using convolutional deep learning, which has been successfully used for spectrogram analysis in the speech recognition domain [AMJ+14]. The main attraction for this kind of learning architecture is that it can be constructed without device-specific feature extraction. This feature representation is instead learned, and we show that parts of the network can even be reused when learning new devices, significantly decreasing the amount of processing needed per device. The labeling process, however, still requires manual effort. Since the specific times when the devices transmit cannot be determined, the spectrogram needs to be manually inspected to ascertain the specific instances when transmissions occurred. This effort scales poorly with the number of samples and target devices. The main contribution of Article IV is the introduction of structured pseudo-labels. Previous work on traditional pseudo-labels has shown that more training data can be extracted from samples with uncertain labels [Lee13]. The work in Article IV shows that adding a temporal aspect to this labeling process further improves accuracy with less data required for each added device.

4.2.1 Data description

To motivate the choice of learning architecture, we first describe the features of the data used for learning and detection. The data consists of spectrum sweeps

provided by a spectrum analyzer from the 2.4 GHz frequency band. Each sweep consists of a set of FFT bins, each bin containing the measured power within a specific frequency interval in the 2.4 GHz band. An example set of measurements from a wireless baby monitor are displayed in Figure 4.1.

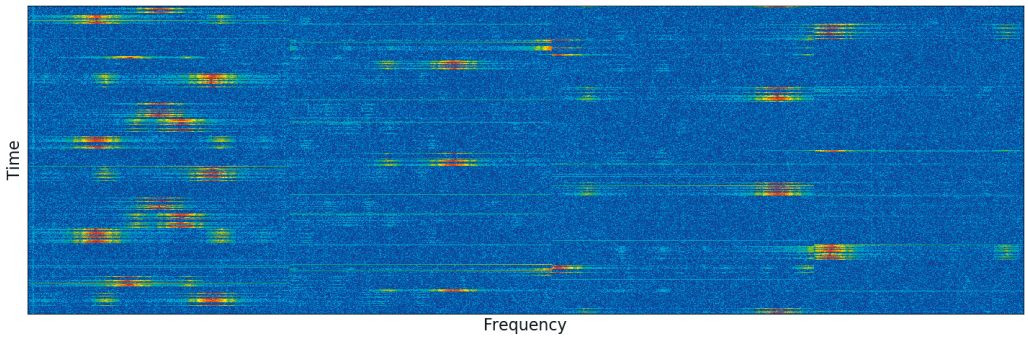

Figure 4.1: Spectrum samples of a wireless baby monitor. The time axis covers roughly 30 seconds of measurements, while the frequency axis covers the 2.4 GHz band. Modified from Article V [PNN20].

As Figure 4.1 shows, individual sweeps of the spectrum – corresponding to the horizontal lines in the image – are not sufficient for recognizing a source of interference since its use of the spectrum might vary over time. Because of this, analysis should be based on multiple sweeps over time.

Given this description, an intuitive way to capture both the frequency dependency and temporal aspects of the signal is to learn a two-dimensional feature representation of the data. This makes convolutional neural networks a particularly good fit since they can learn features that span more than one input [LB98]. These networks are a state-of-the-art machine learning approach that have been shown to yield good results in a wide variety of tasks, particularly in domains where the two-dimensional representation of information is relevant, such as for image recognition [KSH17] or speech recognition from spectrograms [AMJ+14].

4.2.2 Convolutional Neural Networks

Convolutional neural networks, initially nicknamed *neocognitron* [Fuk80], expand upon the traditional artificial neural network by learning so called *kernels* or *filters* instead of linear weights. In the classic artificial neural network, a *multilayer perceptron* (MLP), neurons in the network are connected to each other through

weights and non-linear activation functions. A typical activation function is *tanh*:

$$f(x) = \frac{exp(x) - exp(-x)}{exp(x) + exp(-x)}, \tag{4.1}$$

or, more commonly in recent work (including ours), the rectified linear unit (ReLU) which simply forces positive values through $f(x) = max(0, x)$. At each layer of the network, the activation of each neuron is calculated as

$$n_i = f(\sum_{j=1}^{k} w_j * n_j), \tag{4.2}$$

where k is the number of neurons in the previous layer, n_j is neuron j in the previous layer, w_j is the weight of the connection arriving from neuron j, and f is a non-linear activation function. This activation function is what makes the neural network a non-linear learner: if the activation function is replaced with the identity function it reduces to a linear model [HTF01].

Figure 4.2 depicts a simple MLP network with an input layer, two hidden layers and an output layer. The network learns through a partly unsupervised learning process by passing information from the input layer through a set of hidden layers and reweighting connections based on feedback from the output layer. Specifically, after an iteration where the output layer has calculated a score based on the weighted sum of previous activations, the loss of the network is calculated by comparing the values of the output layer with the corresponding known labels of the input data. This loss is used to adjust the weights of the network through *back-propagation*, which can be implemented using a form of gradient descent [LeC88]. This adjustment alters weights in correspondence with their contribution to the loss. When the next set of inputs is passed through the network, it is likely better equipped to predict the proper class. Feature learning then happens through the weight adjustment in the hidden layers over time as more inputs are evaluated and losses are back-propagated.

In the case of CNNs the weights of the network correspond to filters of varying size. In the image processing domain these filters are also known as kernels, masks, or convolutions. By passing such a filter over the pixels of an image, through steps called *strides*, one can perform operations such as smoothing and edge detection in a systematic way. For instance, a typical 3x3 edge detection filter might have the following shape [Smi97]:

-1/8	-1/8	-1/8
-1/8	1	-1/8
-1/8	-1/8	-1/8

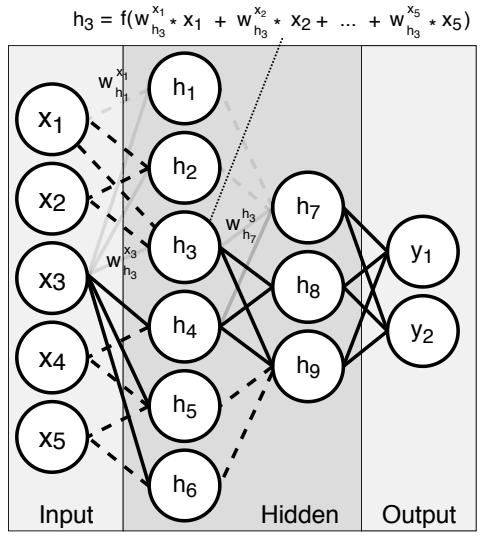

Figure 4.2: Traditional multilayer perceptron neural network. Some connections and labels have been left out for clarity. The super- and subscripts of weights denote the neurons from and to which activations arrive, respectively.

Here, each neighbor of the center pixel is multiplied with -1/8, the center pixel with 1, and the result is summed to provide the new convolved pixel value [NA12]. A CNN for image data might be constructed with multiple filters per layer, which allows it to learn an internal feature representation of the objects in the image. Whereas a learned filter for a network trained on images might correspond to edges of objects, in the domain of WLAN spectrum analysis such a representation encapsulates the correlation between different parts of the frequency band as well as the temporal aspects of this dependency.

Typically, CNNs also contain *pooling* layers which act to downsample the data between convolution layers and help the network learn more generic features and avoid overfitting. For instance, a max-pooling layer passes 2-dimensional windows over the convolved layer and returns the maximum value over the provided window range. Much of the rest of the CNN works in ways similar to the MLP presented above: activation functions such as ReLU are applied on top of the filters and the network learns through backpropagation. In most classification tasks the network employs an MLP, specifically a fully connected layer, for its final layers in order to reduce the multidimensional input into discrete categories.

One of the main benefits of CNNs is their *shift invariance*. In other words, a CNN can learn to recognize features regardless of where they appear in the input. In addition, whereas an MLP network would have no sense of a local topology, i.e. the ordering of the input does not convey information, CNNs can discover local features because filters usually cover more than one input [LB98]. In the context of the WLAN spectrum, this provides an elegant approach to learning a generic feature representation of the frequency use of an interfering device, since only a subset of devices are continuous and fixed frequency transmitters.

The structure of the CNN used for interference detection in Article IV, and with some modifications in Article V, is shown in Figure 4.3. This network is relatively shallow, compared to state-of-the art image classification networks, consisting of only two convolutional layers. This is mainly due to the limited number and size of the data sets available for experiments, but the network still allows capturing non-linear features in the spectrograms. The size of the filters in this network is 3x3, which meant we could model transmissions wider than a frequency bin and lasting longer than one spectrum sweep. Filters were moved three strides between each convolution, and max-pooling was used to induce temporal invariance within the sample window. A drop-out layer – which ignores some neurons in a layer during learning – was used to avoid overfitting [HSK+12]. Training was performed through stochastic gradient descent (SGD) with a l_2 regularizer and a learning rate of 0.1. 50 concurrent samples were provided for learning. For classification, the size of the final layer corresponded to the number of devices in the experimental setup. Following best practices, predictions in the last layer were provided through softmax, meaning the predicted device could be determined by choosing the maximum value in the final layer.

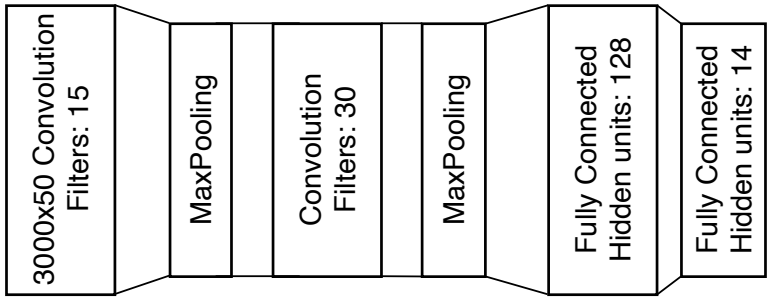

Figure 4.3: Network structure for convolutional neural network. Based on previously published version in Article IV [LPK17].

4.2.3 Structured Pseudo-labels

An inherent difficulty with labeling spectrum samples for training is that even though a device is turned on there is no guarantee that it begins transmitting at the precise moment it receives power, or that it transmits for the full measurement period. The latter issue can be seen e.g. with microwave ovens which often modulate their power by cycling power on and off instead of actually lowering the power itself. This means measurements might contain gaps of information. These issues make labeling spectrum samples tremendously difficult, and would likely require extensive manual effort to ensure a thorough description. The main contribution of the work in Article IV is an extension of the original pseudo-label technique presented in [Lee13], which utilizes unlabeled data to improve the learning process in a semi-supervised way.

The original pseudo-label work provides an approach similar to expectation-maximization (EM) where, between iterations of learning, a set of unlabeled data is labeled with the device with the maximum predicted probability from the previous iteration – providing so-called *pseudo-labels* to uncertain samples. More formally, the optimization problem can be rephrased as

$$\mathcal{L}(\theta, y') = \frac{1}{n}\sum_{m=1}^{n} L(y_m, f_m) + \alpha(t)\frac{1}{n'}\sum_{m=1}^{n'} L(y'_m, f'_m), \tag{4.3}$$

where n corresponds to the labeled set of data and n' the unlabeled data. Similarly, y_m and f_m represent the true labels for labeled data and the output of the network and y'_m and f'_m represent the pseudo-labels for unlabeled data and the outputs, respectively.

The semi-supervised learning of this formulation stems from the latter term, which is optimized over the pseudo-labels. The coefficient $\alpha(t)$ provides a way to balance the extent to which unlabeled data is used for training. A large value might derail the learning process, while a small value essentially reduces the description to the classic supervised learning problem [Lee13]. In the original work an annealing process is used, increasing $\alpha(t)$ over time, but in our work it was found that a constant value of 1 was sufficient to provide improved accuracy.

Structured pseudo-labels enforce a temporal continuity over the chosen pseudo-labels by assuming a Markovian property. The solution is penalized if it chooses pseudo-labels that do not match the labels of the previous instance. Formally,

the following regularizing term is added to Equation 4.3:

$$\lambda \sum_{m=2}^{n'} \mathbb{1}(y_i'^m \neq y_i'^{m-1}), \tag{4.4}$$

where λ determines the extent to which the previous label is enforced and $\mathbb{1}$ is the indicator function that returns 1 if the condition is true and 0 if it is false. The constraint of temporal continuity means the pseudo-label independence assumption is lost. This departure from the original formulation can be overcome by treating the problem as a Hidden Markov Model and solving it through a dynamic programming formulation corresponding to the Viterbi algorithm [Vit67], which involves finding the most likely sequence of states given a state-transition probability matrix. It can then be shown (described in Article IV in more detail) that the prior negative log likelihood of such a sequence reduces to

$$\sum_{m=2}^{n} \mathbb{1}(y_i'^m \neq y_i'^{m-1})(log(p) - log(q)) + D, \tag{4.5}$$

where D is a term that depends only on the self-transition probability p and not the pseudo-label assignment and q is the probability of transitioning to a different device label, i.e. $(1-p)/C$, where C is the number of devices. This formulation then means we can evaluate $\lambda = (log(p) - log(q))$, with $p > q$ to ensure that retaining the label is preferred over switching it. If $p = q$ this solution reverts to the original pseudo-label formulation as $\lambda = 0$. In our experimentation p was set to 0.2 which corresponds to $\lambda = 1.18$. The rest of the algorithm then proceeds as with pseudo-labels, alternating between learning the network and assigning new pseudo-labels with the Viterbi algorithm. This same temporal continuity can then also be enforced when performing predictions.

4.2.4 Signature-based Baseline

The state-of-the-art baseline is the AirShark system [RPB11]. AirShark performs detection by learning individual decision trees for each target device, operating on a set of handcrafted features. Due to differences in hardware, it is not possible to implement AirShark exactly, but our measured performance is mostly in line with the original study despite these differences. The wider range (120 vs 20 MHz), and more granular resolution (40 vs 312.5 kHz) of spectrum sweeps in our measurements meant that the *spectral signature* feature in particular served as a

good fit as our baseline implementation. This signature was calculated through

$$\hat{s} = \frac{\mathbf{s}}{\|s\|}, \tag{4.6}$$

where $\mathbf{s}$ is a vector representing the average power for each bin in the window of samples and $\|s\|$ was the vector norm.

