Machine Learning Techniques for Device-Free Indoor Person Tracking



Doctoral Dissertation
Doctoral Program in Electrical, Electronics and Communications Engineering (33rd cycle)

# Machine Learning Techniques for Device Free Indoor Person Tracking

**Osama Bin Tariq**
* * * * * *

**Supervisors**
Prof. Mihai T. Lazarescu
Prof. Luciano Lavagno

**Doctoral Examination Committee:**
Prof. William Fornaciari, Referee, Politecnico di Milano
Prof. Roberto Passerone, Referee, Università di Trento
Prof. Mario R. Casu, Politecnico di Torino
Prof. Davide Quaglia, Università di Verona
Prof. Dirk Koch, University of Manchester

Politecnico di Torino
March 17, 2021

I hereby declare that, the contents and organisation of this dissertation constitute my own original work and does not compromise in any way the rights of third parties, including those relating to the security of personal data.

.....

Osama Bin Tariq

Turin, March 17, 2021

# Summary

Low-cost, low-maintenance, and accurate indoor localization of persons is important for several applications, such as health care, safety monitoring, and resource usage optimization. For example, assisted living applications can lower assistance costs and improve quality of life, which are increasingly important as the projected ratio between working-age and elderly people keeps on increasing.

Several smart home application can also benefit from indoor localization. For instance, room occupancy information, alongside other factors, can help reducing energy consumption by controlling the ambient temperature, lighting, and water consumption. Health care applications, such as those monitoring the human activity for extended periods, can detect behavioral changes (e.g., gait changes), which can help recognizing the early onset of diseases, e.g., Parkinson's. Moreover, presence monitoring systems can also detect unauthorized intrusions, e.g., through the house windows, which can be a sign of burglary attempt.

Many localization techniques have been researched over the years. However, no *silver bullet* solution emerged mainly because of the complexity of person localization in indoor environments. A large variety of solutions, such as ultra-wide band or Wi-Fi techniques, assume that the person carries a device to be visible to the localization system. However, always wearing or carrying a device is not realistic for all indoor conditions and activities, reducing the acceptance of such systems. Other solutions are privacy-invasive, such as image-based systems, or may require significant infrastructure changes, such as changing floors. Systems based on infrared (IR) radiation, e.g., using passive infrared detectors (PIRs) or thermopiles, can be significantly affected by various environmental heat sources.

Capacitive sensors provide unique advantages, such as simple installation, privacy, and low cost. The capacitive sensors can be used in transmit mode, shunt mode, or load mode. The transmit and shunt modes use two electrodes: the oscillator electrode and the ground electrode. The capacitive sensors operating in load mode use a single oscillator electrode, while the human body and the environment act as ground. While two-plate capacitive sensors operating in transmit and shunt modes can be more accurate, single-plate sensors working in load mode are more attractive because of their simpler installation.

Capacitive sensors in load mode have been used to localize and identify persons

indoors, but their sensitivity decreases steeply with the distance. Long-range sensing, at distances of 10 to 15 times the plate diagonal, is highly susceptible to several environmental factors, such as electromagnetic and electrostatic noise, humidity, or temperature. Hence, the whole sensor data processing chain is crucial to achieving good localization accuracy and stability in variable environmental conditions. In this work, we investigate how machine learning techniques can help mitigating the effects of environmental electromagnetic noise and provide accurate person localization.

Human tracking in indoor environments can be divided in two main groups. The first one classifies the position of a person in a set of predefined locations. Such information can be useful particularly in smart homes to control energy consumption, to analyze the time spent at various locations, and so on. The second one, more complex, continuously tracks the position of a person inside a home. Such information can be useful for assisted living applications, for example, to analyze the gait and behavior deviations which can indicate possible health deterioration.

To determine the suitability of load-mode capacitive sensors for person localization, we designed a 3 m × 3 m *virtual* room divided in 16 equidistant locations on a 60 cm grid. In the middle of each wall of the room, we installed a long-range capacitive sensor, at a height of 115 cm from the floor, four sensors in total. We collected the experimental data through an extensive campaign of experiments, labeled them with the person's position within the room, and then used them to train and test a large set of machine learning (ML) classifiers to infer the location of the person in the room as follows:

1. Collection of time-stamped measurements from the four capacitive sensors in the room labeled with the actual position of the person;

2. Processing the sensor data, both for conditioning and for person localization using different ML classification algorithms from the Weka collection [1];

3. Analyzing the performance of the ML classification algorithms in terms of localization accuracy, average distance error, precision, and recall;

4. Analyzing the effect of the training data size on the localization performance of the algorithms.

We analyze the performance of most ML classification algorithms from the Weka collection for testing machine learning algorithms, with a particular focus on the best performing: k-NN, Bayes Net, Support Vector Machine (SVM), Random Forest, as well as boosting techniques, such as LogitBoost and AdaBoostM1. The best algorithms can provide good localization results even with limited sensor data preprocessing for noise filtering, and we also analyze the effects of the size of the training data on the localization results for different algorithms.

The ML classifiers achieved 93 % accuracy. Random Forest performed the best overall, while AdaBoostM1 used on top of the C4.5 required much less time for inference at the cost of a small accuracy loss.

After exploring the effectiveness of capacitive sensors for classifying the location of a person indoors, we addressed the more complex problem of the inference of the trajectory of a person moving freely inside the same experimental room, using a time- and location X/Y-labeled sequence of continuous data instead of unordered sets of discrete locations. For this, we increased the sampling rate of the capacitive sensors to 3 Hz to better capture the free movement of a person. To record the ground truth of the person's movements and label the data from four capacitive sensors, we used an accurate ultrasound reference system from Marvelmind, which required the person to carry a tag during the experiments.

With the collected data, we explored how advanced data processing can improve the capacitive sensor accuracy and reconstruct the trajectory of a person's movements. As before, we used relatively simple capacitive sensors, which are known to be very susceptible to environmental noise, to better compare the effectiveness of the various data processing chains on the overall accuracy.

We used several signal filtering combinations for preprocessing and then we optimized different neural network (NN) types e.g., (1) autoregressive feedforward, (2) 1D convolutional NN (CNN), and (3) long short-term memory (LSTM), through design space exploration (DSE). For NN training, validation, and testing, we used capacitive sensor data collected while a person moved freely in the 3 m × 3 m experimental room, and we compared the inferred position and trajectory with the reference location acquired using the reference Marvelmind ultrasound localization system.

we report in the thesis the main results on trajectory reconstruction based on data collected from capacitive sensors, which are:

- neural network-based signal processing techniques for indoor person localization and tracking using small capacitive sensors operating in load mode at long ranges (10 to 15 times their plate diagonal);
- noise attenuation using various kinds of digital filters and neural networks for location and trajectory inference;
- comparative analysis of NN-based location and movement dynamics inference accuracy from noisy sensor data.

We achieved a 25.1 cm localization root-mean-square error (RMSE) using the capacitive sensors, which is very good for such small, noisy, low-power, low-cost, privacy-aware sensors.

We also evaluated the effectiveness of proposed methods to reconstruct human trajectory using similar NNs but based on experimental data from 4 pixels × 4 pixels

thermopile infrared sensor. This analysis has two main goals: first to see the effectiveness of the ML methods developed in this thesis for other types of sensors, and second because the capacitive and thermopile sensors have several complementary advantages. In this case, we achieved a much lower localization RMSE of 9.6 cm.

# Acknowledgements

First and foremost I am extremely grateful to my supervisors, Prof. Mihai Lazarescu and Prof. Luciano Lavagno for their invaluable advice, continuous support, and patience during the course of my Ph.D. degree. Their immense knowledge and plentiful experience have encouraged me in all the time of my academic research.

I would like to thank Prof. William Fornaciari and Prof. Roberto Passerone for their comments and suggestions, which have enabled me to make significant improvements to the quality of this thesis. My gratitude extends to Prof. Mario R. Casu, Prof. Davide Quaglia, and Prof. Dirk Koch for being part of the doctoral examination committee.

I would like to thank Junnan Shan for providing knowledge and valuable contribution to the high-level synthesis projects. In addition to the individuals named above, the high-level synthesis and ambient sensing group as a whole has provided a friendly and vibrant atmosphere in which it was a pleasure to work.

# Contents

# List of Tables

# List of Figures

# Publications

The material presented in this thesis has been published in form of the following journals and conferences

- **Bin Tariq, O.**, Lazarescu, M. T., and Lavagno, L. (2021). **Neural Networks for Indoor Person Tracking With Infrared Sensors** IEEE Sensors Letters. DOI 10.1109/LSENS.2021.3049706
- **Bin Tariq, O.**, Lazarescu, M. T., and Lavagno, L. (2020). **Neural Networks for Indoor Human Activity Reconstructions** IEEE Sensors Journal. DOI 10.1109/JSEN.2020.3006009
- **Bin Tariq, O.**, Lazarescu, M. T., and Lavagno, L. (2019, September). **Neural network-based indoor tag-less localization using capacitive sensors.** In Proceedings of the 2019 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2019 ACM International Symposium on Wearable Computers (pp. 9–12). ACM.
- **Bin Tariq, O.**, Lazarescu, M. T., Iqbal, J., and Lavagno, L. (2017). **Performance of machine learning classifiers for indoor person localization with capacitive sensors.** IEEE Access, 5, 12913–12926.

# Chapter 1

# Introduction

Low-cost, low-maintenance, accurate indoor detection and localization of persons is an important enabler for several applications, such as health care and resource usage optimization and security. For instance, room occupancy information, alongside other factors, can help reducing energy consumption by controlling the ambient temperature, lighting and water consumption [2]. For health care, by 2050, the number of elderly persons is estimated to be nearly 2.1 billion worldwide, more than doubled from 2015. Assisted living systems can play an increasingly important role in improving their quality of life, since the ratio between working-age persons and elderly is expected to drop to 3.5 by 2050 [3]. Also, monitoring human activity for extended periods can detect behavioral changes (e.g., gait changes), which can help recognizing the early onset of diseases like Parkinson disease [4]. Moreover, presence monitoring systems can also be used to detect unauthorized intrusions (e.g., through house windows, a sign of burglary attempt [5]).

Indoor person localization can rely on wearable or portable devices [6], such as the IEEE 802.11 (Wi-Fi) [7, 8] or the Bluetooth [9] standards, low power communications using the ZigBee protocol [10], radio frequency identification (RFID) [11], ultra wideband radio (UWB) [12], visible light communication [13], or audible [14] and ultrasound [15] acoustic signals.

Many localization techniques have been proposed over the years [16], such as pressure and load cells [17], sensing mats [18], thermal infrared [19], sound source [20], ultrasound reflections [21, 22], air pressure [23], residential power lines [24], water usage [25], optical [26], carbon dioxide [27], vital functions [28], and data fusion from various sensor types [29]. Many of these methods typically require the person to wear an active tag or require significant infrastructure changes. The tag can often be an important reliability and usability drawback, because the person may forget or be reluctant to wear it [2], leading to missing or incomplete traces. Tag-less indoor localization [2, 30] is necessary whenever the persons may not carry or wear a device to be sensed by the localization system, such as in some smart home applications or assisted living for elderly people. Furthermore, to improve the

localization system acceptance and added value, it should ensure personal privacy, be affordable, unobtrusive, easy to install, and require little or no maintenance (e.g., long battery life or use wireless power).

Video or imaging cameras can be used for human presence detection and localization [31, 32]. However, cameras often require high computational, networking and energy resources, a direct line of sight, and adequate lightning, which increase the installation complexity and system cost. Cameras also raise significant privacy concerns, since the residents are often rejecting constant video monitoring even with blurred images [2]. Other solutions are based on Ultra-Wide Band (UWB) radios or on the measurement of received signal strength variations on narrow channels [33–36]. They typically require the person to wear an active tag or the installation of many mains-powered sensors. Ultrasonic systems have also been used for indoor person localization [37]. However, they also require the user to carry a tag and long-term exposure to ultrasonic noise can cause harmful health effects [38]. Wi-Fi-based systems have been studied for indoor localization [39–41]. They rely on the presence, by now common, of many Wi-Fi-enabled devices in the monitored area to calculate the Time of Arrival (TOA), Angle of Arrival (AOA) and Received Signal Strength (RSS). However, for an adequate accuracy these systems require a large number of Wi-Fi-enabled devices, which have high power consumption. Another limitation is signal attenuation by walls and furniture [41].

Other systems attach tags to the objects that are routinely used by the person, such as the pill box, fridge door or house keys, to monitor when the person uses these items [42]. However, if the person does not interact with the monitored objects, the system will fail to provide any information.

Systems based on passive infrared sensors (PIR) can also be used for tagless localization [43–46]. For effective localization, these solutions require a large number of sensors which increase the installation cost and reduce the user acceptance, because they visually remind them that they are being monitored [2]. Thermopiles are based on several thermocouples joined in series and can measure temperature of different sections of its field of view. A major advantage over PIR sensors is the ability of thermopiles to measure temperature of stationary persons or objects. Thus they can be good candidate for device less indoor localization. However, IR based sensors can give false readings if they are exposed to common infrared (IR) sources, such as sunlight [47], good heat conductors, IR radiation reflectors, incandescence light bulbs [2].

Electric field sensors [48] used for localization include capacitive tiles [5], electric resonance coupling [49], and capacitive coupling [50]. They use the conductive properties of the human body and do not require the person to carry any specific device. Capacitive sensing has been used for human detection, localization and identification [51, 52]. Capacitive coupling has various uses, from musical instruments (Theremin) to precision instruments (e.g., to measure the mechanical vibration of motors and generators) and for user interaction with the touch screens of mobile

phones. The measurements are passive and are not affected by materials with relative permittivity close to that of the air, hence they can operate behind objects made of such materials [47].

Capacitive sensors have various modes of operation. Zimmerman et al. [53] described the human shunt mode and the human transmitter mode. In the shunt mode a voltage is created between two electrodes, where one electrode is an oscillator and the other is a virtual ground. When a hand or any other body comes between the field created between the electrodes, the amount of displacement current reaching the virtual node decreases. In the transmitter mode the human body acts as an electric field emitter when a low frequency energy is coupled into human body. But in this case human body needs to be touching or very close to the oscillating electrode. The displacement current increases in the ground (receiving) electrode when the person moves closer to it. Smith et al. [47] described another mode, named Loading mode, which uses only one plate. In loading mode the current displaced by human body from the transmitter plate is measured. The distance of human body from the transmitter can affect the displaced current, which can be a measure of the distance between transmitter electrode and human body.

Load-mode single-plate capacitive sensors can be self-contained, easy to install, inconspicuous, inexpensive, and do not raise significant privacy concerns. They can sense [54, 55], identify [56, 57], and localize [51, 58] persons indoor, but their sensitivity decreases steeply with the distance. Long range sensing, at distances of 10 to 15 times the plate diagonal, are highly susceptible to several environmental factors, such as electromagnetic and electrostatic noise, humidity, or temperature [59]. Hence, the whole sensor data processing chain is very important to achieve good localization accuracy and stability in variable environmental conditions.

As mentioned earlier, several types of noise can adversely affect sensor data accuracy, from offsets due to changes of indoor objects (e.g., presence, position) or changing environmental conditions (e.g., temperature, humidity, lighting), to noise induced by environmental electromagnetic radiations (e.g., radio, light switches, home appliances). Hence, raw sensor data very often require significant post-processing in order to achieve the localization accuracy needed by applications. Among data processing techniques, the machine learning (ML) algorithms are among the most promising, but their performance (e.g., inference performance, required training, computation complexity) can vary significantly.

Machine Learning (ML) is the branch of Artificial Intelligence (AI) to develop learning techniques [60] to make machines learn and improve to carry out a task. In recent years ML has grown exponentially and is being used in vast number of fields such as image processing, sensor data processing, medicine and many more. Covering the theory of Machine Learning is beyond the scope of this thesis, but a brief overview is presented for the topics that were involved in this work. We will discuss in detail the structure of the algorithms and their interaction with data in the following chapters. Machine Learning can be of various types e.g. *Supervised*

*Learning*, *Unsupervised Learning* or *Reinforcement Learning*.

Supervised learning involves learning a function that maps an input to an output based on example pairs that are provided as the training data. Each pair comprises an input and a desired output (label). [61]. The example pairs, also referred to as labeled training data set, are fed to the learning algorithm. Training data set is analyzed to infer a function that can be used for mapping new data previously unseen in training data. If the function is learnt well it should be able to assign unseen data to the expected output labels. The supervised can be used for classification, where the number of classes is predefined and task of the learnt function is to map the input to one of the classes. Supervised learning can also be used in regression analysis, where contrary to the classification, the output variable is a continuous output variable. A typical example of regression analysis is predicting house price where the predicted output can take any value e.g. between 100000 € to 200000 €.

In unsupervised learning, contrary to supervised learning, example inputs are not labeled. The learning algorithms look for intrinsic structure or some common attributes of the input data. For example, clustering is type of unsupervised learning. Reinforcement learning algorithms are based on learning a policy (mapping from states to actions) that maximize the reward received over time [62]. It does not have labeled training data as is the case with supervised learning and unlike unsupervised learning, it is characterized by a reward measure which must be maximized with trial and error. Common applications of reinforcement learning include strategy planning, robotics for industrial automation etc.

For the purpose of indoor localization we considered supervised learning. We used both classification to classify among pre-defined locations and regression to extract the trajectory of a person's movement. For classification, we used most of the ML classifiers from Weka collection. Prominent among those were Random Forest, Support Vector Machine (SVM), k-NN and Naive Bayes. For extracting trajectory we used Neural Networks in various forms such as Feed-forward, 1D Convolutional Neural Networks (1D-CNN), Long Short Term Memory (LSTM).

Although algorithms and techniques described are for Capacitive sensing and Infrared sensors, we believe these techniques can be applied to other types for sensors as well, particularly those involving time series data.

# Chapter 2

# Human Indoor Localization and Tracking

Indoor localization is a combination of various steps. They include selecting the sensor technology, and processing data through various algorithms to obtain the localization information. In this chapter we give a brief overview on various popular localization methods, and machine learning algorithms.

## 2.1 Main Indoor Localization Techniques

A growing number of smart space applications rely on indoor person localization and activity recognition for safety monitoring, providing added value services, or continuous assistance. For instance, assisted living applications can lower assistance costs, improve safety and quality of life, which are increasingly important with a projected ratio between working-age and elderly people of 3.5 by 2050 [3].

The Global Navigation Satellite Systems (GNSSs) such as Global Positioning System (GPS) from United States and Global Navigation Satellite System (GLONASS) allow mobile devices to position themselves in outdoor environments [63]. However, satellite positioning systems perform poorly in indoor environments. Tracking GPS signals in indoor environments typically require a receiver capable of tracking signals with power levels ranging from -160 dBW to -200 dBW, while a typical mobile receiver has a noise floor of approximately -131 dBW in indoor environments [64]. Even if the receiver can track the signal from a sufficient number of satellites, the indoor multi-path propagation can significantly reduce the correct tracking ability. In outdoor environments, receivers experience multi-path effect only in benign form while in indoor environments the reflected signal can exceed the direct signal [64]. This makes it difficult to achieve the sub-meter accuracy that is usually required in indoor positioning systems. A wide range of indoor technologies have been developed

over the years, but the lack of the single *perfect* method means that indoor localization is still an active area of research. Indoor localization can be broadly classified into two main categories, device based localization (DBL) and device free localization (DFL). Device based localization needs the subject under observation to carry a device with him to get recognized by the localization system. Contrary to DBL, the device free localization methods can track the subject without requiring the subject to carry any device. Many techniques have been proposed on each of these methods involving various types of sensors and methods. Some of them are discussed below.

### 2.1.1 Wi-Fi

The IEEE 802.11 commonly know as Wi-Fi, works in Industrial, Scientific, and Medical (ISM) band and is commonly used these days in all mobile devices, e.g., cell phones and laptops. Because of the existing Wi-Fi infrastructure in many homes and offices, it is an attractive option for use in indoor localization systems. Liu *et al.* in [65], surveyed Wi-Fi based localization systems. They categorized the Wi-Fi based systems in active positioning (device based) and passive positioning (device-less) systems. Active positioning relies on users carrying specific devices while Passive Wi-Fi systems can work without requiring the user to carry any device. They can use fingerprinting methods which involve mapping of the area by storing the received signal strength (RSS) values and the corresponding location information in an offline phase. In the online phase the user location and trajectory are estimated by comparing the RSS value of user's device with earlier stored mapping information during offline phase. Many studies have *fused* Wi-Fi with other sensors commonly present in mobile phones to improve localization accuracy. Examples of such fusion include combining Wi-Fi based localization with camera, inertial sensors (accelerometers and gyroscopes), bluetooth or geomagnetism. Other Wi-Fi based techniques include *range-based* methods. Common range-based methods include, time of arrival (TOA), angle of arrival (AOA) and frequency difference of arrival (FDOA). While the device based localization with Wi-Fi has been widely researched over the years, the device free localization has been relatively less popular. Wi-Fi based DFL utilizes RSSI or channel state information (CSI) combined with trilateration or fingerprinting methods. However, for adequate accuracy, these systems require a large number of Wi-Fi-enabled devices, which have high power consumption. Another limitation is the signal attenuation by walls and furniture [41] and phenomena such as multi-path propagation, which can drastically affect the localization accuracy. Additionally radio mapping of the area under observation is a very time consuming process and it need to be re-done if an access point is added or even removed. Adib *et al.* [66] proposed device-less system WiTrack2.0 which operates in multipath-rich indoor environments and can track multiple persons. They use five transmit antennas and five receive antennas which are used for transmitting RF signals and capturing their reflections in a $5\,\mathrm{m} \times 7\,\mathrm{m}$ room. Although one of the reasons for interest in W-Fi

systems was the preexisting infrastructure, but as it can be noted that the number of access points required in most of the studies are higher than what is required for Wi-Fi communication, for example even in small home at least three access points would be required for positioning while a single access point is enough for Wi-Fi communication to work. Vasisht *et al.* [7] demonstrated a decimeter-level localization with a single Wi-Fi access point by measuring TOF between user's device and the access point. The median positioning error was 65 cm in line-of-sight and 98 cm non-line-of-sight.

### 2.1.2 Ultra-Wide Band (UWB)

Ultra-wide band uses ultra short-pulses with time period of <1 nanosecond, which are transmitted over a large bandwidth (>500 MHz), in the frequency range from 3.1 GHz to 10.6 GHz, using a very low duty cycle [67] resulting in lower power consumption. The UWB has been studied widely for indoor localization [12, 33]. Due to its significantly different signal type and radio spectrum it is immune to interference from other signals, which makes it an attractive technology for indoor localization. UWB signal, particularly of low frequency spectrum, can penetrate various materials, e.g., walls, equipment and clothing (metals and liquids can still cause interference in UWB signals) [6].

UWB can be based on time of flight (TOF), angle of arrival (AOA) or time difference of arrival (TDOA). As mentioned earlier UWB has short duration pulses which are easy to filter so that the correct signals can be separated from the signals generated from multi-path [68]. Short duration pulses also help to accurately determine the precise TOA, which in turn can provide more accurate localization. Moreover, the very short duration of the UWB pulses make them less sensitive to multipath effects, allowing the identification of the main path in the presence of multipath signals and providing accurate estimation of the ToF. Commercially available systems using UWB claim localization accuracy of up to 10 cm [69].

### 2.1.3 Acoustic Localization

Acoustic localization utilizes the microphones in the mobile phones or other devices to capture acoustic signal emitted by reference nodes acting as the sources of sound [6]. Liu *et al.* in [70] surveyed various research methods employed for acoustic localization. Acoustic localization share many characteristics of radio frequency based localization, hence many techniques that are used for RF based localization can be used for acoustic localization. These techniques include time of flight (ToF), time difference of arrival (TDOA), Doppler effect and phase shift, etc. Yet, the acoustic signals have the advantage that their speed is much lower than the speed of RF signals, thus providing potentially higher accuracy [71].

Acoustic-based localization can be achieved by transmitting modulated acoustic signals, containing time stamps or other time related information, which are captured

by microphone sensors to estimate ToF [72]. Other methods involve small phase and frequency shift experienced by the acoustic signal due to the Doppler effect caused by a moving object, which can used to estimate its relative position and velocity [73]. Acoustic based systems have been shown to attain high localization accuracy, but can be limited by the sampling rate/anti-aliasing filter of commonly used microphones, made for audible band acoustic signals (<20 kHz). The transmission power should be low, so that the acoustic signals should not be perceived by the human ear and advanced signal processing algorithms are needed to detect the low power signal [6]. Furthermore, the need of infrastructure such as acoustic sources/reference nodes and the high sampling rate which causes battery drain, make the acoustic signal-based localization not a very popular choice.

### 2.1.4 Visible Light

Visible light communication is high-speed data transfer technology that uses visible light between 400 THz to 800 THz, primarily using light emitting diodes (LEDs) [74]. Localization with visible light is achieved using light sensors (receiver) to estimate the position and direction of the LED emitters which transmit the signal. The angle of arrival (AOA) has been accurately used for localization using visible light [74, 75]. A critical limitation of the visible light-based localization is the requirement of direct line of sight between the LED and the sensors (receivers) for accurate localization [6].

Visible light was used for localization using RFID (which have photo-diodes) sensors placed on the floor by detecting the decrease in the light intensity because of a person [76]. By using the parameters such as the height of the LEDs, the radius of the light zone and height of individuals, localization accuracy of 50 cm was achieved. Hu *et al.* [26] designed proprietary system with changes in existing LED which must be attached to the user's device. They used the RSSI from the LEDs, to estimate person's location with average accuracy of 30 cm. Similarly, visible light from smart LEDs was used for localization where user must carry a device embedded with custom light sensors which receive the energy transmitted by LEDs [77]. The system used frequency higher than 200 Hz to avoid flicker to human eye. However as is common with visible light based systems that the system needs line of sight, moreover at least three LEDs are required to estimate user's location. Zhang *et al.* [78] use fluorescent lights as the reference nodes and user's device camera as sampling device for image processing algorithms. They use difference in fluorescent light's characteristic frequency (>80 kHz) to determine user's proximity to certain reference node. TDOA was used for LED based localization by Jung *et al.* [79], where line of sight is required between transmitter and receiver. They also used the difference of frequency among LEDs to differentiate among transmitting LEDs.

### 2.1.5 Ultrasound

The ultrasound based localization uses ultrasound signals (>20 kHz) to calculate the distance between a transmitter and a receiver node using ToF and sound velocity. It has shown cm level accuracy as claimed by commercially available Marvelmind [80]. RF pulse can be added to ultrasound system to provide synchronization. The sound velocity can change with changes in humidity and temperature; to compensate these changes temperature sensor is usually added to the ultrasound sensor systems [81]. Even though signal processing techniques can filter environmental noise, still a consistent noise source may still reduce the localization performance [6]. Kalman filters have been widely used for robot localization with ultrasound sensing, however the data association step of these methods is very complex and usually requires linearization [6]. The BAT indoor localization is based on ultrasonic signals for indoor localization [82, 83]. It uses lower speed of the sounds waves in the air (330 m/s) which improves it accuracy. The devices are fitted with the transmitter in order to be tracked, and fixed receivers with known positions are used for localization. A combination of RF and ultrasonic signals was used for indoor localization in the Cricket indoor localization system [84]. However, localization of ultrasound systems is very sensitive to the placement of sensors. Moreover, long-term exposure to ultrasonic noise can cause harmful health effects [38].

### 2.1.6 Bluetooth Low Energy (BLE)

BLE is a Bluetooth standard optimized for low-energy consumption. BLE architecture consists of the Physical layer (PHY), Link Layer (LL), Logical Link Control and Adaptation Protocol (L2CAP), and Generic Attribute Protocol (GATT), and Generic Access Profile (GAP). Lower layers PHY and LL, are for bit transceiving and providing medium access, connection establishment, error, and flow control, respectively. L2GAP is for multiplexing and providing fragmentation and reassembly of large data packets.GATT and GAP are the top layers of BLE. Like Bluetooth, BLE uses adaptive frequency hopping spread spectrum to access the shared channel, but in contrast to classic Bluetooth which has 79 hops and 1 MHz channel width, BLE has 43 hops and the channel width is 2 MHz [85]. A BLE device can operate either as a master or slave. BLE network topology is a star, where a master can operate multiple simultaneous connections with several slave devices, but a slave device can only have a single master.

The low energy feature of BLE makes it an attractive option for portable and battery-operated devices. However, just like other device-based localization techniques, BLE localizes other (BLE) devices, not persons directly. BLE-based protocols have been proposed by Apple, *iBeacons*, and Google, *Eddystone*. *iBeacons* is based on RSSI and was primarily designed for proximity-based services. However, it has also been used for localization with accuracy up to 0.95 m [6]. Ayyalasomayajula

*et al.* [86] proposed a CSI-based localization system using BLE-compatible algorithms with an accuracy of 86 cm. Islam *et al.* [87] proposed a multipath profiling algorithm to track any BLE tag in an indoor environment. They used the RSSI information at different hopping channels to obtain the ToF of the direct path signal from an IoT device (peripheral) to an access point (central), with a ranging accuracy of 2.4 m.

### 2.1.7 Electric Field Sensing

Electric field sensors [48] have been used for localization. Localization methods with electric field sensing include capacitive tiles [5], residential power lines [24], electric resonance coupling [49], and capacitive coupling [50]. They use the conductive properties of the human body and do not require the person to carry any specific device. In Section 2.3 we will discuss capacitive sensing in detail.

### 2.1.8 Other Methods

Various other have been used for localizing indoors, for example attaching tags to the objects that are routinely used by the person, such as the pill box, fridge door or house keys, to monitor when the person uses these items [42]. However, if the person does not interact with the monitored objects, the system will fail to provide any information. Other methods include pressure and load cells [17], sensing mats [18], air pressure [23], water usage [25], optical [26], carbon dioxide [27], vital functions [28], the Bluetooth [9] standards, low power communications using the ZigBee protocol [10], and radio frequency identification (RFID) [11].

One major factor to look for indoor positioning is the fact that person inside the home may not be carrying a device, e.g., mobile phone, or may not feel comfortable wearing a device all the time. Similarly, a person can simply forget to wear a device [88]. Thus, tagless indoor localization is necessary whenever the persons may not carry or wear a device to be sensed by the localization system, such as in some smart home applications or assisted living for elderly people. Furthermore, to improve the localization system acceptance and the added value, the localization system should ensure personal privacy, be affordable, unobtrusive, easy to install, and require little or no maintenance (e.g., long battery life or use wireless power). Table 2.1 gives a non-exhaustive summary of indoor localization systems.

