Politecnico di Torino

ScuDo
Scuola di Dottorato ~ Doctoral School
WHAT YOU ARE, TAKES YOU FAR

Doctoral Dissertation
Doctoral Program in Electrical, Telecommunication Engineering (36^{th}cycle)

Efficient Indoor Human Sensing and Continuous Tracking

By

Giorgia Subbicini

******

Supervisors:
Prof. Mihai T. Lazarescu, Supervisor
Prof. Luciano Lavagno, Co-Supervisor

Doctoral Examination Committee:
Prof. Emanuel Popovici, Referee, University College - Cork
Prof. Florenc Demrozi, Referee, University of Stavanger

Politecnico di Torino
2024

Declaration

I hereby declare that, the contents and organization 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.

Giorgia Subbicini
2024

* This dissertation is presented in partial fulfillment of the requirements for **Ph.D. degree** in the Graduate School of Politecnico di Torino (ScuDo).

I would like to dedicate this thesis to my mum, she was, she is, she will be my Ithaca.

Acknowledgements

Firstly, I extend my gratitude to my esteemed supervisors, Professor Mihai Teodor Lazarescu and Professor Luciano Lavagno, whose extensive knowledge and experience have been invaluable in guiding me throughout the progression of my doctoral studies.

Throughout this academic journey, I have had the privilege of interacting with remarkable colleagues and professors who have served as a source of inspiration, offering invaluable counsel and sharing their rich experiences. To them, I extend my heartfelt appreciation.

Furthermore, I wish to express my thanks to the sensing and high-level synthesis groups, whose collective efforts have fostered a welcoming and dynamic environment conducive to productive research endeavors. It has been a pleasure to be part of such an engaging academic community.

Abstract

The proliferation of indoor location-based systems (sLBSs) has been notably accelerated by the successful integration of Internet of Things (IoT) technologies, advancements in wireless network technology, and the unprecedented impact of the COVID-19 pandemic. Indoor LBS encompass a spectrum of hardware and software technologies, rendering them adaptable and applicable across diverse industrial sectors.

While hardware technologies based on electromagnetic waves (EMW), such as Wi-Fi or Bluetooth, dominate the landscape, they predominantly cater to device-based applications, thereby potentially marginalizing the elderly demographic. Indeed, this demographic cohort exhibits a reduced inclination towards mobile device utilization and diminished adeptness in sustained daily engagement with such technology. Furthermore, these technologies primarily serve sales and marketing sectors, focusing on functions like customer preference tracking, inventory management, and supply chain optimization. Conversely, device-free localization (DFL) assumes a pivotal role in facilitating ambient-assisted living (AAL) within intelligent infrastructures, encompassing functionalities such as intrusion detection, fall detection, and occupancy monitoring to promote energy efficiency and illumination management. Furthermore, certain applications such as fall detection and emergency response for elderly or vulnerable individuals, navigation guidance for the visually impaired, and security and access control necessitate high accuracy. These specific fields constitute the focus of our investigation.

This thesis endeavors to implement a proficient device-free indoor location-based system, with focus on accuracy and resource efficiency. The main contributions are:

- Design and characterization of a novel front-end based on slope modulation for long-range load-mode capacitive sensors that rejects sensor drift and increases sensing range and stability.

- Investigation into advanced neural network architectures conducive to continuous indoor human tracking.
- Setting up a neural network optimization methodology, integrating multi-step knowledge distillation and neural architecture search techniques for continuous indoor human tracking.

Capacitive sensors, falling within the electric field sensing domain, present themselves as promising candidates for the implementation of an efficient DFL system. This is due to their attributes of being cost effective, energy efficient, easy to deploy, privacy conscious and unobtrusive. Of the capacitive sensor operating modes, the load mode allows the use of a single sensing plate, making the system architecture lighter and more manageable. Nevertheless, over long distances (e.g., more than 10 times the plate diameter), the measurement field becomes highly susceptible to environmental noise, thereby compromising stability and accuracy. Electric charge variations in their environment cause variable electric fields that can induce a slow, quasi-constant drift current that can alter sensor readings even when their capacitance does not change. To address this challenge, we introduce a novel slope modulation interface. This interface functions on the galvanostatic charge-discharge principle of plate capacitance, akin to period modulation interfaces. However, it utilizes distinct measurements and employs data post-processing techniques to mitigate environmental noise, thereby extending the monitoring range and improving measurement stability. The technique uses a constant current generator driven by a square wave to charge and discharge the transducer plate with a fixed period. Subsequently, the average slope of two consecutive charge-discharge ramps of the triangular waveform of the plate voltage is measured to determine both the transducer capacitance and reject drift (low frequency) noise.

To evaluate the effectiveness of the method, is compared and examined the slope modulator front-end against the constant-current and RC period modulator analytically. Following this, the circuit is simulated and results validated using implementations of the front-ends and a noise generator. The findings are highly encouraging; specifically, the slope modulator exhibits an inversely proportional noise attenuation relative to frequency, rejecting mostly the range of very low frequencies characteristic of quasi-constant drifts. Inversely, the period modulators exhibit a flattening behavior in noise rejection below a specific frequency, where constant drift currents become dominant. Additionally, the slope modulator achieves noise rejection while

preserving high sensitivity, detecting people at distances up to 230 cm. Comparatively, period modulator interfaces can sense people up to 130 cm beyond which the noise increases significantly, surpassing signal levels around 250 cm.