For all devices this linear model consisted of measuring this signature over training samples, including an "off" signature based on all samples with no known sources of interference. This signature could be thought to represent the background noise, i.e. the baseline spectrum information. To perform predictions, the best candidate for each test sample (also reduced to spectral signatures) was the device signature with the smallest angular difference to the test signature [RPB11], i.e. $\text{argmin}_d \, cos^{-1}(\hat{s}_d \cdot \hat{s}_t)$.

4.2.5 Empirical Validation

The described methodology was used for classification of a set of wireless devices in a typical office environment. Specifically, for each device in the experiment – 14 in total, which includes one set of "off" data – measurements were performed over a 3 minute interval. This interval included one minute of background measurements after which the device was turned on for one minute, and turned off again for another minute. The devices consisted of 4 analog video cameras (*video/spy1/spy2/spy3*), narrow- and broadband jammers (*nbjam/bbjam*), a microwave oven (*mwo*), two baby monitors (*baby1/baby2*), a remote control for RC cars (*rc*), an intercom (*inter*), a headset (*head*) and a lapel microphone (*mic*).

In this setup, only one device was turned on at a time. This choice can be justified by the inverse square law of signal power attenuation. Indeed, most interference is extremely localized because path loss ensures a large part of the transmitted power is attenuated before arriving at a receiving device. To cause simultaneous interference devices would essentially have to be co-located.

In our experiments, measurements were made with an Ekahau Sidekick, which contains a WLAN spectrum analyzer capable of measuring the 2.4 GHz spectrum at a rate of about 25 Hz, or once per 40 ms. The measurements consist of 3076 FFT bins for each sweep, covering 120 MHz of bandwidth. To avoid bias caused by measurements at different power levels, the data was normalized using the z-score. Specifically, each value s was scaled through

$$s_Z = \frac{s - \mu_{off}}{\sigma_{off}},$$

where $s - \mu_{off}$ and σ_{off} are summary statistics calculated from samples where no source of interference was transmitting.

The measured data was split into three separate sets through the following scheme. For testing, 20 seconds of samples from the middle of the measurement window were labeled with the device in question. Even though precise onset of transmissions was uncertain, these intervals could more or less be guaranteed to contain relevant transmissions. The first and last 20 seconds of the data was labeled as "off", i.e. these were assumed to not contain any (known) source of interference. The rest of the data was used for training, including a set of data where the label was uncertain. Because labeling was uncertain around device onset/offset times, different versions of the training set were constructed. A more detailed description of the data segmentation can be seen in Figure 4.4. Spectrograms were provided to the network through partially overlapping windows. In our experimental setup, two second windows of data – corresponding to 50 consecutive samples – were used for training, with a new window started every second of measurements.

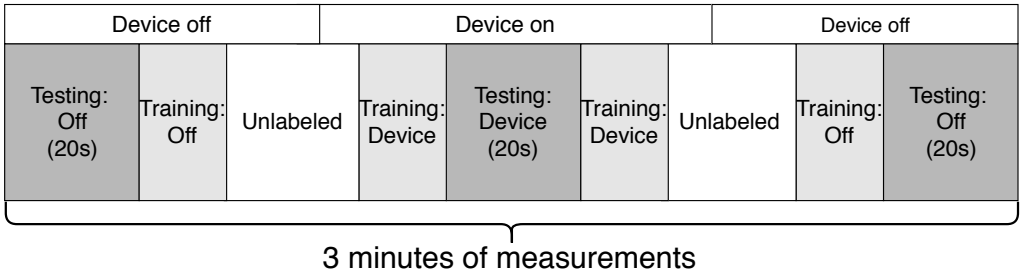

Figure 4.4: Data segmentation for experiments. Modified from previously published version in Article IV [LPK17].

In the first experiment the standard CNN network was compared to the spectrum signature baseline. Here a maximal number of labeled training samples was used in order to measure performance when data is not critically limited. The results of this setup are presented in Table 4.1. The overall accuracy for the CNN approach was 97% compared to 79% for the baseline. The baseline approach has clear issues with devices that are not constantly transmitting (*rc* and *mwo*) or are known to be so-called "frequency hoppers" (*mic, head*). It then finds the "off" label a more reasonable explanation for the lack of information. This is in line with the original work, where frequency hopping devices were detected with lower accuracy, especially at low signal strengths [RPB11].

CNN Classifier

True Label \ Predicted Label	off	video	mic	head	inter	baby1	baby2	mwo	bbjam	nbjam	spy1	spy2	spy3	rc
off	1077	0	1	0	0	0	0	2	0	0	0	0	0	0
video	5	35	0	0	0	0	0	0	0	0	0	0	0	0
mic	0	0	39	1	0	0	0	0	0	0	0	0	0	0
head	0	0	6	34	0	0	0	0	0	0	0	0	0	0
inter	9	0	0	0	31	0	0	0	0	0	0	0	0	0
baby1	0	0	0	0	0	40	0	0	0	0	0	0	0	0
baby2	0	0	0	0	0	0	40	0	0	0	0	0	0	0
mwo	14	0	0	0	0	0	0	26	0	0	0	0	0	0
bbjam	0	0	0	0	0	0	0	0	40	0	0	0	0	0
nbjam	0	0	0	0	0	0	0	0	0	60	0	0	0	0
spy1	0	0	0	0	0	0	0	0	0	0	40	0	0	0
spy2	0	0	0	0	0	0	0	0	0	0	0	40	0	0
spy3	0	0	0	0	0	0	0	0	0	0	0	0	40	0
rc	6	0	0	0	0	0	0	0	0	0	0	0	0	34

Baseline Classifier

True Label \ Predicted Label	off	video	mic	head	inter	baby1	baby2	mwo	bbjam	nbjam	spy1	spy2	spy3	rc
off	778	1	15	47	0	0	1	43	0	0	0	0	0	195
video	0	30	0	0	0	0	0	0	0	0	0	0	0	10
mic	0	0	40	0	0	0	0	0	0	0	0	0	0	0
head	0	0	4	36	0	0	0	0	0	0	0	0	0	0
inter	8	0	0	3	29	0	0	0	0	0	0	0	0	0
baby1	0	0	0	0	0	40	0	0	0	0	0	0	0	0
baby2	0	0	0	1	0	1	38	0	0	0	0	0	0	0
mwo	0	0	0	0	0	0	0	27	0	0	0	0	0	13
bbjam	0	0	0	0	0	0	0	0	40	0	0	0	0	0
nbjam	1	0	0	0	0	0	0	0	0	59	0	0	0	0
spy1	0	0	0	0	0	0	0	0	0	0	40	0	0	0
spy2	0	0	0	0	0	0	0	0	0	0	0	40	0	0
spy3	0	0	0	0	0	0	0	0	0	0	0	0	40	0
rc	0	0	0	0	0	0	0	0	0	0	0	0	0	38

Table 4.1: Confusion matrix of classification with vanilla CNN classifier (left) and spectrum signature baseline (right). Reproduced from Article IV [LPK17].

The CNN clearly outperforms the baseline even with devices using a frequency hopping scheme. It also has issues with detecting the microwave oven (*mwo*) throughout samples where it is not emitting energy into the band. When not operating on full power, microwave ovens typically achieve an average lower power by turning the magnetron on and off in a cyclical manner [GLT03]. In effect, much of this loss of accuracy can actually be attributed to mislabeling: the CNN is correctly detecting an "off" state in the middle of microwave oven operation.

To validate the CNN's capacity for transfer learning, i.e. learning generic features in the initial layers, a leave-one-out form of training was evaluated. All layers of the network were allowed to learn based on all devices but one. Training for the target device, omitted from the initial training set, then consisted of only training the last layer of the network. For most devices, it was shown that nearly equal classification accuracy was achieved with this limited form of training. The results are detailed in Figure 4.5.

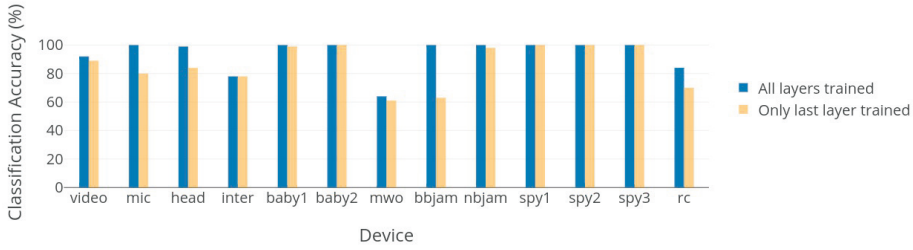

Figure 4.5: Results from classification experiment where entire network was trained vs. only the outer layer. Previously published in Article IV [LPK17].

To measure the improvement gained when using structured over standard pseudo-labels an experiment was performed with training samples of increasing number of labeled measurements. This range at the lowest included only 4 seconds of measurements and in the highest 72 seconds. In the latter case(s), due to the imprecise timing of the measurement onset, we were very likely already including measurements which had in fact been mislabeled. The results are represented in Figure 4.6. When the sample size was small, the structured pseudo label algorithm clearly outperforms both the standard supervised network and the original pseudo-label approach. Once sample sizes reach 40 seconds – the optimal window size with this dataset – more mislabeled samples are encountered and none of the approaches can improve their performance. A small improvement in classification accuracy was also found if the constraint of temporal continuity of sequential samples was extended to testing as well.

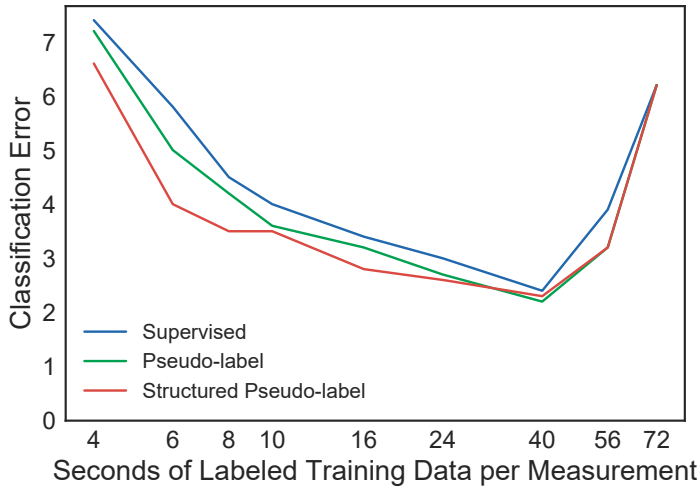

Figure 4.6: Classification errors with different learning strategies. Previously published in Article IV [LPK17].

In conclusion, we showed that convolutional neural networks are a good fit for interference detection, clearly capturing both the temporal and frequency representation of different sources of interference – even when the exact labels for training data could not be ascertained. We showed that uncertain labels can be used to improve prediction accuracy, and improved upon the original work by incorporating a temporal constraint over sequential spectrum samples.

4.3 Deep Learning vs. Signal Modeling

In the previous section we considered a CNN for interference detection and compared it to a linear baseline. Though the accuracy was clearly better for the CNN, some aspects of the solution were not explored during the experimental setup. First, the detection only considered one device at a time and did not take into account the possibility of multiple transmitters in the same location. In Section 4.2.5 this potential limitation was motivated by the inverse square power law, but this scenario could nevertheless warrant further examination.

Second, the work in Article IV only considered the presence of the device, and not the degree of interference. Resolving the device transmit power level in addition to the actual class could improve attempts to localize the device in the environment. This location context of interference could also inform the design of positioning systems. Knowing where the system is likely to face signal degradation helps design more resilient end-user applications. Though Article IV suggested such an extension to the original technique, no further experimentation was performed in that context.

Finally, deep learning approaches for varying tasks, including interference detection, have in recent years mainly focused on experiments in laboratory conditions and rarely measured metrics other than accuracy and training efficiency. This myopic view of testing could mean some issues are overlooked when choosing the algorithm to use, especially in a commercial real-world application.

The following section, summarizing Article V [PNN20], presents a novel signal model for interference detection that can detect multiple sources of interference simultaneously, and determine individual transmit powers for each device. We also highlight instances, equivalent to two real-world scenarios, where a deep-learning approach might fail. To evaluate the performance of this approach compared to the previously established CNN architecture, a set of metrics is used to examine the change in prediction performance when the initial experimental setup is perturbed in a way corresponding to real-world differences in environments.

4.3.1 Experimental setup

The majority of experiments performed in this study consist of altering data in a systematic way and measuring the correlation between this perturbation and the errors that are caused. To ensure that the variance in results could be attributed to testing conditions, the training measurements were performed in a radio-frequency isolated container, a Ramsey Electronics STE3600 RF shielded

enclosure. This meant any energy detected in the spectrum samples could be attributed to the target device alone, and not to background noise, including other common transmitters such as Bluetooth. For testing, measurements were made in an office environment in order to emulate a real-world use case.

This difference in experimental setup compared to Article IV required that, while the overall structure of the network was maintained, certain aspects of the CNN described in Section 4.2.2 were modified. Limited space in the enclosure meant devices like microwave ovens could not be measured, and to ensure that variation in experimental results could be uniquely attributed to a change in testing properties no pseudo-labeling or interleaving of spectrum measurements (into overlapping windows) was performed. Adjustments to the CNN learning parameters were also performed to improve accuracy under this new architecture.

Training and testing data contained 25 sweeps of 2500 spectrum bins per sample. The learning rate of the stochastic gradient descent was set to 0.001, the batch size (concurrent samples) was set to 1, 2 strides were taken per iteration and the max pooling window size was 2x2. Training was performed over 100 epochs. No pseudo-labeling was performed as measurements were done strictly for times when devices were turned on. A fixed seed was used to ensure the effects of each setup could be isolated from the randomness of the learning process.