## 2.2 Machine Learning Techniques for Indoor Localization

Sensor data can be further filtered and processed by the classification algorithms, which ultimately output the approximate location of the person. Before being ready for localization, ML classifiers need to be trained with sensor data sets labelled

Table 2.1: A summary of indoor localization systems.

| Technique | Study | Technique | Accuracy (cm) | User device | Comments |
|---|---|---|---|---|---|
| Wi-Fi | RADAR [89] | RSSI | 294 | ✓ | Requires building radio maps. Accuracy degrades as radio environment changes. |
| | [90] | RSSI | 240 | ✓ | |
| | Horus [91] | RSSI | 39 | ✓ | |
| | Ubicarse [92] | AOA | 39 | ✓ | User device must have two antennas, requires user to rotate device |
| | ArrayTrack [93] | AOA | 23 | ✓ | Requires large number of antennas in APs |
| | Phaser [94] | AOA | 100–200 | ✓ | |
| | WiTrack2.0 [66] | TOF | 11.7 | ✗ | Requires large number of antennas to created multipath-rich environment |
| UWB | [12] | ToA | 19–54 | ✓ | Errors can increase w.r.t. deployment conditions e.g., [95] observes the 0.49 m, 0.71 m, 1.93 m, 1.10 m with DecaWave, BeSpoon, Ubisense, and Ubisense(with AOA), respectively. |
| | Ubisense [96] | AOA/ TDOA | 15 | ✓ | |
| | BeSpoon [97] | N/A | 10 | ✓ | |
| | DecaWave [98] | TOF | 10 | ✓ | |
| Acoustic Localization | Guoguo [99] | TOF | 6–26 | ✓ | Sound pollution can have adverse effects on localization. limitation caused by sampling rate/anti-aliasing filter of commonly used microphones |
| | Beep [100] | TOF | 90 (95 %) | ✓ | |
| | WalkieLokie [101] | N/A | 63 | ✓ | |
| Visible Light | LocaLight [76] | N/A | 50 | ✓ | Needs line of sight. Errors can increase w.r.t. deployment conditions |
| | LiTell [78] | N/A | N/A | ✓ | |
| | LiTell2 [102] | N/A | N/A | ✓ | |
| | Pharos [26] | RSS | 30 | ✓ | |
| | [79] | TDOA | 0.18 | ✓ | |
| Ultrasound | BAT [83] | TOF | 4 | ✓ | Susceptible to reflections and ultrasonic noise. |
| | Cricket [84] | TOF | 10 | ✓ | |
| | Marvelmind [80] | N/A | 2 | ✓ | |
| BLE | [103] | RSSI | 97 | ✓ | Low accuracy. |
| | [86] | CSI | 86 | ✓ | |
| | [87] | RSSI/ TOF | 240 | ✓ | |
| Thermopile | [104] | IR | N/A | ✗ | Only location classification |
| | [105] | IR | N/A | ✗ | Only people count |
| | [106] | IR | 17.5 | ✗ | 8 thermopiles covering 4.9 m × 6.2 m |
| | [107] | IR/AOA | 13.39 | ✗ | Two 16 pixel × 4 pixel thermopiles covering 2.35 m × 3 m |
| Capacitive sensing | [108] | Transmit | 10 | ✗ | Extensive changes are required to the floor of the area under observation. |
| | [109] | Loading | 2.2 | ✗ | |

with the position of the person. After training, the ML classifiers can be used for localization, in which they receive new data sets for which they return the approximate location of the person based on the internal model built during training.

Previous studies evaluated various algorithms for different classifications and using different types of sensors, e.g., GPS and accelerometers. Shafique and Hato in [110] review various studies on using sensor data for training and testing ML classifiers. One such study uses GPS coordinates, speed, heading change, and acceleration among others, and tests these features on five different classification algorithms (Bayesian net, decision tree, random forest, naive Bayesian and multilayer perceptron) [111]. The test results show that random forest outperforms other algorithms. Wenjie *et al.* [36] used k-NN, multivariate Gaussian mixture model

(GMM) and support vector machine (SVM) for classification and for tracking moving person used two hidden Markov model (HMM) based methods, namely GMM-based HMM and k-NN- based HMM for device free localization using RFID based system.

K-nearest neighbors is an instance-based (lazy) learning method that uses a similarity metric between the test and training samples (e.g., Euclidean distance). As most lazy learning algorithms, k-NN assumes little or no knowledge on data distribution because it does not create a generalization of the training data [112]. Hence, the modeling time is reduced, but the algorithm needs to keep all training samples in memory during classification, which can be fairly memory- and processing-expensive for low-resource embedded devices or for large training sets.

Support vector clustering (SVC) [113] is a clustering method based on support vector machines (SVM). SVC maps data points to a high dimensional feature space using Gaussian kernels, where the algorithm searches for the minimal enclosing sphere. This sphere is mapped back to data space, where it forms the contours that contain clusters of data points.

Random forest was proposed by Breiman as one of the ensemble methods [114]. Its internal model is generated by training multiple trees separately with the same distribution and choosing randomly the data samples to ensure that the decision trees are not correlated [115]. The classification is done by a majority vote among the decisions of all trees. The algorithm is robust to noise and outliers, and can work with nonlinear associations in a wide range of application domains, such as environment, ecology, bioinformatics, remote sensing and in physical time-activity classification [115, 116].

The performance of the classifiers can be improved by boosting techniques, such as a majority vote among similar classifiers or a weighted majority vote (AdaBoost) [117]. AdaBoost classifies well new sets, but its performance can degrade for noisy data due to the exponential change of its loss function [118]. LogitBoost uses a logarithmic loss function that changes linearly with the classification error and reduces the algorithm sensitivity to data noise and outliers [117].

Recently the artificial neural network (ANN) have made huge progress in many fields for example image processing, voice recognition, autonomous driving, just to name few. One of the features of ANNs is its ability to extract deep features from data contrary to conventional ML approaches which use hand crafted features extracted on basis of human knowledge [119].

ANNs can be many flavours, e.g., multi-layer perceptron, convolutional (CNN), autoregressive which are feed-forward NNs and long-short term memory (LSTM) which is recurrent NN. A detailed explanation of these varieties is beyond the scope of this thesis, however in Chapter 5 we give an overview of each type along with various parameters that were tuned and the interaction of these NNs with the input data.

Feed-forward NN in its simplest form can contain a neuron in a single hidden layer. An artificial neuron is made of a linear function and an activation function (non-linear), for example rectified linear unit (ReLU) [120], sigmoid [121], etc. For

designing the NN for a specific application various parameters must be tuned, this includes the number of hidden layers, number of neurons in hidden layers, type of activation, type of optimizer, learning rate, optimal number of epochs required to train the NN (more epochs can cause the NN to over-fit and less epochs can leave it under-fit), and the way data is fed to the NN.

Convolutional neural networks (CNN) are based on three main ideas, sparse interactions, parameter sharing, and equivalent representations [122]. CNN have gained huge popularity in image processing, speech and text analysis, thanks to their ability to extract useful features from the input data. While 2D CNNs are popular for image processing, the sensors for human activity produce output as time series, e.g., accelerometer which has multi-dimensional temporal output, thermopile infra red sensor which produce low resolution thermal image which can be treated as a time series as we will see in Chapter 5. On time series, CNN provides two advantages: local dependency, which means neighbouring signals in human activity related data are likely correlated, and scale invariance, which means invariance to changes in scale or pace [119]. CNNs are usually composed of convolution layers, followed by a pooling layer and fully connected layers. Various hyper-parameters must be tuned to optimize the CNN, e.g., the kernel size and the number of filters in convolution layers, the optimizer type, the activation function, the type of pooling layer, the number of fully connected layers and their number of neurons, and the size of input window to the CNN, as we will explain in Chapter 5, section 5.1.4.

The LSTM introduced by Sepp *et al.* [123] is a type of recurrent neural networks (RNN) widely used to extract features from data sequences, such as speech or handwriting. A typical LSTM cell, like a RNN cell, has an internal state that has context information which it passes on to the next time step, so at each time step the output is not only dependent on the current input, but also on the context. However, LSTM differs from RNN because it resolves the problem of vanishing and exploding gradients. Even though the mostly commonly used LSTM architecture (vanilla LSTM) works reasonably well on various datasets [124], there are still crucial parameters which need to be tuned, such as the size of input data window, the size of the LSTM cell and the number of stacked LSTM cells.

1D convolutional NNs (CNNs), long-short term memory NNs and their variants were used in multiple sensing applications. Wang *et al.* [119] survey the use of NN architectures for human activity recognition, including 1D CNNs, recurrent neural networks, and hybrid architectures using data from various sensors to classify human activities. LSTM networks were used for indoor static localization using the magnetic and light sensors that are included in the modern smartphones [125]. They do so in close proximity to the site where they trained the system (a $6\,\mathrm{m} \times 12\,\mathrm{m}$ lab) to preserve the validity of the magnetic field calibration. Chen *et al.* [126] used Wi-Fi fingerprinting for LSTM-assisted indoor discrete localization of multiple persons in a research lab of $35.3\,\mathrm{m} \times 16.0\,\mathrm{m}$ and in an office of $55\,\mathrm{m} \times 50\,\mathrm{m}$.

## 2.3 Capacitive Sensing for Indoor Localization

Capacitive sensing has been used for human detection, localization and identification [51, 52]. Capacitive coupling has various uses, from musical instruments (Theremin) to precision instruments (e.g., to measure the mechanical vibration of motors and generators) and for user interaction with the touch screens of mobile phones. The measurements are passive and are not affected by materials with relative permittivity close to that of the air, hence they can operate behind objects made of such materials [47]. While short-range capacitive sensors have been extensively researched, long-distance capacitive sensors are a relatively new research area. An extensive overview of more than 193 capacitive sensing techniques categorised by application domain includes indoor localization along, e.g., touch, gesture, grip and grasp recognition [55]. Often, indoor localization requires sensor installation in the floor [108, 127–131] or costly changes of the monitored area which are impractical for home use [132–134], on in mats [5]. The latter determines the person location relative to mat position and the monitored area can be extended by deploying more sensor-fitted mats.

Capacitive sensors for localization may also be installed in predefined places, for example near light switches, the study table [135–137], or to detect the presence of the driver in a vehicle [138]. The sensors can be used for close proximity interaction with computers, e.g., gesture recognition and interaction with computer games from short distances [139, 140]. Similarly, capacitive sensors were used for gesture recognition to prevent a patient from falling off a chair [141] or installed in a bed to detect sleep patterns [142]. Haescher *et al.* [143] use capacitive sensors to classify different modes of walking (fast, jogging and walking while carrying weight). In another study, the authors use capacitive sensors to classify various postures of the user [144]. In all these studies the sensor range is too short for the purpose of indoor localization.

Human activity can be detected using capacitive sensors from behind a piece of furniture, without a direct line of sight [59], or to detect variations in environmental fields, e.g., those generated by power lines [145, 146]. Prance *et al.* [146] used the 50 Hz field generated by the power supply lines to localize a human subject in an area of 3.52 m × 3.52 m by correlating the sensor outputs with an accurate camera-based localization, but without machine learning techniques. Spread spectrum capacitive sensors for human detection up to 1 m were also proposed [147], but their suitability for localization is not clear.

Environmental electromagnetic noise and surrounding objects with permittivity different than air may interfere with capacitive sensor measurements. To mitigate these effects, the sensor plate can be guarded by auxiliary fields to reduce the unwanted couplings of the sensor plate with the surrounding objects [59, 140, 148], and by post-processing sensor data to improve the reliability of long-range measurements.

## 2.4 Infrared Sensing for Indoor Localization

Passive infrared sensors or thermocouples have been extensively investigated and are often used to detect indoor presence and localization. PIR sensors are sensitive to movement, while thermopiles can also detect stationary heat sources [149]. Most of the studies involving human indoor tracking with thermopile sensor have been done using using machine learning for classification tasks [104, 105, 150–152], mathematical modelling [106, 153, 154], and image processing techniques for low resolution camera [107, 155–157].

Classification tasks involve selecting between predefined activities or positions in case of localization, for example detecting bed-exit motion [150], estimating people count classifying direction of motion [105], for activity recognition of people within 1 m [151], classifying location of a person with in a grid [104], and classifying gestures such as sitting down, moving forward, walking diagonally etc [152].

A single point thermopile with mechanical sub-masks was used to classify human presence among 16 locations with compressing sensing [158]. Eight thermopiles were used in an area of 4.9 m × 6.2 m to achieve mean error of 17.5 cm using probability hypothesis density filter [106]. Ng [153] used five thermopile sensors in an area of 4.6 m × 2.7 m with detection accuracy of ±50 cm using temperature-distance relationship. Similarly Zhang *et al.* [154] used two thermopile sensors and the voltage-distance relationship for an approximate equation set to estimate the trajectory, but did not mention the mean error.

Using thermopile array sensors as low resolution infrared cameras typically involve image processing, such as background removal and noise reduction [88, 159]. A 4 pixels × 4 pixels thermopile array was used to estimate trajectory using fuzzy logic in an area of 1.58 m × 1.58 m, with mean positioning error of 0.215 m [160]. Two 16 pixel × 4 pixel thermopile sensors were used to detect the angle of arrival with multi-frame averaging, background subtraction, and quadratic regression to localize a person in 60 cm-spaced positions on a snake-like trajectory, with a mean error of 13.39 cm [107]. Qu *et al.* in [156] used thermopile sensor “GridEye” of 8 pixels × 8 pixels in an area of 4 m × 4 m and interpolated it to 71 pixels × 71 pixels. The data was pre-processed by a Gaussian filter and an adaptive threshold, and the trajectory was obtained using a Kalman filter. They report an average error of 7 cm with a single human target walking on a horizontal line. For multiple human targets (up to 3), they show a few straight line trajectories but without reporting the error. Gu *et al.* [155] used a thermopile sensor of 24 pixels × 32 pixels to track a maximum of two persons on pre-defined trajectories. For single-human target, the mean RMSE was 9.5 cm. They used interpolation to enhance the thermal image from 24 pixels × 32 pixels to 93 pixels × 125 pixels, and various other techniques, for example background removal and weighted mean-shift method, to localize human targets. Shetty *et al.* [157] used a thermopile array sensor "GridEye" of 8 pixels × 8 pixels. The authors used a combination of interpolation (from

8 pixels × 8 pixels to 100 pixels × 100 pixels), background subtraction, Gaussian filtering, and iterative threshold algorithm to process the data and used a Kalman filter to localize a human target, but without reporting the localization error.

The existing works do not address continuous tracking of arbitrary person movements using very low resolution infrared images. We, on the other hand, take a different approach. We use the sensor output as a time series, without any filtering and enhancing, and evaluate it on an extensive arbitrary path instead of few straight paths, which resemble more closely the walking patterns of a person.

In Section 5.2 of Chapter 5, we explore the performance of various NNs for indoor tag-less person localization and movement tracking using a very low resolution 4 pixels × 4 pixels thermopile array, for assisted living applications. We explore the localization and movement tracking accuracy and smoothness for several types of neural networks. We search the networks and the configurations that perform best and use the least resources and computation, making them suitable for embedded processing on low power sensors.

# Chapter 3

# Sensors for Tag-less Indoor Human Sensing

This Chapter describes the working principles of both capacitive and thermopile sensors, data acquisition, and filtering methods. We collected the data from the capacitive sensors when the person was in fixed locations. We then collected the data when the person was walking randomly, from both the capacitive sensors and the thermopile sensors, using the same reference system (which requires the user to wear a tag) to provide the ground truth for training and testing.

## 3.1 Long-Range Capacitive Sensors

Electrical capacitance is defined as the electrical charge stored on a conductive object divided by the resulting change of its potential. The capacitance depends primarily on the geometry, distance, and dielectric properties of a system [140].

We use a capacitive sensor in load mode. In this mode, the sensor is connected to one plate of the capacitor, while the other plate is made of the environment and the person body, whose potential is considered constant for the purpose of the measurement. We indirectly measure the changes in the capacitance of the sensor by measuring the free running frequency of an astable multivibrator, which repeatedly charges and discharges the capacitor of the sensor.

A larger plate capacitive sensor has a higher sensitivity, but it typically collects more noise from the environment, which in turn limits the sensor sensitivity. For a given plate size, its capacitance depends on the distance $d$ between the plate and the person body and on the properties of the environment (geometry, permittivity, conductivity). We show that the effects of the environment on the localization can be reduced if the data from several sensors are used for training and testing of the ML classification algorithms, with minimal data filtering.

### 3.1.1 Working Principle

The capacitive sensors are based on an LM555 circuit configured as astable multivibrator and using the sensor plate capacitance (see Fig. 3.1). The oscillation frequency $f$ is inversely proportional to sensor plate capacitance $C$

$$f = \frac{k_c}{C} \tag{3.1}$$

through a constant $k_c$ determined by the resistor values in the multivibrator circuit. Plate capacitance cannot be determined analytically for distances $d$ between the plate and the human body that are much longer than the plate diagonal. Empirically, it changes with the inverse of the distance at a power $n$ ($n \approx 3$) through a constant $\alpha$, sensor area $A$, and with an offset $C_0$, depending on geometric, electric, and dielectric properties of the sensor and the environment [161, 162]

$$C \approx C_0 + \alpha \frac{A}{d^n}, \tag{3.2}$$

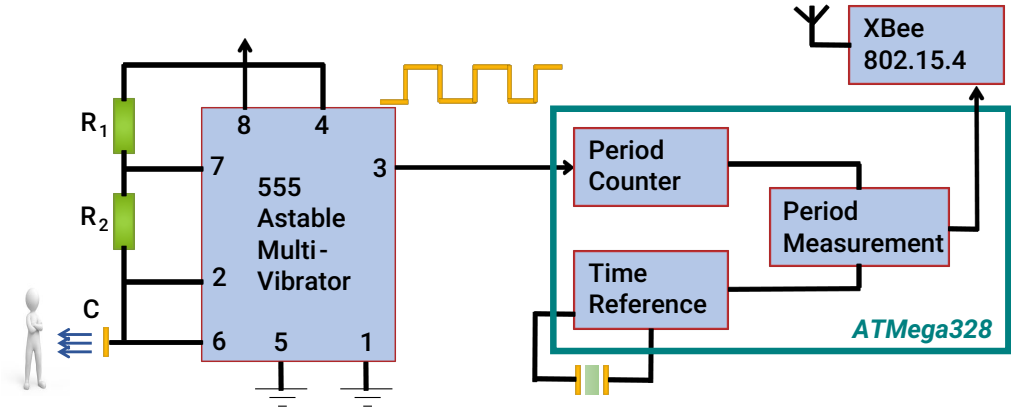



Figure 3.1: Schematic of capacitive sensor front-end using an astable multivibrator to convert the plate capacitance to frequency, measured by a microcontroller.

For the experiments using static positions, each sensor had an 8 cm × 8 cm copper-clad plate attached as a capacitor to a 555 integrated circuit in astable multivibrator configuration, for which the oscillation frequency is given by the formula:

$$\text{Frequency} = \frac{1}{0.7\,(R_1 + 2R_2)\,C}, \tag{3.3}$$

where $R_1 = 200\ k\Omega$ and $R_2 = 560\ k\Omega$. The size for the sensor plate was selected because from previous analysis it provides a good trade-off between the sensor size and its sensitivity [51]. The sampling rate for static position experiments was 1 Hz.

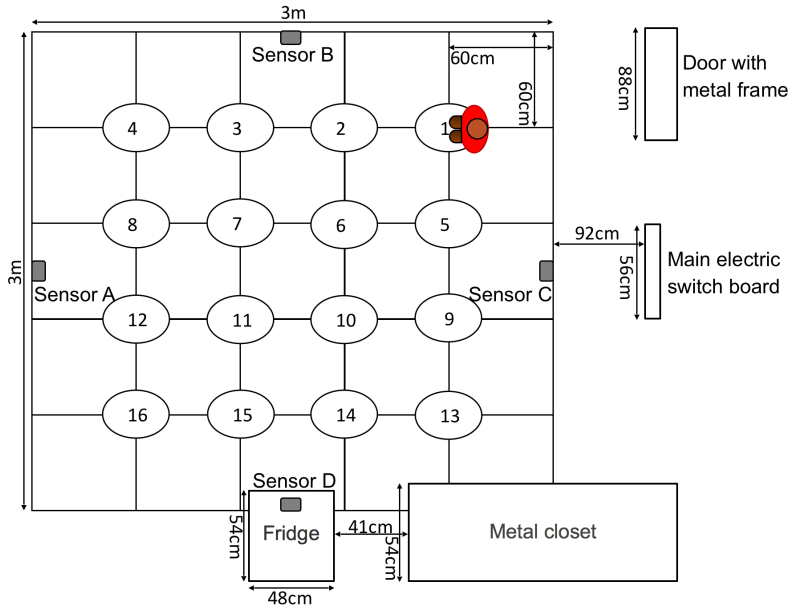



Figure 3.2: Organization of the experiment floor and body orientation for first localization experiment (experiment A). A fridge and metallic cabinet are partially included in the designated room space, while a metallic door and an electric switch board are close to the room space. They emulate the presence of metallic and electric objects in an apartment or house.

However for the free movement experiments, we used a 16 cm × 16 cm copper-clad plate attached to the 555 integrated circuit in astable multivibrator configuration. In addition to that, we improved both the sampling rate of our previous sensor from 1 Hz to 3 Hz, to adequately track a person moving indoors, as well as its discretization error, from 20 ppm to 3 ppm, further lowered to 1.5 ppm through oversampling, decimation and averaging (or about 15 aF while measuring a plate capacitance of roughly 10 pF). Yet, Equation (3.2) shows that sensor distance resolution changes much with the distance.

### 3.1.2 Static Position Experiments

We set up a realistic experiment in order to assess the performance of different ML classification algorithms for the localization of a person in an uncontrolled indoor environment. We designated an area of 3 m × 3 m as the "room" and we positioned four capacitive sensors (A, B, C, and D) at the center of each one of the four "walls" of the room, at a height of 115 cm from the floor, as shown in Fig. 3.2. By "uncontrolled" we mean that we did not prepare the room in any way for the experiment. For instance, we kept in place large metallic objects which may affect the plate capacitance and its sensitivity to the person presence, as well as sources

of electric noise, such as a fridge and an electric switch board on a side wall.

To gather a single set of experimental data, a person stood still for 8 s on each position, while each sensor acquired 8 samples with a sampling frequency of 1 Hz. We kept the sampling rate low to reduce the energy consumption, while still being able to track the daily movements of an elderly person moving with a speed of 1 km/h to 2 km/h indoor. We repeated this procedure for all the 16 positions in the room (shown in Fig. 3.2) to complete the experiment, thus each experiment provided 128 four-tuple samples.

Although the absolute frequency can vary significantly between experiments, the relative variations due to person proximity remain very similar among experiments, as we can see in Fig. 3.6.

We noticed that the base frequency of the sensors (i.e., without a person nearby) may change each time they are turned on. Thus, after gathering the data for each of the 16 positions within one experiment, we reset all the sensor nodes in order to make sure that we include also this type of noise in the experimental data.

We also noticed that the data are afflicted by very low frequency drifts and environmental conditions, hence we split the collection of the experimental data in three sessions. In Session A, we performed 20 experiments from which we obtained 2560 four-tuple samples (20 experiments $\times$ 8 samples per location $\times$ 16 locations). From now on we will refer to these data as *Set A*. After a few months, we used the same equipment to perform 10 additional experiments, in which we collected 1280 four-tuple samples (10 experiments $\times$ 8 samples per location $\times$ 16 locations). One week later, we collected another 10 experiments that added 1280 more four-tuple samples (10 experiments $\times$ 8 samples per location $\times$ 16 locations). Then we grouped the last 20 experiments (2560 samples) in a single set, *Set B*.

Moreover, the orientation of the body can also influence the sensor measurements, because for different rotation angles the distance from the closest body part to the sensor may change for a given position in the room. Thus, the 20 experiments in Set A were actually made of two sets: 10 experiments in which the person orientation was the one shown in Fig. 3.2 (i.e., with the chest towards sensor A), and the other 10 experiments in which we changed the orientation by 90°, as shown in Fig. 3.3 (i.e., with the chest towards sensor B). The latter orientation was kept also for all samples collected in Set B. We considered only two orientations during the experiments, namely either facing the sensors or exposing a shoulder to the sensors, because the human body is roughly symmetric and the capacitance difference between the front and back or between the left and right shoulder are similar.

The capacitive sensors change their base frequency over time even without a person in range, because of changes in the environmental conditions. These changes can significantly offset the acquired data, as shown in Fig. 3.4 and Fig. 3.5, where each plotted line represents a different experiment.

To compensate for these changes, we used the following method: we calculated the standard deviation of all the samples in a given set, then we calculated the

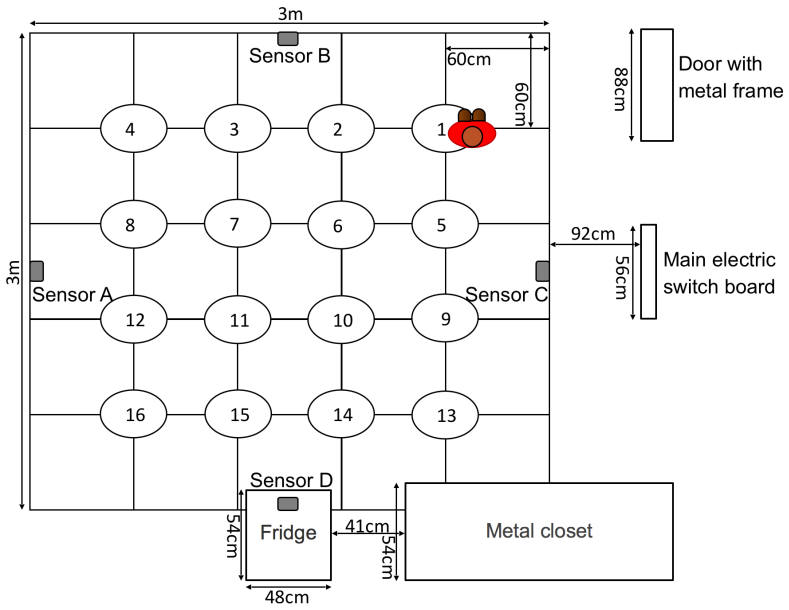



Figure 3.3: Organization of the floor and body orientation for the second localization experiment (B).

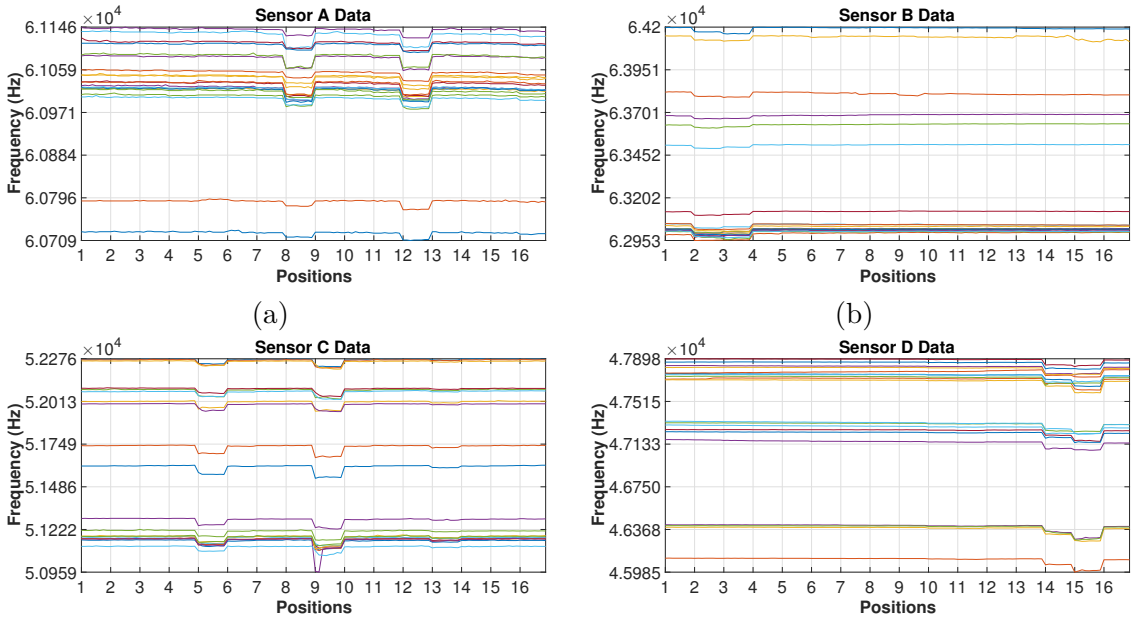



Figure 3.4: Raw data for Set A, sensor A, B, C, D in (a), (b), (c) and (d) respectively. Each color corresponds to one of the 20 data sets collected.

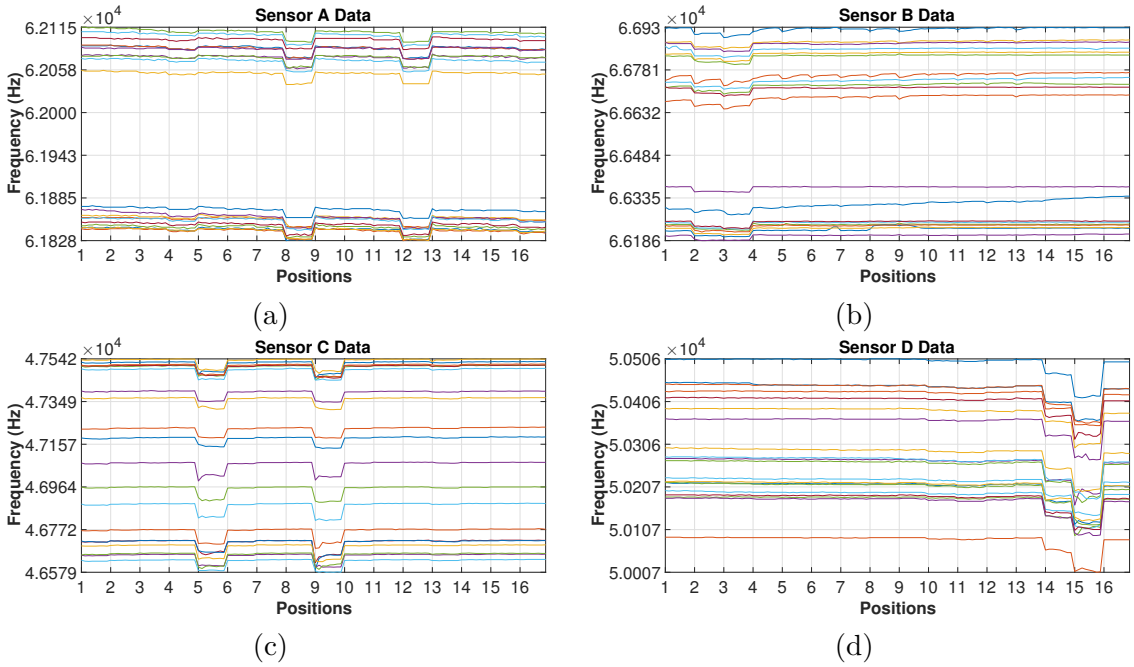



Figure 3.5: Raw data for Set B, sensor A, B, C, D in (a), (b), (c) and (d) respectively. Each color corresponds to one of the 20 data sets collected.

average only for the samples within the bounds of the standard deviation, and then we subtracted the average value for each set from all the samples in that set. We applied this procedure for all experiments and we see in Fig. 3.6 and Fig. 3.7 that the data sets are better aligned.

Then we have used these sets to test the performance of the ML classification algorithms for person localization. This is similar to using a median filter, as in [51], with a window of 128 s.

### Sampling Frequency and Sensor Plate Size

In our previous study [51], we evaluated three different plate sizes: 4 cm × 4 cm, 8 cm × 8 cm, and 16 cm × 16 cm. As mentioned earlier, we calculate the changes in capacitance by measuring the changes in the sensor interface oscillation frequency.