In LBS context, sensor measurements are typically post-processed with general techniques such as outlier removal, frequency filtering and averaging, and then using specialized processing often based on machine learning (ML) techniques. The latter are increasingly used because their ability to process effectively noisy and nonlinear responses, while also offering scalability and robustness. Tariq, Lazarescu, and Lavagno [115] employ digital filters in conjunction with neural neural networks (sNNs) to mitigate noise and extract position and motion dynamics. Their design space exploration encompasses various popular NN architectures, including multilayer perceptron (MLP), 1D convolutional neural network (1DCNN), and long short-term memory (LSTM). Their findings indicate that 1DCNN achieves superior accuracy, whereas LSTM excels in capturing natural movement dynamics.

However, the pooling operations employed in convolutional neural networks (sCNNs) is acknowledged in literature that can lead to the loss of significant relational information, while the recursive nature of LSTM networks may introduce time-consuming complexities. In light of these considerations, we opt to investigate two enhanced network architectures, namely, the capsule network (CAPS) and temporal convolutional network (TCN), both of which avoid the use of pooling and recursion. Through hyperparameter optimization, the CAPS network achieves comparable accuracy to top-performing 1DCNN, utilizing only 78.7 % of the computational resources, while the TCN, albeit with a slight decrease of accuracy, uses merely 26.7 % of the 1DCNN computational resources.

Among the most sought-after characteristics of DFL systems are low cost, low power consumption, ease of deployment and management, and unobtrusiveness. However, their effectiveness may often require resources beyond what can be offered by the typically restricted IoT devices. In this regard, we propose a novel method based on KD and NAS aimed at optimizing the resource requirements of NN models. The KD technique involves the transfer of knowledge from a large NN (referred to as the teacher) to a smaller NN (referred to as the student). While this technique has been extensively studied in the context of classification problems, there is a paucity of research addressing its application to regression tasks and specifically to indoor

continuous human tracking. The NAS optimizes NN and KD parameters for teachers and students.

To evaluate the efficacy of the optimization method, we utilize the enhanced models (TCN and CAPS), along with several LBSs employing diverse sensing principles and data formats, each subject to varying levels of noise, including capacitive, infrared radiation (IR), and radar-based systems. We start with the capacitive environment, considered as the most complex scenario, and iterate through multiple steps, experimenting with different teacher-student configurations in each step of KD.

The results indicate that employing a teacher of the same type as the student in the initial KD stage, followed by a teacher of a different type in the subsequent stage, offers varied perspectives on the features, thereby aiding the student in enhancing accuracy and generalizability. Subsequently, the effectiveness of the optimal workflow is validated through distinct experiments conducted across the capacitive environment and other LBSs. These experiments, conducted under similar yet distinct conditions on different days, confirm the robustness of the proposed methodology.

The experimental outcomes demonstrate that the proposed method frequently reduces the size of NNs significantly, while concurrently improving their accuracy and generalizability in comparison to both state-of-the-art models and the original teacher NNs. Notably, NNs optimized via two-stage KD process tend to demonstrate enhanced generalization, showcasing an average mean squared error (MSE) improvement of 9.57 % for capacitive,14.4 % for infrared, and 4.86 % for radar LBS. This improvement is accompanied by an average reduction in resource utilization of 59.9 %, 56.4 %, and 61.28 % respectively. These findings underscore the effectiveness of the mixed two-step KD approach implemented within the proposed framework.

Contents

List of Figures

List of Tables

Glossary

1DCNN 1D convolutional neural network

AAL ambient-assisted living

ADC analog-to-digital converter

ADE average euclidean distance error

AI artificial intelligence

ANN artificial neural network

AutoML automated machine learning

CNN convolutional neural network

CAPS capsule network

DCNN deep convolutional neural network

DFL device-free localization

DL deep learning

DNN deep neural network

DSE design space exploration

DSP digital signal processor

DT decision tree

EMW electromagnetic waves

FLOP floating point operation per second

FMCW frequency modulated continuous wave

FoV field of view

HAR human activity recognition

HVAC heating, ventilation and air conditioning

IoT Internet of Things

IR infrared radiation

KD knowledge distillation

KNN k-nearest neighbor

LBS location-based system

LSTM long short-term memory

MCU microcontroller unit

ML machine learning

MLP multilayer perceptron

MSE mean squared error

NAS neural architecture search

NN neural network

NSS neuro-symbolic system

PIR passive infrared

PW pulsed wave

RD range-Doppler

RDA range-Doppler-azimuth

RF random forest

RMSE root mean square error

RNN recurrent neural network

RSS received signal strength

RX receive

SVM support vector machine

TCN temporal convolutional network

TDoA time difference of arrival

TOF time-of-flight

TX transmit

VLP visible light positioning

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

- Subbicini, Giorgia; Lavagno, Luciano; Lazarescu, Mihai T. (2022) "Drift Rejection Differential Frontend for Single Plate Capacitive Sensors." In: IEEE SENSORS JOURNAL, vol. 22, pp. 16141-16149. ISSN 1530-437X
- Subbicini, Giorgia; Lavagno, Luciano; Lazarescu, Mihai T. (2023) "Enhanced Exploration of Neural Network Models for Indoor Human Monitoring". In: 2023 9th International Workshop on Advances in Sensors and Interfaces (IWASI), Monopoli (Bari), Italy, 08-09 June 2023, pp. 109-114. ISBN: 979-8-3503-3694-8
- Subbicini, Giorgia; Lavagno, Luciano; Lazarescu, Mihai T. (Date of submission: September 30, 2023) "Comprehensive Evaluation of Mixed Multi-Level Knowledge Distillation for Minimizing Neural Networks for Continuous Indoor Human Tracking". In: IEEE SENSORS JOURNAL, Special Issue on Energy-Efficient Embedded Intelligent Sensor Systems

Chapter 1

Introduction

Indoor LBSs facilitate the tracking of individuals and objects within defined spaces through the integration of hardware and software technologies. The indoor LBS market witnessed substantial growth, particularly in the aftermath of the COVID-19 pandemic, as the surveillance and monitoring of individuals within buildings became imperative to prevent overcrowding and ensure a safe environment.