For both training and testing, 7 devices were measured for 30 seconds and a set of data was measured when no device was turned on. This resulted in a total of 750 spectrum sweeps per device. The selection of devices partly overlapped with those presented in Section 4.2.5 and Article IV, with a few omissions. The microwave oven was not included due to limited space in the enclosure, and some of the other transmitters were no longer available for testing. Nevertheless, the tested devices covered a representative range of potential everyday wireless devices, containing frequency hopping devices as well as periodic and constant transmitters. The set of devices consisted of an analog video camera (*hamy*), an intercom (*skatco*), two baby monitors (*motorola/ babymon*), a headset (*huhd*), a lapel microphone (*boya*) and a remote controller for RC cars (*rc*).

4.3.2 Metrics

We next describe the different metrics used to compare the described approaches. These metrics were inspired by a survey on machine learning testing [ZHML19], and have been adapted to the interference detection domain. We shortly describe what the metrics measure and how they relate to the the topic at hand.

Correctness

The first measured property is the correctness, or accuracy, which is measured by the ratio of correct predictions to the number of total predicted cases [ZHML19];

$$\frac{1}{m}\sum_{i=1}^{m}\mathbb{1}(h(x_i)=y_i). \tag{4.7}$$

Here m is the total number of classes, x_i is the unknown label of the class, $h(x_i)$ the algorithm's estimation of said class and y_i the true class of the data in question. $\mathbb{1}$ is the indicator function. In our experimental setup we evaluated classification through the use of a confusion matrix, as before in Section 4.2.5, which helps describe which classes the algorithm is struggling with. The correctness can easily be calculated from this matrix from the proportion of labels that fall along the diagonal, i.e. the ratio of true positive cases to all other cases.

Overfitting

High training accuracy is desirable for machine learning algorithms. When the model otherwise performs well during training but fails to generalize to unseen cases, the most likely cause is overfitting. Zhang et al. [ZHML19] suggest overfitting is a result of the model having learned to fit against the noise instead of the relevant features of the data. A possible way to determine the level of overfitting is to perturb the training data in a systematic way and to measure the loss in training accuracy with respect to the level of perturbation [ZBG+19]. Specifically, the authors of [ZBG+19] suggests a form of *accuracy decrease rate* that is measured by the absolute value of the gradient of the best fitting line – in the least squares sense – of the point line formed by (r_i, S_{ri}), i.e. the point pairs of the injected noise rate and measured training accuracy. In our work, the perturbation scheme consisted of randomly permuting training labels for each class, taking care not to select the original, true, label. This random perturbation approach is suggested by a work on adversarial attacks [CAD+18].

Robustness

The robustness of an algorithm essentially measures its capacity to perform in the face of noise. For neural networks, robustness can be measured for instance through *pointwise robustness* or *adversarial frequency* [BIL+16]. The former

determines the minimum level of noise required to fail (change in predicted label) whereas the latter corresponds to the rate of failure when noise is encountered [BIL+16]. We adopt these descriptions for our evaluation, but use a domain-specific source of noise for testing. Specifically, we use *additive white gaussian noise* (AWGN), a common noise model for information channels [Gal08].

Efficiency

The efficiency of an algorithm can be measured through its real-world performance, or timing, as well as its capacity to use information in an efficient way. The ability for deep-learning algorithms to learn on limited data has previously been explored from the perspective of optimal subset selection [SG19]. Loss-based models might work better for deep-learning, but we opt for random subset selection to ensure a fair comparison. This is considered best practice [SG19]. The algorithms were also compared in terms of the time spent in training and prediction relative to the number of classes in the model.

Interpretability

The degree to which an algorithm's internal logic is traceable, or interpretable, is increasingly important for deep learning. Interpretability can be gauged by the transparency of the algorithm model, and how well its predictions can be explained [ZHML19]. Linear regression could be thought to be inherently more interpretable [Mol19], but interpreting the weights in a neural network can be complex, especially in a way that is quantifiable. In our experimental evaluation, we focus on the interpretability of predictions in two separate experiments. We first instrumented an experiment with two concurrent transmitters and analyzed the prediction weights qualitatively. We then measured the algorithms' capacity to estimate the distance to a transmitting device.

4.3.3 Deep Learning Evaluation

CNN architectures, and deep-learning in general, can fail when the data provided for training is not rich or extensive enough. A disconnect between training and testing environments, for instance through an increase in overall noise levels, will cause further uncertainty in predictions. To evaluate the extent of these issues, we performed experiments where these experimental conditions were varied.

To validate the chosen architecture, we first performed a traditional accuracy estimation. The confusion matrix resulting from this baseline evaluation is

shown in Table 4.2. An overall accuracy of 80% was reached, which is in line with previous work, but also shows the impact of the change in environments. When trained on the completely noise-free training samples from the RF shielded enclosure, exposure to the real world is enough to decrease the model's accuracy.

truth/predicted	babymon	boya	hamy	huhd	motorola	skatco	rc	none
babymon	19	0	1	3	4	0	3	0
boya	1	29	0	0	0	0	0	0
hamy	1	0	29	0	0	0	0	0
huhd	0	0	0	29	0	1	0	0
motorola	0	0	0	0	30	0	0	0
skatco	0	2	0	21	0	7	0	0
rc	0	9	0	0	0	0	21	0
none	0	0	0	0	0	0	0	30

Table 4.2: Confusion matrix of CNN classification. Overall accuracy: 0.80. Reproduced from Article V [PNN20].

This weakness in terms of **robustness** was further evaluated by injecting Gaussian noise of increasing standard deviation into the testing set. The rate of failure was measured as the number of changed labels from the initial set of predictions. The results of this testing are shown to the left of Figure 4.7. Here, the CNN suffers greatly. The initial failure occurs at noise levels of 1.5 dB. Thereafter the CNN's accuracy drops sharply and is essentially predicting the same label once noise levels reach 10 dB. When the Z-score (see Equation 4.2.5) used to normalize CNN is recalculated between iterations (denoted CNN-Z in the figure) the situation is improved to some extent. In a real-world application this would require recalibration in every new environment. To put noise values into perspective, propagation models consider shadowing effects (X_g in Equation 2.2 in Section 2.2.2) ranging from 5 dB to 14 dB [Rap01], depending on the environment type and topology. In the extreme range, noise values such as this can have a severe impact even if a renormalization is performed.

Efficiency was evaluated through random subset sampling, i.e. by iteratively reducing the number of datasets used for training over 30 trials for each subset, and performance was re-evaluated. As shown to the right in Figure 4.7, a great degree of stochasticity is still embedded in the learning process. The convergence to the best known performance happens relatively quickly, but the accuracy can drop as low as 30% when only one sample is removed from training data.

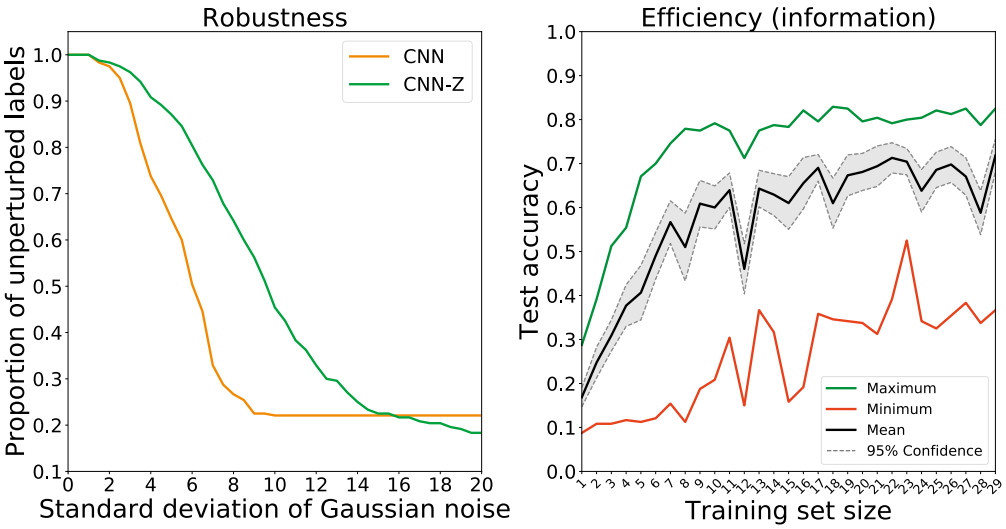

Figure 4.7: The deep learning approach fails when the original testing data is injected with synthetic, but realistic, noise and when data is not rich enough. Modified from original in Article V [PNN20].

This suggests the CNN architecture is not only more sensitive to which specific datasets are used for training, but is also volatile even when almost the entire training dataset is available. A good subset selection strategy, could achieve close to best-case accuracy with only 25% of the complete data, meaning we are less concerned about the absolute size of training data than its richness.

4.3.4 Signal Modeling using Multiple Linear Regression

The deep-learning approach generally performed with good accuracy, but faltered in aspects that are relevant for real-world applications. The performance of the algorithm was critically dependent on the type and number of data samples it was provided. It also stumbled when faced with a degree of noise that no longer matched the training conditions. In Article V [PNN20], an alternate technique that is simpler in composition but adheres more closely to the underlying domain of the measurements was designed. This provides better interpretability of the inner workings, which directly translates to features useful for real-world applications. This signal model is based on the least-squares fitting of pre-defined spectrum signatures to spectrum samples averaged over the measurement win-

dow. Whereas the approach described in Section 4.2.4 also calculated a signature over the relevant frequency band, it was modeled as a unit vector and matching was performed through cosine similarity. The key contribution of our approach is to act directly on top of raw signal values, which allows the fit to correspond to the transmit power of multiple concurrently transmitting devices. The computed signatures were calculated in the linear domain as opposed to the decibel domain. This encapsulates the underlying intuition that all sources of interference in the wireless domain could be described through a linear combination of spectrum signatures. Conceptually this is similar to spectral unmixing employed in hyperspectral imaging analysis [Cha13].

Figure 4.8 displays two example device signatures, the latter calculated from the spectrum samples in Figure 4.1 (Section 4.2.1). In this case, these signatures correspond to an analog video camera (orange graph, single spike) and a frequency-hopping baby monitor (blue graph, multiple spikes).

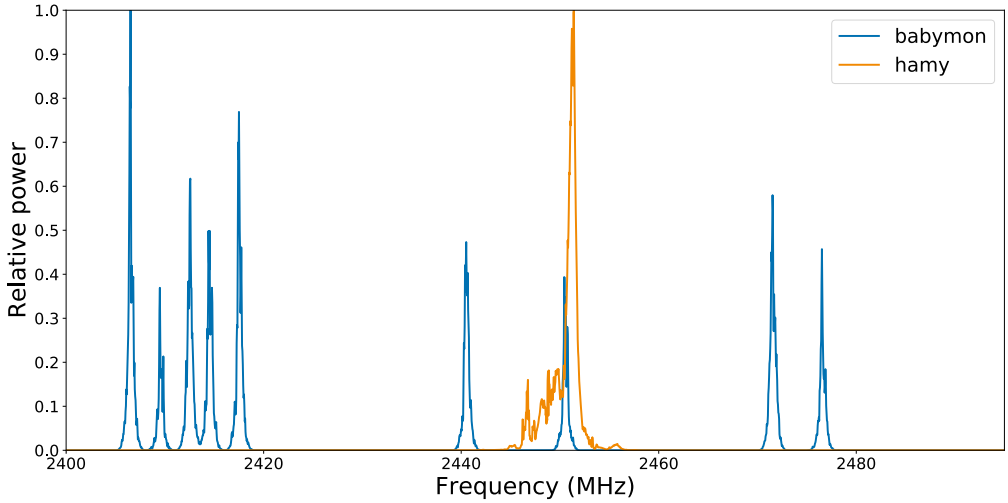

Figure 4.8: Two example device spectrum signatures used for the linear regression approach. Previously published in Article V [PNN20].

More formally, each device signature was measured as the average power, per spectrum bin, over the time domain. Signatures were then scaled by the strongest bin, providing a vector $v = [s_1, s_2, ..., s_n]$ in the range $[0, 1]$ for each device. Collecting these signatures into a design matrix means we could solve the detected transmit powers for each device using standard least squares (see

e.g. Equation 3.4), given a new spectrum sample vector, consisting of the average (linear) spectrum power over the measurement window. Since measured power cannot be expected to extend to the negative range in the linear domain, a specific non-negative variant (non-negative least squares, NNLS) was used which is based [NNL20] on FORTRAN code published in [LH95].

4.3.5 Empirical Validation

To show the improved performance afforded by the NNLS approach, we first validated its accuracy. The confusion matrix for NNLS is presented in Table 4.3. At 73%, the overall accuracy is lower than the best-case performance of CNN, but in line with previous work. The random initialization of the CNN network weights, and the stochastic gradient descent used for backpropagation, meant the absolute difference between the algorithms varied to some degree.

truth/predicted	babymon	boya	hamy	huhd	motorola	skatco	rc	none
babymon	12	3	7	0	0	0	4	4
boya	0	21	0	0	0	0	0	9
hamy	0	0	30	0	0	0	0	0
huhd	0	0	0	29	0	1	0	0
motorola	4	1	1	0	13	0	2	9
skatco	0	3	0	1	1	19	0	6
rc	0	0	0	0	0	0	22	8
none	0	0	0	0	0	1	0	29

Table 4.3: Confusion matrix of NNLS classification. Overall accuracy: 0.73. Reproduced from Article V [PNN20].

Due to our measurement setup we were unable to include devices that are known to be problematic for signature-based detection methods, such as microwave ovens (as described in Section 4.2.5). Nevertheless, it is apparent that both approaches suffer from the difference between the clean environment used for training and the real-world office environment measurements used for testing.

An interesting aspect of the classification accuracies is the disagreement over labels. In about 10% of the cases one algorithm predicted the correct class while the other did not. In five cases the approaches predicted the same, wrong, label. In three of those cases the reason is likely that the fixed-frequency controller *rc* momentarily transmits in the same frequency range as the frequency-hopping *babymon* which makes some measurements look deceptively similar.

Before delving into metrics that more clearly set the approaches apart, we first validated their performance in terms of **overfitting**, justifying the use of the chosen architectures. As described in Section 4.3.2, 1-15 labels per class were permuted to a random (wrong) class, after which the training accuracy was recalculated and the accuracy decrease rate was determined as the absolute value of the best fitting gradient. The results are depicted in Figure 4.9. In both cases, the gradient was clearly decreasing, which suggests no significant overfitting occurred in either case. Though the precise rate for CNN was better than for NNLS (1.35 vs 0.6), no specific benchmark for overfitting was provided in the original work [ZBG+19], as the rate was used for relative comparison. That said, no overfitting was determined for rates above 0.5.