The relation between a parallel-plate capacitor area and the capacitance is

$$\mathrm{C} = \frac{\varepsilon_o k A}{d}, \tag{3.4}$$

where $C$ is the capacitance between the two plates (in Farad), $k$ is the relative dielectric permittivity of the material between plates ($k = 1$ in case of free space), $\varepsilon_o$ is the absolute dielectric permittivity of free space ($8.854 \times 10^{-12}$ F/m), $A$ is the effective area of capacitor plates (in $\mathrm{m}^2$), and $d$ is the distance between the capacitor

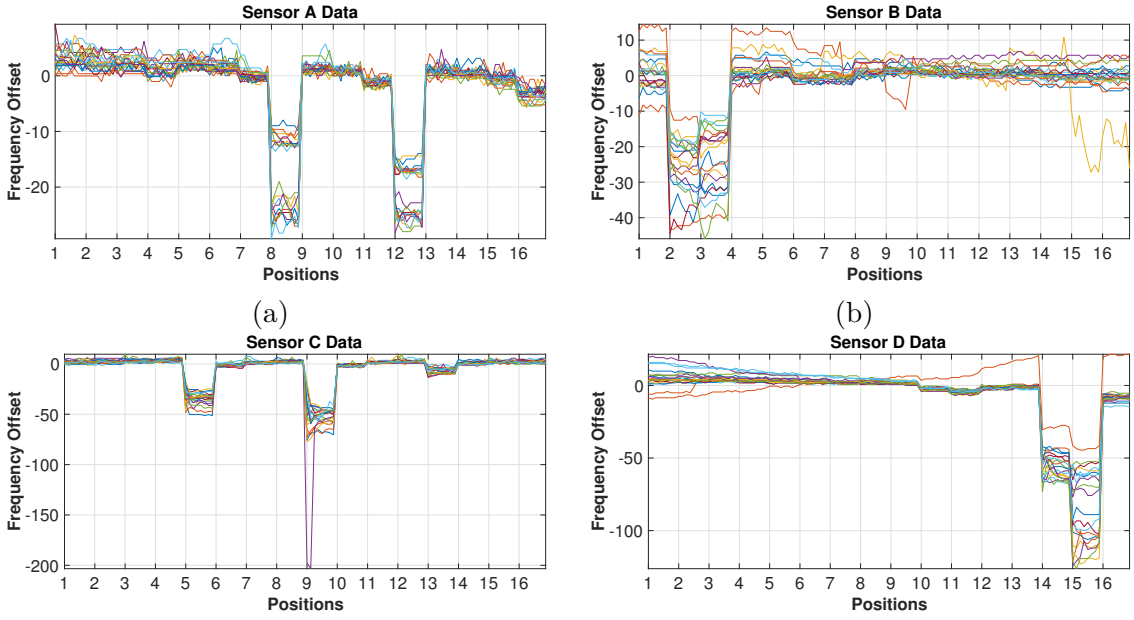



Figure 3.6: Offset-compensated data for Set A, sensor A, B, C, D in (a), (b), (c) and (d) respectively. Each colour corresponds to one of the data sets collected.

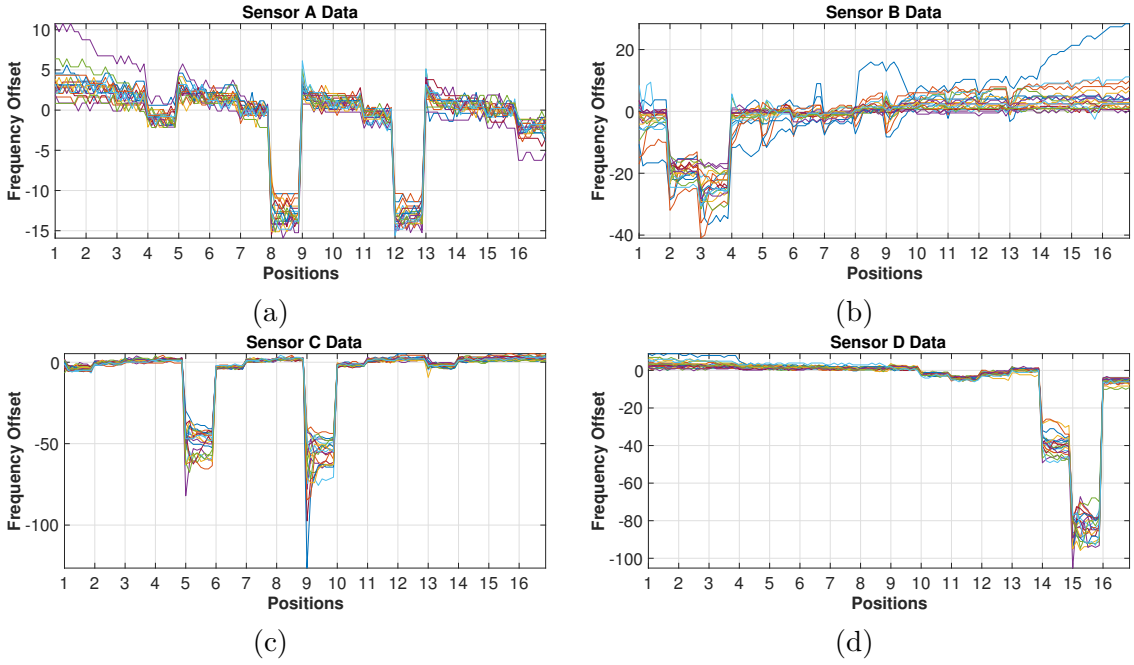



Figure 3.7: Offset-compensated data for Set B, sensor A, B, C, D in (a), (b), (c) and (d) respectively. Each colour corresponds to one of the data sets collected.

plates (in meters), assumed to be much smaller than the plate sizes. This assumption is not valid in our experiments, for which we determined empirically (3.2).

The effective area of the capacitor plates $A$ affects the sensitivity (to both the signal and the environmental noise). For a smaller plate size, we observe smaller changes in the output frequency, which are closer to the noise level. Thus we need to adopt a larger measurement period for smaller plates to attenuate noise and extend the sensing range.

A measurement period of 1 s is sufficient to monitor persons in static positions, but it is too long to properly capture the dynamics of a moving person. Larger plates improve the signal-to-noise ratio for the same measurement distance allowing to reduce the measurement period. This makes the larger plate size more suitable to observe the dynamic movements with higher sampling frequencies.

The optimal sampling frequency also depends on the application domain [163]. For example, if the target application involves observing the movements of limbs then the sampling frequency must be higher. This is because the movements of limbs tend to be faster than the movement of the whole body, e.g., hands and fingers move faster as compared to the displacement of the whole body.

The specter of indoor walking is below 1 Hz [164]. So for trajectory tracking experiments, we use a larger plate (16 cm × 16 cm) which allows us to increase the sampling frequency to 3 Hz, sufficient for tracking normal indoor walking [12].

### 3.1.3 Free Movement Experiments

We emulate in our laboratory a small (yet realistic with respect to a typical apartment room where an elderly person may live) room as an empty space of 3 m × 3 m. We monitor the position of the person within the room using two systems. The target system uses four capacitive sensors, each one with a sensing plate of 16 cm × 16 cm installed at chest level in the center of a "wall" of the virtual room (as shown in Fig. 3.8), providing readings three times per second. The reference system (from Marvelmind Robotics [80]) is based on four ultrasound anchors that can localize a mobile tag with ±2 cm accuracy at 15 Hz.

We characterize the localization accuracy of the ultrasound-based reference system in our environment by acquiring four times per second for five seconds the location of a person that wears the tag on the head, while standing on each one of 16 predefined locations inside the experimental room space (see Fig.3.9). The average localization error of the system is ±3.9 cm, with a maximum error of ±6.4 cm, and a maximum standard deviation (calculated over the norm) of ±0.7 cm. We note that the absolute localization error in our setting is higher than the ±2 cm reported by the producer, but with a good stability.

Fig. 3.10 shows the actual room setup with four capacitive sensors around the experimental space and a person walking with *tag* of ultrasound reference system. In the lower part of the figure, the arbitrary trajectory is shown, obtained after

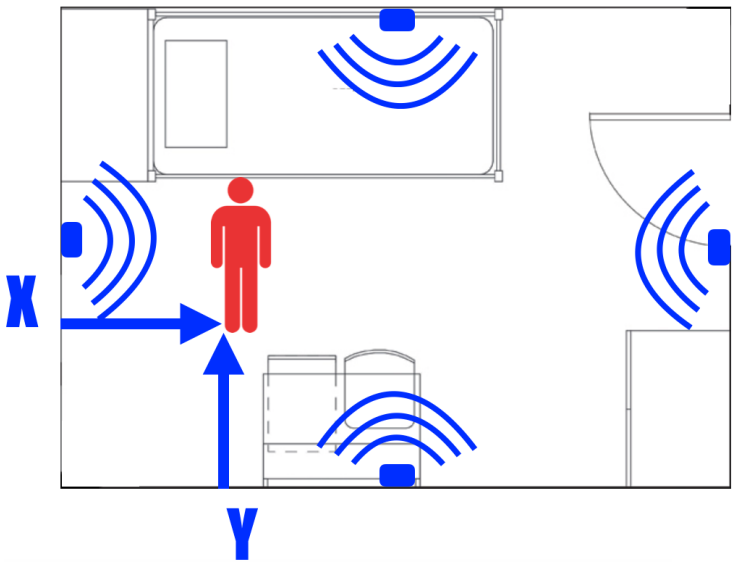



Figure 3.8: Four capacitive sensors centered on the walls of a 3 m × 3 m virtual room in the lab trace the position of a person moving in the space.

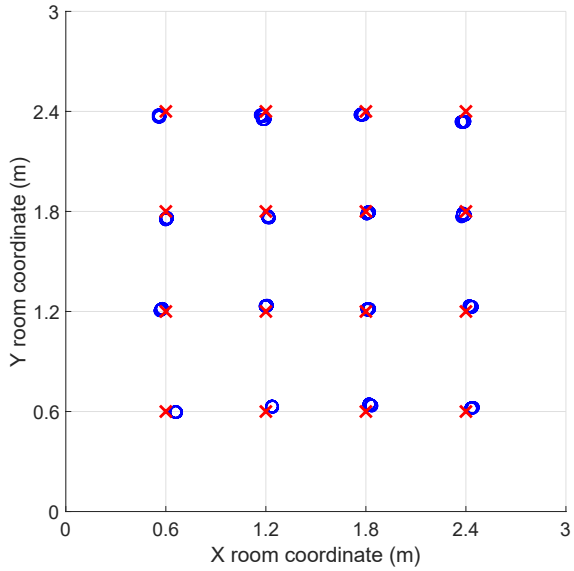



Figure 3.9: Static test of the reference system in 16 locations (red 'x'). The results shows very good stability (blue circles) and adequate accuracy.

walking in room for about nine minutes (1626 tuples at 3 tuples/s).

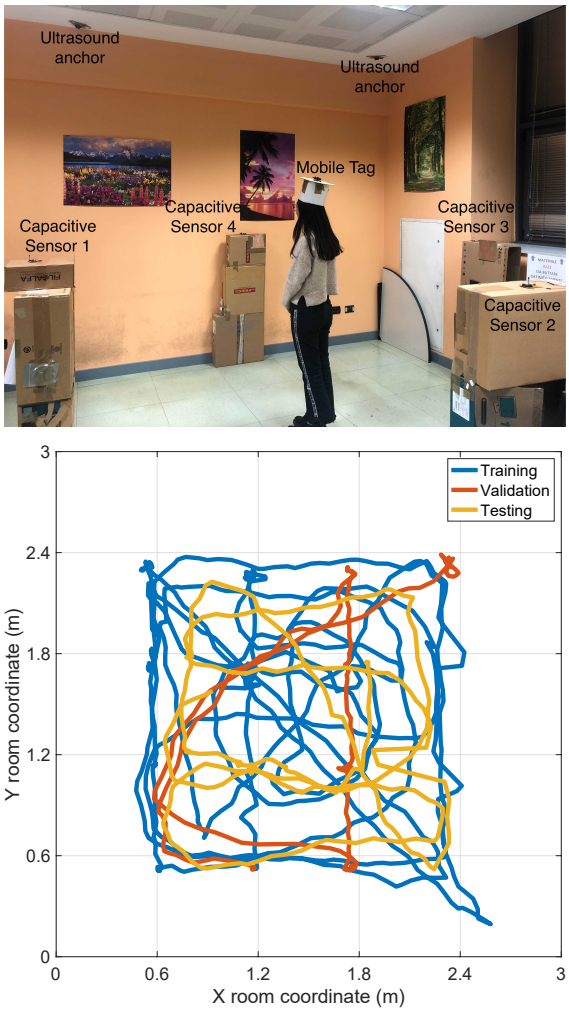



Figure 3.10: Virtual room used for movement tracking experiments and person trajectory (split into segments for NN training, validation and testing).

### 3.1.4 Sources of Errors and Filtering Techniques

However, this sensor front-end is sensitive to both high pitch and drift environmental noise, which limits its range and stability over time. In Fig. 3.11, we show an example of sensor data, where the high-pitch noise is mostly visible at long range, at the top of the plot, while the drift is mostly visible at the beginning, up to around the 200 sample mark, and towards the end, especially beyond the 1200 sample mark. Environmental noise typically reduces considerably the sensing range, e.g., in laboratory tests we were able to detect a person standing at a distance up to 1.6 m to 1.8 m in front of the sensor. Nevertheless, the higher noise susceptivity of the sensors allows us to better compare how efficiently different signal processing techniques can reject environmental noise. For instance, instead of using period

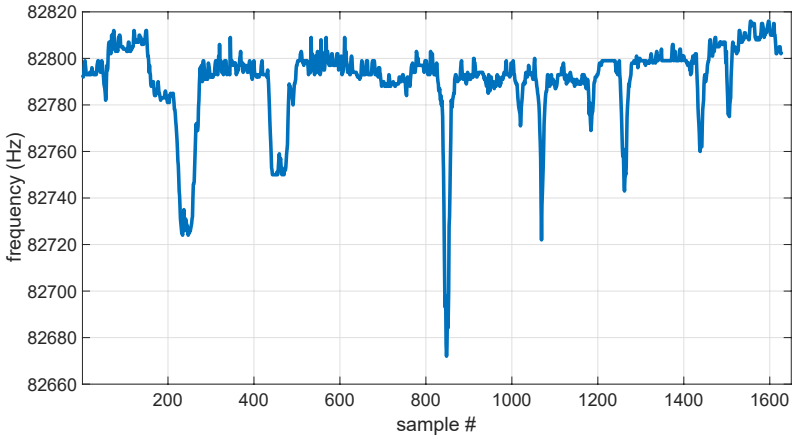



Figure 3.11: Example of raw sensor data acquired at 3 Hz while a person was moving in the room. High-pitch noise is visible at the top (far end of sensor range). Slow drift is mostly visible before sample #200 and after #1200.

modulation to measure the capacitance of the sensor transducer, we can explore front-ends based on carrier modulation in amplitude and/or phase, as well as other techniques that are less susceptible to and reject better the environmental noise.

In this setting, we record concurrently the reference position of the person using the ultrasound system (ground truth) and the capacitive sensor readings (see Fig. 3.12). Then we:

1. translate the average of the latter to zero,
2. pass it through a wide window (50 s) median filter (MF), to extract the slow drift,
3. pass it through a low-pass filter (LPF) with a pass-band edge of 0.1 Hz and a stop-band edge of 0.6 Hz, to reduce high-pitch noise (see both traces in Fig. 3.13(a)),
4. and finally, we subtract the median filter output from the LPF output, and
5. normalize the values to the $[0, 1]$ range to use them to train and test the performance of different NN types.

Note that the best values for the window and the cutoff frequency were found via an extensive design space exploration, as reported in Section 5.1.2.

In the semi-logarithmic scale plot of the inverted normalized output $(1 - y)$ of the filter block shown in Fig. 3.13(b), we can see a rugged but relatively flat low level as effect of the median filter reducing much of the drift visible on the top side of raw sensor output in Fig. 3.11. Filtering effects can also be seen in Fig. 3.14 comparing the frequency spectrum of the raw and the filtered sensor signals. As

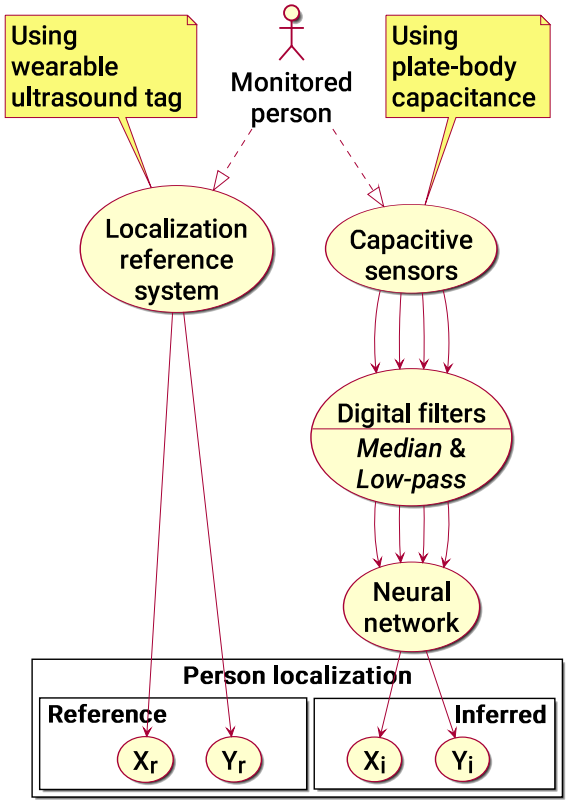
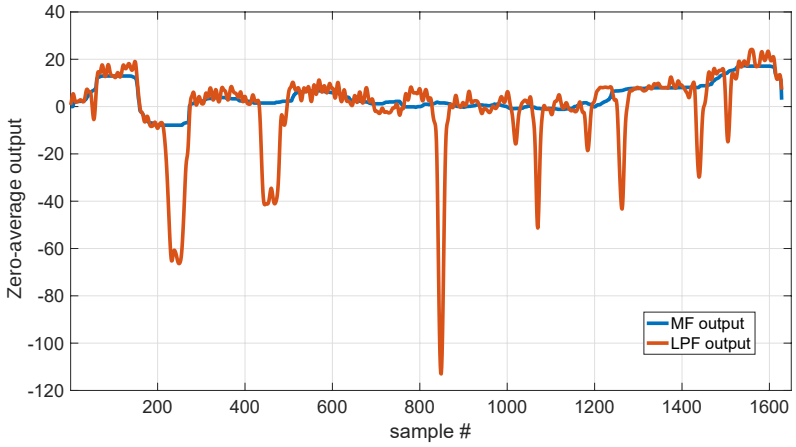


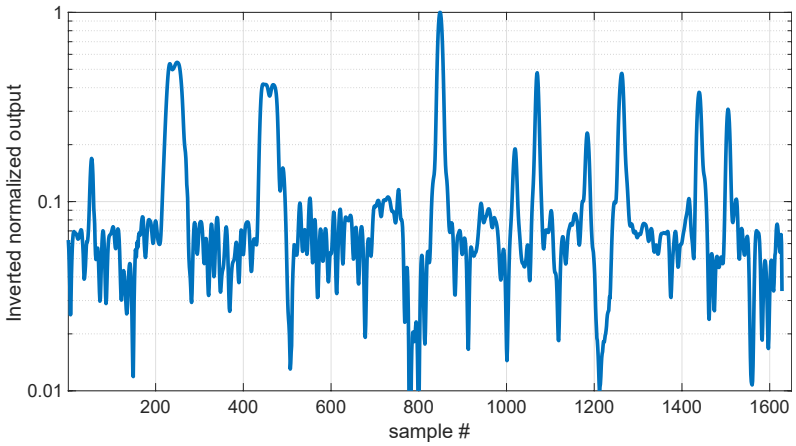

Figure 3.12: Experimental data processing uses an accurate ultrasound-based reference (for training data labelling and inference testing), and the capacitive sensor processing chain with digital filters and the neural network under test.

noted in Fig. 3.13(b), the very low frequency components are reduced by the median filter, while the components above 0.3 Hz are attenuated by the high-pass filter below the noise level around −60 dB.

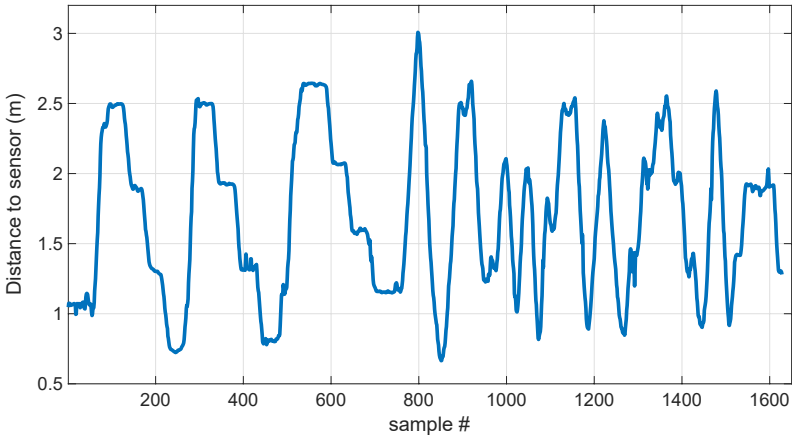
Comparing the frequency spectrum of the capacitive sensors with the spectrum of the location data from the ultrasound-based reference localization system, shown in Fig. 3.15, we note that also the noise floor of the reference system is around −60 dB, and that the signal emerges above it for frequencies below 0.3 Hz. However, while the sensor spectrum flattens around −40 dB for lower frequencies, the reference signal starts to increase below 0.1 Hz and is about 20 dB stronger than the sensor signal for lower frequencies, around 0.02 Hz. This part of the spectrum is important because most of the movement in the room during the experiment was slow, as can be expected of an elderly person, and likely contributed to the lower end of the frequency spectrum.




(a)




(b)




(c)

Figure 3.13: Sensor output sampled at 3 Hz after (a) filtering [50 s-window median (MF) and 0.1 Hz low-pass (LPF) filters], (b) normalization (shown inverted and in semi-logarithmic scale to expose the noise), and (c) distance to person body as they roam the room.

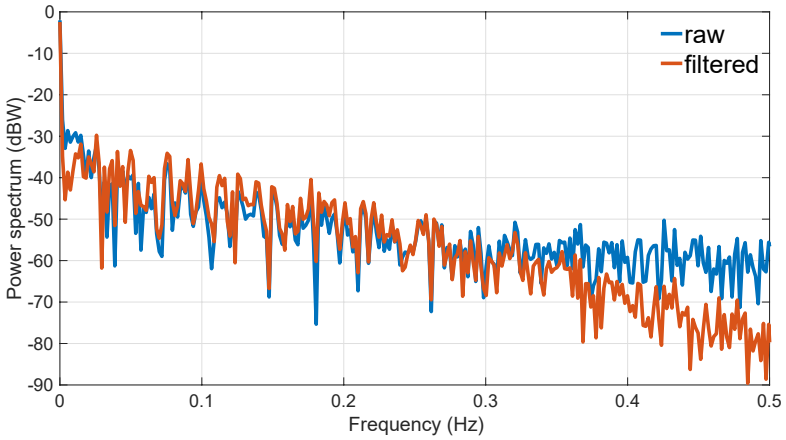



Figure 3.14: Power frequency spectrum of one sensor output before ("raw") and after digital filtering ("filtered"). The median filter attenuates frequencies below 0.02 Hz and the low-pass filter attenuates frequencies above 0.3 Hz.

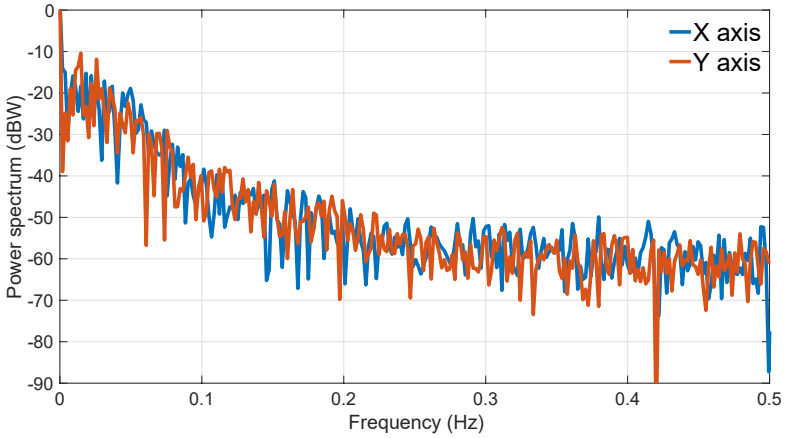



Figure 3.15: Power frequency spectrum of the reference localization system output coordinates, X and Y. Below 0.3 Hz the signal rises above the noise floor and increases especially at lower frequencies, corresponding to slow movements.

Comparing the filtered sensor signal in Fig. 3.13(b) to the plot of the distance between the person and the sensor in Fig. 3.13(c) (calculated from the reference system measurements), we can see a strong and well correlated sensor response when the person comes closer [the top peaks in Fig. 3.13(b) match the bottom peaks in Fig. 3.13(c)], but noise still limits the sensor sensitivity at longer distances [which can be seen from the poor correlation with the distance sensor-person of the rugged lower part of sensor response in Fig. 3.13(b)].

Considering the high noise level of this type of sensors, we are mostly interested in how well various neural network types can extract position and trajectory information from them.

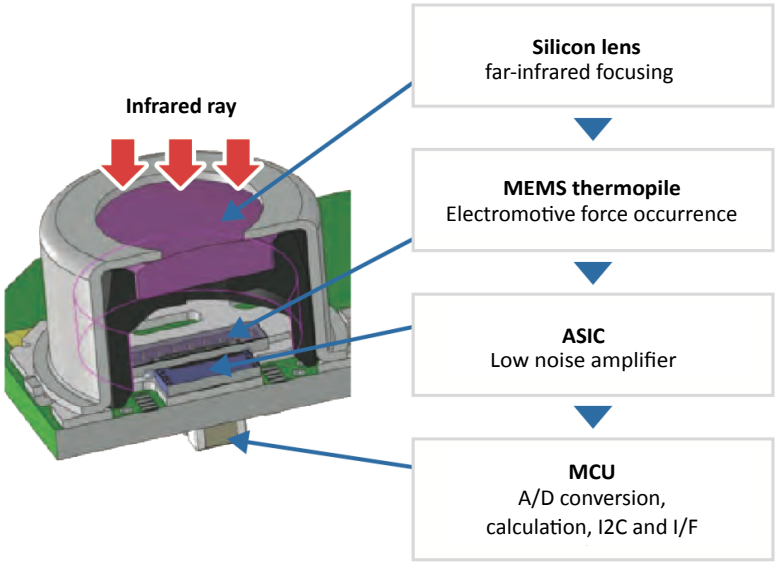



Figure 3.17: Cross-section view of the Omron D6T thermopile [166].

## 3.2 Infrared Sensing and Thermopiles

### 3.2.1 Working Principle

Thermopiles are made of several thermocouples connected usually in series. Thermocouples produce voltage proportional that is dependent on temperature difference as a result of the thermoelectric effect. We used a 4 pixel × 4 pixel Omron D6T-44L-06 thermopile infrared sensor with temperature resolution 0.14 °C and accuracy ±1.5 °C [165]. The sensor utilizes the Seebeck effect, in which a thermoelectric force is generated because of the temperature difference on the opposite ends of the junction point of two different types of metal [166].


Figure 3.16: Detection principle of thermopile sensor [166]: Seebeck effect in which thermoelectric force is generated because of the temperature difference on the opposite ends of the junction point of two different types of metal.

As shown in Fig. 3.16, the cold junctions are attached to a thermally isolated membrane, while the hot junction is influenced by ambient temperature. A voltage difference is generated which is proportional to the temperature gradients in the thermocouple between the hot and cold junctions. Fig. 3.17 shows the cross section

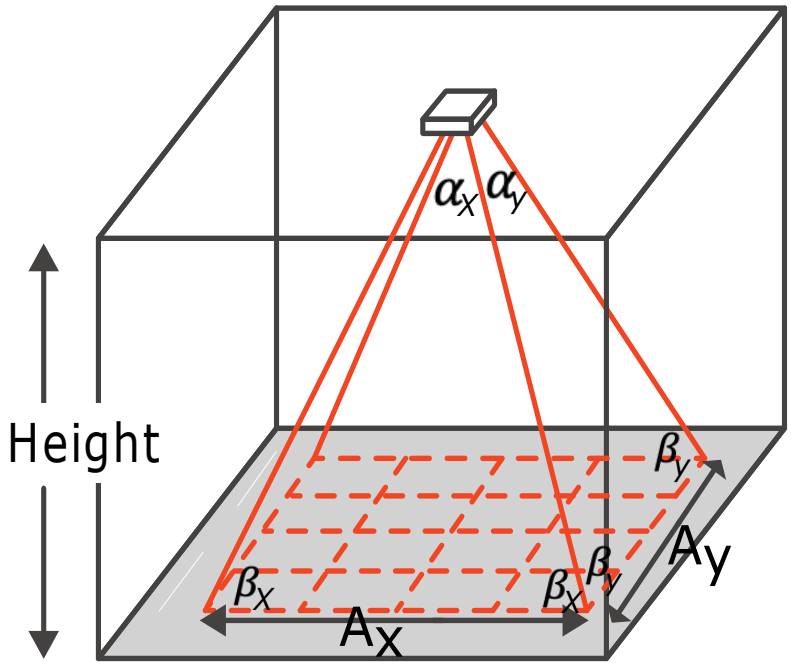



Figure 3.18: Conceptual view of the experimental space [166]. $\alpha_x$ and $\alpha_y$ show the view angles for X and Y direction respectively, while *Height* is the height of ceiling where sensor was placed. $A_x$ and $A_y$ show the area covered by field of view.

view of the Omron D6T thermopile, displaying various components of the sensor. The silicon lens on the top focus the radiant energy from objects or humans onto the thermopile. The radiant energy focused on the MEMS thermopile generates electromotive force. The values of the generated electromotive force and the internal thermal sensors are measured, and are then used to calculate the temperature of the object through interpolation, which compares the measured values with an internally stored lookup table [167]. The measured values are sent out through an I2C bus. We used an Arduino Uno as the host system, which received the measurements from the thermopile and sent them to the base station via XBee 802.15.4.

### 3.2.2 Data Acquisition

The D6T thermopile outputs the temperature of the objects in its field of view (FOV) in the form of 16 channels. These 16 channels cover the FOV in a $4 \times 4$ grid, as shown in Fig. 3.18.