According to a report from Grand View Research, the global market size for indoor LBSs reached USD 15.48 billion in 2022, with a projected compound annual growth rate (CAGR) of 14.1 % from 2023 to 2030. The report emphasizes that the increasing prevalence of interconnected devices worldwide, coupled with the rising deployment of local and home area networks such as Wi-Fi and WLAN, is anticipated to drive market expansion throughout the forecast period.

The application of LBSs extends across various industries, encompassing field such as sales and marketing for comprehending customer behavior and interests, inventory and supply chain management, as well as considerations related to energy efficiency and safety. Additionally, persistent indoor person tracking proves beneficial in the domain of assisted living, particularly for the elderly or individuals with health concerns. In this context, sensors play a crucial role in detecting falls, monitoring vital signs, and analyzing alterations in daily activities. Such monitoring mechanisms serve to identify potential indicators of emerging health conditions, thereby necessitating timely alerts to caregivers or emergency services.

Localization methodologies are categorized into active and passive systems based on their sensing processes. Active methodologies involve the generation and

transmission of signals, while passive methodologies rely on the detection of existing signals. Additionally, these methodologies can be further classified into tag-based or tagless categories, depending on the level of human cooperation necessary for effective monitoring.

Methods that are both active and tag-based often require substantial computational, networking, and energy resources, which may have implications for reliability and usability. Examples of such methods include those utilizing EMW, such as Wi-Fi [125], ZigBee [25], Bluetooth [23], ultra-wide band [38], and radio frequency identification (RFID) [35]. Ultrasonic systems [16] have also been explored as active solutions, with the flexibility of implementation with or without individuals wearing physical tags. However, challenges arise from ambient noise and multipath effects, necessitating the use of intricate algorithms to address potential inaccuracies [58].

Active tagless solutions, exemplified by Infrared (IR) sensors emitting infrared light beams [48], and radar systems [61] utilizing radio waves for emission and reflection, provide viable options for non-intrusive monitoring.

On the passive tagless front, approaches include video or image-based methods and passive infrared (PIR) sensors [5]. While these methods require a direct field of view, the former may raise privacy concerns, and the latter may lack reliable stationary occupancy detection.

An alternative passive tagless system can be devised by employing capacitive sensors capable of detecting and tracking both conductive and non-conductive objects. Long-range capacitive sensors operating in load mode, as detailed in [108], utilize single-plate transducers, with the electrically conductive human body serving as the other plate. These sensors are characterized by their cost-effectiveness, ease of aesthetic concealment, and privacy considerations. However, at prolonged distances (e.g., approximately 10 times the plate diameter), the measurement field encounters challenges in effectively mitigating drift and noise, as evidenced by studies such as [15, 98]. Consequently, various environmental conditions can impact measurement stability and accuracy, this issue becomes particularly pronounced due to the minute capacitance variations of interest at long sensing ranges, falling below 0.01 %, as reported in [117, 56]. Within this study, a novel frontend is introduced, employing slope modulation and differential measurements. This innovative approach demonstrates efficacy in mitigating noise arising from slow drifts, concurrently maintaining the sensitivity of capacitance measurements. Analytical comparisons of sensitivity

and noise rejection are conducted with period modulation techniques, subsequently corroborated through simulations and experimental validations. The findings reveal a twofold expansion in the human body sensing range and a noteworthy reduction in the noise floor when juxtaposed with the prevailing state-of-the-art method (period modulation).

In contemporary applications of LBSs, machine learning techniques have been increasingly employed in post-processing stages. This strategic integration serves to enhance adaptability to complex indoor environments, address non-line-of-sight challenges, and improve the overall accuracy and reliability of location estimates by leveraging diverse data sources and learning from dynamic conditions.

Indoor tracking/localization has traditionally been achieved through ML algorithms such as k-nearest neighbor (KNN) and support vector machine (SVM) [128], which require hand-crafted features and have limited representational power. Today, deep learning (DL) architectures are widely used [12, 128] that can operate on raw sensor data, extract valuable features, and identify spatial and temporal patterns. However, they require massive training datasets and many resources for inference, which are difficult to provide on resource-constrained IoT systems [128].

The acquired data from capacitive sensors exhibit pronounced noise and substantial non-linearity, necessitating extensive post-processing. Tariq, Lazarescu, and Lavagno [115] integrate digital filters with neural networks to effectively attenuate noise and deduce both position and motion dynamics. Design space exploration (DSE) performed on the most popular feedforward (MLP, autoregressive and 1DCNN) and recurrent (LSTM) neural networks reveals that sequence-aware (LSTMs) networks capture the movement dynamics best while 1DCNNs have the least error.