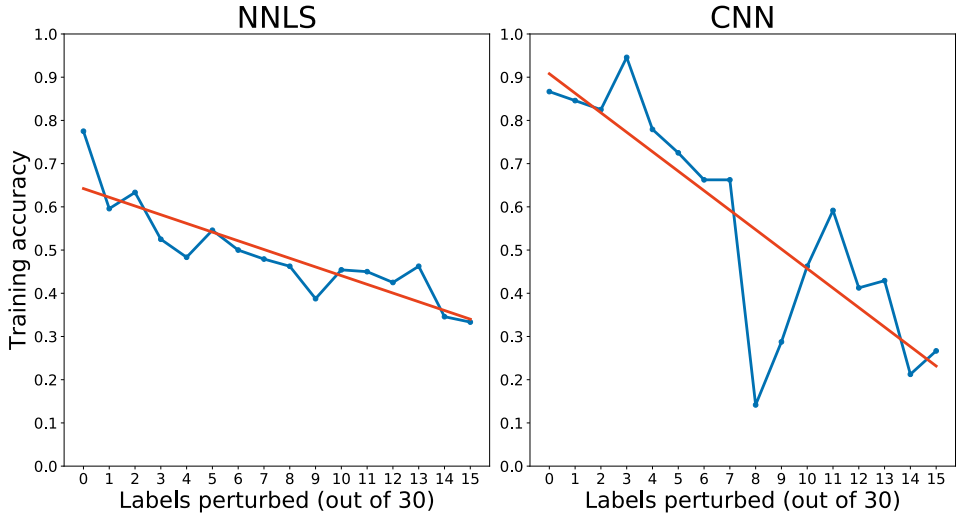

Figure 4.9: Comparison of the overfitting metric (see Section 4.3.2). Previously published in Article V [PNN20].

The algorithms display their first clear deviation in performance when **robustness** is re-evaluated for both approaches, as displayed in Figure 4.10. The range of noise values has been extended to fully evaluate the NNLS behavior. Whereas CNN struggles to perform beyond 10-15 dB, NNLS has more than random chance performance at noise levels of 20 dB. A slight increase in local variance is likely due to the linear scaling of signatures, which can momentarily give great significance to individual spectrum bins. Nevertheless, overall the simpler NNLS model's capacity to handle noise is much greater than that of CNN.

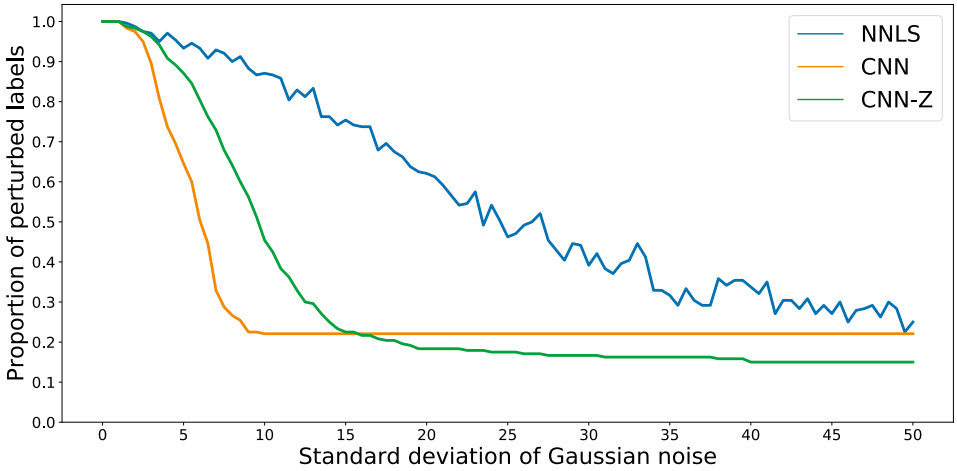

Figure 4.10: Comparison of robustness. CNN-Z corresponds to a version of CNN where the dataset is renormalized between noise injections. Previously published in Article V [PNN20].

In terms of **efficiency**, the approaches were evaluated as shown in Figure 4.11. Whereas CNN struggled even when almost all data was available, NNLS could almost reach its best-case accuracy with only one well selected sample per device, and 5 samples in the best-case could be enough for maximal performance.

Efficiency in terms of performance was measured relative to a binary classification scenario. Starting from two classes, the number of classes used for training and testing was increased up to 102, and the relative time used was calculated. This allowed for a comparison that marginalizes the impact of the underlying experiment hardware. The results of this comparison are shown in Figure 4.12. In terms of training time, both algorithms have a similar, linear, dependency with respect to the number of classes, as seen from the best fitting lines. The major difference between algorithms is shown for prediction speed. Whereas for CNN increasing the number of classes has at most a linear impact, the complexity of NNLS is on the order of $O(n^2)$ and the best fitting curve for NNLS relative testing times is $0.01n^2$. On a standard off-the-shelf laptop this complexity would result in a detection model containing 900 devices resolving sources of interference once per second. This is well within range of the 500 unique devices suggested in Article IV, but could be considered a minor drawback of the NNLS approach. For more details on the performance evaluation, see Article V.

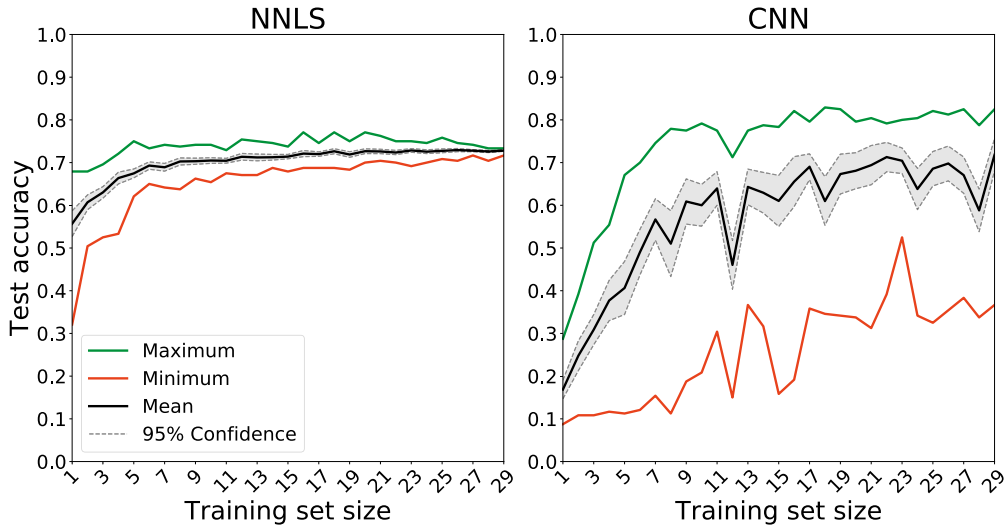

Figure 4.11: Comparing information theoretical efficiency. Previously published in Article V [PNN20].

Interpretability of the chosen approaches was measured qualitatively by examining two problematic cases with real-world relevance. First, a new dataset was measured where two of the trained devices were transmitting concurrently, and performed predictions again without retraining the algorithms. Heatmaps of the final weights (scaled power in the case of NNLS) are shown in Figure 4.13.

The softmax-trained CNN has difficulty resolving both devices at the same time, as could be expected. An alternate network trained with sigmoid as the activation function for the final layer provides a smoother estimate for one of the devices, but is also not able to handle concurrency in a robust way. NNLS recovers this setup more distinctly, providing a consistent estimate of the multi-label classification scenario.

Next, the weights themselves were analyzed with respect to distance to the *hamy* device. In this test setup the device was turned on and measurements were made while walking away, turning around and walking back towards the device. The predicted weights are shown in Figure 4.14. Both algorithms have a level of understanding of the proximity to the device, but the CNN approach loses this context relatively fast. Because the fit provided by the NNLS approach corresponds to the power of each transmitting device, the values can be directly

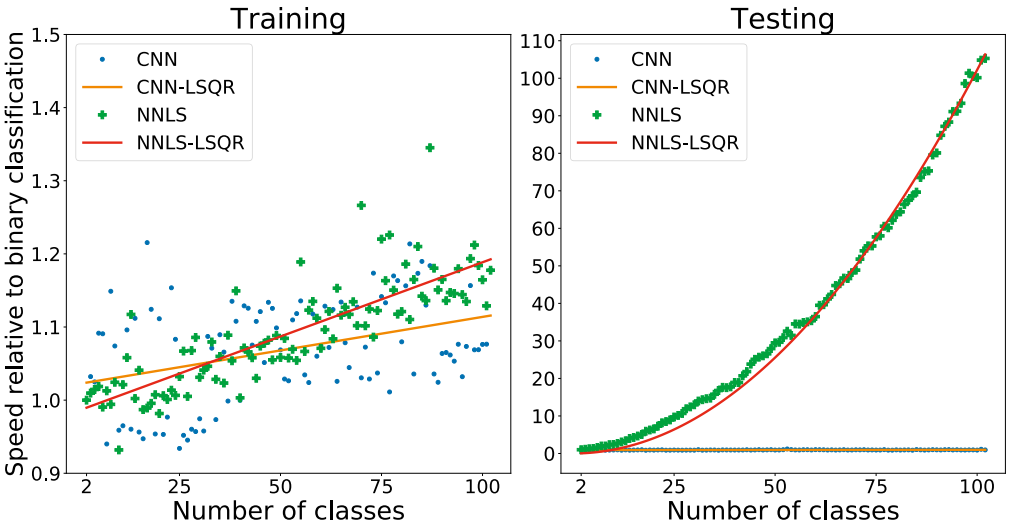

Figure 4.12: Comparing efficiency in terms of time. Previously published in Article V [PNN20].

interpreted and modeled through the standard log distance path loss formula (described in Equation 2.2 in Section 2.2.2). This dependency is depicted on the left graph in Figure 4.14, where the orange line represents the expected theoretical path loss. The NNLS predictions adhere to this line quite consistently, indicating that our signal model is consistent with the real world phenomena. This representation could help determine the location of the device, but also provides a principled interference detection threshold.

In conclusion, NNLS is able to compensate for many of the weaknesses of the CNN approach. It handles noise much more robustly, and provides a faster convergence to optimal performance when training data is limited. Though its accuracy is slightly lower than that of CNN, it provides better interpretability through its built-in capacity for multi-class detection and adherence to a theoretical path loss model. The latter property provides applications with a distance-dependent quantity to use for locating the source of interference.

The CNN likely provides a better option for sources of interference with complex or unknown transmitter characteristics, and in scenarios where a sufficient amount of data is easy to obtain. Its computational complexity is also less dependent on the number devices in the model. It is clear that the training and testing

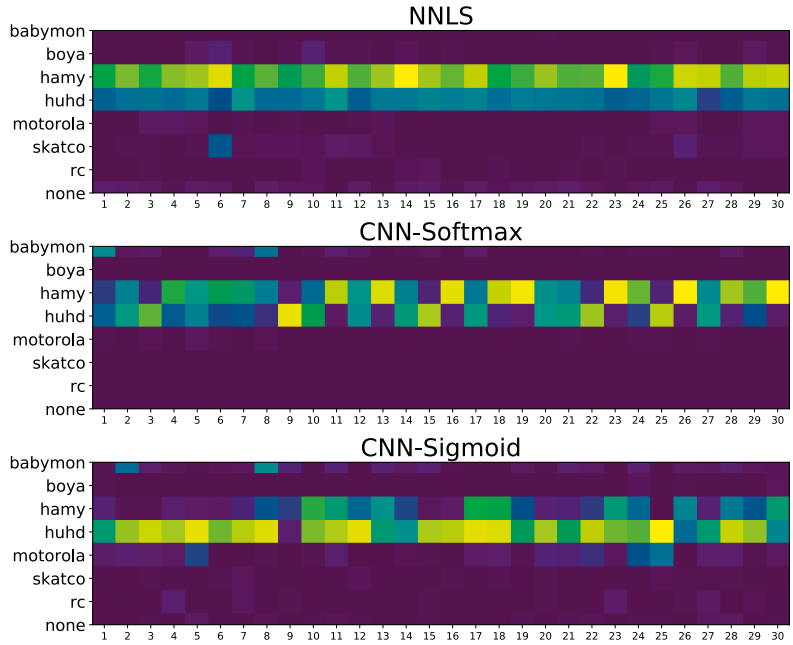

Figure 4.13: Comparing interpretability of output vectors. Devices *hamy* and *huhd* were turned on simultaneously. Previously published in Article V [PNN20].

environments should match as closely as possible. This might prove difficult to ensure particularly in commercial deployments since the environment used for training the algorithm will rarely match that of the end user.

4.4 Discussion

The dependency of deep-learning algorithms on data – specifically, as we've shown, on informative samples – could to some extent be lessened through transfer learning. Section 4.2.5 explored this concept from the perspective of computation time, showing that after the neural network had been trained on initial sources of interference, subsequent device classes could be trained to nearly optimal performance using only the final layer of the network. The inherent stochasticity in the architecture described in Article V could also be reduced by pre-training the initial layers and then re-using them in subsequent training sessions. This approach is common in the image recognition domain [SZ15].