The FOV varies as the distance between the sensor and the measured area changes. When the sensor is far from the ground, the FOV becomes large, but consequently, it will reduce the occupancy ratio of the person in the FOV. Thus, the environmental temperature can have more influence than the temperature of the person, which can adversely affect the localization performance. The X and Y view angles are provided

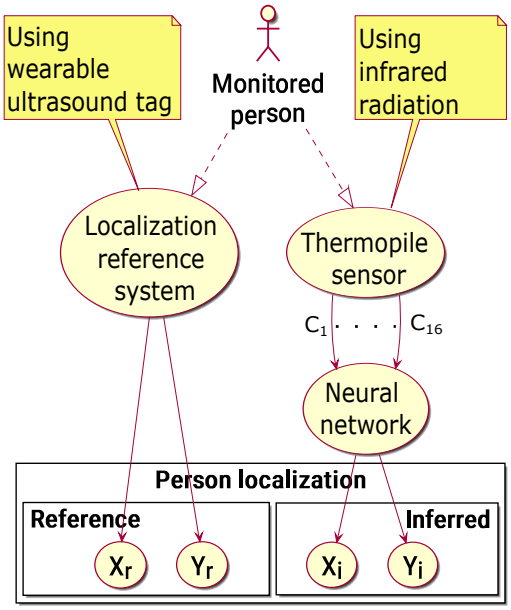



Figure 3.19: Experimental data processing uses an accurate ultrasound-based reference (for training data labelling and inference testing), and the thermopile sensor processing with the neural network under test.

in the catalog [166] which can be used along with the height of the ceiling, where the sensor is placed, to calculate the precise FOV through the trigonometric functions. The view angle in X direction is 44.2° and in Y direction is 45.7°, which we denote by $\alpha_x$ and $\alpha_y$ respectively. The FOV forms an isosceles triangle, which has two sides of equal length, and both base angles are the same. We denote base angles as $\beta_x$ and $\beta_y$. Using the trigonometric identity in (3.5) we get $\beta_x = 67.90$ and $\beta_y = 67.15$.

$$\beta = \frac{180 - \alpha}{2} \tag{3.5}$$

The sensor was placed at height of 3.05 m which we can denote as $h = 3.05$. Using (3.6) and (3.7), we obtain the field of view (FOV) 2.48 m × 2.57 m.

$$A_x = \frac{2.h}{tan(\beta_x)} = 2.48m \tag{3.6}$$

$$A_y = \frac{2.h}{tan(\beta_y)} = 2.57m \tag{3.7}$$

Fig. 3.18 shows the conceptual overview of the experimental room and corresponding field of view. Like for the experiments we conducted for capacitive sensing in Section 3.1.3, we used the same 3 m × 3 m experiment space. Similarly, we collected the reference person location with a tag of the Marvelmind Starter Set HW v4.9

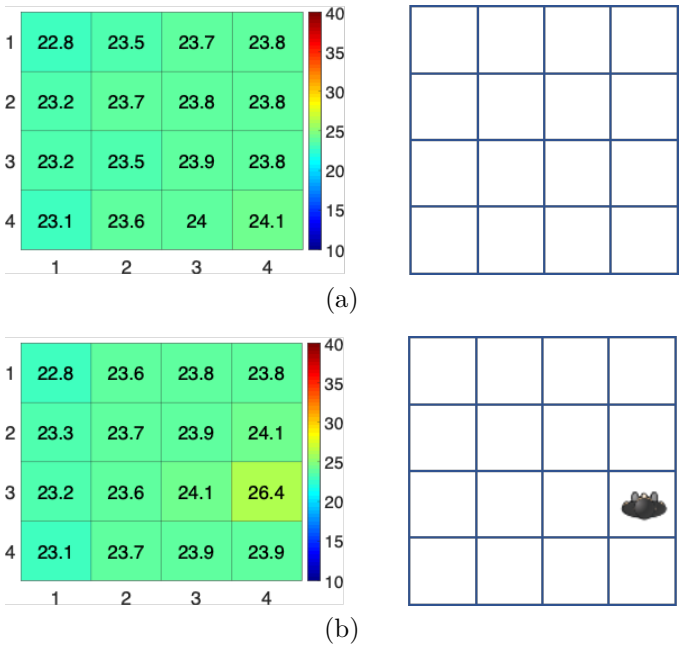
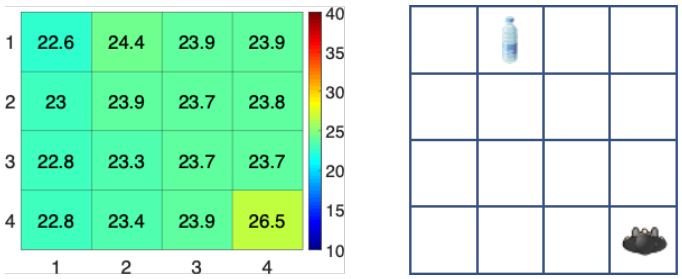



Figure 3.20: Thermopile output sample for the experiment conducted in the morning with subject 1 (a) with no person (b) with a person under FOV.

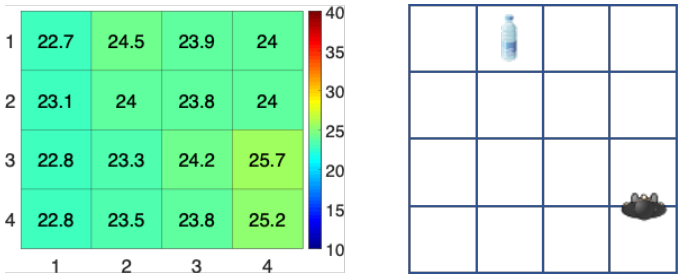
ultrasound-based system [80], with ±2 cm localization accuracy at 15 Hz. As described earlier, the average localization accuracy in our environment is ±3.9 cm (max ±6.4 cm and ±0.7 cm standard deviation) measured by acquiring four times per second for five seconds the location of a person standing in turn on 16 equidistant predefined locations inside the 3 m × 3 m experiment space. We conducted four data gathering campaigns with the thermal sensor. Two of these campaigns were conducted in the morning, while the other two were conducted the next day in the evening. Two human subjects were involved in gathering the data, each performed one experiment in the morning and one in the evening. During each experiment, the subject walked for around 30 min an arbitrary path with variable speed, and we collected synchronous readings from both the IR sensor and the ultrasound reference at 5 Hz (see Fig. 3.19). The specific conditions of the experiments are following:

1. **Morning, subject 1:** The experiment was conducted in the afternoon on a sunny day. The room had a window on one side from where sunlight was entering the room. There was no other object on the floor. As can be seen in Figure 3.20, the temperature difference between the human subject and the background is around 2.4 °C.

2. **Morning, subject 2:** This experiment was also conducted in the afternoon on the same day. Besides the sunlight entering through the window, an additional source of heat was introduced in this experiment by placing a bottle of warm

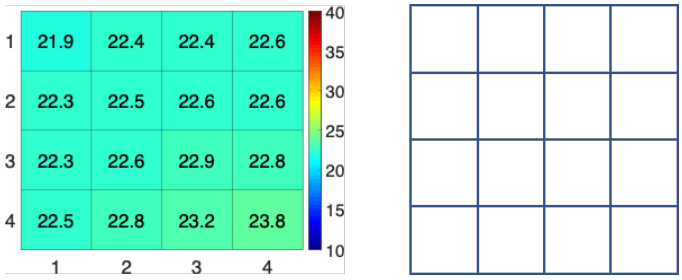



(a)

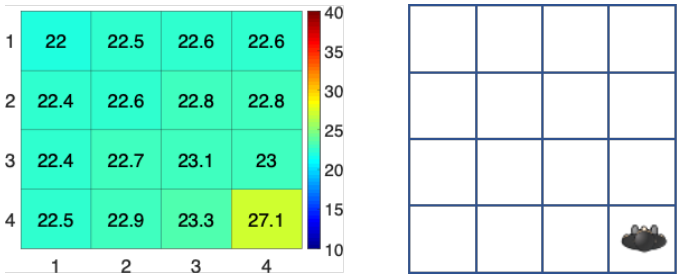



(b)

Figure 3.21: Thermopile output sample for the experiment conducted in the morning with subject 2 with a bottle of warm water on the floor(a) with no person (b) with a person on the border between two FOVs.

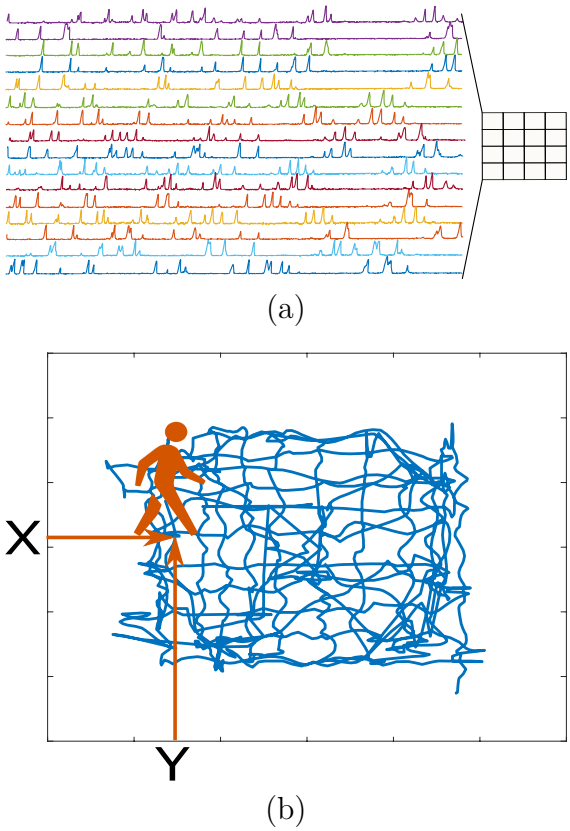



(a)




(b)

Figure 3.22: Thermopile output sample for the experiment conducted in the evening with subject 1 (a) with no person (b) with a person under FOV.




(a)

(b)

Figure 3.23: (a) Sixteen output waveforms from the thermopile sensor, (b) arbitrary path followed by a person.

water on the floor. This bottle can be observed as a warmer point in Figure 3.21.

3. **Evening, subject 1:** This experiment was conducted on the following day in the evening, after sunset. A greater difference between the human subject and the ambient temperature can be observed in Figure 3.22

4. **Evening, subject 2:** This experiment was also conducted in the evening on the same day, but by a different human subject.

Each experiment lasted for almost half an hour and the sensor data was collected with a sampling rate of 5 Hz. So, for each experiment we collect 8500 to 9000 tuples made of 16 thermal sensor readings and two room co-ordinates from the ultrasound reference. Fig. 3.23 shows a part of 16 output waveforms from the thermopile sensor as a person moves inside its FOV arbitrarily and a part of the arbitrary path followed by a person. The ground truth in the figure is recorded with the reference localization system (Marvelmind Starter Set HW v4.9 ultrasound-based system).

# Chapter 4

# Machine Learning for Indoor Position Classification With Capacitive Sensors

In this chapter we compare the performance of most ML classifiers in the Weka collection [1] in order to support the selection of the optimal ML algorithms to process sensor data for person localization. The performance of the of ML classification algorithms was analyzed in terms of localization accuracy, average distance error, precision, recall. Moreover, the effect of training data size on localization performance of the algorithms was examined. We use the data collected in a 3 m × 3 m room on 16 predefined positions, as explained in Section 3.1.2.

## 4.1 Machine Learning Based Classification

We executed all machine learning (ML) classifiers in the current study with their default parameter values used in the Weka collection, except for the boosting algorithms, where we tried only the base algorithms which are mentioned in the results tables along with the names of boosting algorithms. These default parameters can be found in Weka collection documentation [168].

- In BayesNet, the search algorithm is set by default to K2 and the maximum number of parents of a node is set to 1.
- For Random Forest, the number of iterations was set to 100 and unlimited tree depth. We also tried with 200 iterations but that gave an improvement of less than 0.5 % which is very limited when compared to doubling of computational cost.
- For SVM, we used the SVC clustering method with the radial basis function from the Weka collection LibSVM package [169].

Table 4.1: Average localization accuracy and error for data Set A for 100 runs of Weka collection best performing ML classification algorithms and boost methods.

| **Algorithm** | *Set A* | | | |
|---|---|---|---|---|
| | **Accuracy** | | **Error** | |
| | % | $\sigma$ | (m) | $\sigma$ |
| **Bayes Net** | 85.07 | 1.34 | 0.13 | 0.014 |
| **k-Nearest Neighbors** | 91.14 | 1.07 | 0.07 | 0.010 |
| **Support Vector Machine** | 91.75 | 1.01 | 0.07 | 0.011 |
| **Random Forest** | 92.81 | 0.968 | 0.06 | 0.009 |
| **LogitBoost(Random Forest)** | 93.55 | 0.918 | 0.05 | 0.008 |
| **AdaBoostM1(Random Forest)** | 92.96 | 1.01 | 0.06 | 0.009 |
| **AdaBoostM1(C4.5)** | 92.49 | 0.95 | 0.06 | 0.009 |

- For k-NN, we set $k = 1$. We also tried $k$ values up to 100, but higher values degraded the performance in our case.
- For LogitBoost running on top of Random Forest, the number of boosting iterations was set to 10, on top of Random Forest 100 iterations.
- Similarly, for AdaBoostM1 the number of boosting iterations was set to 10 on top of Random Forest 100 iterations and on top of C4.5.

## 4.2 Localization Performance

We first evaluated the performance of the Weka collection ML classifiers for indoor person localization using data sets A and B (see Section 3.1.2). Then, we merged Set A and Set B in a new set, *Set C*, which had a higher variance than each of its composing sets A and B, and we processed Set C with Weka algorithms as well. For each algorithm, Weka splits the input data in two parts: 75 % for algorithm training and 25 % for algorithm testing. We executed each algorithm 100 times, reshuffling the input data before each run, then we averaged the localization results over all 100 runs for each algorithm.

We show in Tables 4.1, 4.2, and 4.3, the results for data sets A, B and C of the four best performing ML classification algorithms in the Weka collection: Random Forest, k-Nearest Neighbors (for $k = 1$, i.e., one neighbor), Bayes Net and Support Vector Machine with SVC. We also report the results of LogitBoost used on top of Random Forest and AdaBoostM1 on top of Random Forest and C4.5. For all of them, we compare the localization performance in terms of accuracy and average distance error, calculated by summing all localization errors for all room locations and for all test samples, and dividing by the total number of test samples. For Set C, we also show in parentheses the results of 10-fold cross-validation, averaged over 100 runs.

Table 4.2: Average localization accuracy and error for data Set B for 100 runs of Weka collection best performing ML classification algorithms and boost methods.

| **Algorithm** | *Set B* | | | |
|---|---|---|---|---|
| | **Accuracy** | | **Error** | |
| | % | $\sigma$ | (m) | $\sigma$ |
| **Bayes Net** | 87.25 | 1.38 | 0.11 | 0.012 |
| **k-Nearest Neighbors** | 87.35 | 1.12 | 0.11 | 0.010 |
| **Support Vector Machine** | 87.84 | 1.32 | 0.11 | 0.013 |
| **Random Forest** | 91.53 | 1.09 | 0.07 | 0.009 |
| **LogitBoost(Random Forest)** | 92.06 | 1.02 | 0.07 | 0.009 |
| **AdaBoostM1(Random Forest)** | 91.81 | 1.01 | 0.07 | 0.009 |
| **AdaBoostM1(C4.5)** | 90.65 | 1.08 | 0.08 | 0.010 |

Table 4.3: Average localization accuracy and error for data Set C for 100 runs (and average of 100 runs of 10-fold cross-validation in parentheses) of Weka collection best performing ML classification algorithms and boost methods.

| **Algorithm** | *Set C* | | | |
|---|---|---|---|---|
| | **Accuracy** | | **Error** | |
| | % | $\sigma$ | (m) | $\sigma$ |
| **Bayes Net** | 83.39 (84.08) | 0.980 (0.242) | 0.14 (0.14) | 0.009 (0.002) |
| **k-Nearest Neighbors** | 87.64 (88.56) | 0.766 (0.200) | 0.10 (0.09) | 0.007 (0.002) |
| **Support Vector Machine** | 87.80 (88.35) | 0.903 (0.173) | 0.11 (0.10) | 0.009 (0.002) |
| **Random Forest** | 91.56 (92.10) | 0.861 (0.173) | 0.07 (0.07) | 0.007 (0.002) |
| **LogitBoost(Random Forest)** | 92.34 (92.83) | 0.753 (0.158) | 0.06 (0.06) | 0.007 (0.001) |
| **AdaBoostM1(Random Forest)** | 91.61 (92.20) | 0.836 (0.191) | 0.07 (0.07) | 0.008 (0.002) |
| **AdaBoostM1(C4.5)** | 90.98 (91.66) | 0.910 (0.257) | 0.08 (0.07) | 0.008 (0.002) |

Average distance error calculations were based on the confusion matrix generated by Weka for each tested algorithm. Fig. 4.1 shows one confusion matrix for Random Forest applied to Set A. The top row lists the correct positions and the rightmost column lists the positions determined by the algorithm. In absence of localization errors, the confusion matrix is diagonal. Each number outside the diagonal represents the number of erroneous predictions. We use these numbers together with the distance between the actual and the predicted position to calculate the total distance error.

Random Forest was consistently the best performing algorithm of the Weka collection, with accuracies of 92.81 %, 91.53 % and 91.56 % for Set A, Set B, and Set C respectively, and the lowest average distance error. The algorithm performance generally decreased on Set B because it is noisier [see Fig. 3.7, especially sensor B data in Fig. 3.7(b)]. SVM and k-NN were generally the second best performing algorithms with almost similar results. Bayes Net performance on Set B was almost the same, unlike k-NN and SVM whose performances decreased on Set B. Among the boosting algorithms, both LogitBoost and AdaBoostM1, showed slight improvements in terms of accuracy and average distance error. However, LogitBoost can be fairly

```
 a  b  c  d  e  f  g  h  i  j  k  l  m  n  o  p   <-- classified as
53  1  0  1  0  3  0  0  0  0  0  0  0  0  0  0 |  a = location1
 1 41 12  0  0  0  0  0  0  0  2  0  0  0  0  0 |  b = location2
 0  7 38  0  0  0  0  0  0  0  0  0  0  0  0  0 |  c = location3
 1  1  3 44  0  0  5  0  0  0  0  0  0  0  0  0 |  d = location4
 1  1  0  1 39  4  2  0  2  0  0  0  0  0  0  0 |  e = location5
 3  0  0  1  2 37  1  0  0  0  0  0  0  0  0  0 |  f = location6
 0  0  0  3  0  2 48  0  0  0  0  0  0  0  0  0 |  g = location7
 0  0  0  0  0  0  2 35  0  0  0  0  0  0  0  0 |  h = location8
 0  0  0  0  4  5  0  0 40  1  0  2  1  0  0  0 |  i = location9
 0  0  0  0  0  0  0  0  0 45  1  0  4  0  0  0 |  j = location10
 0  0  0  0  0  0  0  0  0  3 56  0  0  0  0  0 |  k = location11
 0  0  0  0  0  0  0  1  0  0  0 61  1  0  0  0 |  l = location12
 0  0  0  0  0  0  0  0  1  2  0  0 40  1  0  1 |  m = location13
 0  0  0  0  0  0  0  0  0  2  0  0  0 45  0  0 |  n = location14
 0  0  0  0  0  0  0  0  0  0  0  0  0  2 45  0 |  o = location15
 0  0  0  1  0  0  0  0  0  0  2  0  0  0  0 36 |  p = location16
```

Figure 4.1: One Random Forest confusion matrix generated by Weka for data Set A. The top row lists the correct positions and the rightmost column shows the positions determined by the algorithm. Each non-diagonal number represents the number of erroneous predictions.

expensive during both training and inferring, as we will discuss in Section 4.3.1.

### 4.2.1 Average Distance Error Per Position

Fig. 4.2 shows the distribution of the total distance error of each algorithm between the 16 room positions defined in Fig. 3.2. For each room position, we added the distance between the actual and the predicted position, and divided the sum by the total number of test samples. The average distance error is shown both quantitatively (in meters, below each position) and qualitatively (as dot intensity, darker for higher errors). Random Forest remains the best performing in terms of error among all locations. Note that the idea behind calculating the distance error (also reported in Table 4.1, 4.2 and 4.3) is to give a comparative idea on which algorithms have erroneous classification but closer in distance to the actual position. This information can be better seen through confusion matrix, but to provide a single comparative metric we used distance error.

### 4.2.2 Precision and Recall

Recall and precision are calculated as follows:

$$\text{Recall } (\%) = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}} \times 100 \tag{4.1}$$

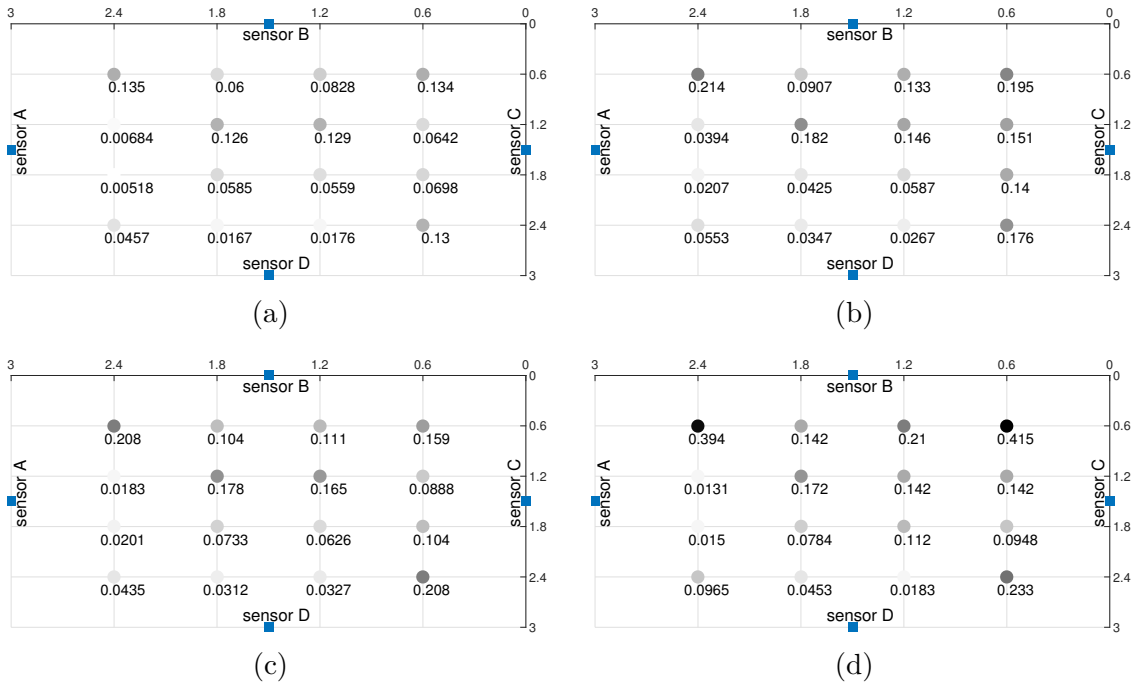



Figure 4.2: Localization error (in meters) for each position for Set C: (a) Random Forest, (b) SVM, (c) k-NN (for k=1), (d) Bayes Net. Darker dots for higher errors.

Table 4.4: Average precision and recall for Set A for 100 runs of Weka collection best performing ML classification algorithms and boost methods.

| **Algorithm** | *Set A* | |
|---|---|---|
| | **Precision (%)** | **Recall (%)** |
| **Bayes Net** | 85.80 | 85.08 |
| **k-Nearest Neighbors** | 91.41 | 91.15 |
| **Support Vector Machine** | 92.42 | 91.76 |
| **Random Forest** | 93.04 | 92.82 |
| **LogitBoost(Random Forest)** | 93.77 | 93.55 |
| **AdaBoostM1(Random Forest)** | 93.20 | 92.97 |
| **AdaBoostM1(C4.5)** | 92.75 | 92.50 |

$$\text{Precision (\%)} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}} \times 100 \tag{4.2}$$

where True Positives is the number of 4-tuples that are correctly classified, False Negatives is the number of 4-tuples pertaining to a position that are incorrectly classified as other positions, and False Positives is the number of 4-tuples pertaining to other positions that are incorrectly classified as a given position.

Tables 4.4, 4.5, and 4.6 show the average precision and recall of the algorithms for Set A, B, and C respectively. As mentioned above, 75 % of the samples in each each set was used for training and 25 % was used for testing. For Set C, we

Table 4.5: Average precision and recall for Set B for 100 runs of Weka collection best performing ML classification algorithms and boost methods.

| **Algorithm** | *Set B* | |
|---|---|---|
| | **Precision (%)** | **Recall (%)** |
| **Bayes Net** | 87.64 | 87.26 |
| **k-Nearest Neighbors** | 87.71 | 87.36 |
| **Support Vector Machine** | 88.43 | 87.85 |
| **Random Forest** | 91.78 | 91.55 |
| **LogitBoost(Random Forest)** | 92.30 | 92.07 |
| **AdaBoostM1(Random Forest)** | 92.07 | 91.82 |
| **AdaBoostM1(C4.5)** | 90.92 | 90.66 |

Table 4.6: Average precision and recall for Set C for 100 runs (and average of 100 runs of 10-fold cross-validation in parentheses) of Weka collection best performing ML classification algorithms and boost methods.

| **Algorithm** | *Set C* | |
|---|---|---|
| | **Precision (%)** | **Recall (%)** |
| **Bayes Net** | 83.70 (84.13) | 83.39 (84.08) |
| **k-Nearest Neighbors** | 87.81 (88.58) | 87.64 (88.57) |
| **Support Vector Machine** | 88.12 (88.46) | 87.80 (88.35) |
| **Random Forest** | 91.72 (92.13) | 91.56 (92.10) |
| **LogitBoost(Random Forest)** | 92.49 (92.86) | 92.35 (92.84) |
| **AdaBoostM1(Random Forest)** | 91.77 (92.22) | 91.62 (92.20) |
| **AdaBoostM1(C4.5)** | 91.13 (91.67) | 90.98 (91.66) |

also show in parenthesis the average of 100 runs of 10-fold cross-validation results. LogitBoost on top of Random Forest performed best for all sets, followed closely by AdaBoostM1 on top of Random Forest and then by their base algorithm, Random Forest. LogitBoost precision and recall are above 93 % for Set A, and above 92 % for sets B and C. Random Forest precision is above 93 % and the recall is above 92 % for Set A, and above 91 % for Set B and C.

The slightly lower performance for Set B is likely due to its noisier data, as can be seen in Fig. 3.7. Note, however, that all best performing ML algorithms considered are very robust to the significant amount of noise exhibited by our data sets.

### 4.2.3 Sensitivity To Training Data Size

The performance of the ML classifiers strongly depends on their training. However, there is no agreement on the optimal size of the training data in the scientific literature. The influence of the training data size on the performance of various algorithms is summarized by [110] from various previous studies.

In the following, we investigate how different Weka collection ML classification

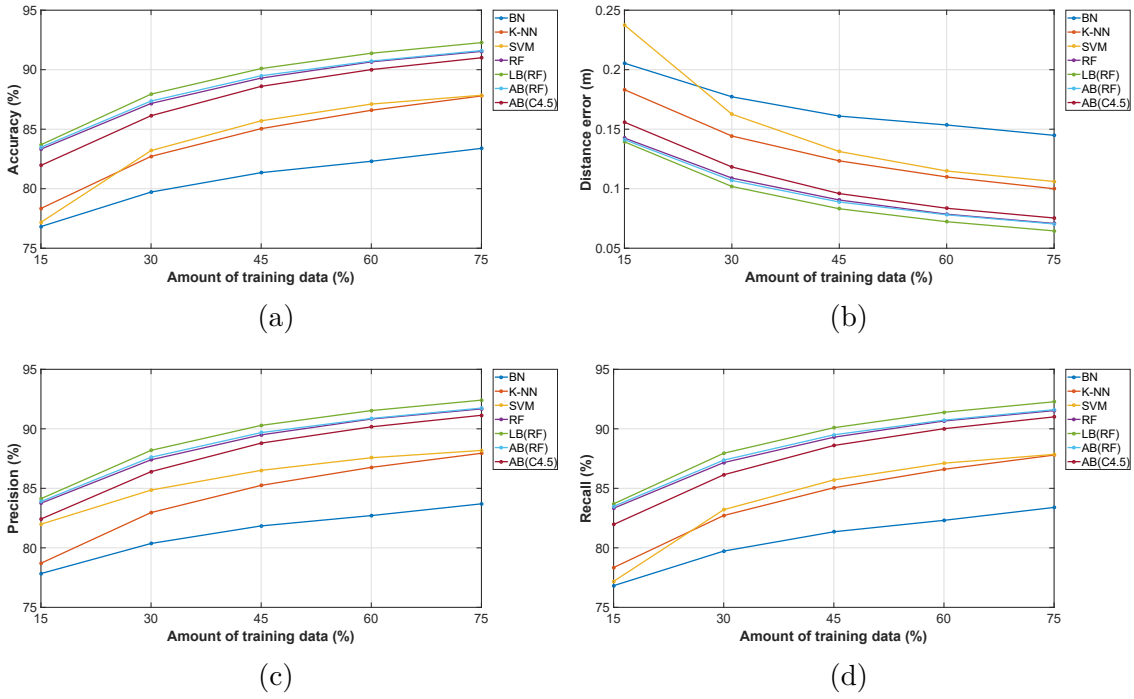



Figure 4.3: Training data size dependency of average accuracy (a), distance error (b), precision (c) and recall (d) for set C for best performing machine language classification algorithms in Weka collection: Bayes Net (BN), k-Nearest Neighbors (k-NN with $k = 1$), Random Forest (RF), Support Vector Machine (SVM), LogitBoost (LB(RF)) and AdaBoostM1(AB(RF)) running on top of Random Forest, and AdaBoostM1(AB(C4.5)) running on top of C4.5.

algorithms perform when trained with reduced data sets. The purpose is to explore if we can reduce the duration of training with a low impact on performance, so that the end users do not have to spend too much time training the system in actual deployments. For this purpose, we split the 5120 four-tuples samples in Set C in 25 % (1280 four-tuple samples) for testing and a variable size for training as follows:

1. 15 % (768 four-tuple samples) for training and 25 % (1280 four-tuple samples) for testing
2. 30 % (1536 four-tuple samples) for training and 25 % (1280 four-tuple samples) for testing
3. 45 % (2304 four-tuple samples) for training and 25 % (1280 four-tuple samples) for testing
4. 60 % (3072 four-tuple samples) for training and 25 % (1280 four-tuple samples) for testing

Table 4.7: Accuracy, precision and recall after using set A for training and set B for testing.

| **Algorithm** | Training *Set A*, testing *Set B* | | |
|---|---|---|---|
| | **Accuracy (%)** | **Precision (%)** | **Recall (%)** |
| **Bayes Net** | 58.98 | 59.60 | 59.00 |
| **k-Nearest Neighbors** | 60.31 | 63.10 | 60.30 |
| **Support Vector Machine** | 55.12 | 68.10 | 55.10 |
| **Random Forest** | 67.11 | 70.30 | 67.10 |
| **LogitBoost(Random Forest)** | 64.10 | 65.50 | 64.10 |
| **AdaBoostM1(Random Forest)** | **67.85** | **71.70** | **67.90** |
| **AdaBoostM1(C4.5)** | 67.27 | 71.10 | 67.30 |

5. 75 % (3840 four-tuple samples) for training and 25 % (1280 four-tuple samples) for testing

For each training ratio above, we shuffled all data in Set C before splitting it into training and testing samples, then we ran the localization algorithm. We repeated this process 100 times for each ratio.

The results in Fig. 4.3, Fig. 4.5, and Fig. 4.6 show that different algorithms are affected differently by the size of the training set. Generally, a larger training set improves the performance up to a point of near saturation. Precision and recall follow a similar trend, again with LogitBoost improving slightly the performance of its base algorithm, Random Forest.

## 4.3 Effect of Train and Test Sets From Different Distributions

So far in all the evaluations we performed, the train and test sets were taken either from the same distribution, or were mixed, with 75 % of the data used for training and 25 % of the data used for testing.

We also evaluated the ML algorithms where the one data set was used for training and the other data set was used for testing, i.e. *Set A* was used for training and *Set B* was used for testing and vice versa. Note that these two sets were taken months apart, with different human subjects, the sensor were taken off the experimental area and the reinstalled. Thus, this test represents a more challenging scenario.

In these conditions, we obtained the best accuracy of 67.85 % when *Set A* was used for training and *Set B* was used for testing with AdaBoostM1 running on top of Random Forest. When *Set B* was used for training and *Set A* was used for testing we obtained the best accuracy of 71.05 %, also with AdaBoostM1 running on top of Random Forest.