However, the pooling operations used in CNNs are known to lose important relations [4]. In the literature, two promising candidate neural networks for time-series tasks are identified: TCNs [9, 45, 66] and CAPS [59, 112]. TCNs leverage multiple layers of dilated convolutions with progressively increasing intervals between adjacent taps, enabling them to encompass broader input ranges while maintaining resource efficiency. CAPSs address fundamental limitations associated with CNN, including challenges in comprehending hierarchical and relational structures, inadequacies in handling rotations and resizing, and weakened inference resulting from average/max pooling layers. The innovative approach of CAPS involves decompos-

ing intricate objects into a hierarchical representation of pre-learned patterns and employing dynamic routing mechanisms instead of traditional pooling operations. When appropriately tuned, both TCN and CAPS models demonstrate superior accuracy and resource efficiency (evaluated in terms of the number of parameters) in comparison to 1DCNN for capacitive sensors [115]. However, the relatively high resource utilization motivates ongoing research endeavors aimed at optimizing these models to minimize size without compromising accuracy and generalization.

Numerous compression techniques, including quantization and pruning [71], have been proposed to diminish the size and complexity of neural networks. However, these methods often entail a trade-off with network accuracy. A relatively recent approach, referred to as KD [50, 40], has emerged, aiming to transfer knowledge from a larger network (teacher) to a smaller one (student), mimicking a teacher-student relationship. While many existing KD methods primarily focus on classification problems [67, 92, 121], where softened logits from the teacher offer valuable insights into class relationships absent in one-hot encoded ground-truth labels, this advantage is not present in regression problems with continuous and unbounded ground-truth predictions. A study exploring the amalgamation of teacher, student, and ground-truth losses in the context of visual odometry regression [101] revealed that the most effective formulation is the attentive imitation loss. This approach considers the uncertainty associated with teacher predictions and adjusts the weight of the distillation loss accordingly.

Limited research has delved into the application of KD specifically in the context of indoor human tracking. Existing studies have predominantly concentrated on location classification, rather than regression for the continuous estimation of location coordinates. Our emphasis, however, is directed towards the latter, as the estimation of movement trajectories provides a more comprehensive and informative representation of human dynamics. This approach demonstrates superior adaptability to dynamic environments. Moreover, an enhanced understanding of motion patterns enables applications in various fields, empowering them to make more informed decisions, predictions, and interactions.

To address this objective, we propose a multi-step KD approach coupled with optimization based on NAS tailored specifically for regression-based indoor human tracking. We aim to achieve fine-grained continuous tracking of an individual's position. The proposed methodology incorporates NAS to simultaneously optimize both

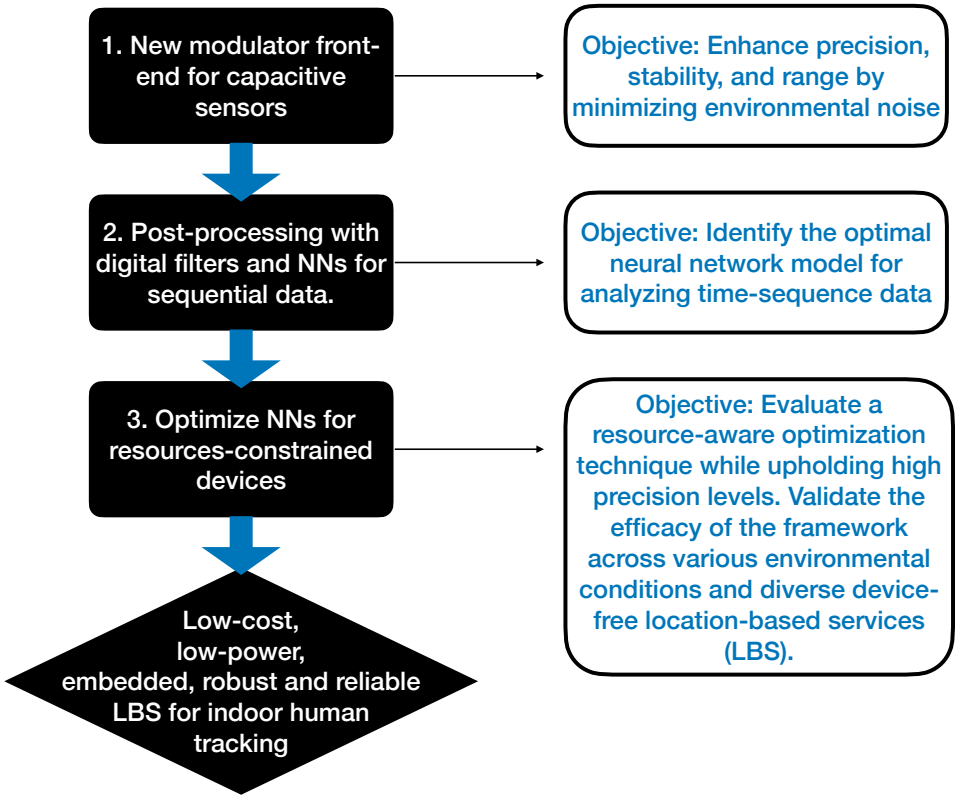


Fig. 1.1 Flowchart illustrating the organization of work and outlining objectives.

the hyperparameters of the NNs and the KD strategy. This dual optimization is designed to minimize resource requirements and enhance the localization performance of the NN models. Prior to determining the most effective multi-step optimization procedure, we explore multiple types and combinations of teachers and students. The validation and testing of the proposed method are conducted using various LBSs. These systems employ different sensing principles, data formats, and are exposed to diverse types and levels of noise, including long-range capacitive, IRs, and radar.