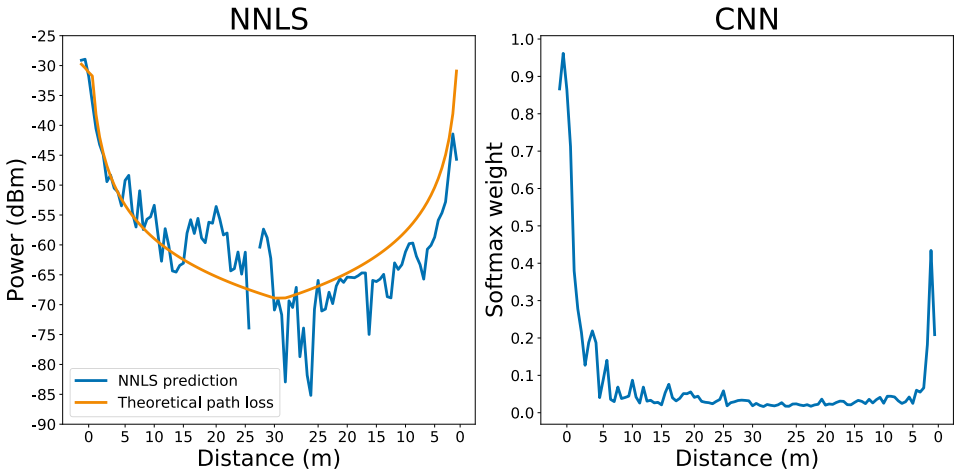

Figure 4.14: Comparing distance interpretation of predicted value. Note that the path doubles back around the midpoint of the distance axis. Previously published in Article V [PNN20].

In this work we have we have exclusively considered spectrum sweeps over the entire frequency band. This provides a complete view of any potential source of interference in terms of frequency, but sacrifices some granularity in the time domain. Given a more narrow, and faster [RPB11], view of the spectrum a hybrid deep-learning approach could learn to classify sources of interference at different time scales. Separate networks could be trained on the different sources and predictions could be made as a weighted combination in a form of multi-source classification [SSW15]. Both perspectives could even be incorporated into the same network, with different dimensions for the convolutional filters. As presented in work on singing voice separation [GZP19], separate filters could be trained for high frequency resolution and high time resolution sources of information.

Measuring the frequency distribution of energy with a spectrum analyzer is an approach that focuses wholly on the physical layer of wireless network communication. Anomaly detection has previously also been performed using packet traffic [MMH+17]. In addition to a multi-resolution spectrum view into interference classification, such techniques could also be used as an additional source of information when the source of interference is a device with a known communication protocol. For instance, the open source libraries *tcpdump* and *libpcap* [Tcp20] could be used to decode Bluetooth and ZigBee traffic.

As touched upon briefly in Section 3.2.5, the algorithm for suggesting access point placement could potentially be used to find candidate locations outside of the range of known sources of interference. Given the localization capability of the NNLS approach to interference detection, these approaches could potentially work in tandem when designing the access point layout for positioning. An interference heatmap could be used as an added layer to inform access point placement when interference could not otherwise be mitigated. For navigation applications, like the one described in Section 2.4, knowing the location of potential sources of interference could also inform navigation instructions. When the underlying positioning technology is known to conflict with that of the interfering transmitters, such as industrial microwave ovens in the bakery section of a supermarket, pathing could be designed around potential trouble spots.

Chapter 5

Discussion & Conclusions

This thesis has provided novel solutions to some key challenges that have prevented the proliferation of WLAN positioning as the de facto indoor positioning solution. Though the presented work has focused exclusively on WLAN as the underlying technology, the provided contributions are not specific to WLAN and could support the deployment of systems based on other technologies as well. Other promising sources of location information could follow or compliment the presented contributions, or work in tandem to provide a holistic view of ubiquitous positioning. Some of these are described in Section 5.1.

In addition to these technical aspects of an indoor positioning solution, we also described a study on navigation instructions in Section 2.4, where we illustrated the real-world issues faced by a typical location-based service. Naturally, the domain of location-based services is broader than this, and is an active topic of research in itself. As such, it provides many unexplored opportunities, but also a wide range of challenges – some of which are unique to the indoor context. We discuss some of these aspects in Section 5.2. Finally, we conclude the thesis in Section 5.3 and summarize our contributions in Section 5.4.

5.1 On Indoor Positioning

In this work we have largely focused on signal modeling aspects relating specifically to a WLAN positioning system. Other radio frequency solutions operating in the same band as WLAN could adopt our contributions on signal modeling as well, because propagation characteristics can be assumed to be similar. One crucial attraction of WLAN is a set of ubiquitous dedicated base stations without

the need to monitor battery life – increasing the potential transmit power and by proxy coverage. Through the advent of access points with Bluetooth Low Energy beacons this restriction is quickly eroding, however. ZigBee-enabled lights [Wan13] could also serve as a source of location information arguably even more ubiquitous than WLAN. The work in this thesis is not limited to WLAN as a technology, and can complement such efforts since similar aspects of calibration effort, signal topology modeling and interference detection are relevant for other wireless technologies. Operating within the same frequency range as WLAN, the path loss characteristics – and thus the non-linear nature of measurements – and challenges with cross-technology interference remain relevant.

The specific impact of interference on WLAN positioning was discussed but not evaluated directly, largely because the effects on communication – and by proxy beacon transmissions – are well known. An interesting avenue for research would nevertheless be to instrument different interference scenarios and measure their impact specifically on positioning. The disconnect between the data used for training the positioning algorithm and the real world measurements will inevitably cause uncertainty in position estimates, but such an endeavor could measure this change systematically. Because interference is typically localized, due to propagation losses, the impact could not be assumed to be constant over the entire positioning environment. This would likely require non-linear modeling of the resulting positioning error. Furthermore, client devices capable of sensing the energy of the WLAN channel could potentially mine constant but low-impact sources of interference for location information, potentially providing yet another organic landmark for techniques such as UnLoc [WSE+12].

The focus of this thesis has largely been on signal strength measurements of WLAN beacon frames. A recent trend in the domain of WLAN sensing is the use of channel state information (CSI) for various purposes [HHSW11, MZW19], which provides a richer view into the phase and amplitude distribution over the wireless channel. This more detailed view comes at the cost of lower compatibility, however, due to the need for custom wireless drivers and specific hardware. We previously discussed its use for positioning in Section 2.3.4, but other interesting tasks have also been described in the literature. CSI has been used to remotely detect breathing rates of patients [WYM17], sense different types of materials [FLC+18], as well as recognize gait and gestures of people [ZTL+18]. In terms of positioning, knowing that the degree of multipath in a specific environment could be extracted from amplitude data through clustering [WGMP15] provides an opportunity to recognize and account for potential sources of uncertainty as part of the modeling process.

We have contributed novel techniques to aid WLAN positioning to become a low-cost, low-effort and robust indoor positioning solution. This paves the way for bridging the gap between indoor and outdoor spaces for truly ubiquitous positioning. In the outdoor space, some attractive options for model-based positioning include GSM positioning, which could provide better energy efficiency than GPS [NBK10], and outdoor WLAN positioning [LQD08], which operates well in so-called *urban canyons*, where GPS is known to struggle [Gro11]. To provide for a ubiquitous solution, however, requires designing a global location model that can take the intricacies of indoor environments into account. For the outdoor case, a global environment topology can be described in a deterministic way, using geographic coordinates or by utilizing a grid [NBK10]. Because movement outdoors is much less restricted, this generic topology is likely to cause conflicts with the constraints of indoor movement. Future endeavors would need to incorporate these restrictions to provide a smooth transition between contexts.

5.2 On Location-based Services

Our work has focused on the improvement of position accuracy and consistency, but other considerations such as privacy and standardization are in many ways equally important aspects to consider if positioning solutions are to receive widespread adoption for location-based services. In the following we describe some important topics in the domain and discuss ways in which the contributions of this thesis could form avenues for future research.

Privacy

Collecting data for a WLAN positioning system, whether during the training or the deployment phase, exposes the users to divulging information that could be used to identify them. This might decrease the uptake of the positioning solution by privacy-conscious individuals. The issue of privacy, in addition to ethical and social challenges, is one of the directions into which research on location-based services has expanded in recent years [HGK+18]. In many cases, aspects that would provide an improved location-based experience conflict with consumers' reluctance to share personal information. A typical social networking application of location-based services is "checking in" at venues or events. Such uses for location information need to be moderated with proper privacy controls to ensure mainstream support of the service [RW13]. In shopping scenarios the sometimes intrusive nature of promotions might be mitigated by personalizing

the application. This, however, might also conflict with the consumers desire for privacy, depending on the level of control they have over the information they provide [ARG16]. While crowdsourcing WLAN measurements could provide a way to distribute the effort of initializing the positioning system, consumers might not be willing to report their location or the access point information they are receiving. Given that we have shown that a large quantity of measurements can in fact be gathered without specific location information in Section 3.1 and Article II, we can in effect provide a way to perform such crowdsourcing in an anonymous way. Since none of the contributions directly require the location or identifying information of access points in the environment, our solutions could painlessly integrate a deterministic obfuscation of identifiers, such as access point MAC addresses, in the measurements. Finally, reducing the size of the location model by optimizing the size of the environment topology, as described in Section 3.2 and Article III, also provides means to ensure that positioning can run on client devices without interfacing with external services.

Context

We previously investigated navigation semantics from the perspective of cognitive load in Section 2.4 and Article I, with the intention of finding a balance between the potentially orthogonal goals of the supermarket proprietor and the customer. That is, efficient navigation might decrease the number of advertising opportunities within the store. Another potential mediator is that of location-based advertising (LBA), i.e. presenting product promotions at opportune moments. Properly accounting for all aspects of an LBA solution might require multidisciplinary effort, including insights from marketing and psychology research. Contexts like the size of the retail environment [RNG16] and the complex psychology behind promotion timing [MAP07, SI13] have been found to affect purchase behavior. In order for location to provide a context for other effects, the granularity of location information needs to match the demands of the application. In supermarket environments a minimum resolution for position updates would be the width of individual shelf units, as product categories can change multiple times along a single shelf. In terms of advertising, on the other hand, position estimates need to fall within visible range of the product for the promotion context to matter. In Chapter 3 we highlighted the difficulty of reconciling such human location contexts with the limitations of the spatial variability of the measured signal. In some cases it is simply impossible to distinguish two locations in signal space that have semantic differences in the real world. This issue could be

mitigated by adding enriching information to poorly performing regions of the location model, which we showed can be performed in a cost-effective way by placing access points based on region fitness scores in Section 3.2.3. Even in cases where this is not practicable, if the regions which are likely to violate the expected contexts are known in advance, recognizing and accounting for them in the design of the LBS could provide a more consistent experience.

Standardized operation

Finding an automatic way of discretizing the position environment was previously motivated, in Section 3.2, by the lack of a connection to the underlying signal space. Another aspect to consider is how a chosen discretization, or environment topology in general, influences the generalizability of the algorithms and the evaluation of position errors across deployments. This consistency of design is particularly important in the industrial sector, where standardization and interoperability are deemed important characteristics of potential LBS [FJAD18]. This calls for positioning systems to have a common set of standards in order to provide a consistent LBS experience, something which ISO has recently recognized through its publication of testing and evaluation standards for localization systems [ISO16]. Our strategy for optimizing the indoor topology for positioning purposes in Section 3.2 provides for a systematic way of designing the location model around the measured signal instead of heuristics, which could change from one location to the next. This perspective into the environment topology could arguably prove even more fair than a set grid size for systematic testing, because the impact of the unique signal topology of each location is effectively marginalized as part of the dynamic partitioning scheme.

Real-world deployments

We explored the issue of validating algorithms in real-world environments as part of our investigation into interference detection techniques in Section 4.3 and Article V. The disconnect between how the system has been developed and the use cases in which it is deployed is typical for intelligent systems. In some extreme scenarios the assumptions made during design can completely break down. In [GFCM17] the authors highlight several complicating scenarios when indoor positioning solutions are used by emergency responders, especially firefighters. In such catastrophic circumstances the existing infrastructure for wireless connectivity is at risk of failing completely due to interruptions in electrical systems, and

fires might change the layout of the environment to the extent that established positioning models no longer apply. The authors also recognize the disconnect between a model trained for the average pedestrian and the movement patterns that are unique to emergency responders – such as crawling or hunched walking. Such modeling disconnects are relevant to other domains as well. Consumers in a supermarket might require different pedestrian models based on which type of cart – if any – they use to carry their shoppings. This behavior might even change as the cart grows heavier. Similarly, when traveling the transportation of luggage will likely not match unhindered pedestrian gait. A testament to the need for real-world validation is our navigation study described in Section 2.4 and Article I, which instrumented an established positioning system in a complex everyday environment, and faced issues with consistency of position estimates. This disconnect between design and deployment motivated other contributions in this thesis as well. We advocated for a modeling of the real-world signal space instead of anchoring the positioning model to a predefined topology in Section 3.2 and Article III, and our work in Section 4.3 and Article V further highlighted the need to test solutions in real-world conditions, or at least simulate the change in expectations by varying testing properties. A wider examination of positioning solutions could discover the impact of missing infrastructure, variance in locomotion or location model uncertainty before deployment actually proceeds.

5.3 Conclusions

Location-based services such as fitness tracking, navigation and augmented reality gaming thrive outdoors where GPS is readily available. Indoors, GPS struggles to provide accurate location information, which reduces the range of available location-aware applications. WLAN positioning, often considered the de facto solution for many indoor LBS [HGK+18], has seen limited uptake despite decades of research. This can likely be attributed to the many real-world obstacles it has to face for widespread adoption.

Our experiences from real world deployments, which have motivated the contributions of this thesis, have shown that the WLAN positioning lifecycle contains several vulnerable phases, all ultimately relating to uncertainty of the location-dependent signal. Depending on which phase of the lifecycle this uncertainty manifests, different approaches are required for mitigating them. First, the early intuition of substituting the complex description of the environment through a purely physical modeling of the signal with empirical approaches has only bought

a limited reprieve from effort. To provide high-resolution position estimates, a significant amount of data covering all traversable regions of the environment is required. During this so-called *calibration survey*, measurements are labeled with real-world locations, often through rigorous manual effort. Before this measurement phase even takes place, however, the *environment topology* and the *location labels* it relates to needs to be defined. This has often been engineered with the end application in mind, instead of constructing it based on the limitations of the underlying signal. This has caused great uncertainty for the algorithms trying to choose the best of two similar but not necessarily co-located options. This environment description has traditionally involved manual effort in defining which parts of the environment to model, and at which level of granularity.