Table 4.8: Accuracy, precision and recall after using set B for training, and set A for testing.

| **Algorithm** | Training *Set B*, testing *Set A* | | |
|---|---|---|---|
| | **Accuracy** (%) | **Precision** (%) | **Recall** (%) |
| **Bayes Net** | 63.63 | 66.20 | 63.60 |
| **k-Nearest Neighbors** | 63.09 | 65.70 | 63.10 |
| **Support Vector Machine** | 38.40 | 65.80 | 38.40 |
| **Random Forest** | 69.77 | 71.50 | 69.80 |
| **LogitBoost(Random Forest)** | 70.16 | 72.10 | 70.20 |
| **AdaBoostM1(Random Forest)** | **71.05** | **72.30** | **71.10** |
| **AdaBoostM1(C4.5)** | 66.84 | 69.70 | 66.80 |

Table 4.9: Average processing effort during training and inferring for set C for 100 runs of the best performing Weka collection algorithms. Random Forest seems to be the best trade-off between processing effort and performance.

| **Algorithm** | **Time** | |
|---|---|---|
| | **Training** (s) | **Test** (s) |
| **Bayes Net** | 0.4365 | 0.2699 |
| **k-Nearest Neighbors** | 0.0799 | 2.3113 |
| **Support Vector Machine** | 5.4715 | 3.1431 |
| **Random Forest** | 4.4159 | 1.1028 |
| **LogitBoost(Random Forest)** | 99.5197 | 42.634 |
| **AdaBoostM1(Random Forest)** | 26.2447 | 9.0915 |
| **AdaBoostM1(C4.5)** | 5.3893 | 0.3608 |

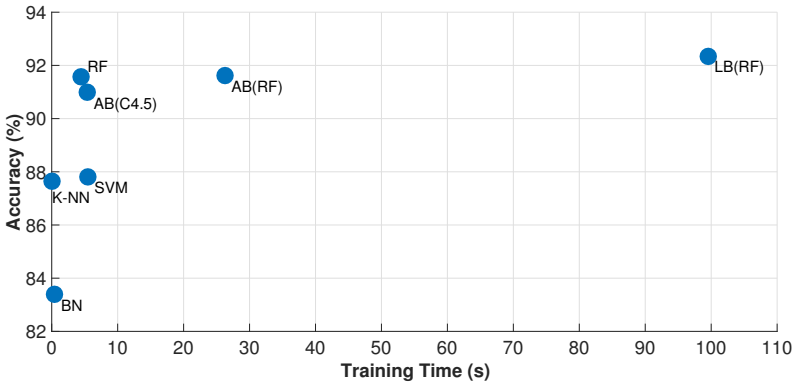
### 4.3.1 Training and Testing Effort

We compare the training and inferring effort required by some of the best performing localization algorithms. The performance during the inferring (localization) phase is by far the most critical for most applications, since it typically lasts for the entire exploitation phase of a deployed system (years), while the training phase is generally much shorter.

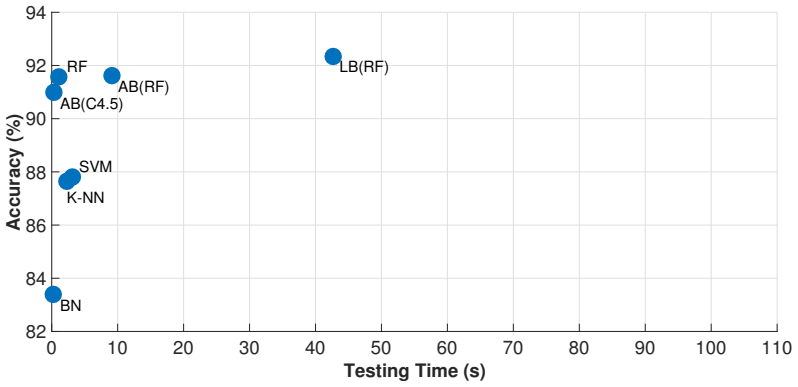
The Weka collection ML algorithm suite was run on a Virtual Machine running Ubuntu (64 bit). The Virtual Machine was allocated 2 GB of physical memory and 1 CPU. The host system had an AMD Athlon 64 X2 Dual Core processor, 4 GB RAM and was running Windows 10.

Table 4.9 shows the time taken by different algorithms to build the model during training and the time taken to infer the location using the test data. LogitBoost performs slightly better than AdaBoostM1, both on top of Random Forest, but at the cost of much higher modelling and inferring time since it computes the weights after every iteration based on the obtained classifier [170]. AdaBoostM1 on top of C4.5 performs slightly worse than Random Forest, but it infers faster.

K-nearest neighbor is a non-parametric lazy learning algorithm that keeps all




(a)




(b)

Figure 4.4: Processing effort in terms of CPU time during training (a) and processing effort during inferring (b) versus accuracy for set C. Bayes Net (BN), k-Nearest Neighbors (k-NN with $k = 1$), Random Forest (RF), Support Vector Machine (SVM), LogitBoost [LB(RF)] and AdaBoostM1[AB(RF)] running on top of Random Forest, and AdaBoostM1[AB(C4.5)] running on top of C4.5. Random Forest and AdaBoostM1 with C4.5 seems the best trade-off between localization processing effort and performance.

training data in memory for inferring instead of building a model during training. Hence, it trains fast, but it is computing- and RAM-intensive during inferring.

Random Forest is an ensemble method whose overall training complexity is close to the sum of the complexities of building the individual trees. The actual complexity varies with parameters like the number of trees (100 in our case).

Fig. 4.4 shows the training and inferring times versus accuracy. As can be observed, Random Forest and AdaBoostM1(C4.5) are the best trade-offs between localization processing effort and performance, especially during the testing phase.

## 4.4 Summary

We tested the performance of ML classification algorithms in the Weka collection for indoor person localization using capacitive sensors. We compared localization accuracy, precision and recall, distance error, classification error, and resource requirements (processing, memory and training set size). We used two sets of 2560 four-tuples of samples gathered from four sensors at different times. We first measured the localization accuracy and distance error for most Weka collection classification algorithms (see Fig. 4.5 and Fig. 4.6). Then, we analyzed in detail the most promising ones: Bayes Net, k-Nearest Neighbors, Support Vector Machine, Random Forest, LogitBoost (running on top of Random Forest) and AdaBoostM1 (running on top of Random Forest and C4.5).

Generally, we can conclude that Random Forest was performing best. Both LogitBoost and AdaBoostM1 running on top of Random Forest showed slightly better performance than Random Forest. However, they required significantly more processing time for training and inferring.

It is worth noting, however, that AdaBoostM1 used on top of C4.5 required much less inferring time than Random Forest, with only a slight loss of accuracy and requiring a comparable training time. Hence, as mentioned earlier, AdaBoostM1 on top of C4.5 can be best for energy-constrained localization applications, e.g., to reduce the maintenance requirements of battery-powered nodes.

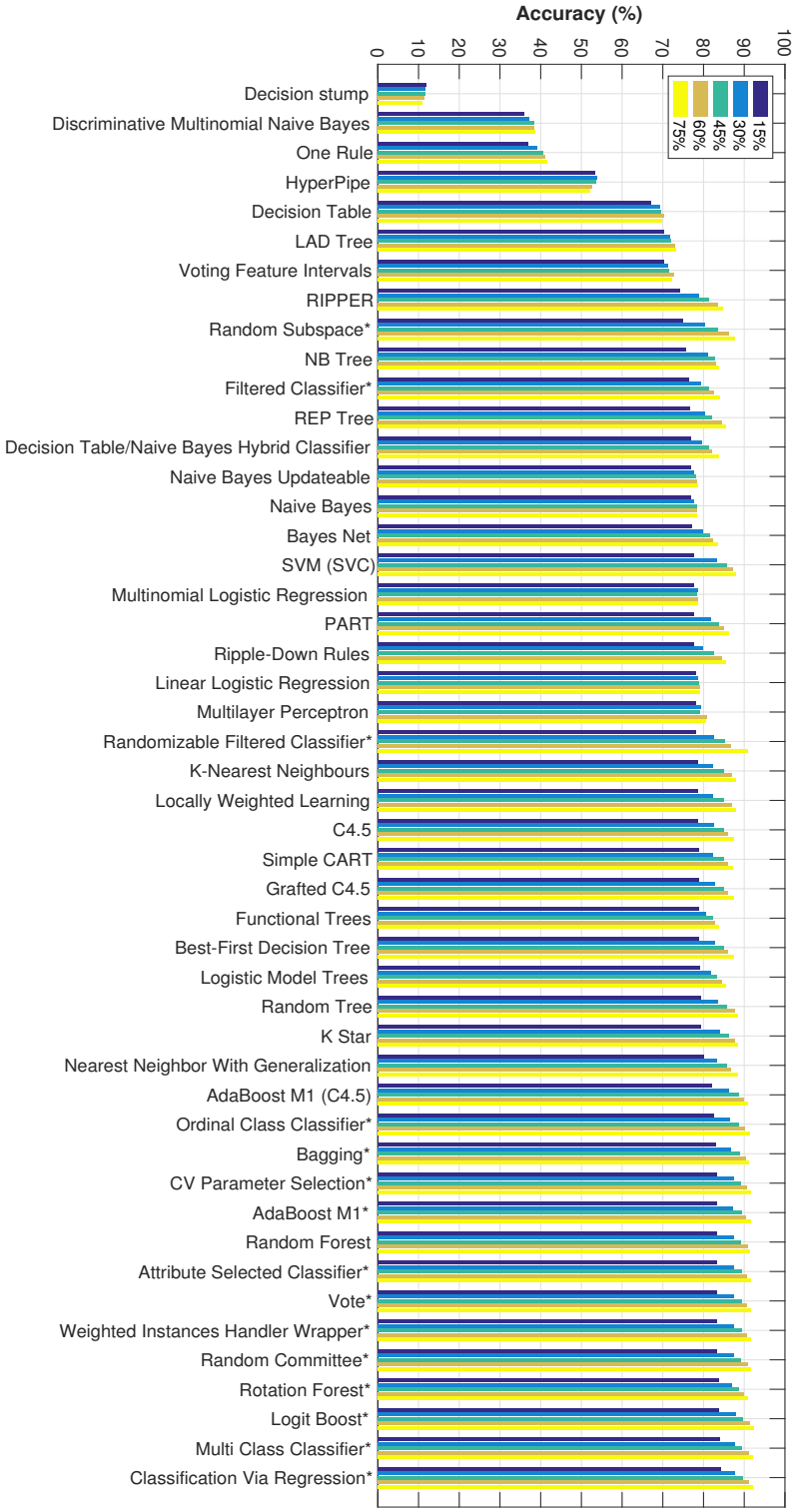



Figure 4.5: Training data size dependency of average accuracy for set C for Weka collection ML classification algorithms. Starred algorithms are built on top of Random Forest.

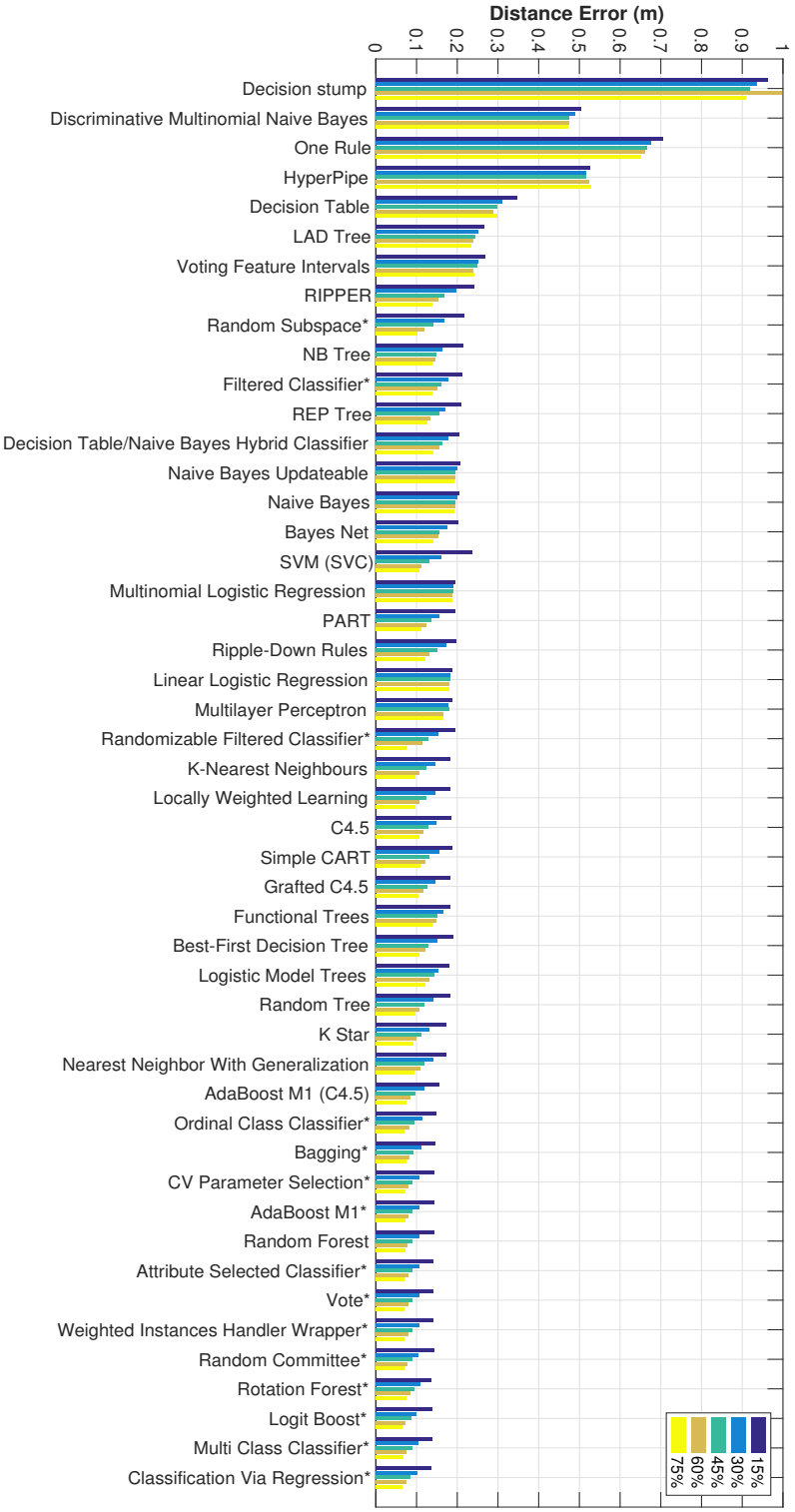



Figure 4.6: Training data size dependency of distance error for set C for Weka collection ML classification algorithms. Starred algorithms are built on top of Random Forest.

# Chapter 5

# Neural Network Architectures and Localization Performance for Indoor Position Tracking

In this chapter we explore how well different types of neural networks in basic configurations can extract location and movement information from noisy experimental data (with both high-pitch and slow drift noise) obtained from capacitive sensors operating in loading mode at ranges much longer than the diagonal of their plates. Through design space exploration, we optimize and analyze the location and trajectory tracking inference performance of multilayer perceptron (MLP), autoregressive feedforward, 1D Convolutional, and Long-Short Term Memory neural networks on experimental data collected using four capacitive sensors with 16 cm × 16 cm plates deployed on the boundaries of a 3 m × 3 m open space in our laboratory. We evaluate the localization and tracking performance, as well as resource and processing requirements, of various neural network (NN) types. Then, we evaluate the performance of the same NN architectures optimized for data from a low resolution 16-pixel thermopile sensor array monitoring the same experimental room.

## 5.1 Neural Network Architectures and Localization Performance for Capacitive Sensors

We use two types of tests: static localization and movement tracking. For the former, we use the data that was collected for classification experiments, as described in Section 3.1.2, i.e. 320 tuples collected while a person was standing in each one of the 16 positions shown in Fig. 5.3, i.e. $320 \times 16 = 5120$ total tuples.

For the latter, we use the sensor and reference data collected while the person moves along arbitrary paths in the room for about nine minutes (1626 tuples at 3 tuples/s, as was described in Section 3.1.3).

Generally, the trajectory does not come too close to the walls to better reflect the person movements in actual rooms and because the sensor measurements closer to the center of the room (far from the sensors) are noisier, hence more interesting for us, because they are more difficult to interpret.

Note that in both tests we train and test the neural networks to report the location estimation as a pair of X/Y coordinates, not to classify the position into a predefined set of locations.

We optimize and compare the performance of several types of neural networks in terms of mean square error (MSE) and average Euclidean distance error (ADE) between the inferred position and the reference position (ground truth). For movement tracking, we also compare graphically the plots of the ground truth (as reported by the reference system) and the NN inference. We do this separately for the X and Y coordinates instead of 2D plots of full trajectories to visualize better inference discrepancies from ground truth and to comparatively analyze the accuracy of different NNs.

We discretize the sensor data (shown on top-left of Fig. 5.1) at 3 Hz into four-sample tuples, $S_1, \ldots, S_4$, holding one sample for each capacitive sensor. Then we concatenate the tuples in chronological order and provide them (with suitable windowing, as we will discuss later) to the tested neural networks.

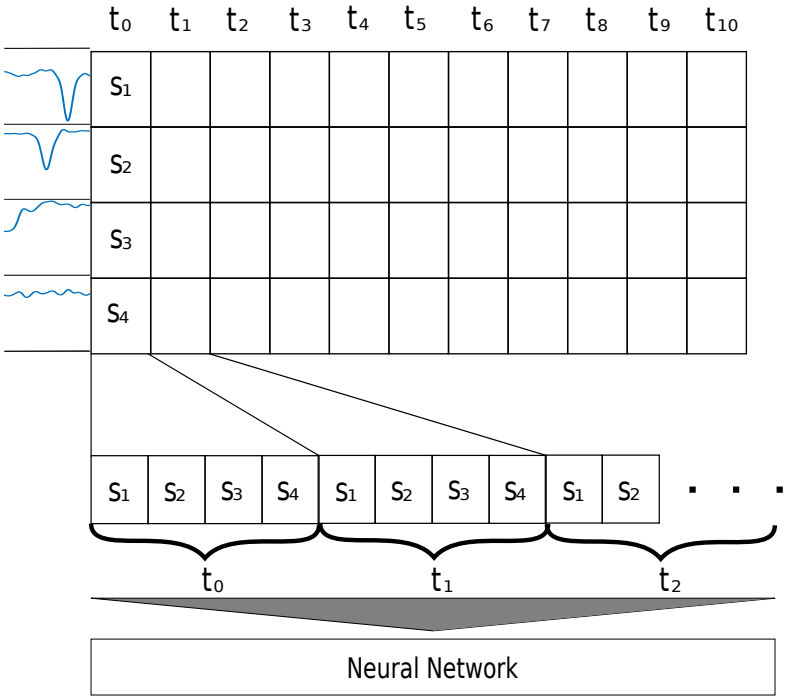



Figure 5.1: Sensor data (top-left) is discretized at 3 Hz in four-sample tuples, $S_1, \ldots, S_4$, which are then concatenated in chronological order and input to the neural network with appropriate windowing.

### 5.1.1 Static Position Classification With Multilayer Perceptron Neural Networks

We use the preprocessed experimental data from our previous classification experiment (see Section 3.1.2). They hold 320 capacitive sensor tuples collected when the person stood in each of the 16 positions shown in Fig. 5.3 (320 tuples × 16 positions = 5120 tuples) labelled with the coordinates of each position.

Because this experiment monitors static positions, we can consider the tuples independent and use a multilayer perceptron NN (see Fig. 5.2), which does not consider the tuple temporal sequence. Hence, we split the experimental data randomly in subsets of size 60 %, 20 %, and 20 % to use for NN training, validation, and testing, respectively. We use the validation set to stop the NN training before it starts overfitting, i.e. when the NN inference error on the validation set starts to increase while the error on the training set continues to decrease.

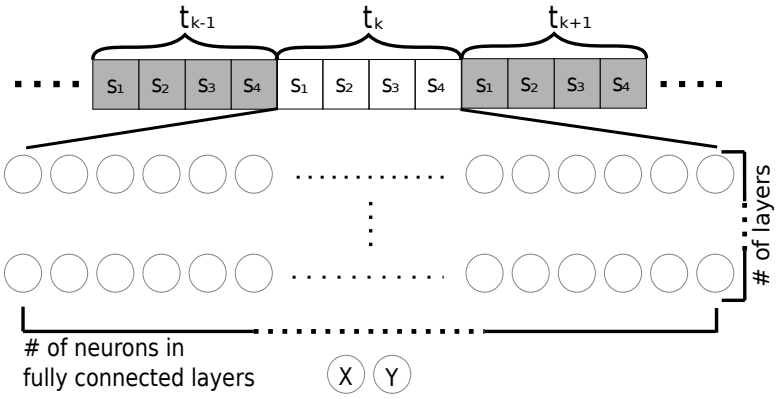



Figure 5.2: Network structure and data access for the multilayer perceptron network. The input layer receives sensor data tuples from randomly selected time frames (with labels during training, not shown for readability) and reports the inferred $x$ and $y$ coordinates of the person.

We keep the size of the input layer fixed at four neurons, equal to the number of sensors. We vary the structure of the rest of the neural network from one hidden layer with four neurons up to five hidden layers with 64 neurons each (all hidden layers have always the same number of neurons). For each configuration, we train and test the NN ten times using random initializations.

Table 5.1 shows the best MSE performance of the multilayer perceptron NN. Both performance metrics (MSE and ADE) improve as either the number of hidden layers or the number of neurons per hidden layer increase, albeit with diminishing returns beyond four hidden layers with 32 neurons each.

Fig. 5.3 shows the location inferred by the best NNs for each one of the 16 static locations. Standard deviation is from 0.040 m to 0.227 m and we can see that most

Table 5.1: Mean Square Error (MSE) and Average Distance Error (ADE) for multilayer perceptron neural network inferring static locations.

| | **Number of hidden layers** | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 1 | | 3 | | 4 | | 5 | |
| **Neurons per layer** | MSE ($m^2$) | ADE (m) | MSE ($m^2$) | ADE (m) | MSE ($m^2$) | ADE (m) | MSE ($m^2$) | ADE (m) |
| 4 | 0.116 | 0.373 | 0.074 | 0.299 | 0.073 | 0.298 | 0.065 | 0.271 |
| 8 | 0.084 | 0.324 | 0.062 | 0.262 | 0.053 | 0.235 | 0.051 | 0.225 |
| 16 | 0.076 | 0.307 | 0.048 | 0.224 | 0.042 | 0.203 | 0.039 | 0.188 |
| 32 | 0.072 | 0.302 | 0.037 | 0.186 | 0.030 | 0.159 | 0.026 | 0.142 |
| 64 | 0.063 | 0.273 | 0.026 | 0.150 | 0.024 | 0.137 | **0.022** | **0.124** |

inferences are close to the actual location. We also note that the inference spread appears to be higher in the upper and the right parts of the room. This may be due to higher environmental noise in that areas of the experimental space.

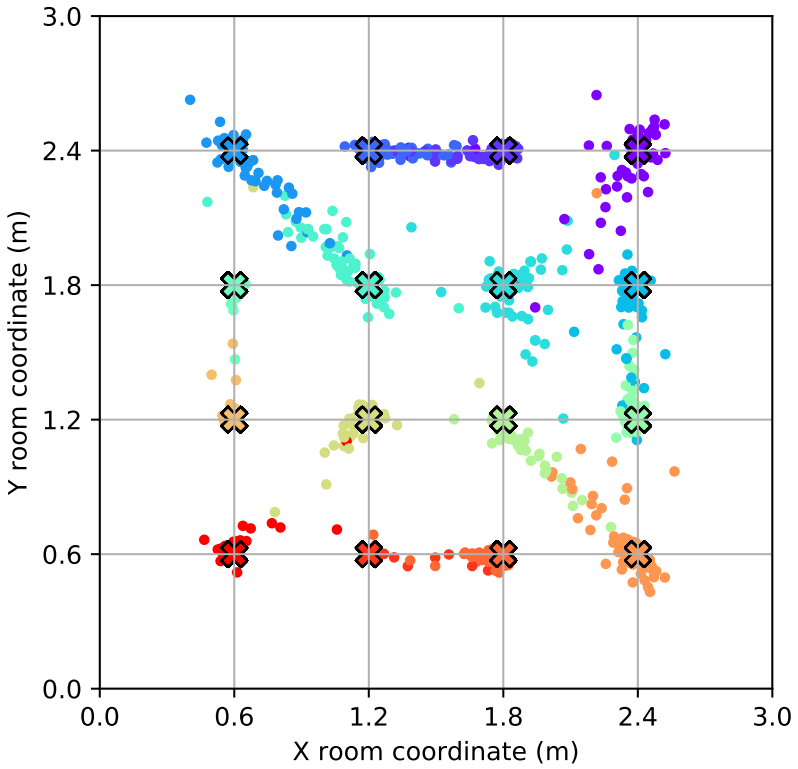



Figure 5.3: Inference of 16 static positions (black 'x') using multilayer perceptron neural networks. Inference results (dots) are colored according to their reference positions.

### 5.1.2 Filter Optimization and Trajectory Tracking With Multilayer Perceptron Neural Networks

We will show later that neural networks that track the position of the person considering the past behaviour can have better accuracy than those that consider one tuple at a time, as in Section 5.1.1. But we evaluate first the tracking performance of the multilayer perceptron neural networks (see Fig. 5.2), which do not consider the past behaviour, to establish a baseline for the next experiments.

We split the whole trajectory of the person in the room (after applying the median and low-pass filter) in three contiguous segments 60 %, 20 %, and 20 % long for neural network training, validation, and testing, respectively, as shown in Fig. 3.10. Note that the trajectory segments are different, as they would be in the case of a real-life deployment.

We use the best NN structure that we found in Section 5.1.1, five hidden layers with 64 neurons each. We next optimize the parameters of the filters (see Fig. 3.13) by analyzing the NN performance for all combinations of:

- *median filter window*: 50 s, 100 s, and 150 s
- *low-pass filter pass-band edge*: 0.1 Hz, 0.2 Hz, 0.3 Hz, 0.4 Hz, and 0.6 Hz
- *low-pass filter stop-band edge*: 0.2 Hz, 0.3 Hz, 0.4 Hz, 0.5 Hz, 0.6 Hz, and 0.7 Hz.

We obtain the best NN performance (MSE 0.111 $m^2$ and ADE 0.405 m) for a median filter window of 50 s, and low-pass filter pass-band edge of 0.3 Hz and stop-band edge of 0.4 Hz,Hz (see the top ten results shown in Table 5.2). The NN performance seems to be more dependent on the median filter window (windows longer than 50 s generally lead to poorer NN results) than on the parameters of the low-pass filter (almost all present in the NN top ten best results). This can be explained because the amplitude of the drift can be much higher than that of the high-pitch noise (see Fig. 3.11). For these parameters, we show in Fig. 5.4 the NN inference separately for the X and Y coordinates compared with the ground truth. We notice the ragged look of both X and Y inference, which seems not to smooth enough the sensor noise. We also notice increasing discrepancies in the latter part of the X and Y tracks (roughly after sample 250).

### 5.1.3 Trajectory Tracking With Autoregressive Feedforward Neural Networks

This is the first experiment to infer the trajectory using a neural network that considers some aspects of the movement history. We choose autoregressive feedforward neural networks [171, 172] because they are non-recurrent (i.e., feedback-free) sequence-aware models that can be used to infer sequential data as a simpler

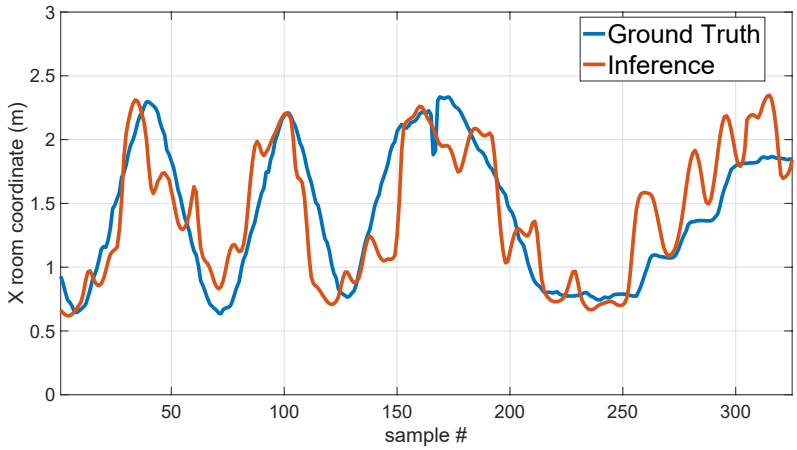
Table 5.2: Design space exploration results for filter optimization for the best multi-layer perceptron neural network Mean Square Error (MSE) and the corresponding Average Distance Error (ADE).

| **Low-pass filter** | | **Median filter** | **Error** | |
|---|---|---|---|---|
| Passband edge (Hz) | Stopband edge (Hz) | Window (s) | MSE ($m^2$) | ADE (m) |
| 0.3 | 0.4 | 50 | **0.111** | **0.405** |
| 0.5 | 0.7 | 50 | 0.115 | 0.418 |
| 0.2 | 0.6 | 50 | 0.116 | 0.432 |
| 0.1 | 0.3 | 50 | 0.121 | 0.429 |
| 0.6 | 0.7 | 50 | 0.122 | 0.428 |
| 0.4 | 0.5 | 50 | 0.122 | 0.431 |
| 0.1 | 0.6 | 50 | 0.125 | 0.432 |
| 0.1 | 0.4 | 50 | 0.126 | 0.445 |
| 0.2 | 0.6 | 100 | 0.127 | 0.449 |
| 0.3 | 0.6 | 50 | 0.128 | 0.447 |

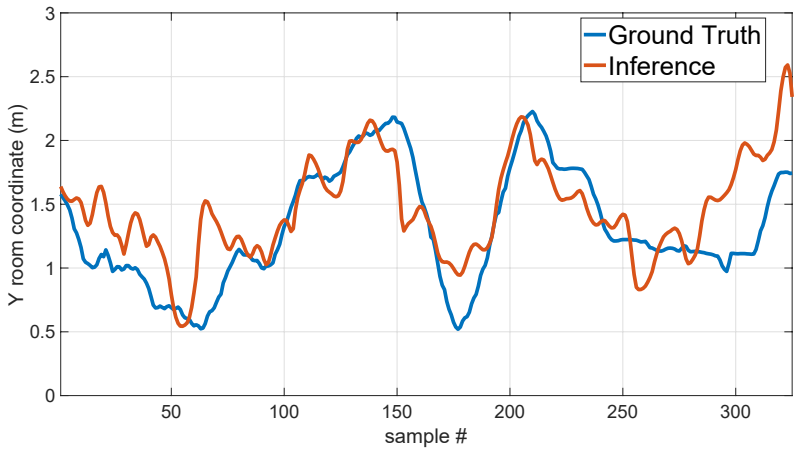
alternative to recurrent neural networks (see Fig. 5.5). They are akin to finite impulse response filters in digital signal processing, while recurrent NNs are akin to infinite impulse response filters. We provide the NN with capacitive sensor tuples that fall within a temporal window and train the NN to infer the X and Y coordinates corresponding to the *middle* tuple in each window. Hence, the window gives the NN access to both past and future readings in the sensor time series. Using these, during training the NN can refine the best weights for both the past and future sensor readings (around the current position) to better reject the noise and perhaps also to learn the dynamic characteristics of person movements, such as maximum speed, acceleration, movement patterns or direction changes. Of course, the NN may also significantly overfit in a real deployment, hence we put a lot of attention to the "natural looking" aspects of our sample trajectories.

Since the NN needs a window width of samples to produce a valid inference, it will start inferring after seeing a full window of samples at the beginning of the trajectory and stop when the last sample of the trajectory enters the window. But it infers the position corresponding to the middle of the window, hence the inferred trajectory in Fig. 5.6 starts and stops half a window from trajectory extremes. This behavior is not a problem for our target applications, which are not particularly sensitive to delays of a few seconds. The same applies to the other NNs based on windows that are discussed later. We implement the best network structure that we found in Section 5.1.1 (see Fig. 5.5), i.e., five hidden layers with 64 neurons each. The input layer receives all sensor samples within the input window and is fully connected to the first hidden layer, like in Fig. 5.2.