This research endeavors to achieve several objectives (Fig. 1.1). Firstly, it seeks to identify a cost-effective, integrated, and robust solution for noise rejection in a capacitive sensor operating in load mode. Secondly, it aims to enhance the reliability of long-range measurements by employing advanced time-sequence NNs. Thirdly, it aims to evaluate a resource-conscious optimization technique for the efficient deployment of NNs in resource-constrained devices, adaptable to diverse environmental conditions and various types of LBSs. All of these advancements contribute to the development of an IoT-deployable LBS that is device-free, unobtrusive, and low-

cost, while providing fine-grained accuracy. This system is particularly effective for applications aimed at assisting individuals facing various challenges. Aligned with these goals, the principal contributions of this dissertation encompass:

- Development of a novel front-end based on slope modulation tailored for long-range capacitive sensors operating in load mode, designed to mitigate environmental noise interference, thereby augmenting the monitoring range and stability.
- Investigation into advanced neural network architectures conducive to continuous indoor human tracking.
- Setting up a neural network optimization methodology, integrating multi-step knowledge distillation and neural architecture search techniques for fine-grained continuous indoor human tracking.

Chapter 2

Sensing and Processing for Indoor Human Tracking

Indoor localization holds promising applicability across diverse sectors, including but not limited to retail, manufacturing, healthcare, smart buildings, and sports industries. The design phase of LBS necessitates meticulous consideration of numerous parameters such as cost, energy efficiency, reception range, localization/tracking accuracy, latency, scalability, and privacy. These factors are pivotal in configuring a deployable system tailored to quality of service requirements. This chapter will expound upon and analyze various existent technologies employed for the provision of device-free indoor localization services, along with machine learning techniques applied in the postprocessing of the sensor readings.

2.1 Sensors Technologies for Device-Free Localization

DFL denotes a methodology for localizing a target without the requisite use of a wearable tag or device to facilitate the localization process. This mode of localization holds notable utility in fostering AAL within intelligent structures. Moreover, its applicability extends to diverse domains such as intrusion detection, fall detection, and remote monitoring of the elderly. Furthermore, DFL serves as an effective tool for occupancy detection in optimizing energy utilization within heating, ventilation, and heating, ventilation and air conditioning (HVAC) systems, as well as for lighting control. In emergency scenarios, it proves valuable for occupancy counting, notably

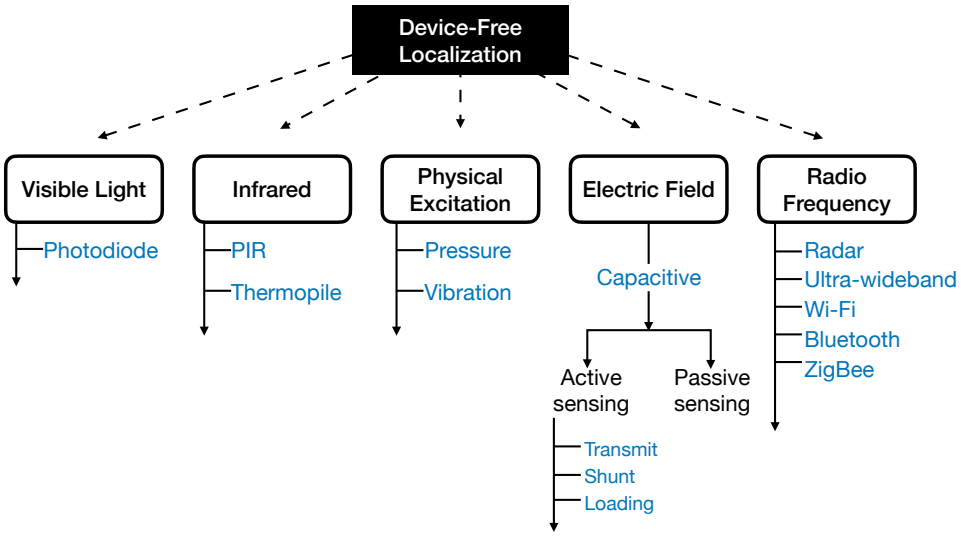


Fig. 2.1 Categories of DFL technologies categorized by sensing modality.

during the emergency evacuation of offices and public buildings. Beyond these applications, DFL contributes to business analytics in retail settings and provides assistance in enhancing accessibility for visually impaired individuals, among other potential uses [3].

The methodologies employed within DFL encompass a diverse array of sensing modalities. This section undertakes an examination and analysis of these modalities (with focus on those utilized later in our experiments), identifying five primary categories: visible light, infrared, physical excitation, electric field, and radio frequency (Fig. 2.1). Each category may involve either an active or passive sensing approach. The former entails the emission of energy or signals into the environment by the sensor itself, followed by the detection of the reflected or scattered signal. In contrast, the latter simply observes and records the existing energy sources or signals.

Visible Light Sensing

Device-free visible light positioning (VLP) systems rely on light sources and light sensors, such as photodiodes. These systems operate by detecting alterations in the optical channel between the light source and sensor caused by the movement of a target. Typically, these changes manifest as variations in light level or illuminance recorded by a sensor in proximity to the target. The positioning system's architecture involves strategically placing sensors across different surfaces, including the floor

(for shadow detection) [69], ceiling (for analyzing reflected signals) [85], or wall (for observing variations in received signal strength (RSS) of ambient light) [37].

The key advantages of these technologies include immunity from radio frequency interference, the absence of small-scale fading, and low power consumption. However, it is important to note that the infrastructure required for implementation is relatively complex, and maintenance can pose challenges.