Finally, though the ubiquity of the WLAN protocol as a communication medium can to a large extent be attributed to its unlicensed use of spectrum [Rit03] – requiring very little upfront cost especially if the network is already established – this free use of the frequency band has revealed itself to be a double-edged sword. Other technologies, many not adhering to as strict or polite a protocol as IEEE 802.11, vying for the same set of frequencies can hamper the connectivity of the network. For everyday communication, this has the effect of decreased throughput in the network or, in the worst case, a complete communications breakdown [RPB11]. In the positioning domain, this can have varying impact depending on when the disruption occurs. In practice, interference can raise the level of noise on the wireless channel, which in turn decreases the *signal-to-noise-ratio* available for communication. This decreases the potential range of access points and by proxy location information coverage. In morbid cases, a strong source of interference might even *jam* communication entirely, causing a loss of information in both the modeling and application phases of positioning. Any mismatch between the calibration and testing phase will inherently imbue the system with a level of unwanted uncertainty, but unexpected gaps of information can also violate the smoothness assumption of many approaches based on the signal strength fingerprinting approach. The end result of such uncertainty is an inconsistent performance in the end-user application.

This thesis has presented ways to alleviate these issues, and bring WLAN positioning closer to the ubiquitous solution initially envisioned. Machine learning algorithms need to include the inherent non-linearity of the domain in their models of the environment. Harnessing this notion can alleviate, or even eliminate, much of the efforts that are traditionally manual. Additionally, aspects external to the actual positioning model, such as other technologies, need also be taken into account to ensure robust location-based services.

5.4 Summary of Contributions

In the following, we revisit the topics explored in this thesis and summarize the contributions of each publication.

5.4.1 Location-based Service in a Supermarket Environment (Section 2.4, Article I)

We provided insights into the inaccuracies of commercial positioning systems in everyday contexts of practical interest. We furthermore detailed the added effort required to mitigate these inadequacies in order to enable location-based services in a supermarket environment. Finally, we investigated means to bridge the gap caused by the cost of deploying a positioning system and the interests of different stakeholders. By instrumenting navigation instructions with brands instead of generic markers – providing an opportunity for marketing for the system owner – an increased amount of attention was captured without disrupting the navigation experience. The lessons learned from this real-world deployment in a complex everyday environment provided motivation for our further contributions.

5.4.2 Signal Space Modeling (Chapter 3, Article II & Article III)

Though it was shown that a state-of-the-art positioning system could be instrumented post-hoc to serve as a provider of location context for commercial applications, it nevertheless required manual effort that does not easily generalize to other environments. The survey effort was to some extent reduced due to a convenient segmented path survey, but still required multiple passes from different directions to achieve the required accuracy. Furthermore, the discretization of the environment was initially performed to serve the product categories of the supermarket rather than the constraints of the underlying positioning technology. This resulted in the need for further abstraction in order to ensure that navigation instructions were not only played in the correct location but also with proper timing in order to provide a smooth navigation experience.

Semi-supervised Learning

We showed that by extracting a two-dimensional embedding from an inherently multidimensional signal space, a rough approximation of the environment topology could be discovered in an unsupervised way. A further semi-supervised so-

phistication was then able to propagate labels from a select number of anchor points to measurements that have no real-world mapping. This discovery showed that a spatial dependency within the signal space exists and can be exploited in a simple yet principled way to decimate the calibration effort.

Automatic Environment Partitioning

The previously described spatial dependency of the signal space was further exploited to detect anomalies in the signal data. By exploring the way signal strength fingerprints interacted with their neighbors in an automatically constructed signal space mapping, we could identify areas in the environment with poor variability. Sequentially merging weak co-located areas ensured that the size of the positioning model could be reduced by up to 60% without impacting the accuracy of the positioning system, all in a completely automated way. This same intuition could even be extended to suggesting locations for further access point placements, in order to enrich poorly separable areas.

5.4.3 Detecting Competing Technologies (Chapter 4, Article IV & Article V)

When interference impacts a WLAN network, the subject of this disruption is often the very thing that most WLAN positioning systems use as a source of context awareness: the signal strength of received frames. Any source of remediation for these issues thus needs to analyze a medium that is independent of the actual WLAN protocol. Measurements from a spectrum analyzer provide a rich description of the spectrum and can provide enough information for approaches like CNNs – previously employed to great success for image recognition – to infer the likely source of non-conformant energy.

Semi-supervised Detection

We showed that even a relatively shallow convolutional network can learn the features of a wide variety of potential interferers, classifying the correct device in 97% of cases. When this network is extended with a temporal modeling aspect through a Markovian interpretation of sequential measurements, the approach is improved even further, especially when learning with a constrained set of data.

Real-world Interference Detection

Bringing a similar deep learning approach to the real world and comparing it to a relatively simple, yet effective, linear baseline shows that in many respects a deep learning approach requires more varied data to provide a solution equivalent to the linear approach. Though this linear alternative in extreme cases suffers from a polynomial computational complexity, it also allows for more interpretable results and real-world practicality by providing a consistent estimate of the interferer transmit power. This could significantly reduce efforts to locate the device in question, since known path loss propagation formulas can be used to translate the solved power to real-world proximity. The added location context of interference could also help inform the design of a wireless network for positioning, even if the direct effect could not be mitigated.

References

[AAHA11] Andam-Arkoful, S., Hill, C. J. and Atuahene, I., Evaluation of ultra wide-band for indoor positioning. *Annual Conference and Expo of the Institute of Industrial Engineers*, 2011, pages 1–8.

[ACK94] Asthana, A., Crauatts, M. and Krzyzanowski, P., An indoor wireless system for personalized shopping assistance. *Workshop on Mobile Computing Systems and Applications*, 1994, pages 69–74.

[AER+19] Abbas, M., Elhamshary, M., Rizk, H., Torki, M. and Youssef, M., WiDeep: WiFi-based accurate and robust indoor localization system using deep learning. *IEEE International Conference on Pervasive Computing and Communications*, 2019, pages 1–10.

[AM98] Aylott, R. and Mitchell, V.-W., An exploratory study of grocery shopping stressors. *International Journal of Retail and Distribution Management*, 26, pages 362–373.

[AMJ+14] Abdel-Hamid, O., Mohamed, A., Jiang, H., Deng, L., Penn, G. and Yu, D., Convolutional neural networks for speech recognition. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 22,10(2014), pages 1533–1545.

[ARG16] Aguirre, E., Roggeveen, A. L. and Grewal, D., The personalization-privacy paradox: implications for new media. *Journal of Consumer Marketing*, 33,2(2016), pages 98–110.

[ASVN12] Asp, A., Sydorov, Y., Valkama, M. and Niemelä, J., Radio signal propagation and attenuation measurements for modern residential buildings. *IEEE Globecom Workshops*, Dec 2012, pages 580–584.

[BC13] Banerji, S. and Chowdhury, R., On IEEE 802.11: Wireless LAN technology. *International Journal of Mobile Network Communications & Telematics*, 3.

[BFF+12] Bhattacharya, S., Floréen, P., Forsblom, A., Hemminki, S., Myllymäki, P., Nurmi, P., Pulkkinen, T. and Salovaara, A., Ma$$iv€ – an intelligent mobile grocery assistant. *International Conference on Intelligent Environments*, June 2012, pages 165–172.

[BGVGT+17] Brena, R. F., García-Vázquez, J. P., Galván-Tejada, Carlos, E., Muñoz Rodriguez, D., Vargas-Rosales, C. and Fangmeyer, James, J., Evolution of indoor positioning technologies: A survey. *Journal of Sensors*, 2017.

[BIL+16] Bastani, O., Ioannou, Y., Lampropoulos, L., Vytiniotis, D., Nori, A. and Criminisi, A., Measuring neural net robustness with constraints. In *Advances in Neural Information Processing Systems 29*, Curran Associates, Inc., 2016, pages 2613–2621.

[BJJA05] Bohnenberger, T., Jacobs, O., Jameson, A. and Aslan, I., Decision-theoretic planning meets user requirements: Enhancements and studies of an intelligent shopping guide. *Pervasive Computing*, Berlin, Heidelberg, 2005, Springer Berlin Heidelberg, pages 279–296.

[BK08] Bucur, D. and Kjærgaard, M. B., Gammasense: Infrastructureless positioning using background radioactivity. *European Conference on Smart Sensing and Context*. Springer Berlin Heidelberg, 2008, pages 69–82.

[BLB05] Bocquet, M., Loyez, C. and Benlarbi-Delai, A., Using enhanced-TDOA measurement for indoor positioning. *IEEE Microwave and Wireless Components Letters*, 15,10(2005), pages 612–614.

[BLM+17] Basiri, A., Lohan, E. S., Moore, T., Winstanley, A., Peltola, P., Hill, C., Amirian, P. and e Silva, P. F., Indoor location based services challenges, requirements and usability of current solutions. *Computer Science Review*, 24,C(2017), pages 1–12.

[BMR17] Bitar, N., Muhammad, S. and Refai, H. H., Wireless technology identification using deep convolutional neural networks. *IEEE An-*

nual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC), Oct 2017, pages 1–6.

[BOB07] Birch, C. P., Oom, S. P. and Beecham, J. A., Rectangular and hexagonal grids used for observation, experiment and simulation in ecology. *Ecological Modelling*, 206,3(2007), pages 347 – 359.

[BP00] Bahl, P. and Padmanabhan, V. N., RADAR: an in-building RF-based user location and tracking system. *Annual Joint Conference of the IEEE Computer and Communications Societies*, volume 2, 2000, pages 775–784 vol.2.

[CA19] Chidlovskii, B. and Antsfeld, L., Semi-supervised variational autoencoder for WiFi indoor localization. *International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, 2019, pages 1–8.

[CAD+18] Chakraborty, A., Alam, M., Dey, V., Chattopadhyay, A. and Mukhopadhyay, D., Adversarial attacks and defences: A survey, arXiv, 2018. URL `http://arxiv.org/abs/1810.00069`.

[CBM10] Chan, E. C. L., Baciu, G. and Mak, S. C., Effect of channel interference on indoor wireless local area network positioning. *IEEE 6th International Conference on Wireless and Mobile Computing, Networking and Communications*, 2010, pages 239–245.

[CCKM01] Castro, P., Chiu, P., Kremenek, T. and Muntz, R. R., A probabilistic room location service for wireless networked environments. *International Conference on Ubiquitous Computing*, UbiComp '01, Berlin, Heidelberg, 2001, Springer-Verlag, pages 18–34.

[Cha13] Chang, C.-I. *Hyperspectral Data Processing: Algorithm Design and Analysis*, chapter 3, page 351. Wiley, 2013.

[Cis20] Cisco, Wi-Fi location-based services 4.1 design guide, `https://www.cisco.com/c/en/us/td/docs/solutions/Enterprise/Mobility/WiFiLBS-DG/wifich5.html`, 2020. Accessed: 2020-05.

[Cor09] Cordeiro, C., Evaluation of medium access technologies for next generation millimeter-wave WLAN and WPAN. *IEEE Interna-*

tional Conference on Communications Workshops, June 2009, pages 1–5.

[CPIP10] Chintalapudi, K., Padmanabha Iyer, A. and Padmanabhan, V. N., Indoor localization without the pain. *International Conference on Mobile Computing and Networking*, MobiCom '10. ACM, 2010, pages 173–184.

[CT06] Cover, T. M. and Thomas, J. A., *Elements of Information Theory (Wiley Series in Telecommunications and Signal Processing)*. Wiley-Interscience, New York, NY, USA, 2006.

[dBQAG+17] de Blasio, G., Quesada-Arencibia, A., García, C. R., Molina-Gil, J. M. and Caballero-Gil, C., Study on an indoor positioning system for harsh environments based on Wi-Fi and bluetooth low energy. *Sensors (Basel, Switzerland)*, 17,6(2017), page 1299.

[DCSM16] Dickinson, P., Cielniak, G., Szymanezyk, O. and Mannion, M., Indoor positioning of shoppers using a network of bluetooth low energy beacons. *International Conference on Indoor Positioning and Indoor Navigation*, Oct 2016, pages 1–8.

[DNXY19] Dong, J., Noreikis, M., Xiao, Y. and Ylä-Jääski, A., ViNav: A vision-based indoor navigation system for smartphones. *IEEE Transactions on Mobile Computing*, 18,6(2019), pages 1461–1475.

[DP17] Davidson, P. and Piché, R., A survey of selected indoor positioning methods for smartphones. *IEEE Communications Surveys Tutorials*, 19,2(2017), pages 1347–1370.

[DWWZ18] Ding, L., Wang, S., Wang, F. and Zhang, W., Specific emitter identification via convolutional neural networks. *IEEE Communications Letters*, 22,12(2018), pages 2591–2594.

[DYWY19] Dai, P., Yang, Y., Wang, M. and Yan, R., Combination of DNN and improved KNN for indoor location fingerprinting. *Wireless Communications and Mobile Computing*, 2019, pages 1–9.

[EKSX96] Ester, M., Kriegel, H.-P., Sander, J. and Xu, X., A density-based algorithm for discovering clusters a density-based algorithm for

discovering clusters in large spatial databases with noise. *International Conference on Knowledge Discovery and Data Mining*. AAAI Press, 1996, pages 226–231.

[EYU+16] Elhamshary, M., Youssef, M., Uchiyama, A., Yamaguchi, H. and Higashino, T., TransitLabel: A crowd-sensing system for automatic labeling of transit stations semantics. *Annual International Conference on Mobile Systems, Applications, and Services*. Association for Computing Machinery, 2016, pages 193–206.

[FCC47] FCC, Thirteenth annual report, fiscal year ending June 30, 1947. Technical Report, Federal Communications Commission, 1947.

[FFL07] Ferris, B., Fox, D. and Lawrence, N., WiFi-SLAM using gaussian process latent variable models. *International Joint Conference on Artifical Intelligence*, IJCAI'07, San Francisco, CA, USA, 2007, Morgan Kaufmann Publishers Inc., pages 2480–2485.

[FHF06] Ferris, B., Hähnel, D. and Fox, D., Gaussian processes for signal strength-based location estimation. *Robotics: Science and Systems*.