We explore the performance of this neural network by setting the duration of its input window to 5 s, 10 s, and 15 s. They give the number of input tuples (thus the




(a)




(b)

Figure 5.4: Multilayer perceptron neural network trajectory tracking inference and ground truth for the (a) X axis and (b) Y axis.

size of the NN input layer) that are seen by the NN at any moment at 3 Hz sampling rate. For example, for an input window of 5 s we have on the NN input layer

$$5\,\text{s} \times 3\,\text{tuples/s} \times 4\,\text{samples/tuple} = 60\ \text{input neurons}. \tag{5.1}$$

We also vary the parameters of the low-pass and median filters in the ranges shown in Section 5.1.2. As can be seen from the top-ten best results shown in Table 5.3, the best MSE is 0.079 $\text{m}^2$ for an ADE of 0.342 m. They show a marked improvement compared to the multilayer perceptron NN results shown in Table 5.2, which is attributable to allowing the NN to infer the position while examining the sensor tuples of a segment of the trajectory instead of just the current tuple. We also note that the optimal length of the trajectory segment (input window) in our DSE seems to be 10 s (the performance for windows of 5 s are equal or marginally better).

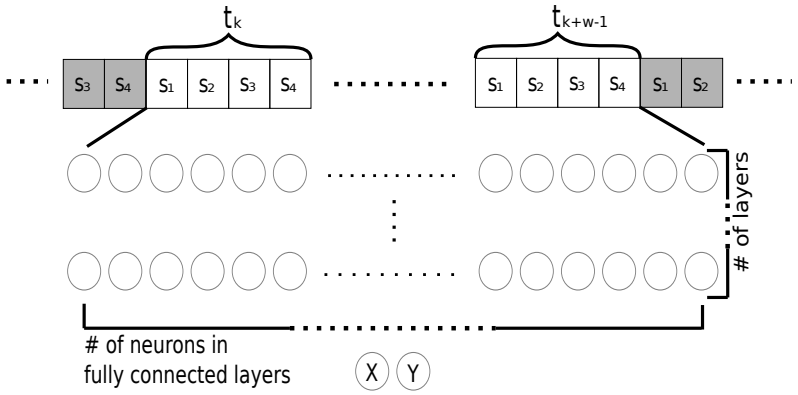
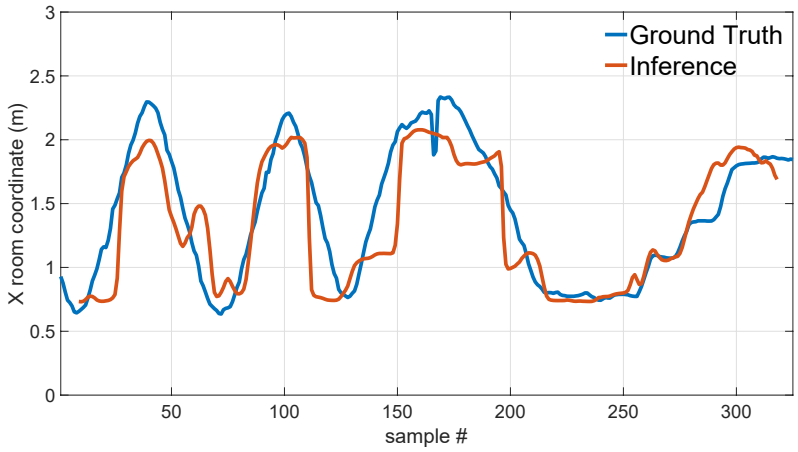



Figure 5.5: Network structure and data access for the autoregressive feedforward network. The input layer receives sensor data tuples from sequential time frames that fall into network window (labelled with the $x$ and $y$ coordinates of the central sample during training, not shown for readability) and reports the inferred coordinates of the person.

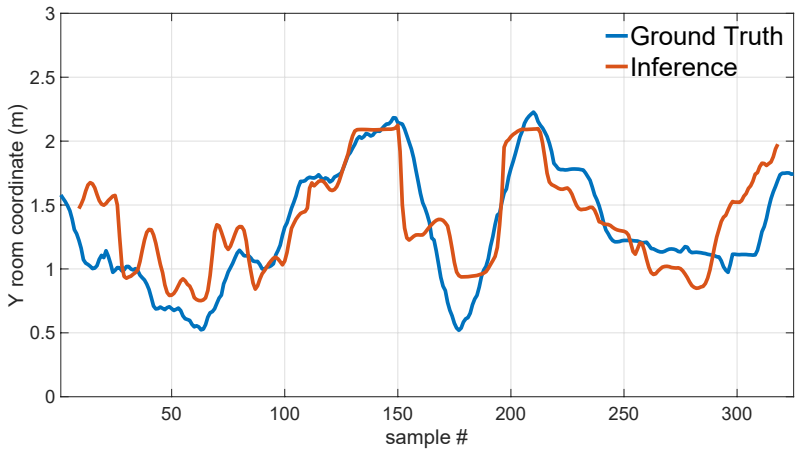
Table 5.3: Design space exploration results for filter optimization for the best autoregressive feedforward neural network Mean Square Error (MSE) and the corresponding Average Distance Error (ADE).

| **NN Input** | **Low-pass filter** | | **Median filter** | **Error** | |
|---|---|---|---|---|---|
| Window (s) | Passband edge (Hz) | Stopband edge (Hz) | Window (s) | MSE ($m^2$) | ADE (m) |
| 5 | 0.1 | 0.6 | 50 | **0.079** | 0.342 |
| 10 | 0.2 | 0.6 | 50 | 0.082 | 0.347 |
| 10 | 0.3 | 0.6 | 50 | 0.083 | **0.340** |
| 10 | 0.4 | 0.7 | 50 | 0.085 | 0.365 |
| 10 | 0.6 | 0.7 | 100 | 0.086 | 0.371 |
| 5 | 0.3 | 0.6 | 50 | 0.092 | 0.358 |
| 10 | 0.1 | 0.5 | 100 | 0.092 | 0.380 |
| 10 | 0.4 | 0.6 | 50 | 0.092 | 0.373 |
| 5 | 0.5 | 0.6 | 50 | 0.093 | 0.358 |
| 10 | 0.4 | 0.7 | 100 | 0.093 | 0.395 |

In Fig. 5.6 we see that the inferences of the X and Y coordinates are smoother and tend to follow closer the ground truth almost everywhere, and especially towards the end (roughly after sample 250) than the multilayer perceptron NN shown in Fig. 5.4.




(a)




(b)

Figure 5.6: Autoregressive feedforward neural network trajectory tracking inference and ground truth for the (a) X axis and (b) Y axis.

### 5.1.4 Trajectory Tracking With 1D Convolutional Neural Networks

We extend the tests of neural networks that infer the trajectory based on movement history using 1D Convolutional NNs [173]. They are known to be effective for deriving meaningful features from fixed-length segments (input windows) of data sequences. Typical applications include sequences of sensor data (e.g., accelerometer, audio), which are similar to the ones that we have in this application.

We use 1D CNNs with the structure shown in Fig. 5.7. We set the window size to 5 s, which we determined to be among the best options in Section 5.1.3. We scan this window with several 1D convolution filters with kernels of the same size, which make one convolutional layer. Convolutional processing in our NN uses several such

layers, followed by a pooling layer, and a fixed size MLP network (of two layers with 64 neurons each), before the output layer. During the DSE, we change the number

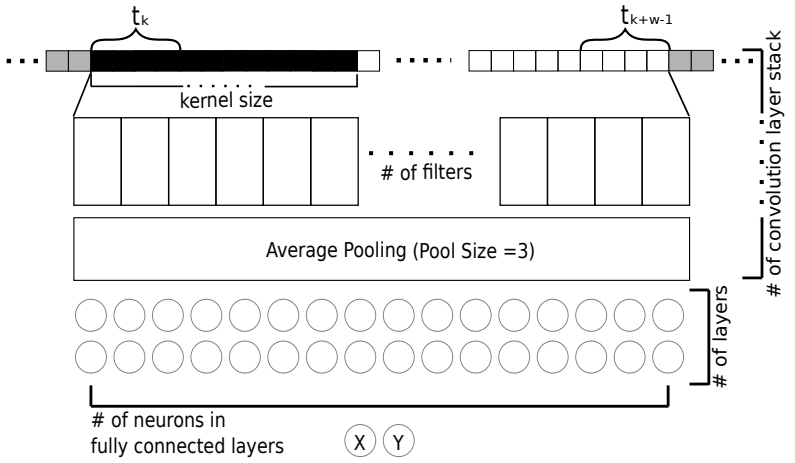



Figure 5.7: Network structure and data access for the convolutional network. Each filter processes the data tuples within the kernel width, then the kernel slides one tuple to the right until the end of the input window. When done, the window moves one tuple to the right and the kernel restarts its scanning. For network training, each window is labelled (not shown for readability) with the person coordinates corresponding to the middle tuple in the window.

of convolutional layers (in groups of two convolutional layers followed by a 50 % dropout), the convolution kernel size, and the number of filters. In each experiment, we keep constant for each network architecture the kernel size and the number of filters per layer. We use the LPF and MF parameters optimized in Section 5.1.2.

We train and test ten times the neural network for each combination of hyperparameters. We show in Table 5.4 the results of the best network for each hyperparameter combination. We note that most of the best configurations have four convolutional layers, while the network performance tends to degrade for either smaller or larger number of layers. Also, for a given number of convolutional layers, the network configurations with fewer filters appear to have the best performance. The best overall network configuration has the least number of filters (eight) and the smallest convolutional kernel size (three). We show in Fig. 5.8 the inference of the X and Y coordinates of the best networks for each number of convolutional layers, two, four, and six (highlighted in Table 5.4). We note how the six-layer network matches well the last part of the X trace, but less so in the middle. The two-layer network appears to have the highest ripples, while the four-layer seems to match best almost the whole trace, except for the last part, roughly after sample 250. On the Y trace, the four-layers network appears to stay closest to the ground truth overall.

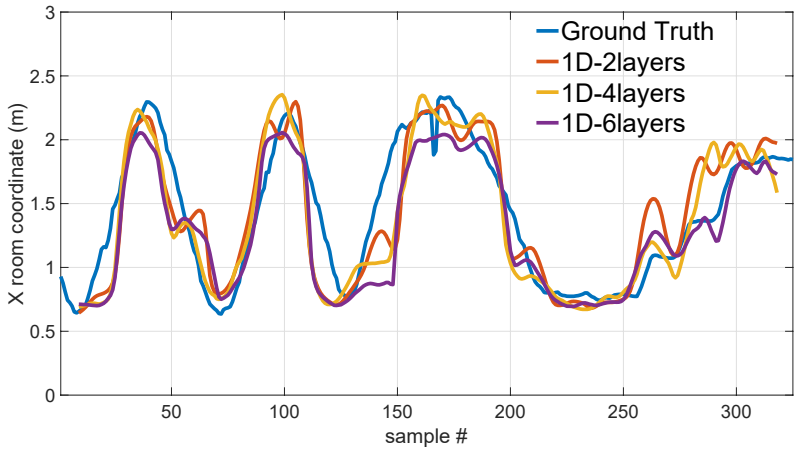
Table 5.4: Design space exploration results for 1D convolutional neural networks Mean Square Error (MSE) and the corresponding Average Distance Error (ADE) for different convolutional kernel sizes, number of convolutional layers, and filters.

| | **Number of filters** | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **8** | | **16** | | **32** | | **64** | |
| **Kernel size** | MSE ($m^2$) | ADE (m) | MSE ($m^2$) | ADE (m) | MSE ($m^2$) | ADE (m) | MSE ($m^2$) | ADE (m) |
| | Two 1D convolutional layers | | | | | | | |
| **3** | 0.086 | 0.351 | 0.092 | 0.370 | 0.090 | 0.364 | 0.085 | 0.357 |
| **5** | 0.084 | 0.346 | 0.094 | 0.373 | 0.093 | 0.369 | 0.092 | 0.371 |
| **7** | 0.080 | 0.347 | 0.093 | 0.359 | **0.078** | **0.343** | 0.097 | 0.384 |
| | Four 1D convolutional layers | | | | | | | |
| **3** | ***0.063*** | ***0.307*** | 0.089 | 0.372 | 0.090 | 0.366 | 0.085 | 0.351 |
| **5** | 0.081 | 0.350 | 0.088 | 0.358 | 0.088 | 0.371 | 0.091 | 0.365 |
| **7** | 0.093 | 0.369 | 0.090 | 0.351 | 0.091 | 0.388 | 0.078 | 0.342 |
| | Six 1D convolutional layers | | | | | | | |
| **3** | **0.078** | **0.328** | 0.101 | 0.377 | 0.087 | 0.364 | 0.092 | 0.385 |
| **5** | 0.086 | 0.365 | 0.090 | 0.372 | 0.099 | 0.379 | 0.093 | 0.378 |
| **7** | 0.098 | 0.387 | 0.107 | 0.397 | 0.092 | 0.366 | 0.092 | 0.373 |

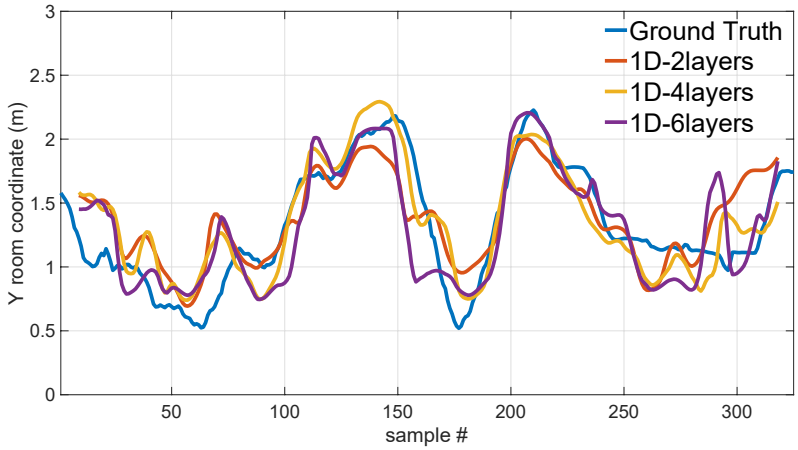
### 5.1.5 Trajectory Tracking With Long-Short Term Memory Networks

After testing feedforward neural networks (multilayer perceptron, autoregressive, and 1D convolutional types, discussed in previous sections), we explore recurrent neural networks. Of these, the LSTMs [123] are widely used to extract features from data sequences, such as speech or handwriting.

Fig. 5.9 shows the structure of our LSTM network [123]. It is a typical LSTM, in which the cells transfer the state horizontally and receive inputs either from the sensors (the first layer) or from the outputs of the previous layer. To explore the LSTM network performance, we vary the hyperparameters known to have most influence [124], namely the number of neurons in the hidden layers and the number of hidden layers. During the DSE, we run ten times the LSTM training (with random initialization) and testing for each combination of hyperparameters, and report the best results in the top half of Table 5.5. We note that the network performance, MSE, does not change much with the number of hidden layers or their number of neurons, and especially so for smaller numbers of neurons (8 or 16 per each hidden layer). An LSTM with one hidden layer with 16 neurons appears to perform best. Bidirectional LSTMs (BD-LSTMs) [174] can improve the LSTM performance leveraging future samples in their inference (e.g., handwriting recognition can improve by looking




(a)




(b)

Figure 5.8: Best 1D convolutional neural network trajectory tracking inferences and the ground truth for the (a) X axis and (b) Y axis for different number of convolutional layers.

also at letters after the current one). We test the BD-LSTM performance for our problem using the same DSE parameters and report the results in the second half of Table 5.5. Performance seems to be more sensitive to hyperparameters, and especially to the number of hidden layers. The best appears again the configuration with one hidden layer with 16 neurons.

We show in Fig. 5.10 the inference of the X and Y coordinates of the best LSTM and BD-LSTM network (highlighted in Table 5.5). Generally, we note very little differences between them. They both miss the first two peaks of the X coordinate and the central peak of the Y coordinate, as well as the beginning (up to sample 70 or so) and end (from around sample 250) of the Y coordinate. In some occasions BD-LSTM appears to come closer than LSTM to the ground truth, such as around sample 50

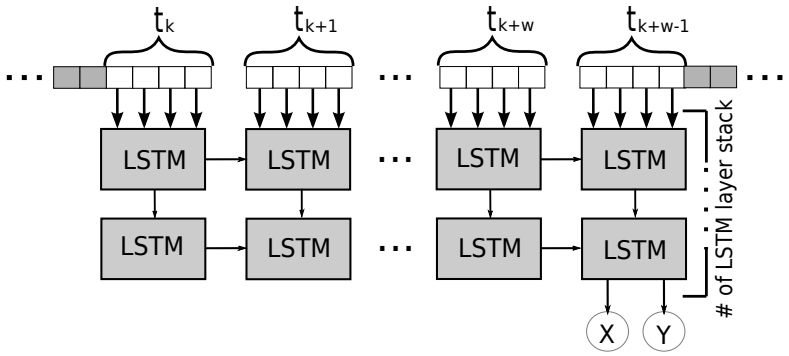
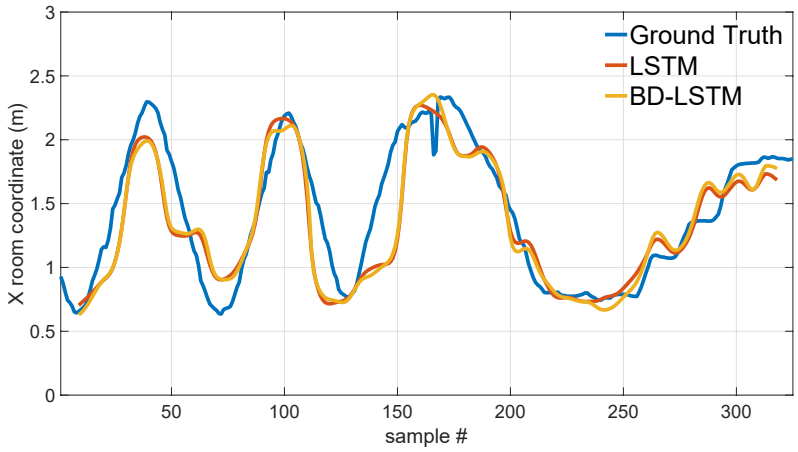



Figure 5.9: Network structure and data access for the Long-Short Term Memory (LSTM) network, in which the LSTM cells in the input layer process the data tuples from the input window. Each window is labelled for training with the person coordinates corresponding to the middle tuple in the window (not shown for readability).

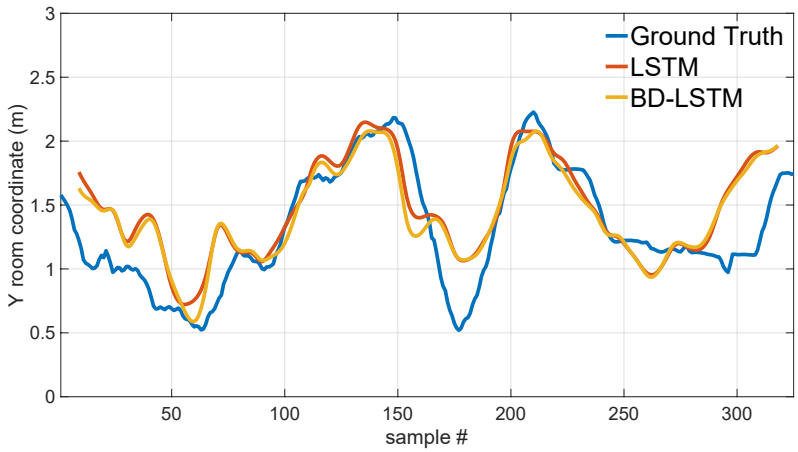
Table 5.5: Design space exploration results for uni- and bi-directional long-short term memory networks Mean Square Error (MSE) and the corresponding Average Distance Error (ADE) while varying the number of hidden layers and neurons per layer.

| | **Internal units of LSTM layer** | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **8** | | **16** | | **32** | | **64** | |
| **Layers** | MSE ($m^2$) | ADE (m) | MSE ($m^2$) | ADE (m) | MSE ($m^2$) | ADE (m) | MSE ($m^2$) | ADE (m) |
| | Unidirectional long-short time memory neural network | | | | | | | |
| **1** | 0.085 | 0.339 | **0.080** | **0.325** | 0.085 | 0.333 | 0.089 | 0.352 |
| **2** | 0.083 | 0.345 | 0.082 | 0.335 | 0.088 | 0.347 | 0.091 | 0.357 |
| **3** | 0.084 | 0.350 | 0.083 | 0.342 | 0.096 | 0.366 | 0.087 | 0.355 |
| | Bidirectional long-short time memory neural network | | | | | | | |
| **1** | 0.083 | 0.342 | **0.079** | **0.326** | 0.091 | 0.341 | 0.095 | 0.362 |
| **2** | 0.095 | 0.362 | 0.092 | 0.352 | 0.099 | 0.378 | 0.102 | 0.376 |
| **3** | 0.080 | 0.339 | 0.099 | 0.372 | 0.110 | 0.408 | 0.107 | 0.401 |

on the Y trace and sample 160 on the X trace. We note that the trajectories inferred by the LSTM network are the smoothest among all networks that we explored. They seem to reflect more closely the movement dynamics of a person, albeit with slightly higher error than the best inference (LSTM 0.079 $m^2$ MSE and 0.326 m ADE versus 1D convolutional 0.063 $m^2$ MSE and 0.307 m ADE). We intend to investigate in future work if LSTM networks indeed capture better movement dynamics and if their performance improves using less noisy readings from capacitive sensors that are more robust to environmental noise.




(a)

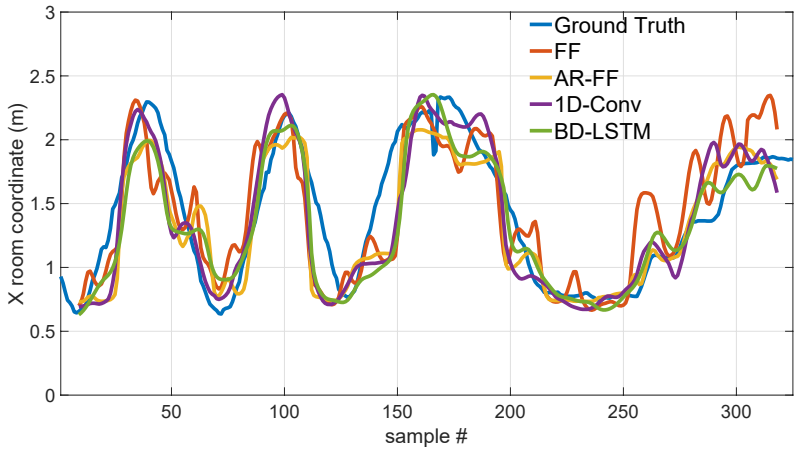



(b)

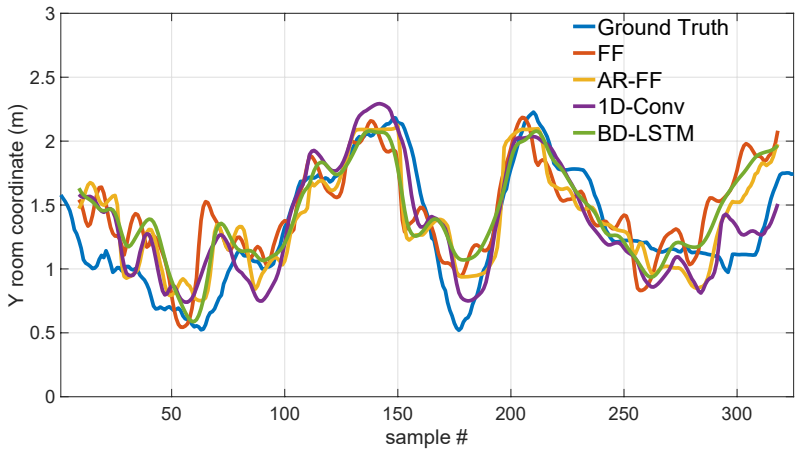
Figure 5.10: Best Long-Short Term Memory (LSTM) and Bidirectional LSTM (BD-LSTM) neural network trajectory tracking inferences and the ground truth for the (a) X axis and (b) Y axis.

### 5.1.6 Summary

In Fig. 5.11, we can compare visually the quality of the inferred X and Y traces of the room trajectory of the best configurations of all neural network types. Their performance metrics are shown in Table 5.1 and in Table 5.3 to Table 5.6. In Table 5.6 we report the correlation between the network inferences and the ground truth, as a measure of the inference replication of the actual person trajectory regardless of systematic distance errors. We also report the RMS of the first and second derivative of the inferences, as inverse measures of the speed and acceleration smoothness of the inferences, respectively (lower numbers are associated with better smoothness) [175, p. 62]. Note that the figures for the smoothness of the ground truth itself are




(a)




(b)

Figure 5.11: Ground truth and best neural network trajectory tracking inferences of the (a) X axis and (b) Y axis for different types of neural networks: multilayer perceptron (FF), autoregressive feedforward (AR-FF), 1D convolutional (1D-Conv), and bidirectional long-short term memory (BD-LSTM).

rather high, mostly because the localization data collected from the ultrasound-based reference system has some centimeter-level jitter which we did not filter, but which seems to be filtered well by all neural networks. The 1D convolutional network appears to follow best the X and Y components of the reference trajectory. The four-layer 1D CNN inference has the lowest MSE and the best correlation with the ground truth. The 1D CNN inference is also among the smoothest, closely matching the dynamics of the actual movement of the person, as can be seen in Fig. 5.11. In fact, the RMS of the first and second derivatives of the inferred location are bested only by networks in the LSTM class, as shown in Table 5.6.

The inference of the recurrent networks, LSTMs, seems to be the smoothest,

Table 5.6: Characterization of movement inference quality in terms of correlation with the ground truth, and speed and acceleration smoothness calculated as the root mean square (RMS) of the first and second derivatives, respectively.

| | **Inference characterization** | | |
|---|---|---|---|
| **Neural network type** | **Ground truth correlation** (%) | **RMS first derivative** (m/s) | **RMS second derivative** ($m/s^2$) |
| Multilayer perceptron | 77.1 | 0.215 | 0.370 |
| Autoregressive | 83.1 | 0.211 | 0.475 |
| 1D CNN (2 layers) | 83.3 | 0.157 | 0.172 |
| 1D CNN (4 layers) | **87.5** | 0.162 | 0.187 |
| 1D CNN (6 layers) | 84.5 | 0.176 | 0.259 |
| LSTM | 85.0 | **0.129** | **0.106** |
| Bidirectional LSTM | 84.0 | 0.133 | 0.129 |
| *Ground truth* | | *0.143* | *0.333* |

closely matching the actual person movement dynamics, as shown by the low RMS of first and second derivatives of the inferred location in Table 5.1. LSTM ground truth correlation is also very good, but its MSE is higher because it does not follow well all parts of the person trajectory. We intend to investigate in future work if the discrepancy is due to the low signal-to-noise ratio of the sensor data.

Fig. 5.11 shows that the multilayer perceptron network infers a trajectory with the largest oscillations. Hence, it has the lowest correlation with the ground truth and distinctly high RMSs of the inference first and second derivatives. This can be because the network has no means to understand the physical movement dynamics because it is trained with separate points of the trajectory, which carry no dynamic information. But even in autoregressive configuration, where it is trained using segments of trajectory, the performance of the multilayer perceptron network does not improve, most likely because it lacks filtering capabilities, unlike the convolution filters of the 1D CNN or the intrinsic recurrent memory of the LSTMs. However, the multilayer perceptron networks can infer with good accuracy static positions, as shown in Table 5.1 and Fig. 5.3. Besides the noise in the sensor data, we should note that the accuracy of neural network inferences was affected both by errors of the reference system and by differences in the posture of the person. As discussed in Section 3.1.3, we measured the former to average at ±3.9 cm with peaks of ±6.4 cm in our experimental conditions. The latter can depend on head inclination (the person was wearing the mobile tag on the head) or body rotation, which can change the distance between the part of the body that is closest to the sensors for the same position of the person in the room. In fact, the human body can have complex postures and irregular shape, making it difficult to accurately define its position. Hence part of the reported neural network errors can be attributed to this

Table 5.7: Number of parameters, floating-point operations (FLOPs), and inference error [mean square (MSE) and average distance (ADE)] for the best neural networks of each type.

| **Neural network** | **Parameters** | **FLOPs** | **MSE** ($m^2$) | **ADE** (m) |
|---|---|---|---|---|
| Multilayer perceptron | 17 090 | 34 180 | 0.111 | 0.405 |
| Autoregressive | 20 674 | 40 382 | 0.079 | 0.342 |
| 1D CNN (2 layers) 32 filters, kernel size 7 | 14 530 | 22 318 | 0.078 | 0.343 |
| 1D CNN (4 layers) 8 filters, kernel size 3 | 7618 | 34 078 | **0.063** | **0.307** |
| 1D CNN (6 layers) 8 filters, kernel size 3 | 8018 | 45 838 | 0.078 | 0.328 |
| LSTM | **1378** | **16800** | 0.080 | 0.325 |
| Bidirectional LSTM | 2754 | 33 600 | 0.079 | 0.326 |

Table 5.8: Training, validation, and test errors (RMSE) for the best configuration of each neural network type trained on capacitive sensors.

| **Neural network** | **Training** (m) | **Validation** (m) | **Test** (m) |
|---|---|---|---|
| Multilayer perceptron | 0.422 | 0.433 | 0.333 |
| Autoregressive | 0.365 | 0.514 | 0.282 |
| 1D CNN (2 layers) 32 filters, kernel size 7 | 0.368 | 0.463 | 0.280 |
| 1D CNN (4 layers) 8 filters, kernel size 3 | 0.340 | 0.533 | 0.252 |
| 1D CNN (6 layers) 8 filters, kernel size 3 | 0.334 | 0.559 | 0.279 |
| LSTM | 0.341 | 0.471 | 0.283 |
| Bidirectional LSTM | 0.347 | 0.475 | 0.281 |

intractable application domain-specific uncertainty.

In Table 5.7, we show the number of parameters, processing effort (estimated as the number of floating-point operations), and the inference accuracy for the best neural network of each type. The four layer 1D convolutional has the best accuracy, but needs more parameters and higher processing effort. The single layer LSTM provides very good localization estimation, the smoothest movement tracking, and also requires the fewest parameters and lowest processing effort (both important for embedded applications). The training, validation, and test errors in terms of RMSE for the best configuration of each neural network type are reported in Table 5.8.

Table 5.9: Edge devices suitable to deploy the best NN architecture.

| Device | Processor | Memory | Frequency | Latency |
|---|---|---|---|---|
| Ambiq Apollo | Arm Cortex-M4F with FPU | 64 kB | 24 MHz | 1.42 ms |
| MAX32620FTHR | Arm Cortex-M4F with FPU | 256 kB | 96 MHz | 0.35 ms |
| EVK-NINA-B1 | Arm Cortex-M4 with FPU | 64 kB | 64 MHz | 0.53 ms |

### 5.1.7 Edge Devices for NN Deployment

There are several platforms available in the market for edge computing integrating accelerators specifically designed for AI applications. Considering the FLOPS and parameters requirements in Table 5.7, we can evaluate what platforms are suitable for NN deployment. For evaluation, we select the parameters and FLOPS of the NN architecture with the best MSE and ADE, i.e. the four-layers 1D CNN with 8 filters each and kernel size of 3. It requires 34 078 FLOPS, 7618 parameters and 650 activations. If we consider parameters and activations floating-point numbers, then we need at least 32.29 KiB of memory.