Infrared Sensing

Device-free infrared technology relies on passive infrared (PIR) sensors, which exploit IR emitted by a warm target, typically a human, within the sensor's field of view (FoV). Unlike configurations with distinct senders and receivers, in this technology the PIR sensor directly localizes the signal source. A PIR sensor comprises a pair of pyroelectric sensing elements separated by a small gap. The variation in IR signals received by the two sensing elements is analyzed through a differential amplifier, rendering the system capable of detecting dynamic sources while being immune to static sources.

Two processing modes are available for investigation utilizing PIR sensors: binary output [47] and analog voltage output [83]. The former entails the generation of a voltage signal proportional to IR change, which is subsequently compared with a predefined threshold to ascertain the presence or absence of a target. In contrast, the latter mode directly employs the raw voltage analog output for localization purposes. The binary technique, characterized by its coarser resolution, necessitates the installation of numerous sensors, whereas the analog output, leveraging cluster-based strategies, can achieve finer-grained localization with a reduced number of devices.

Thermopile sensors are also based on IR, but unlike PIR sensors, they employ a configuration of series-connected thermocouples. This configuration enables the measurement of incident IR flux, thereby facilitating the generation of a thermal image. Most of the works use a ceiling mounted 8 x 8pixel array.

Hevesi et al. [49] use "Grid Eye" from Panasonic with 8x8pixel, 0.25 °C sensitivity. In conjunction with decision tree and KNN models, the researchers employed this sensor for household activity recognition, with a tracking accuracy of 1 m.

In [22], an 8x8pixel "Grid Eye" thermopile sensor is integrated into a rotational platform, which also incorporates a time-of-flight (TOF) VL53L0X sensor module and placed above a table. The primary objective of this system is to discern the facing direction of a target utilizing thermal imaging data, followed by feature extraction using CNN techniques and classification employing SVM algorithms. Simultaneously, the system tracks the occupancy within a rectangular area measuring 1.2x2.4 using the distance sensor. Remarkably, the system achieves a localization root mean square error (RMSE) of 0.19 m.

Kowalski. et al. [62] employ a wall-mounted configuration of thermopile sensors. In order to achieve comprehensive area coverage spanning 180°, they devise an interconnected cluster comprising three 8x8pixel "Grid Eye" sensors. Positioned at a height of approximately 1.5 m, two such clusters are affixed onto the wall, one on the upper-hand and another on the right-hand side, to monitor a rectangular region measuring 2x2.5. Subsequently, employing SVM classification techniques, they undertake the task of localizing targets within a grid composed of 20 discernible 0.5x0.5 segments. The classification accuracy for tracking targets within a single cell and within a neighborhood comprising four cells is reported as 73.1 % and 93 %, respectively.

In the work presented by Gu, Yang, and Li [43], the utilization of a high-resolution thermopile sensor with dimensions of 24x32pixel is demonstrated to confer wider coverage and enhanced accuracy. Positioned at a height of 3 m on the ceiling, this sensor configuration effectively surveils a rectangular area measuring 2x3.72. Employing techniques such as interpolation, filtering, and adaptive threshold-based background removal, the authors enhance the efficacy of the system. Experimental evaluation conducted along predetermined paths reveals an average RMSE below 0.15 m, indicative of the system's robust performance. Conversely,

Kuki et al. [63] utilize a thermopile sensor with markedly lower resolution, specifically 4x4pixel, for area monitoring purposes. Their system covers an area of 1.58x1.58. Employing fuzzy logic for background removal and centroid calculation for target positioning, the authors report a mean positioning error of 0.215 m. This demonstrates the compromise in accuracy inherent in employing sensors of lower resolution.

Physical Excitation Sensing

In closed environments, the act of taking a single step on the floor generates a distinct physical excitation, which can be leveraged to localize the movement of individuals. Consequently, the system operates by discerning and pinpointing the source of this excitation to achieve accurate localization.

The earliest reported technique for target localization involves pressure excitation, which necessitates an intelligent floor embedded with piezoelectric or resistive pressure sensors. These sensors, typically arranged in a grid-like array, analyze the ground reaction force by means of small tiles composed of load cells [6, 29, 84, 1].

Alternatively, vibration-based sensing systems [80, 81, 27, 105], utilize geophones or seismic sensors to detect footstep-induced vibrations. Multiple sensors capture these vibrations, enabling the application of time difference of arrival (TDoA) algorithms for source localization. To discern footstep events amidst other excitation events (such as the fall of a chair), classifiers are commonly employed.

While pressure-based systems entail a complex architecture and costly deployment, vibration-based systems may capitalize on preexisting sensors. However, the latter is susceptible to interference from ambient noise.

Electric Field Sensing

Electric field sensing relies on capacitive coupling and offers two distinct sensing modes: 1) Active sensing involves the use of a transmitter that actively generates an electric field. Measurements are then conducted to evaluate changes in the electric field caused by the movement of the target; 2) Passive sensing, on the other hand, does not utilize a transmitter. Instead, it leverages the ambient electric field and measures variations in it resulting from the presence or movements of the target.

The active sensing technique offers three distinct modes, namely transmit, shunt and loading mode [108]. In the transmit mode, a transmitter generates the electric field, and a receiver measures the field. When the target is in close proximity to the transmitter, the field between the transmitter and the receiver may be interrupted. In such cases, the subject's body can act as an emitter, affecting the field and enabling detection. The shunt mode involves both a transmitter and an emitter. Here, the target obstructs a portion of the field, redirecting it to the ground. However, the receiver

can still detect the remaining field, allowing for localization. Conversely, the loading mode solely utilizes the transmitter. In this mode, the target's body effectively serves as the second plate of the capacitor to the ground, influencing the electric field and enabling detection. Many studies in the field of loading mode sensing predominantly rely on floors outfitted with sensors.