[FJAD18] Falkowski, T., Jürgenhake, C., Anacker, H. and Dumitrescu, R., Feature model for the specification of industrial indoor location-based services. *International Conference on System-Integrated Intelligence: Intelligent, Flexible and Connected Systems in Products and Production*, volume 24, 2018, pages 141 – 146.

[FLC+18] Feng, C., Li, X., Chang, L., Xiong, J., Chen, X., Fang, D., Liu, B., Chen, F. and Zhang, T., Material identification with commodity Wi-Fi devices. *ACM Conference on Embedded Networked Sensor Systems*, SenSys '18. ACM, 2018, pages 382–383.

[Fuk80] Fukushima, K., Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological Cybernetics*, 36,4(1980), pages 193–202.

[GA15] Gudonavičienė, R. and Alijošienė, S., Visual merchandising impact on impulse buying behaviour. *Procedia - Social and Behavioral Sciences*, 213, pages 635 – 640. International Scientific Conference on Economics and Management.

[Gal08] Gallager, R. G., *Principles of Digital Communication*. Cambridge University Press, 2008.

[GFCM17] Gonçalves Ferreira, A. F. G., Fernandes, D. M. A., Catarino, A. P. and Monteiro, J. L., Localization and positioning systems for emergency responders: A survey. *IEEE Communications Surveys Tutorials*, 19,4(2017), pages 2836–2870.

[GH16] Gao, C. and Harle, R., Easing the survey burden: Quantitative assessment of low-cost signal surveys for indoor positioning. *International Conference on Indoor Positioning and Indoor Navigation*, Oct 2016, pages 1–8.

[GLT03] Ghammaz, A., Lefeuvre, S. and Teissandier, N., Spectral behavior of domestic microwave ovens and its effects on the ISM band. *Annales Des Télécommunications*, 58,7(2003), pages 1178–1188.

[GLY14] Grover, K., Lim, A. and Yang, Q., Jamming and anti-jamming techniques in wireless networks: A survey. *International Journal of Ad Hoc and Ubiquitous Computing*, 17,4(2014), pages 197–215.

[Gol05] Goldsmith, A., *Wireless Communications*. Cambridge University Press, 2005.

[Gro11] Groves, P. D., Shadow matching: A new GNSS positioning technique for urban canyons. *The Journal of Navigation*, 64,3(2011), pages 417–430.

[GSS93] Gordon, N. J., Salmond, D. J. and Smith, A. F. M., Novel approach to nonlinear/non-gaussian bayesian state estimation. *IEE Proceedings F - Radar and Signal Processing*, 140,2(1993), pages 107–113.

[GWGS07] Gummadi, R., Wetherall, D., Greenstein, B. and Seshan, S., Understanding and mitigating the impact of RF interference on 802.11 networks. *Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, SIGCOMM 07, New York, NY, USA, 2007, Association for Computing Machinery, pages 385–396.

[GZP19] Grais, E. M., Zhao, F. and Plumbley, M. D., Multi-band multi-resolution fully convolutional neural networks for singing voice separation, 2019.

[Hay98] Haykin, S., *Neural Networks: A Comprehensive Foundation*. Prentice Hall PTR, second edition, 1998.

[HB04] Hightower, J. and Borriello, G., Particle filters for location estimation in ubiquitous computing: A case study. *International Conference on Ubiquitous Computing*, Davies, N., Mynatt, E. D. and Siio, I., editors, Berlin, Heidelberg, 2004, Springer Berlin Heidelberg, pages 88–106.

[HFL+04] Haeberlen, A., Flannery, E., Ladd, A. M., Rudys, A., Wallach, D. S. and Kavraki, L. E., Practical robust localization over large-scale 802.11 wireless networks. *Annual International Conference on Mobile Computing and Networking*. ACM, 2004, pages 70–84.

[HGK+18] Huang, H., Gartner, G., Krisp, J. M., Raubal, M. and de Weghe, N. V., Location based services: ongoing evolution and research agenda. *Journal of Location Based Services*, 12,2(2018), pages 63–93.

[HHSW11] Halperin, D., Hu, W., Sheth, A. and Wetherall, D., Tool release: Gathering 802.11n traces with channel state information. *SIGCOMM Computer Communication Review*, 41,1(2011), page 53.

[HK09] Haverinen, J. and Kemppainen, A., Global indoor self-localization based on the ambient magnetic field. *Robotics and Autonomous Systems*, 57,10(2009), pages 1028 – 1035. International Conference on Computational Intelligence, Robotics and Autonomous Systems.

[HK11] Hong, S. S. and Katti, S. R., DOF: A local wireless information plane. *ACM SIGCOMM Conference*, SIGCOMM '11, New York, NY, USA, 2011, ACM, pages 230–241.

[HPAP09] Honkavirta, V., Perälä, T., Ali-Löytty, S. and Piché, R., A comparative survey of WLAN location fingerprinting methods. *Workshop on Positioning, Navigation and Communication*, March 2009, pages 243–251.

[HSK+12] Hinton, G. E., Srivastava, N., Krizhevsky, A., Sutskever, I. and Salakhutdinov, R. R., Improving neural networks by preventing co-adaptation of feature detectors, 2012.

[HTF01] Hastie, T., Tibshirani, R. and Friedman, J., *The Elements of Statistical Learning*. Springer Series in Statistics. Springer New York Inc., New York, NY, USA, 2001.

[HTKM17] Hayashi, T., Taniuchi, D., Korpela, J. and Maekawa, T., Spatio-temporal adaptive indoor positioning using an ensemble approach. *Pervasive and Mobile Computing*, 41, pages 319 – 332.

[IEE16] IEEE, IEEE standard for information technology - telecommunications and information exchange between systems local and metropolitan area networks - specific requirements - part 11: Wireless LAN medium access control (MAC) and physical layer (PHY) specifications. *IEEE Std 802.11-2016 (Revision of IEEE Std 802.11-2012)*.

[Int20] Intel, Usb 3.0 radio frequency interference on 2.4 GHz devices, `https://www.intel.com/content/www/us/en/products/docs/io/universal-serial-bus/usb3-frequency-interference-paper.html`, 2020. Accessed: 2020-01.

[ISO16] ISO, Information technology - Real time locating systems - Test and evaluation of localization and tracking systems. Standard, International Organization for Standardization, Geneva, CH, November 2016. ISO/IEC 18305:2016.

[JSPG09] Jimenez, A. R., Seco, F., Prieto, C. and Guevara, J., A comparison of pedestrian dead-reckoning algorithms using a low-cost MEMS IMU. *IEEE International Symposium on Intelligent Signal Processing*, Aug 2009, pages 37–42.

[KBG+10] Kjærgaard, M. B., Blunck, H., Godsk, T., Toftkjær, T., Christensen, D. L. and Grønbæk, K., Indoor positioning using GPS revisited. *Pervasive Computing*. Springer Berlin Heidelberg, 2010, pages 38–56.

[KC11] Koo, J. and Cha, H., Autonomous construction of a WiFi access point map using multidimensional scaling. *Pervasive Computing*, Berlin, Heidelberg, 2011, Springer Berlin Heidelberg, pages 115–132.

[KH04] Krumm, J. and Horvitz, E., LOCADIO: inferring motion and location from Wi-Fi signal strengths. *International Conference on Mobile and Ubiquitous Systems: Networking and Services*, Aug 2004, pages 4–13.

[KHLH03] Kotanen, A., Hännikäinen, M., Leppäkoski, H. and Hämäläinen, T. D., Positioning with IEEE 802.11b wireless LAN. *IEEE Proceedings on Personal, Indoor and Mobile Radio Communications*, volume 3, Sep. 2003, pages 2218–2222.

[Kjæ07] Kjærgaard, M. B., A taxonomy for radio location fingerprinting. *International Conference on Location- and Context-Awareness*, LoCA 07, Berlin, Heidelberg, 2007, Springer-Verlag, pages 139–156.

[Kjæ12] Kjærgaard, M. B., Location-based services on mobile phones: minimizing power consumption. *IEEE Pervasive Computing*, 11,1(2012), pages 67–73.

[KJBK15] Kotaru, M., Joshi, K., Bharadia, D. and Katti, S., SpotFi: Decimeter level localization using WiFi. *ACM Conference on Special Interest Group on Data Communication*, New York, NY, USA, 2015, Association for Computing Machinery, pages 269–282.

[KK04] Kaemarungsi, K. and Krishnamurthy, P., Properties of indoor received signal strength for WLAN location fingerprinting. *International Conference on Mobile and Ubiquitous Systems: Networking and Services*, 2004, pages 14–23.

[KKMD18] Kulin, M., Kazaz, T., Moerman, I. and De Poorter, E., End-to-end learning from spectrum data: A deep learning approach for wireless signal identification in spectrum monitoring applications. *IEEE Access*, 6, pages 18484–18501.

[KM08] Kjærgaard, M. B. and Munk, C. V., Hyperbolic location fingerprinting: A calibration-free solution for handling differences in signal strength. *International Conference on Pervasive Computing*

and Communications, Washington, DC, USA, 2008, IEEE Computer Society, pages 110–116.

[KNO+01] Klepeis, N. E., Nelson, W. C., Ott, W. R., Robinson, J. P., Tsang, A. M., Switzer, P., Behar, J. V., Hern, S. C. and Engelmann, W. H., The national human activity pattern survey (NHAPS): a resource for assessing exposure to environmental pollutants. *Journal of Exposure Science & Environmental Epidemiology*, 11,3(2001), pages 231–252.

[KNS14] Kim, K.-H., Nam, H. and Schulzrinne, H., WiSlow: A Wi-Fi network performance troubleshooting tool for end users. *IEEE Conference on Computer Communications*, 04 2014, pages 862–870.

[Koh82] Kohonen, T., Self-organized formation of topologically correct feature maps. *Biological Cybernetics*, 43,1(1982), pages 59–69.

[Koh97] Kohonen, T., The basic SOM. In *Self-Organizing Maps*, Springer Berlin Heidelberg, 1997, pages 85–144.

[KP03] Krumm, J. and Platt, J., Minimizing calibration effort for an indoor 802.11 device location measurement system. Technical Report MSR-TR-2003-82, Microsoft Research, November 2003.

[KSH17] Krizhevsky, A., Sutskever, I. and Hinton, G. E., Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60,6(2017), pages 84–90.

[Kup05] Kupper, A., *Location-based Services: Fundamentals and Operation*. John Wiley & Sons, Inc., USA, 2005.

[LB98] LeCun, Y. and Bengio, Y., Convolutional networks for images, speech, and time series. In *The Handbook of Brain Theory and Neural Networks*, Arbib, M. A., editor, MIT Press, Cambridge, MA, USA, 1998, pages 255–258.

[LeC88] LeCun, Y., A theoretical framework for back-propagation. *Connectionist Models Summer School*. Morgan Kaufmann, 1988, pages 21–28.

[Lee13] Lee, D.-H., Pseudo-label : The simple and efficient semi-supervised learning method for deep neural networks. *International Conference on Machine Learning Workshop : Challenges in Representation Learning*, 2013.

[LH95] Lawson, C. L. and Hanson, R. J., *Solving least squares problems*, volume 15 of *Classics in Applied Mathematics*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1995.

[LJW14] Li, Y. and Jack Wang, J., A pedestrian navigation system based on low cost IMU. *Journal of Navigation*, 67,6(2014), pages 929–949.

[LKE08] Lemelson, H., King, T. and Effelsberg, W., Pre-processing of fingerprints to improve the positioning accuracy of 802.11-based positioning systems. *ACM International Workshop on Mobile Entity Localization and Tracking in GPS-Less Environments*, New York, NY, USA, 2008, ACM, pages 73–78.

[LKHK09] Lemelson, H., Kjærgaard, M. B., Hansen, R. and King, T., Error estimation for indoor 802.11 location fingerprinting. *Location and Context Awareness*, Choudhury, T., Quigley, A., Strang, T. and Suginuma, K., editors, Berlin, Heidelberg, 2009, Springer Berlin Heidelberg, pages 138–155.

[LKM13] Li, M., Kim, B. and Mourikis, A., Real-time motion tracking on a cellphone using inertial sensing and a rolling-shutter camera. *IEEE International Conference on Robotics and Automation*, 05 2013, pages 4712–4719.

[LMH18] Le, D. V., Meratnia, N. and Havinga, P. J. M., Unsupervised deep feature learning to reduce the collection of fingerprints for indoor localization using deep belief networks. *International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, 2018, pages 1–7.

[LPK17] Longi, K., Pulkkinen, T. and Klami, A., Semi-supervised convolutional neural networks for identifying Wi-Fi interference sources. *Proceedings of Machine Learning Research*, volume 77, 11 2017, pages 391–406.

[LQD08] Li, B., Quader, I. and Dempster, A., On outdoor positioning with Wi-Fi. *Journal of Global Positioning Systems*, 7, pages 18–26.

[LSE10] Lemelson, H., Schnaufer, S. and Effelsberg, W., Automatic identification of fingerprint regions for quick and reliable location estimation. *IEEE International Conference on Pervasive Computing and Communications Workshops*, March 2010, pages 540–545.

[LV07] Lee, J. A. and Verleysen, M., Method comparisons. In *Nonlinear Dimensionality Reduction*, Springer New York, New York, NY, 2007, chapter 6, pages 173–223.

[MAP07] Mogilner, C., Aaker, J. L. and Pennington, G. L., Time will tell: The distant appeal of promotion and imminent appeal of prevention. *Journal of Consumer Research*, 34,5(2007), pages 670–681.

[MBMS17] Molina, L., Blanc, A., Montavont, N. and Simić, L., Identifying channel saturation in Wi-Fi networks via passive monitoring of IEEE 802.11 beacon jitter. *ACM International Symposium on Mobility Management and Wireless Access*, New York, NY, USA, 2017, ACM, pages 63–70.

[MD14] Moghtadaiee, V. and Dempster, A. G., Indoor location fingerprinting using FM radio signals. *IEEE Transactions on Broadcasting*, 60,2(2014), pages 336–346.

[Mis07] Misikangas, P., Location determination techniques, 2007. US20070149216A1.