In Table 5.9, we list some devices which can be the suitable candidates for the deployment of the NN, along with their available memory and CPU frequency, and the estimated inference latency. The latter is within a few milliseconds, which is fast eough for low-energy real-time processing. Devices with dedicated NN accelerators, like MAX 78000, are currently beyond the requirements, but may be considered to support more complex NNs in future developments.

## 5.2 Neural Network Architectures and Localization Performance for IR Sensors

In this section we evaluate the effectiveness of methods devised in Sections 5.1.1, 5.1.3, 5.1.4, and 5.1.5 to reconstruct human trajectory using similar NNs but based on experimental data from a thermopile infra-red sensor. Following the similar pattern, we test three feedforward NN types: MLP, autoregressive, and 1D-CNN, and one recurrent, LSTM. The neurons use the ReLU activation function, except some LSTM gates that use the default activations [123]. We split the experimental data in 60 % sequential tuples for training, 20 % sequential tuples for validation, 20 % sequential tuples for testing. We used 50 % dropout layers where appropriate to avoid training data overfitting. We train 50 times for each hyperparameter combination with the Keras library and TensorFlow v2.2 back-end, and Adamax first order gradient-based optimization with default parameters. We also tried the Adam optimizer which had an essentially identical RMSE, within ±8 %, or ±0.006 m, of Adamax.

We evaluate (1) the inference quality by the accuracy RMSE, (2) the smoothness of the inferred trajectory by the average of the second derivative [175, p. 62], and

Table 5.10: Required parameters (memory), floating-point operations (FLOPs), train, validation, and test accuracy (RMSE), and smoothness for the best configuration of each neural network type.

| **Neural network type** | **Param.** | **FLOPs** | **Train** RMSE (m) | **Validation** RMSE (m) | **Test** RMSE (m) | **Smooth** ($m/s^2$) |
|---|---|---|---|---|---|---|
| **Multilayer perceptron** | | | | | | |
| 128 neurons, 3 layers | 35 458 | 70 149 | 0.196 | 0.203 | 0.103 | 1.329 |
| **Autoregressive** | | | | | | |
| 1 s win., 64 neur./layer, 3 layers | 13 634 | 26 885 | 0.195 | 0.197 | 0.117 | 0.990 |
| 1 s win., 256 neur./layer, 3 layers | 152 834 | 304 133 | 0.183 | 0.196 | 0.098 | 0.638 |
| **1D CNN** | | | | | | |
| 3 s win., 16 filt., 1 conv. lay., ker. 5 | 4562 | 8964 | 0.177 | 0.199 | 0.108 | 0.305 |
| 1 s win., 32 filt., 1 conv. lay., ker. 3 | **3810** | **7428** | 0.192 | 0.196 | **0.096** | 0.515 |
| **LSTM** | | | | | | |
| 3 s window, 2 layers, 64 units | 53 890 | 172 300 | 0.186 | 0.199 | 0.105 | **0.163** |
| 1 s window, 2 layers, 64 units | 53 890 | 172 300 | 0.191 | 0.195 | 0.109 | 0.765 |
| ***Ground truth*** | | | | | | |
| *smoothed over* 1 s *window* | | | | | | 0.443 |
| *smoothed over* 3 s *window* | | | | | | 0.292 |

(3) the NN computation and memory resource requirements by the total number of operations and parameters. The results, including the RMSE of training, validation, and test errors for the best configuration of each neural network type are reported in Table 5.10.

### 5.2.1 Multilayer Perceptron Neural Networks

Similar to Section 5.1.1, the NN receives one sensor tuple, 16 temperature readings on 16 input neurons, and infers the X and Y co-ordinates of the person in the room on two output neurons. For design space exploration (DSE), we vary the network depth from one to five hidden layers and the number of neurons per hidden layer from 4 to 512, in powers of two.

The best network has three hidden layers with 128 neurons each, inference accuracy RMSE 0.103 m, and smoothness 1.329 $m/s^2$, which is much higher than the ground truth smoothness (see Table 5.10).

### 5.2.2 Autoregressive Feedforward Neural Networks

Similar to Section 5.1.3, the NN receives a sliding window of inputs containing multiple sequential 16-sensor reading tuples and infers the X and Y co-ordinates of the middle tuple on two output neurons. The NN accesses both past and future samples, which can help it to learn the movement dynamics. The DSE varies the

NN depth from one to five hidden layers, from 4 to 512 neurons per hidden layer in powers of two, and window widths of 1 s and 3 s (covering 5 and 15 tuples respectively, thus changing the input layer size from 80 neurons for the 1 s window to 240 neurons for the 3 s window).

An autoregressive NN with three hidden layers, 256 neurons per layer, and 1 s input window has among the lowest inference RMSE, 0.098 m (see Table 5.10). But smaller networks, e.g., with 64 neurons per hidden layer, have also small RMSEs, of 0.117 m. Compared to MLP, the autoregressive NN significantly improves the smoothness of the inferred trajectory, to 0.638 $m/s^2$ from 1.329 $m/s^2$, thus better capturing the movement dynamics.

### 5.2.3 1D Convolutional Neural Networks

Convolutional NNs can efficiently extract relevant data patterns and are widely used in image and data time series processing e.g., accelerometer sensors. Efficient pattern recognition helps significantly reduce the computation effort compared to fully connected NNs. Similar to 5.1.4, the 1D-CNN receives a sliding window of inputs containing multiple sequential 16-sensor tuples and infers the X and Y co-ordinates of the middle tuple on two output neurons. The NN accesses both past and future samples, which can help it to learn the movement dynamics. The DSE varies the number of kernels from 2 to 64, in powers of two, the kernel size (3, 5, and 7 tuples), the number of convolution layers (1, 2, and 4), and the window width (1 s and 3 s). The hidden layers have convolution layers, an average pooling layer of size five, and a fully connected layer with 64 neurons.

A 1 s window CNN with 32 filters and kernel size of 3 tuples has the best RMSE of 0.096 m and a smoothness of 0.515 $m/s^2$, both better than the autoregressive NN and requiring only about a quarter of the resources (see Table 5.10). With a larger 3 s window, the RMSE increases slightly and the smoothness improves markedly, at the expense of more resource requirements.

### 5.2.4 Long-Short Term Memory Neural Networks

LSTMs are recurrent networks used mostly where history and context awareness can improve the inference, e.g., for handwriting and speech recognition, or translation. Similar to previous work 5.1.5, the LSTM receives a sliding window of inputs containing multiple sequential 16-sensor tuples and infers the X and Y co-ordinates of the middle tuple. In the DSE we vary the LSTM layers (1, 2, and 3), LSTM units from 2 to 64, in powers of two, and the input window width (1 s and 3 s).

The LSTM achieves by far the best smoothness, 0.163 $m/s^2$, with a good RMSE of 0.105 m using a 3 s input window (see Table 5.10), but requires 15 to 20× more resources than the 1D-CNN. With a smaller window of 1 s, the RMSE changes only slightly to 0.109 m, but the smoothness lowers significantly to 0.765 $m/s^2$ for

Table 5.11: Evaluation of the best NN models in RMSE on the data sets obtained at various times and with a different person.

| **Neural network type** | **Evening Person 2** RMSE (m) | **Evening Person 1** RMSE (m) | **Morning Person 2** RMSE (m) |
|---|---|---|---|
| **Multilayer perceptron** | | | |
| 128 neurons, 3 layers | 0.109 | 0.230 | 0.163 |
| **Autoregressive** | | | |
| 1 s win., 64 neur./layer, 3 layers | 0.120 | 0.232 | 0.208 |
| 1 s win., 256 neur./layer, 3 layers | 0.127 | 0.239 | 0.172 |
| **1D CNN** | | | |
| 3 s win., 16 filt., 1 conv. lay., ker. 5 | 0.174 | 0.254 | 0.251 |
| 1 s win., 32 filt., 1 conv. lay., ker. 3 | 0.124 | 0.221 | 0.260 |
| **LSTM** | | | |
| 3 s window, 2 layers, 64 units | 0.149 | 0.220 | 0.228 |
| 1 s window, 2 layers, 64 units | 0.179 | 0.294 | 0.278 |

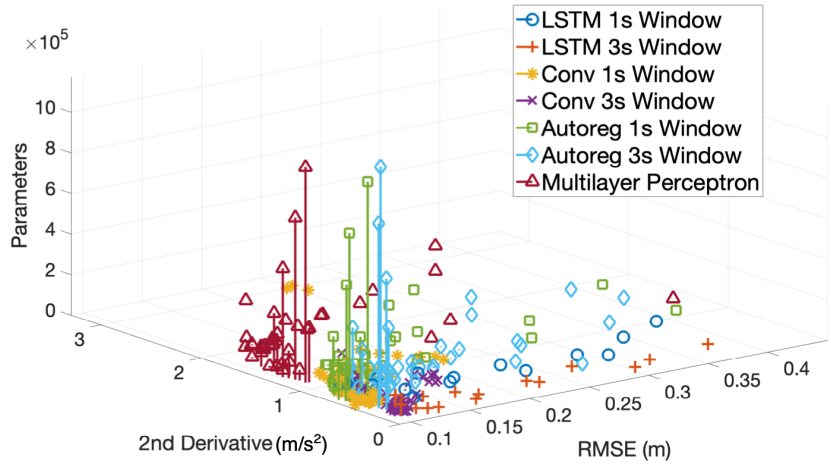
virtually the same resource requirements (being a recurrent network, the resource requirements are largely independent on the input window size).

### 5.2.5 Effects of Room Temperature Variations

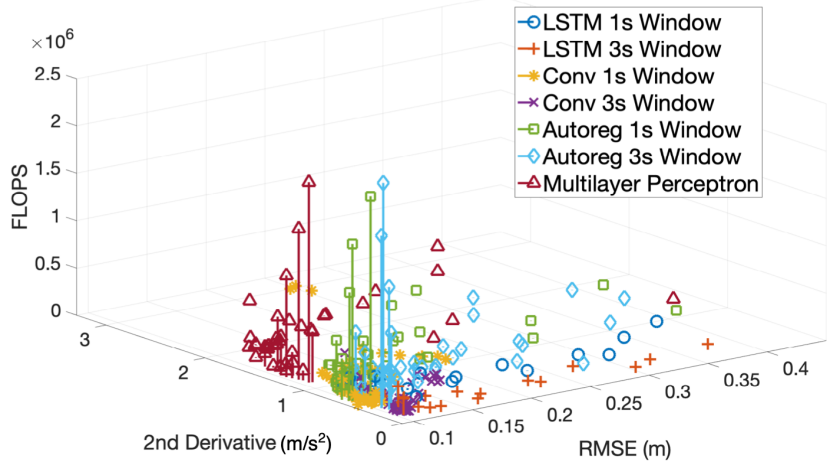
As mentioned in Section 3.2.2, we gathered two data sets in the morning and on the following day two data sets in the evening. The results in Table 5.10 were obtained after training and testing by using 60 % and 20 % of the data from the same experiment, i.e., *Morning, Person 1*. We evaluated the trained NN models on the other data sets, to see the robustness of the trained NN architectures. This means, the NN models trained on 5400 tuples of data from *Morning Person 1* were tested on 8664, 8818, and 8982 tuples of *Evening Person 2*, *Evening Person 1*, and *Morning Person 2* respectively. Table 5.11 shows the inference RMSE obtained on the best configuration of various NNs. The experiments show that minimum RMSE of 0.109 m, 0.220 m, and 0.163 m were obtained for *Evening Person 2*, *Evening Person 1*, and *Morning Person 2* respectively, which makes it suitable for a realistic localization system.

### 5.2.6 Summary

We summarize in Fig. 5.12 (further zoomed around the origin in Fig. 5.13) the dependence of a) memory requirements (parameters) and b) processing requirements (FLOPs) on the localization inference RMSE and the inference smoothness (2nd derivative) for all the NNs. Each NN allows a distinct performance-resource trade-off (note that memory and processing are closely correlated). The MLP NNs can




(a)

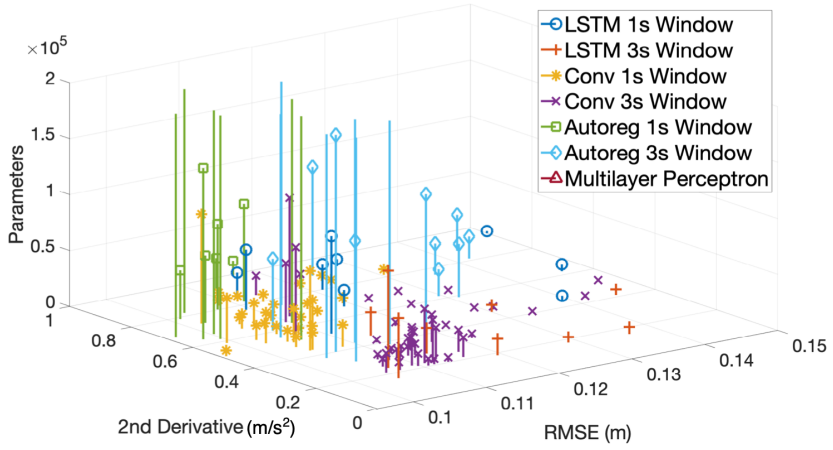



(b)

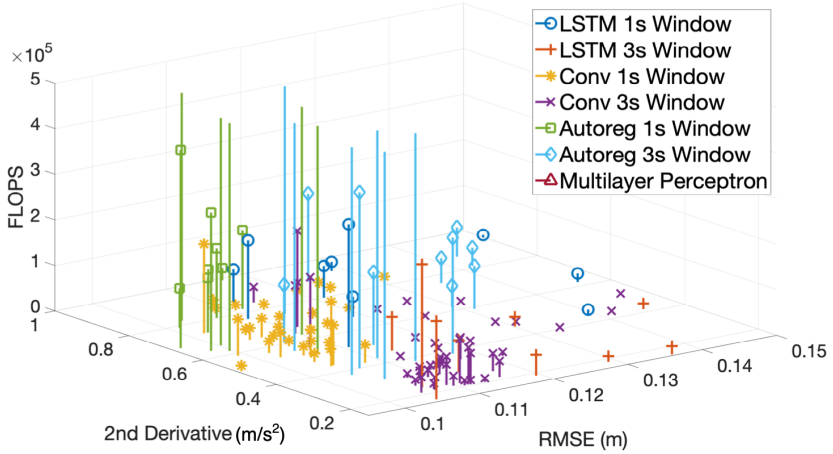
Figure 5.12: Inference localization accuracy (RMSE) and trajectory smoothness (second derivative) as a function of (a) memory (parameters) and (b) processing (FLOPS) requirements for multilayer perceptron, autoregressive feedforward, 1D convolutional, and long-short term memory (LSTM) neural networks.

have low RMSEs, but mostly poor smoothness and high resource requirements. The autoregressive NNs have better inference smoothness, especially with 3 s windows, but still high resource requirements (see Fig. 5.13). The 1D-CNN and LSTM NNs perform best. The former generally have better performance-resource trade-offs with 3 s windows.

Fig. 5.14 comparatively shows the inference of the X and Y co-ordinates of the person while moving within the space. The MLP and the autoregressive NNs seem to be the most "noisy." The LSTM looks the smoothest, but leaves some extremes uncovered, while the 1D-CNN seems a good compromise between trajectory coverage




(a)



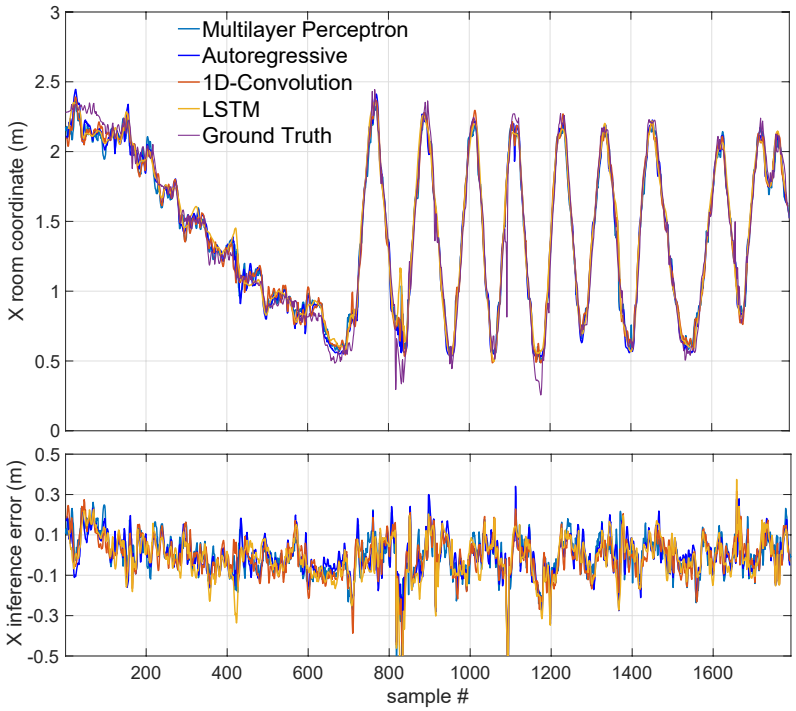

(b)

Figure 5.13: Detail of inference localization accuracy (RMSE) and trajectory smoothness (second derivative) as a function of (a) memory (parameters) and (b) processing (FLOPS) requirements for multilayer perceptron, autoregressive feedforward, 1D convolutional, and long-short term memory (LSTM) neural networks.

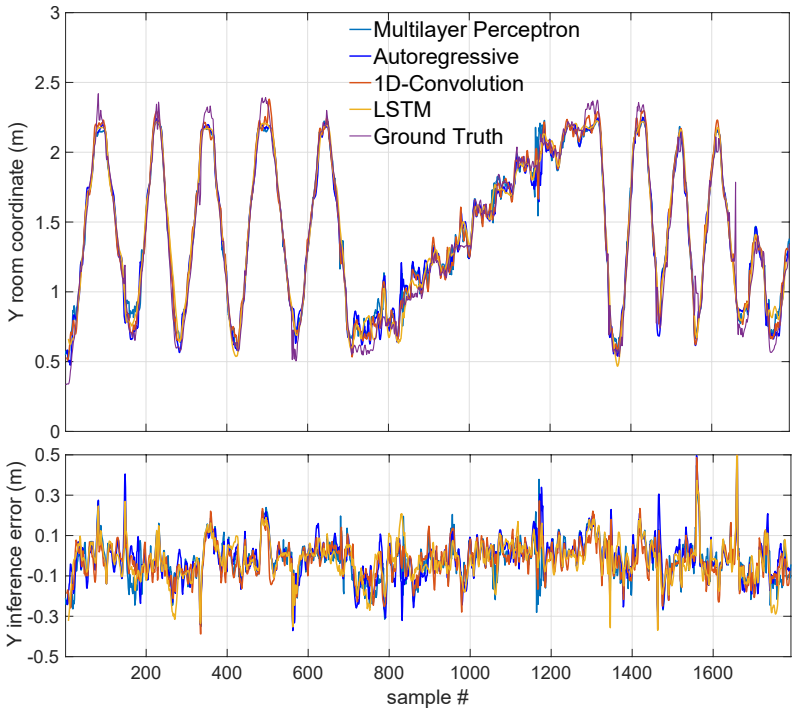
and smoothness.

Considering the above, the 1D-CNN with 3 s input window seems the best trade-off between inference performance and resource requirements for embedded implementation (see Table 5.10). Moreover, our most accurate tracking inference over an area of 3 m × 3 m using one 4 pixel × 4 pixel sensor has 0.096 m RMSE (see Table 5.10), which is sufficient for most assisted living or home automation applications. The reports in the state-of-the-art used tens of sensors to monitor a space 15 times larger with a much higher average error of 0.322 m [176], or used two higher-resolution sensors to classify the location in predefined 60 cm-spaced positions with higher mean error of 0.134 m [107], or a sensor with comparable resolution over

a quarter of our monitored space, with much higher average error of 0.246 m [160], or more expensive sensors with four times higher resolution, further enhanced with interpolation, achieving comparable localization accuracy over comparable areas, but for predefined (not arbitrary) trajectories [156, 157].




(a)




(b)

Figure 5.14: Ground truth, trajectory tracking inference and its error for the (a) X axis and (b) Y axis for the best NN configurations.

# Chapter 6

# Conclusions and Future Work

We tested various machine learning techniques for device free indoor person tracking with capacitive sensors and for a low resolution thermopile infrared sensor. For classification of a person's location inside a 3 m × 3 m room, we tested various aspects of the performance of most Weka collection ML classification algorithms for indoor person localization using capacitive sensors.

The data sets used for training and testing were collected during experiments in an uncontrolled noisy environment, at three separate times and with different body orientations, in order to acquire realistic data sets. We used these data sets with very little preprocessing to test Weka collection machine learning classification algorithms. We found that Random Forest was performing best overall, while AdaBoostM1 used on top of C4.5 requires much less time for inference at the cost of a small accuracy loss.

For reconstructing a person's trajectory, we tested the inference accuracy of several neural network types, both feedforward and recurrent, while tracking the location and movement of a person using data from four capacitive sensors placed in the middle of the "walls" of a 3 m × 3 m empty laboratory area. Sensor sensitivity was limited by noise level, and their stability was also affected by a slow but significant drift. While we used filters to reduce both the drift and the high-pitch noise, we were especially interested in how much the remaining noise affects the accuracy of the inference of person location and trajectory for various types of neural networks.

The best inference, evaluated both as mean square error and as smoothness and closeness to the actual person movement in the room, was obtained by neural networks trained on trajectory segments, processing either windows (feedforward autoregressive and 1D convolutional) or sequences (long-short term memory). The latter kind seems to capture best the movement dynamics, while the 1D convolutional network has the smallest error. Networks that consider trajectory points in isolation perform well with data collected for static positions, but have the worst trajectory inference error and do not capture the movement dynamics at all.

1D-CNN with four convolution layers has the best localization accuracy, of

0.251 m RMSE. However a single layer LSTM achieved the best inference smoothness, 0.106 $m/s^2$, and 0.283 m localization accuracy RMSE, and also needs much fewer computing resources (1378 FLOPs compared to 7618 FLOPs for 1D-CNN) and memory resources (16 800 parameters compared to 34 078 parameters for 1D-CNN). Thus, the single layer LSTM provides very good localization estimation, the smoothest movement tracking, and also requires the fewest parameters and lowest processing effort (both important for embedded applications).

It is hard to define accurately the position of the human body, especially while moving, due to its complex shapes and postures. This can explain part of the inference errors, in addition to the limited accuracy of the reference system (±3.9 cm average and ±6.4 cm max in our setting).

We note that even with these noisy sensors, the best average localization RMSE of 0.251 m is suitable for our main target application, namely assisted living of elderly persons.

We also showed that low resolution infrared thermopile sensors can also be used for low cost privacy-aware indoor person localization and movement tracking using neural networks. Network architecture and hyperparameter values greatly influence the sensor performance. We explored trade-offs between location accuracy, trajectory smoothness, computing cost and memory resources, in order to find the best compromise for embedded implementations with limited resources.

As we showed for capacitive sensors, networks that consider a sequence of sensor readings, such as the autoregressive, 1D-CNN, or LSTM, have smoother inferences that better follow the actual dynamics of the movements of a person. Among these, the recurrent networks, such as LSTMs, achieved the best inference smoothness, 0.163 $m/s^2$, and 0.105 m localization accuracy RMSE. However, the 1D-CNN with a 1 s input window has the best localization accuracy, of 0.096 m RMSE, needs much fewer computing resources (7428 FLOPs compared to 172 300 FLOPs for LSTMs) and memory resources (3810 parameters compared to 53 890 parameters for LSTMs), thus being better suited for embedded implementation.

We discussed two different approaches that can be used for human indoor localization, capacitive sensing-based and infrared-based. It is difficult to make one to one comparison of these approaches as they have different working principles and both have their advantages and disadvantages. Thermopile sensors can provide real-time localization because no filtering-related delays are involved to remove environmental noise and fewer computational resources are required to infer localization information. In terms of deployment, a single thermopile can cover the area for which multiple capacitive sensors would be required. However, infrared sensing can be problematic in areas where there are many warm objects, for example, heated floor, pipes of hot water, pets, etc. In such scenarios, the capacitive sensors can provide better localization, especially if used as a proximity sensors. For example, capacitive sensors placed on the walls will not be afflicted by a pet, since pets of human height are very rare. In the future, we can test such scenarios by fusing data from both these sensors

and compare their performance with the performance of a single sensor type.

Indoor localization is still an active area of research and many aspects still need to be investigated. For example, we plan to extend the duration of the experiments and to increase the size of the experimental room beyond $3\,\mathrm{m} \times 3\,\mathrm{m}$, and use real rooms, instead of an empty space in a lab, thus imposing much more stress on the algorithms. We also plan to look into transfer learning techniques, to ease the post-deployment training after the installation of the sensors in a new place, e.g., an apartment or an office.

The implementation of these localization systems on embedded platforms is an important aspect toward their real-world deployment. Reduced processing and memory requirements can help save power (which can enhance battery life) and improve inference time. In the future, we plan to explore how quantized ML algorithms can affect the accuracy and processing requirements.

Moreover, the localization techniques described here were limited to a single person tracking. In the future, these localization techniques can be extended to multiple persons. We also plan to fuse capacitive sensor data with infrared based thermopile data because both these have features complementary to each other. Similarly, fusion with other types of sensors can be experimented for presence, movement and distance tracking to improve the quality of the results.

# Nomenclature

**Acronyms / Abbreviations**

ADE Average Distance Error

ANN Artificial Neural Network

AOA Angle of Arrival

AP Access Point

BLE Bluetooth Low Energy

BT Bluetooth

CNN Convolutional Neural Networks

CSI Channel State Information

DBL Device Based Localization

DFL Device Free Localization

DSE Design Space Exploration

FDOA Frequency Difference of Arrival

FIR Finite Impulse Response

FLOPS Floating Point Operations per Second

GCD Greatest Common Divisor

GLONASS Global Navigation Satellite System

GMM Gaussian Mixture Model

GNSS Global Navigation Satellite System

GPS Global Positioning System

HMM Hidden Markov Model

IIR Infinite Impulse Response

IoT Internet of Things

ISM Industrial, Scientific, and Medical

LED Light Emitting Diode

LPF Low-Pass Filter

LSTM Long-Short Term Memory

MF Median Filter

MLP Multi-Layer Perceptron

MSE Mean Square Error

NN Neural Networks

ReLU Rectified Linear Unit

RFID Radio-frequency Identification

RSS Received Signal Strength

SVC Support Vector Clustering

SVM Support Vector Machine

TDOA Time Difference of Arrival

TOA Time of Arrival

TOF Time of Flight

UWB Ultra-Wide Band

# Bibliography


[1] Mark Hall et al. "The WEKA data mining software: an update". In: *ACM SIGKDD explorations newsletter* 11.1 (2009), pp. 10–18.

[2] Tero Kivimäki et al. "A review on device-free passive indoor positioning methods". In: *International Journal of Smart Home* 8.1 (2014), pp. 71–94.

[3] United Nations. *World Population Ageing 2019 Highlights*. 2019, p. 41. DOI: https://doi.org/https://doi.org/10.18356/9df3caed-en.

[4] Boštjan Kaluža et al. "An agent-based approach to care in independent living". In: *Ambient intelligence*. Springer, 2010, pp. 177–186.

[5] Andreas Braun, Henning Heggen, and Reiner Wichert. "CapFloor–a flexible capacitive indoor localization system". In: *Evaluating AAL Systems Through Competitive Benchmarking. Indoor Localization and Tracking*. Springer, 2011, pp. 26–35.

[6] Faheem Zafari, Athanasios Gkelias, and Kin K Leung. "A survey of indoor localization systems and technologies". In: *IEEE Communications Surveys & Tutorials* 21.3 (2019), pp. 2568–2599.

[7] Deepak Vasisht, Swarun Kumar, and Dina Katabi. "Decimeter-level localization with a single WiFi access point". In: *13th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 16)*. 2016, pp. 165–178.

[8] Elahe Soltanaghaei, Avinash Kalyanaraman, and Kamin Whitehouse. "Multipath triangulation: Decimeter-level WiFi localization and orientation with a single unaided receiver". In: *Proceedings of the 16th annual international conference on mobile systems, applications, and services*. 2018, pp. 376–388.

[9] Tatsuya Ishihara et al. "Inference machines for supervised Bluetooth localization". In: *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2017, pp. 5950–5954.

[10] Paolo Baronti et al. "Wireless sensor networks: A survey on the state of the art and the 802.15.4 and ZigBee standards". In: *Computer communications* 30.7 (2007), pp. 1655–1695.

[11] Mohammed El-Absi et al. "High-accuracy indoor localization based on chipless RFID systems at THz band". In: *IEEE Access* 6 (2018), pp. 54355–54368.

[12] Yuan Xu et al. “UWB-based indoor human localization with time-delayed data using EFIR filtering”. In: *IEEE Access* 5 (2017), pp. 16676–16683.

[13] Ye-Sheng Kuo et al. “Luxapose: Indoor positioning with mobile phones and visible light”. In: *Proceedings of the 20th annual international conference on Mobile computing and networking*. 2014, pp. 447–458.

[14] Wenchao Huang et al. “Swadloon: Direction finding and indoor localization using acoustic signal by shaking smartphones”. In: *IEEE Trans. on Mobile Computing* 14.10 (2014), pp. 2145–2157.

[15] Faheem Ijaz et al. “Indoor positioning: A review of indoor ultrasonic positioning systems”. In: *2013 15th International Conference on Advanced Communications Technology (ICACT)*. IEEE. 2013, pp. 1146–1150.

[16] Biying Fu et al. “Sensing Technology for Human Activity Recognition: a Comprehensive Survey”. In: *IEEE Access* (2020).

[17] Lee Middleton et al. “A floor sensor system for gait recognition”. In: *Fourth IEEE Workshop on Automatic Identification Advanced Technologies*. IEEE. 2005, pp. 171–176.

[18] Joseph Paradiso et al. “The magic carpet: physical sensing for immersive environments”. In: *CHI’97 Extended Abstracts on Human Factors in Computing Systems*. 1997, pp. 277–278.

[19] Xin Zhou, Qi Hao, and Hu Fei. “1-bit walker recognition with distributed binary pyroelectric sensors”. In: *2010 IEEE Conf. on Multisensor Fusion and Integration*. IEEE. 2010, pp. 168–173.

[20] J-M Valin et al. “Localization of simultaneous moving sound sources for mobile robot using a frequency-domain steered beamformer approach”. In: *IEEE Int. Conf. on Robotics and Automation*. Vol. 1. IEEE. 2004, pp. 1033–1038.

[21] Sverre Holm and Carl-Inge C Nilsen. “Robust ultrasonic indoor positioning using transmitter arrays”. In: *2010 Int. Conf. on Indoor Positioning and Indoor Navigation*. IEEE. 2010, pp. 1–5.

[22] Yoshifumi Nishida et al. “Minimally privacy-violative system for locating human by ultrasonic radar embedded on ceiling”. In: *2004 IEEE Int. Conf. on Systems, Man and Cybernetics (IEEE Cat. No. 04CH37583)*. Vol. 2. IEEE. 2004, pp. 1549–1554.

[23] Shwetak N Patel, Matthew S Reynolds, and Gregory D Abowd. “Detecting human movement by differential air pressure sensing in HVAC system ductwork: An exploration in infrastructure mediated sensing”. In: *Int. Conf. on Pervasive Computing*. Springer. 2008, pp. 1–18.