SensFloor [65] organizes the floor into a grid comprising independently functioning modules. Each module features a microcontroller board positioned centrally, connected to a power supply, and surrounded by eight triangular sensor shapes. The authors subsequently employ a LSTM network to discern various walking gaits.

Conversely, CapFloor [19] situates electronic components at the periphery, with passive elements (wires) positioned beneath the floor. For their prototype, they utilize passive floor mats of rectangular shape furnished with eight active sensor elements on two adjacent outer sides, covering a total area of 6 m^2. According to [19], the estimated positioning error falls below 0.5 m.

The SensingFloor technology discussed in [36] exhibits significant promise, capable of achieving a median position error of 0.013 m and a median angular error of 10.4° for stationary foot placement. The authors employ thirty-six copper foil squares arranged in a panel measuring 0.6x0.6, with four such panels constituting the testbed. The acquired data undergoes interpolation, and subsequently, the detection of the "blob" is performed.

Ramezani Akhmareh et al. [95] employs a configuration comprising four loading mode capacitive sensors affixed to the walls. Each sensor, crafted from small copper-clad tiles measuring 0.16x0.16, is positioned centrally on each wall at a height equivalent to that of the chest. These tiles are interconnected to a period modulator front-end, which is implemented using a 555 timer, thereby facilitating a frequency modulation response in correspondence with alterations in capacitance. The evaluation of this system involves conducting tests along pre-defined trajectories delineated by fixed grid positions. The experimental setting encompasses diverse sources of interference, such as a refrigerator and a metal cupboard within a 3x3 room. The reported findings indicate a mean localization error of 0.2 m for static positioning under these conditions.

TileTrack in [119], implements transmit mode sensing in conjunction with a distinct receiver, thereby enabling the mitigation of stray capacitances that may form with the environment. The transmitting electrodes are positioned beneath the floor, while two types of receiving electrodes are affixed to a wooden frame adjacent to the

floor tiles: either a copper plate or a power-line cable. Both receiver configurations demonstrate validity and exhibit comparable performance characteristics. Regarding the positioning of a standing individual, an accuracy of 0.143 m is attained, while in the context of a tracking task, the error consistently remains below 0.41 m.

Passive electric field sensing relies on detecting the electric field emitted by the mains power line (operating at 50/60 Hz). Movement of a charged target leads to changes in the electric field or the generation of a potential buildup due to the triboelectric effect. Positioning is determined by measuring these alterations in the ambient electric field caused by the target, as outlined by Prance et al. [93], Grosse-Puppendahl et al. [42], and Tang and Mandal [114].

Radio Wave Sensing

Radar systems consist of key components: transmit (TX) and receive (RX) radio frequency antennas, along with analog-to-digital converters (sADCs), microcontroller units (sMCUs), digital signal processors (sDSPs), and clocks. These systems operate by transmitting shaped radio waves and analyzing their modified back-scattered signals to determine parameters like distance, angle, velocity, and target shape. TX and RX components are usually housed together, with the transmitter emitting a modulated signal and the receiver detecting its back-scattered version after a short delay. Modern radar systems commonly utilize pulsed wave (PW) and frequency modulated continuous wave (FMCW) functions, with mmWave-based radar emerging as a low-power option for short-range environmental sensing. Most of the applications deal with human activity recognition (HAR) and tracking.

Pegoraro, Meneghello, and Rossi [88] utilizes a radar operating at a frequency of 77 GHz. Signal processing procedures encompass a sequence of steps including micro-Doppler mapping, the DBSCAN algorithm, Kalman filtering, and the utilization of a deep convolutional neural network (DCNN). Integration of identification data with trajectory tracking enhances identification accuracies, particularly in scenarios involving multiple targets. Results indicate an identification accuracy of 95.26 % for the range-Doppler (RD) configuration with three targets and 98.27 % for the range-Doppler-azimuth (RDA) and four targets.

In the same vein, the study presented in [54] focuses on movement pattern detection for one or two patients as a primary outcome. Employing DBSCAN and

Kalman filtering techniques, the trajectory of each patient is tracked. The proposed CNN model classifies movement patterns such as walking, falling, swinging, seizure, and restless movements, achieving accuracy values ranging from 82.77 % to 95.74 %.

Similarly, Zhang and Cao [130] employ a 77 GHz radar device to recognize various human behaviors. Their processing methodology involves micro-Doppler analysis, DBSCAN, and CNN, resulting in an accuracy exceeding 90 %.

In their work [44], the authors introduce a framework named "mmSense", which utilizes an LSTM-based classification model for localization purposes. Initially, the framework conducts environment fingerprinting both with and without human presence. Subsequently, utilizing the LSTM model, it estimates the presence and location of individuals within the environment. Additionally, a novel approach is developed, which combines human outline profile with vital sign measurements extracted from the reflected signal strength series at 60 GHz, aiming to identify the targets. Testing of the mmSense framework involves scenarios with five individuals concurrently sharing the same physical space, resulting in an accuracy of 97.73 % for classification tasks and 93 % for identification tasks, respectively. Radar sensors at 77 GHz were used in a 2x8 space with a denoising autoencoder and a sequence-to-sequence NN to infer the position of a moving target with 15 cm accuracy [89].

Other works use 3D point clouds and deep neural networks (sDNNs) to classify activities, postures, or human gaits [107, 102, 79, 60].