[MMH+17] Miettinen, M., Marchal, S., Hafeez, I., Asokan, N., Sadeghi, A. and Tarkoma, S., IoT SENTINEL: Automated device-type identification for security enforcement in IoT. *International Conference on Distributed Computing Systems*, June 2017, pages 2177–2184.

[Mol19] Molnar, C., *Interpretable Machine Learning*, 2019 (accessed October 20th, 2019). URL `https://christophm.github.io/interpretable-ml-book/`.

[MRSN18] Merchant, K., Revay, S., Stantchev, G. and Nousain, B., Deep learning for RF device fingerprinting in cognitive communication networks. *IEEE Journal of Selected Topics in Signal Processing*, 12,1(2018), pages 160–167.

[MZW19] Ma, Y., Zhou, G. and Wang, S., WiFi sensing with channel state information: A survey. *ACM Computing Surveys*, 52,3(2019).

[NA12] Nixon, M. S. and Aguado, A. S., Chapter 3 - basic image processing operations. In *Feature Extraction & Image Processing for Computer Vision (Third Edition)*, Academic Press, Oxford, third edition edition, 2012, pages 83 – 136.

[NBK10] Nurmi, P., Bhattacharya, S. and Kukkonen, J., A grid-based algorithm for on-device GSM positioning. *International Conference on Ubiquitous Computing*, UbiComp '10. ACM, 2010, pages 227–236.

[NGL+13] Niu, J., Gu, Y., Lu, B., Cheng, L. and Jun, J. H., WiFi fingerprint localization in open space. *IEEE Conference on Local Computer Networks*, 2013.

[Nie94] Nielsen, J., Enhancing the explanatory power of usability heuristics. *SIGCHI Conference on Human Factors in Computing Systems*, New York, NY, USA, 1994, Association for Computing Machinery, pages 152–158.

[NLB+08] Nurmi, P., Lagerspetz, E., Buntine, W., Floréen, P. and Kukkonen, J., Product retrieval for grocery stores. *Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '08, New York, NY, USA, 2008, ACM, pages 781–782.

[NNL20] NNLS, Non-negative least squares solver, `https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.nnls.html`, 2020. Accessed: 2020-01.

[NSB+11] Nurmi, P., Salovaara, A., Bhattacharya, S., Pulkkinen, T. and Kahl, G., Influence of landmark-based navigation instructions on user attention in indoor smart spaces. *International Conference on Intelligent User Interfaces (IUI)*. ACM, 2011, pages 33–42.

[NSF+14] Nurmi, P., Salovaara, A., Forsblom, A., Bohnert, F. and Floréen, P., Promotionrank: Ranking and recommending grocery product promotions using personal shopping lists. *ACM Transactions on Interactive Intelligent Systems*, 4,1(2014).

[OAAJ+18] Oguntala, G., Abd-Alhameed, R., Jones, S., Noras, J., Patwary, M. and Rodriguez, J., Indoor location identification technologies for real-time IoT-based applications: An inclusive survey. *Computer Science Review*, 30, pages 55 – 79.

[PCC+10] Park, J.-g., Charrow, B., Curtis, D., Battat, J., Minkov, E., Hicks, J., Teller, S. and Ledlie, J., Growing an organic indoor location system. *International Conference on Mobile Systems, Applications, and Services*, MobiSys '10, New York, NY, USA, 2010, ACM, pages 271–284.

[Phi00] Phillips Business Information Corporation, President ends GPS selective availability. *Global Positioning & Navigation News*, 10,9(2000).

[PLC+17] Pei, L., Liu, J., Chen, Y., Chen, R. and Chen, L., Evaluation of fingerprinting-based WiFi indoor localization coexisted with bluetooth. *The Journal of Global Positioning Systems*, 15,1(2017), page 3.

[PN12] Pulkkinen, T. and Nurmi, P., AWESOM: Automatic discrete partitioning of indoor spaces for WiFi fingerprinting. *International Conference on Pervasive Computing (Pervasive)*. Springer-Verlag, 2012, pages 271–288.

[PNN20] Pulkkinen, T., Nurminen, J. K. and Nurmi, P., Understanding WiFi cross-technology interference detection in the real world. *International Conference on Distributed Computing Systems (ICDCS)*. IEEE, 2020. In press.

[POM10] Popleteev, A., Osmani, V. and Mayora, O., Indoor positioning using FM radio. *International Journal of Handheld Computing Research*, 1, pages 19–31.

[PRM11] Pulkkinen, T., Roos, T. and Myllymäki, P., Semi-supervised learning for WLAN positioning. *International Conference on Artificial Neural Networks (ICANN)*. Springer-Verlag, 2011, pages 355–362.

[PVN15] Pulkkinen, T., Verwijnen, J. and Nurmi, P., WiFi positioning with propagation-based calibration. *International Conference on Information Processing in Sensor Networks (IPSN)*. ACM, 2015, pages 366–367.

[Quu20] Quuppa, Intelligent locating system, `https://quuppa.com/`, 2020. Accessed: 2020-01.

[Rap01] Rappaport, T., *Wireless Communications: Principles and Practice*. Prentice Hall PTR, second edition, 2001.

[RCPS12] Rai, A., Chintalapudi, K. K., Padmanabhan, V. N. and Sen, R., Zee: Zero-effort crowdsourcing for indoor localization. *Annual International Conference on Mobile Computing and Networking*, Mobicom '12, New York, NY, USA, 2012, ACM, pages 293–304.

[Rit03] Ritter, M. W., The future of WLAN. *Queue*, 1,3(2003), pages 18–27.

[RMT+02] Roos, T., Myllymäki, P., Tirri, H., Misikangas, P. and Sievänen, J., A probabilistic approach to WLAN user location estimation. *International Journal of Wireless Information Networks*, 9, pages 155–164.

[RNG16] Roggeveen, A. L., Nordfält, J. and Grewal, D., Do digital displays enhance sales? role of retail format and message content. *Journal of Retailing*, 92,1(2016), pages 122 – 131.

[RPB11] Rayanchu, S., Patro, A. and Banerjee, S., Airshark: Detecting non-WiFi RF devices using commodity WiFi hardware. *ACM SIGCOMM Conference on Internet Measurement Conference*, IMC '11. ACM, 2011, pages 137–154.

[RSIC18] Riyaz, S., Sankhe, K., Ioannidis, S. and Chowdhury, K., Deep learning convolutional neural networks for radio identification. *IEEE Communications Magazine*, 56,9(2018), pages 146–152.

[RSK19] Ranasinghe, C., Schiestel, N. and Kray, C., Visualising location uncertainty to support navigation under degraded gps signals: A comparison study. *21st International Conference on Human-Computer Interaction with Mobile Devices and Services*. Association for Computing Machinery, 2019.

[RW13] Roback, D. and Wakefield, R. L., Privacy risk versus socialness in the decision to use mobile location-based applications. *ACM SIGMIS Database*, 44,2(2013), pages 19–38.

[SBT+06] Schrooyen, F., Baert, I., Truijen, S., Pieters, L., Denis, T., Williame, K. and Weyn, M., Real time location system over WiFi in a healthcare environment. *Journal on Information Technology in Healthcare*, 4,6(2006), pages 401–416.

[SG15] Sati, S. and Graffi, K., Adapting the beacon interval for opportunistic network communications. *International Conference on Advances in Computing, Communications and Informatics*, Aug 2015, pages 6–12.

[SG19] Spieker, H. and Gotlieb, A., Towards testing of deep learning systems with training set reduction, arXiv, 2019. URL `https://arxiv.org/abs/1901.04169`.

[SHW07] Shao, C., Huang, H. and Wan, C., Selection of the suitable neighborhood size for the isomap algorithm. *International Joint Conference on Neural Networks*, Aug 2007, pages 300–305.

[SI13] Sheehan, D. and Ittersum, K. V., Close, yet so far away: the influence of temporal distance on mobile promotion redemption during a shopping experience. *Advances in Consumer Research*, 41, pages 223 – 224.

[SKI17] Soomro, Y. A., Kaimkhani, S. A. and Iqbal, J., Effect of visual merchandising elements of retail store on consumer attentione. *Journal of Business Strategies*, 11,1(2017).

[Smi97] Smith, S. W., *The Scientist and Engineer's Guide to Digital Signal Processing*. California Technical Publishing, San Diego, CA, USA, 1997.

[SMR06] Samko, O., Marshall, A. and Rosin, P., Selection of the optimal parameter value for the isomap algorithm. *Pattern Recognition Letters*, 27,9(2006), pages 968 – 979.

[SNSK05] Shido, M., Nakamura, T., Serita, T. and Kando, M., High-pressure microwave discharges in a compact quartz tube for an electrodeless high intensity discharge lamp. *IEEJ Transactions on Fundamentals and Materials*, 125,6(2005), pages 495–500.

[SSW15] Sun, S., Shi, H. and Wu, Y., A survey of multi-source domain adaptation. *Information Fusion*, 24, pages 84 – 92.

[SZ15] Simonyan, K. and Zisserman, A., Very deep convolutional networks for large-scale image recognition. *International Conference on Learning Representations*, 2015.

[Tcp20] Tcpdump Group, tcpdump, libpcap, `https://www.tcpdump.org/`, 2020. Accessed: 2020-01.

[TSL00] Tenenbaum, J. B., Silva, V. d. and Langford, J. C., A global geometric framework for nonlinear dimensionality reduction. *Science*, 290,5500(2000), pages 2319–2323.

[UAAL19] Ubom, E., Akpanobong, A., Abraham, I. and Leo, A., Characterization of indoor propagation properties and performance evaluation for 2.4Ghz band Wi-Fi. *SSRN Electronic Journal*, 11, page 10.

[vdMH08] van der Maaten, L. and Hinton, G., Visualizing high-dimensional data using t-SNE. *Journal of Machine Learning Research*, 9, pages 2579–2605.

[Vit67] Viterbi, A., Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Transactions on Information Theory*, 13,2(1967), pages 260–269.

[Wal20] Walkbase, Asset, `https://www.walkbase.com/`, 2020. Accessed: 2020-01.

[Wan13] Wang, J., Zigbee light link and its applicationss. *IEEE Wireless Communications*, 20,4(2013), pages 6–7.

[WB15] Wang, M. and Brassil, J., Managing large scale, ultra-dense beacon deployments in smart campuses. *IEEE Conference on Computer Communications Workshops*, April 2015, pages 606–611.

[WGMP15] Wang, X., Gao, L., Mao, S. and Pandey, S., DeepFi: Deep learning for indoor fingerprinting using channel state information. *IEEE Wireless Communications and Networking Conference*, March 2015, pages 1666–1671.

[Wi-20] Wi-Fi Alliance, `https://www.wi-fi.org/`, 2020. Accessed: 2020-01.

[WKM08] Wong, C., Klukas, R. and Messier, G. G., Using WLAN infrastructure for angle-of-arrival indoor user location. *2008 IEEE 68th Vehicular Technology Conference*, 2008, pages 1–5.

[WOK14] Weng, Z., Orlik, P. and Kim, K. J., Classification of wireless interference on 2.4GHz spectrum. *IEEE Wireless Communications and Networking Conference*, 2014, pages 786–791.

[WS19] Wiklundh, K. and Stenumgaard, P., EMC challenges for the era of massive Internet of Things. *IEEE Electromagnetic Compatibility Magazine*, 8,2(2019), pages 65–74.

[WSE+12] Wang, H., Sen, S., Elgohary, A., Farid, M., Youssef, M. and Choudhury, R. R., No need to war-drive: Unsupervised indoor localization. *International Conference on Mobile Systems, Applications, and Services*, MobiSys '12. ACM, 2012, pages 197–210.

[WTD+13] Wang, B., Toobaei, M., Danskin, R., Ngarmnil, T., Pham, L. and Pham, H., Evaluation of RFID and Wi-Fi technologies for RTLS applications in healthcare centers. *Proceedings of Technology Management in the IT-Driven Services*, July 2013.

[WYM17] Wang, X., Yang, C. and Mao, S., TensorBeat: Tensor decomposition for monitoring multiperson breathing beats with commodity WiFi. *ACM Transactions on Intelligent Systems and Technology*, 9,1(2017), pages 8:1–8:27.

[YA05] Youssef, M. and Agrawala, A., The horus WLAN location determination system. *International Conference on Mobile Systems, Applications, and Services*. ACM, 2005, pages 205–218.

[YH10] Yazdandoost, K. Y. and Hamaguchi, K., Antenna polarization mismatch in body area network communications. *European Conference on Antennas and Propagation*, April 2010, pages 1–4.

[YOT+05] Yamasaki, R., Ogino, A., Tamaki, T., Uta, T., Matsuzawa, N. and Kato, T., TDOA location system for IEEE 802.11b WLAN. *IEEE Wireless Communications and Networking Conference*, volume 4, 2005, pages 2338–2343.

[ZB11] Zandbergen, P. A. and Barbeau, S. J., Positional accuracy of assisted GPS data from high-sensitivity GPS-enabled mobile phones. *The Journal of Navigation*, 64,3(2011), pages 381–399.

[ZBG+19] Zhang, J. M., Barr, E. T., Guedj, B., Harman, M. and Shawe-Taylor, J., Perturbed model validation: A new framework to validate model relevance, arXiv, 2019. URL `https://arxiv.org/abs/1905.10201v2`.

[ZFL+04] Zhou, C., Frankowski, D., Ludford, P., Shekhar, S. and Terveen, L., Discovering personal gazetteers: An interactive clustering approach. *Annual ACM International Workshop on Geographic Information Systems*, GIS '04, New York, NY, USA, 2004, ACM, pages 266–273.

[ZHML19] Zhang, J. M., Harman, M., Ma, L. and Liu, Y., Machine learning testing: Survey, landscapes and horizons, arXiv, 2019. URL `http://arxiv.org/abs/1906.10742`.

[ZTL+18] Zhang, J., Tang, Z., Li, M., Fang, D., Nurmi, P. and Wang, Z., CrossSense: Towards cross-site and large-scale WiFi sensing. *Proceedings of the 24th Annual International Conference on Mobile Computing and Networking*, 2018.