[24] Shwetak N Patel et al. “At the flick of a switch: Detecting and classifying unique electrical events on the residential power line”. In: *Int. Conf. on Ubiquitous Computing*. Springer. 2007, pp. 271–288.

[25] James Fogarty, Carolyn Au, and Scott E Hudson. “Sensing from the basement: a feasibility study of unobtrusive and low-cost home activity recognition”. In: *Proceedings of the 19th ACM symposium on User interface software and technology*. 2006, pp. 91–100.

[26] Pan Hu et al. “Pharos: Enable physical analytics through visible light based indoor localization”. In: *Proceedings of the Twelfth ACM Workshop on Hot Topics in Networks*. 2013, pp. 1–7.

[27] Irvan B Arief-Ang, Flora D Salim, and Margaret Hamilton. “CD-HOC: indoor human occupancy counting using carbon dioxide sensor data”. In: *arXiv preprint arXiv:1706.05286* (2017).

[28] Mari Zakrzewski, Harri Raittinen, and Jukka Vanhala. “Comparison of center estimation algorithms for heart and respiration monitoring with microwave Doppler radar”. In: *IEEE Sensors J.* 12.3 (2011), pp. 627–634.

[29] Daniel Hauschildt and Nicolaj Kirchhof. “Improving indoor position estimation by combining active TDOA ultrasound and passive thermal infrared localization”. In: *2011 8th Workshop on Positioning, Navigation and Communication*. IEEE. 2011, pp. 94–99.

[30] Nasrullah Pirzada et al. “Device-free localization technique for indoor detection and tracking of human body: A survey”. In: *Procedia-Social and Behavioral Sciences* 129.422-429 (2014), 2nd.

[31] Jose Rivera-Rubio, Ioannis Alexiou, and Anil A Bharath. “Appearance-based indoor localization: A comparison of patch descriptor performance”. In: *Pattern Recognition Letters* 66 (2015), pp. 109–117.

[32] Guoyu Lu et al. “Where am i in the dark: Exploring active transfer learning on the use of indoor localization based on thermal imaging”. In: *Neurocomputing* 173 (2016), pp. 83–92.

[33] Sinan Gezici et al. “Localization via ultra-wideband radios: a look at positioning aspects for future sensor networks”. In: *IEEE signal processing magazine* 22.4 (2005), pp. 70–84.

[34] Alessandro Bay et al. “Block-sparsity-based localization in wireless sensor networks”. In: *EURASIP Journal on Wireless Communications and Networking* 2015.1 (2015), p. 1.

[35] Moustafa Seifeldin et al. “Nuzzer: A large-scale device-free passive localization system for wireless environments”. In: *Mobile Computing, IEEE Transactions on* 12.7 (2013), pp. 1321–1334.

[36] Wenjie Ruan et al. "TagTrack: device-free localization and tracking using passive RFID tags". In: *Proceedings of the 11th International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services*. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering). 2014, pp. 80–89.

[37] Luca Mainetti, Luigi Patrono, and Ilaria Sergi. "A survey on indoor positioning systems". In: *Software, Telecommunications and Computer Networks (SoftCOM), 2014 22nd International Conference on*. IEEE. 2014, pp. 111–120.

[38] Bożena Smagowska and Małgorzata Pawlaczyk-Łuszczyńska. "Effects of ultrasonic noise on the human body—a bibliographic review". In: *International Journal of Occupational Safety and Ergonomics* 19.2 (2013), pp. 195–202.

[39] Chenshu Wu et al. "Non-invasive detection of moving and stationary human with WiFi". In: *IEEE Journal on Selected Areas in Communications* 33.11 (2015), pp. 2329–2342.

[40] Manikanta Kotaru et al. "Spotfi: Decimeter level localization using wifi". In: *ACM SIGCOMM Computer Communication Review*. Vol. 45. 4. ACM. 2015, pp. 269–282.

[41] Junyi Ma et al. "A Survey on Wi-Fi Based Contactless Activity Recognition". In: *Ubiquitous Intelligence & Computing, Advanced and Trusted Computing, Scalable Computing and Communications, Cloud and Big Data Computing, Internet of People, and Smart World Congress*. IEEE. 2016, pp. 1086–1091.

[42] *Lively Activity Monitoring System*. http://www.mylively.com/. Accessed on December 1, 2017.

[43] Jaeseok Yun and Sang-Shin Lee. "Human movement detection and identification using pyroelectric infrared sensors". In: *Sensors (Basel)* 14.5 (2014), pp. 8057–81. DOI: 10.3390/s140508057.

[44] Changqiang Jing et al. "Performance Evaluation of an Indoor Positioning Scheme Using Infrared Motion Sensors". In: *Information* 5.4 (2014), pp. 548–557.

[45] *Canary Monitoring System*. https://www.canarycare.co.uk/. Accessed on December 1, 2017.

[46] *Maricare Localization System*. http://www.maricare.com/elea/. Accessed on December 1, 2017.

[47] Joshua Smith et al. "Electric field sensing for graphical interfaces". In: *Computer Graphics and Applications, IEEE* 18.3 (1998), pp. 54–60.

[48] Xinyao Tang and Soumyajit Mandal. "Indoor occupancy awareness and localization using passive electric field sensing". In: *IEEE Trans. on Instrumentation and Measurement* 68.11 (2019), pp. 4535–4549.

[49] Sousuke Nakamura et al. "Electric-field resonance coupling between human and transmitter for human position estimation system". In: *ICCAS 2010*. IEEE. 2010, pp. 109–114.

[50] Antti Ropponen, Matti Linnavuo, Raimo Sepponen, et al. "LF indoor location and identification system". In: *Int. J. on Smart Sensing and Intelligent Systems* 2.1 (2009), pp. 94–117.

[51] Alireza Ramezani Akhmareh et al. "A Tagless Indoor Localization System Based on Capacitive Sensing Technology". In: *Sensors* 16.9 (2016), p. 1448. ISSN: 1424-8220. DOI: 10.3390/s16091448. URL: http://www.mdpi.com/1424-8220/16/9/1448.

[52] Javed Iqbal et al. "A contactless sensor for human body identification using RF absorption signatures". In: *Sensors Applications Symposium (SAS), 2017 IEEE*. IEEE. 2017, pp. 1–6.

[53] Thomas G Zimmerman et al. "Applying electric field sensing to human-computer interfaces". In: *Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM Press/Addison-Wesley Publishing Co. 1995, pp. 280–287.

[54] Andreas Braun et al. "Capacitive proximity sensing in smart environments". In: *J. of Ambient Intelligence and Smart Environments* 7.4 (2015), pp. 483–510.

[55] Tobias Grosse-Puppendahl et al. "Finding Common Ground: A Survey of Capacitive Sensing in Human-Computer Interaction". In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI'17)*. ACM. 2017.

[56] Javed Iqbal et al. "A contactless sensor for human body identification using RF absorption signatures". In: *2017 IEEE Sensors Applications Symposium (SAS)*. IEEE. 2017, pp. 1–6.

[57] J. Iqbal et al. "Capacitive Sensor for Tagless Remote Human Identification Using Body Frequency Absorption Signatures". In: *IEEE Transactions on Instrumentation and Measurement* 67.4 (Apr. 2018), pp. 789–797. ISSN: 0018-9456. DOI: 10.1109/TIM.2017.2789078.

[58] Atika Arshad et al. "An activity monitoring system for senior citizens living independently using capacitive sensing technique". In: *2016 IEEE Int. Instrumentation and Meas. Tech. Conf. Proc.* IEEE. 2016, pp. 1–6.

[59] Raphael Wimmer et al. "A capacitive sensing toolkit for pervasive activity detection and recognition". In: *Pervasive Computing and Communications, 2007. PerCom'07. Fifth Annual IEEE International Conference on*. IEEE. 2007, pp. 171–180.

[60] Arthur L Samuel. "Some studies in machine learning using the game of checkers". In: *IBM Journal of research and development* 44.1.2 (2000), pp. 206–226.

[61] "Supervised Learning". In: *Encyclopedia of Machine Learning*. Ed. by Claude Sammut and Geoffrey I. Webb. Boston, MA: Springer US, 2010, pp. 941–941. ISBN: 978-0-387-30164-8. DOI: 10.1007/978-0-387-30164-8_803. URL: https://doi.org/10.1007/978-0-387-30164-8_803.

[62] Peter Stone. "Reinforcement Learning". In: *Encyclopedia of Machine Learning*. Ed. by Claude Sammut and Geoffrey I. Webb. Boston, MA: Springer US, 2010, pp. 849–851. ISBN: 978-0-387-30164-8. DOI: 10.1007/978-0-387-30164-8_714. URL: https://doi.org/10.1007/978-0-387-30164-8_714.

[63] Scott Gleason, Demoz Gebre-Egziabher, and Demoz Gebre Egziabher. "GNSS applications and methods". In: (2009).

[64] George Dedes and Andrew G Dempster. "Indoor GPS positioning-challenges and opportunities". In: *VTC-2005-Fall. 2005 IEEE 62nd Vehicular Technology Conference, 2005.* Vol. 1. Citeseer. 2005, pp. 412–415.

[65] Fen Liu et al. "Survey on WiFi-based indoor positioning techniques". In: *IET Communications* 14.9 (2020), pp. 1372–1383.

[66] Fadel Adib, Zachary Kabelac, and Dina Katabi. "Multi-person localization via {RF} body reflections". In: *12th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 15)*. 2015, pp. 279–292.

[67] Hui Liu et al. "Survey of wireless indoor positioning techniques and systems". In: *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* 37.6 (2007), pp. 1067–1080.

[68] Hui Liu et al. "Survey of wireless indoor positioning techniques and systems". In: *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* 37.6 (2007), pp. 1067–1080.

[69] *Decawave. REAL TIME LOCATION SYSTEMS An Introduction.* https://www.decawave.com/sites/default/files/resources/aps003_dw1000_rtls_introduction.pdf. Accessed on December 28, 2020.

[70] Manni Liu et al. "Indoor acoustic localization: a survey". In: *Human-centric Computing and Information Sciences* 10.1 (2020), p. 2.

[71] Janos Sallai et al. "Acoustic Ranging in Resource-Constrained Sensor Networks." In: *International Conference on Wireless Networks*. 2004, p. 467.

[72] Kaikai Liu, Xinxin Liu, and Xiaolin Li. "Guoguo: Enabling fine-grained indoor localization via smartphone". In: *Proceeding of the 11th annual international conference on Mobile systems, applications, and services*. 2013, pp. 235–248.

[73] Wenchao Huang et al. “Swadloon: Direction finding and indoor localization using acoustic signal by shaking smartphones”. In: *IEEE Transactions on Mobile Computing* 14.10 (2014), pp. 2145–2157.

[74] Ye-Sheng Kuo et al. “Luxapose: Indoor positioning with mobile phones and visible light”. In: *Proceedings of the 20th annual international conference on Mobile computing and networking*. 2014, pp. 447–458.

[75] Jean Armstrong, Y Ahmet Sekercioglu, and Adrian Neild. “Visible light positioning: a roadmap for international standardization”. In: *IEEE Communications Magazine* 51.12 (2013), pp. 68–73.

[76] Elena Di Lascio et al. “LocaLight-A Battery-free Passive Localization System Using Visible Light”. In: *2016 15th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*. IEEE. 2016, pp. 1–2.

[77] Liqun Li et al. “Epsilon: A visible light based positioning system”. In: *11th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 14)*. 2014, pp. 331–343.

[78] Chi Zhang and Xinyu Zhang. “LiTell: Robust indoor localization using unmodified light fixtures”. In: *Proceedings of the 22nd Annual International Conference on Mobile Computing and Networking*. 2016, pp. 230–242.

[79] Soo-Yong Jung, Swook Hann, and Chang-Soo Park. “TDOA-based optical wireless indoor localization using LED ceiling lamps”. In: *IEEE Transactions on Consumer Electronics* 57.4 (2011), pp. 1592–1597.

[80] *Starter Set HW v4.9*. en-US. URL: https://marvelmind.com/product/starter-set-hw-v4-9/ (visited on 11/12/2019).

[81] Faheem Ijaz et al. “Indoor positioning: A review of indoor ultrasonic positioning systems”. In: *2013 15th International Conference on Advanced Communications Technology (ICACT)*. IEEE. 2013, pp. 1146–1150.

[82] Andy Ward, Alan Jones, and Andy Hopper. “A new location technique for the active office”. In: *IEEE Personal communications* 4.5 (1997), pp. 42–47.

[83] Robert K Harle and Andy Hopper. “Deploying and evaluating a location-aware system”. In: *Proceedings of the 3rd international conference on Mobile systems, applications, and services*. 2005, pp. 219–232.

[84] Nissanka B Priyantha, Anit Chakraborty, and Hari Balakrishnan. “The cricket location-support system”. In: *Proceedings of the 6th annual international conference on Mobile computing and networking*. 2000, pp. 32–43.

[85] Matti Siekkinen et al. “How low energy is bluetooth low energy? comparative measurements with zigbee/802.15. 4”. In: *2012 IEEE wireless communications and networking conference workshops (WCNCW)*. IEEE. 2012, pp. 232–237.

[86] Roshan Ayyalasomayajula, Deepak Vasisht, and Dinesh Bharadia. "BLoc: CSI-based accurate localization for BLE tags". In: *Proceedings of the 14th International Conference on emerging Networking EXperiments and Technologies*. 2018, pp. 126–138.

[87] Bashima Islam et al. "Rethinking ranging of unmodified BLE peripherals in smart city infrastructure". In: *Proceedings of the 9th ACM Multimedia Systems Conference*. 2018, pp. 339–350.

[88] Fakhrul Alam, Nathaniel Faulkner, and Baden Parr. "Device Free Localization: A Review of Non-RF Techniques for Unobtrusive Indoor Positioning". In: *IEEE Internet of Things Journal* (2020).

[89] Paramvir Bahl and Venkata N Padmanabhan. "RADAR: An in-building RF-based user location and tracking system". In: *Proceedings IEEE INFOCOM 2000. Conference on computer communications. Nineteenth annual joint conference of the IEEE computer and communications societies (Cat. No. 00CH37064)*. Vol. 2. Ieee. 2000, pp. 775–784.

[90] Ismail Güvenc et al. "Enhancements to RSS based indoor tracking systems using Kalman filters". In: *International Signal Processing Conference (ISPC) and Global Signal Processing Expo (GSPx)*. 2003, pp. 91–102.

[91] Moustafa Youssef and Ashok Agrawala. "The Horus WLAN location determination system". In: *Proceedings of the 3rd international conference on Mobile systems, applications, and services*. 2005, pp. 205–218.

[92] Swarun Kumar et al. "Accurate indoor localization with zero start-up cost". In: *Proceedings of the 20th annual international conference on Mobile computing and networking*. 2014, pp. 483–494.

[93] Jie Xiong and Kyle Jamieson. "Arraytrack: A fine-grained indoor location system". In: *10th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 13)*. 2013, pp. 71–84.

[94] Jon Gjengset et al. "Phaser: Enabling phased array signal processing on commodity WiFi access points". In: *Proceedings of the 20th annual international conference on Mobile computing and networking*. 2014, pp. 153–164.

[95] Antonio Ramón Jiménez Ruiz and Fernando Seco Granja. "Comparing ubisense, bespoon, and decawave uwb location systems: Indoor performance analysis". In: *IEEE Transactions on instrumentation and Measurement* 66.8 (2017), pp. 2106–2117.

[96] *Ubisense*. https://ubisense.net/en. Accessed on March 11, 2021.

[97] *BeSpoon*. https://bespoon.xyz. Accessed on March 11, 2021.

[98] *Decawave*. https://www.decawave.com. Accessed on March 11, 2021.

[99] Kaikai Liu, Xinxin Liu, and Xiaolin Li. "Guoguo: Enabling fine-grained indoor localization via smartphone". In: *Proceeding of the 11th annual international conference on Mobile systems, applications, and services*. 2013, pp. 235–248.

[100] Atri Mandal et al. "Beep: 3D indoor positioning using audible sound". In: *Second IEEE Consumer Communications and Networking Conference, 2005. CCNC. 2005*. IEEE. 2005, pp. 348–353.

[101] Wenchao Huang et al. "Walkielokie: Sensing relative positions of surrounding presenters by acoustic signals". In: *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. 2016, pp. 439–450.

[102] Chi Zhang, Shipei Zhou, and Xinyu Zhang. "Visible light localization using incumbent light fixtures: Demo abstract". In: *Proceedings of the 14th ACM Conference on Embedded Network Sensor Systems CD-ROM*. 2016, pp. 304–305.

[103] Faheem Zafari and Ioannis Papapanagiotou. "Enhancing ibeacon based micro-location with particle filtering". In: *2015 IEEE Global Communications Conference (GLOBECOM)*. IEEE. 2015, pp. 1–7.

[104] Christian Kowalski et al. "Multi Low-resolution Infrared Sensor Setup for Privacy-preserving Unobtrusive Indoor Localization." In: *ICT4AWE*. 2019, pp. 183–188.

[105] Chandrayee Basu and Anthony Rowe. "Tracking motion and proxemics using thermal-sensor array". In: *arXiv preprint arXiv:1511.08166* (2015).

[106] Jürgen Kemper and Daniel Hauschildt. "Passive infrared localization with a probability hypothesis density filter". In: *2010 7th Workshop on Positioning, Navigation and Communication*. IEEE. 2010, pp. 68–76.

[107] Wei-Han Chen and Hsi-Pin Ma. "A fall detection system based on infrared array sensors with tracking capability for the elderly at home". In: *2015 17th International Conference on E-health Networking, Application & Services (HealthCom)*. IEEE. 2015, pp. 428–434.

[108] Miika Valtonen, Jaakko Maentausta, and Jukka Vanhala. "Tiletrack: Capacitive human tracking using floor tiles". In: *Pervasive Computing and Communications, 2009. PerCom 2009. IEEE International Conference on*. IEEE. 2009, pp. 1–10.

[109] Nathaniel Faulkner et al. "CapLoc: Capacitive Sensing Floor for Device-Free Localization and Fall Detection". In: *IEEE Access* 8 (2020), pp. 187353–187364.

[110] Muhammad Awais Shafique and Eiji Hato. "Travel Mode Detection with Varying Smartphone Data Collection Frequencies". In: *Sensors* 16.5 (2016), p. 716.

[111] Leon Stenneth et al. "Transportation mode detection using mobile phones and GIS information". In: *Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. ACM. 2011, pp. 54–63.

[112] Leif E Peterson. "K-nearest neighbor". In: *Scholarpedia* 4.2 (2009), p. 1883.

[113] Asa Ben-Hur et al. "Support vector clustering". In: *Journal of machine learning research* 2.Dec (2001), pp. 125–137.

[114] Leo Breiman. "Random forests". In: *Machine learning* 45.1 (2001), pp. 5–32.

[115] Timo Sztyler and Heiner Stuckenschmidt. "On-body localization of wearable devices: An investigation of position-aware activity recognition". In: *2016 IEEE International Conference on Pervasive Computing and Communications (PerCom)*. IEEE. 2016, pp. 1–9.

[116] Maogui Hu et al. "Refining Time-Activity Classification of Human Subjects Using the Global Positioning System". In: *PLoS One* 11.2 (2016), e0148875. DOI: 10.1371/journal.pone.0148875.

[117] Jerome Friedman, Trevor Hastie, Robert Tibshirani, et al. "Additive logistic regression: a statistical view of boosting (with discussion and a rejoinder by the authors)". In: *The annals of statistics* 28.2 (2000), pp. 337–407.

[118] Guangya Zhang and Baishan Fang. "LogitBoost classifier for discriminating thermophilic and mesophilic proteins". In: *Journal of biotechnology* 127.3 (2007), pp. 417–424.

[119] Jindong Wang et al. "Deep learning for sensor-based activity recognition: A survey". In: *Pattern Recognition Letters* 119 (2019), pp. 3–11.

[120] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. "Deep sparse rectifier neural networks". In: *Proceedings of the fourteenth international conference on artificial intelligence and statistics*. 2011, pp. 315–323.

[121] Jun Han and Claudio Moraga. "The influence of the sigmoid function parameters on the speed of backpropagation learning". In: *International Workshop on Artificial Neural Networks*. Springer. 1995, pp. 195–201.

[122] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. "Deep learning". In: *nature* 521.7553 (2015), pp. 436–444.

[123] Sepp Hochreiter and Jürgen Schmidhuber. "Long short-term memory". In: *Neural computation* 9.8 (1997), pp. 1735–1780.

[124] Klaus Greff et al. "LSTM: A search space odyssey". In: *IEEE transactions on neural networks and learning systems* 28.10 (2016), pp. 2222–2232.

[125] Xuyu Wang, Zhitao Yu, and Shiwen Mao. "DeepML: Deep LSTM for indoor localization with smartphone magnetic and light sensors". In: *2018 IEEE Int. Conf. on Communications*. IEEE. 2018, pp. 1–6.

[126] Zhenghua Chen et al. “WiFi Fingerprinting Indoor Localization Using Local Feature-Based Deep LSTM”. In: *IEEE Systems J.* (2019).

[127] Miika Valtonen and Jukka Vanhala. “Human tracking using electric fields”. In: *Pervasive Computing and Communications, 2009. PerCom 2009. IEEE International Conference on*. IEEE. 2009, pp. 1–3.

[128] Miika Valtonen et al. “Unobtrusive human height and posture recognition with a capacitive sensor”. In: *Journal of Ambient Intelligence and Smart Environments* 3.4 (2011), pp. 305–332.

[129] Miguel Sousa et al. “Human tracking and identification using a sensitive floor and wearable accelerometers”. In: *Pervasive Computing and Communications (PerCom), 2013 IEEE International Conference on*. IEEE. 2013, pp. 166–171.

[130] Nan-Wei Gong, Steve Hodges, and Joseph A Paradiso. “Leveraging conductive inkjet technology to build a scalable and versatile surface for ubiquitous sensing to build a scalable and versatile surface for ubiquitous sensing”. In: *Proceedings of the 13th international conference on Ubiquitous computing*. ACM. 2011, pp. 45–54.

[131] Tero Kivimaki et al. “Reliability of the TileTrack capacitive user tracking system in smart home environment”. In: *Telecommunications (ICT), 2013 20th International Conference on*. IEEE. 2013, pp. 1–5.

[132] C Lauterbach and A Steinhage. “SensFloor®-A Large-area Sensor System Based on Printed Textiles Printed Electronics”. In: *Ambient Assisted Living Congress. VDE Verlag*. 2009.

[133] Miika Valtonen, Tero Kivimäki, and Jukka Vanhala. “Capacitive 3D user tracking with a mobile demonstration platform”. In: *Proceeding of the 16th International Academic MindTrek Conference*. ACM. 2012, pp. 61–63.

[134] Biying Fu et al. “Indoor Localization Based on Passive Electric Field Sensing”. In: *European Conference on Ambient Intelligence*. Springer. 2017, pp. 64–79.

[135] Gabe Cohn et al. “Humantenna: using the body as an antenna for real-time whole-body interaction”. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM. 2012, pp. 1901–1910.

[136] Gabe Cohn et al. “Your noise is my command: sensing gestures using the body as an antenna”. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM. 2011, pp. 791–800.

[137] Adiyan Mujibiya and Jun Rekimoto. “Mirage: exploring interaction modalities using off-body static electric field sensing”. In: *Proceedings of the 26th annual ACM symposium on User interface software and technology*. ACM. 2013, pp. 211–220.

[138] Takeshi Togura et al. “Long-Range Human-Body-Sensing Modules with Capacitive Sensor”. In: *Fujikura Technical Review* (2009).

[139] Raphael Wimmer et al. “Thracker-using capacitive sensing for gesture recognition”. In: *26th IEEE International Conference on Distributed Computing Systems Workshops (ICDCSW'06)*. IEEE. 2006, pp. 64–64.

[140] Raphael Wimmer. “Capacitive sensors for whole body interaction”. In: *Whole Body Interaction*. Springer, 2011, pp. 121–133.

[141] Heather Knight, Jae-Kyu Lee, and Hongshen Ma. “Chair alarm for patient fall prevention based on gesture recognition and interactivity”. In: *Engineering in Medicine and Biology Society, 2008. EMBS 2008. 30th Annual International Conference of the IEEE*. IEEE. 2008, pp. 3698–3701.

[142] Maxim Djakow, Andreas Braun, and Alexander Marinc. “MoviBed-sleep analysis using capacitive sensors”. In: *International Conference on Universal Access in Human-Computer Interaction*. Springer. 2014, pp. 171–181.

[143] Marian Haescher et al. “Capwalk: A capacitive recognition of walking-based activities as a wearable assistive technology”. In: *Proceedings of the 8th ACM International Conference on PErvasive Technologies Related to Assistive Environments*. ACM. 2015, p. 35.

[144] Tobias Alexander Große-Puppendahl, Alexander Marinc, and Andreas Braun. “Classification of user postures with capacitive proximity sensors in AAL-Environments”. In: *International Joint Conference on Ambient Intelligence*. Springer. 2011, pp. 314–323.

[145] William Buller and Brian Wilson. “Measuring the capacitance of electrical wiring and humans for proximity sensing with existing electrical infrastructure”. In: *Electro/information Technology, 2006 IEEE Int. Conf. on*. IEEE. 2006, pp. 93–96.

[146] H Prance et al. “Position and movement sensing at metre standoff distances using ambient electric field”. In: *Measurement Science and Technology* 23.11 (2012), p. 115101.

[147] R MacLachlan. “Spread Spectrum Capacitive Proximity Sensor”. In: *Human Condition Wiki* (2004).

[148] Tobias Grosse-Puppendahl et al. “OpenCapSense: A rapid prototyping toolkit for pervasive interaction using capacitive sensing”. In: *Pervasive Computing and Communications (PerCom), 2013 IEEE International Conference on*. IEEE. 2013, pp. 152–159.

[149] Dehui Xu et al. “MEMS-based thermoelectric infrared sensors: A review”. In: *Frontiers of Mechanical Engineering* 12.4 (2017), pp. 557–566.

[150] S. Chiu et al. “A Convolutional Neural Networks Approach with Infrared Array Sensor for Bed-Exit Detection”. In: *2018 International Conference on System Science and Engineering (ICSSE)*. 2018, pp. 1–6.

[151] Peter Hevesi et al. “Monitoring household activities and user location with a cheap, unobtrusive thermal sensor array”. In: *Proceedings of the 2014 ACM international joint conference on pervasive and ubiquitous computing*. 2014, pp. 141–145.

[152] Yordanka Karayaneva et al. “Use of low-resolution infrared pixel array for passive human motion movement and recognition”. In: *Proceedings of the 32nd International BCS Human Computer Interaction Conference 32*. 2018, pp. 1–2.

[153] Hock M Ng. “Human localization and activity detection using thermopile sensors”. In: *2013 ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*. IEEE. 2013, pp. 337–338.

[154] Xipeng Zhang, Hiroaki Seki, and Masatoshi Hikizu. “Detection of human position and motion by thermopile infrared sensor”. In: *International Journal of Automation Technology* 9.5 (2015), pp. 580–587.

[155] Nanhao Gu, Bo Yang, and Tianfu Li. “High-resolution Thermopile Array Sensor-based System for Human Detection and Tracking in Indoor Environment”. In: *2020 15th IEEE Conference on Industrial Electronics and Applications (ICIEA)*. IEEE. 2020, pp. 1926–1931.

[156] Dongning Qu, Bo Yang, and Nanhao Gu. “Indoor multiple human targets localization and tracking using thermopile sensor”. In: *Infrared Physics & Technology* 97 (2019), pp. 349–359.

[157] Akshaya D Shetty, B Shubha, K Suryanarayana, et al. “Detection and tracking of a human using the infrared thermopile array sensor—“Grid-EYE””. In: *2017 International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICICT)*. IEEE. 2017, pp. 1490–1495.

[158] Libo Wu and Ya Wang. “Compressive Sensing Based Indoor Occupancy Positioning Using a Single Thermopile Point Detector With a Coded Binary Mask”. In: *IEEE Sensors Letters* 3.12 (2019), pp. 1–4.

[159] Nanhao Gu, Bo Yang, and Tong Zhang. “Dynamic fuzzy background removal for indoor human target perception based on thermopile array sensor”. In: *IEEE Sensors Journal* 20.1 (2019), pp. 67–76.

[160] Masato Kuki et al. “Human movement trajectory recording for home alone by thermopile array sensor”. In: *2012 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. IEEE. 2012, pp. 2042–2047.

[161] Larry K Baxter. “Capacitive sensors”. In: *Design and Applications* (1997).

[162] Hitoshi Nishiyama and Mitsunobu Nakamura. "Capacitance of a strip capacitor". In: *IEEE Trans. on Components, Hybrids, and Manufacturing Technology* 13.2 (1990), pp. 417–423.

[163] Aftab Khan et al. "Optimising sampling rates for accelerometer-based human activity recognition". In: *Pattern Recognition Letters* 73 (2016), pp. 33–40.

[164] Yuanchao Shu et al. "Gradient-based fingerprinting for indoor localization and tracking". In: *IEEE Transactions on Industrial Electronics* 63.4 (2015), pp. 2424–2433.

[165] *D6T MEMS Thermal Sensors*. en-US. URL: https://www.components.omron.com/product-detail?partNumber=D6T (visited on 02/12/2020).

[166] *D6T MEMS Thermal Sensors*. en-US. URL: https://omronfs.omron.com/en_US/ecb/products/pdf/en_D6T_catalog.pdf (visited on 02/12/2020).

[167] *MEMS Thermal Sensors D6T*. en-US. URL: https://omronfs.omron.com/en_US/ecb/products/pdf/en_D6T_users_manual.pdf (visited on 02/12/2020).

[168] Ian H Witten et al. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, 2016.

[169] Chih-Chung Chang and Chih-Jen Lin. "LIBSVM: a library for support vector machines". In: *ACM Trans. on Intelligent Systems and Technology (TIST)* 2.3 (2011), p. 27.

[170] Yazhene Krishnaraj and Chandan K Reddy. "Boosting methods for protein fold recognition: an empirical comparison". In: *Bioinformatics and Biomedicine, 2008. BIBM'08. IEEE International Conference on*. IEEE. 2008, pp. 393–396.

[171] Terrence J Sejnowski and Charles R Rosenberg. "Parallel networks that learn to pronounce English text". In: *Complex systems* 1.1 (1987), pp. 145–168.

[172] C Freeman, RD Dony, and SM Areibi. "Audio environment classication for hearing aids using artificial neural networks with windowed input". In: *2007 IEEE Symposium on Computational Intelligence in Image and Signal Processing*. IEEE. 2007, pp. 183–188.

[173] Serkan Kiranyaz et al. "1D Convolutional Neural Networks and Applications: A Survey". In: *arXiv preprint arXiv:1905.03554* (2019).

[174] Mike Schuster and Kuldip K Paliwal. "Bidirectional recurrent neural networks". In: *IEEE Transactions on Signal Processing* 45.11 (1997), pp. 2673–2681.

[175] JO Ramsay, G Hooker, and S Graves. *Functional data analysis with R and MATLAB*. New York, NY: Springer, 2009.

[176] Shuai Tao et al. “Recording the activities of daily living based on person localization using an infrared ceiling sensor network”. In: *2011 IEEE International Conference on Granular Computing*. IEEE. 2011, pp. 647–652.

This Ph.D. thesis has been typeset by means of the TEX-system facilities. The typesetting engine was pdfLATEX. The document class was `toptesi`, by Claudio Beccari, with option `tipotesi=scudo`. This class is available in every up-to-date and complete TEX-system installation.