Device-free ultra-wideband (UWB) technology employs ultra-wideband radio waves for the detection and tracking of objects or individuals, eliminating the need for them to carry specific devices or tags. Instead, it relies on analyzing variations in the UWB signals propagating through the environment, which interact with surrounding objects and people, undergoing reflections, diffractions, and scattering.

UWB technology utilizes extremely short-duration pulses of electromagnetic energy spread across a broad frequency band, typically lasting only nanoseconds. This bandwidth is significantly wider than that of traditional RF signals.

Through the analysis of received UWB signals, including factors like time-of-arrival, angle-of-arrival, signal strength, and multipath effects, device-free UWB systems can estimate the presence, location, and movement of objects or individuals within the sensing area [26, 122].

In the radio frequency spectrum, various wireless communication standards such as Wi-Fi, ZigBee, and Bluetooth operate. These systems are distinguished by the presence of numerous access points (APs) within the relevant environment. Employing signal processing techniques like fingerprinting or trilateration, these systems analyze and compare signal characteristics observed with a database of signal maps or reference measurements.

Wi-Fi typically delivers superior localization accuracy and broader range compared to ZigBee and Bluetooth owing to its higher data rates. Additionally, Wi-Fi signals possess the ability to traverse obstacles, facilitating localization across multiple rooms or floors. Conversely, Wi-Fi generally entails higher power consumption and necessitates intricate configuration and calibration processes [30, 111, 68].

2.2 Post-processing Machine Learning Techniques for Indoor Localization

Indoor human tracking/localization has conventionally relied on ML classification algorithms like KNN and SVM. These methods typically necessitate manual feature engineering and exhibit limited representational capacity.

KNN [46], for instance, is a lazy learner, meaning it does not have a training phase as such. It simply memorizes the training dataset. Given a new, unlabeled data point, KNN predicts the label (classification) or value (regression) based on the majority vote (for classification) or averaging (for regression) of the labels or values of its nearest neighbors. Its performance can degrade with high-dimensional data or when the dataset size is large due to the computational cost of finding nearest neighbors. Additionally, it does not perform well with imbalanced datasets or when feature scaling is required.

SVM [90] works by finding the optimal hyperplane that best separates different classes in the feature space. SVM can handle non-linearly separable data by mapping the input features into a higher-dimensional space through a kernel function. This allows SVM to find a linear decision boundary in the higher-dimensional space, which corresponds to a non-linear decision boundary in the original feature space. SVM performance can be sensitive to the choice of kernel function and its parameters, and it may become computationally expensive with large datasets.

Random forest (RF) [87] is a versatile ensemble learning method used for both classification and regression tasks. It is based on the idea of building multiple decision trees during training and merging them to get a more accurate and stable prediction. Data samples are randomly selected to ensure tree independence. Classification is then performed through a majority vote among the decisions of all trees. This algorithm is known for its robustness, scalability, and ability to handle high-dimensional data with ease, however, it may not perform well on very imbalanced datasets.

Today, DL architectures are widely used [99] that can operate on raw sensor data, extract valuable features, and identify spatial and temporal patterns. However, they require massive training datasets and many resources for inference, which are difficult to provide on resource-constrained IoT systems.

Artificial neural networks (sANNs) encompass various architectures, such as the MLP, CNN, and autoregressive models. Additionally, LSTM networks represent a recurrent neural network (RNN) architecture.

An MLP [21] consists of input, hidden, and output layers, where neurons in adjacent layers are fully connected via weighted connections. Each neuron (excluding input layer neurons) has an associated bias term. Activation functions introduce nonlinearity to the network, enhancing its ability to learn complex patterns. MLPs find wide application in domains such as image recognition, natural language processing, and time series prediction.

In an autoregressive [118] model, the value of a variable at a specific time point is predicted based on its past values, reflecting the model dependence on its own historical data. Autoregressive models are adept at capturing temporal dependencies and patterns in time series data. However, their effectiveness may be diminished in cases where there is nonlinearity or nonstationarity in the data.

CNNs [70] consist of convolutional layers, which use filters to extract features from input images, followed by activation functions like ReLU. Pooling layers then downsample the feature maps while preserving important information. Finally, fully connected layers combine these features for classification or regression tasks. CNNs excel in various computer vision tasks, such as image classification and object detection.

LSTM [127] networks, a subset of RNNs, are adept at handling sequential data by addressing the vanishing gradient problem and capturing long-term dependencies.

Widely used in tasks like time series prediction and natural language processing, LSTMs represent each data point in a sequence as a vector. They employ memory cells with input, forget, and output gates to effectively capture and retain information over extended sequences.

While CNNs can extract sophisticated features with simple computations that are invariant to translations, they (1) fail at rotations and shrinking/enlargement transformations, (2) cannot understand hierarchical and relational structures, and (3) their inference is brittle mainly because of their average/max pooling layers [34], which increase their field of view but may discard relevant features where they are not the maximum or overlook complex patterns that require finer resolution.

CAPSs overcome the CNN major limitations due to information loss using dynamic routing instead of max pooling operations. They decompose a complex novel object into a hierarchical representation of previously learned patterns. While a CNN neuron outputs a scalar that signals only if a feature is recognized, without relative object relationships, CAPSs explicitly model these relationships in the vector-form of the neuron outputs: the vector length (modulus) encodes the detection probability, while the direction (placement in space) encodes the feature/object state (instantiation parameters e.g., pose, deformation, velocity). Capsule networks, originally designed for understanding spatial relationships in 2D data, have also been explored for time-series analysis [52, 100, 103, 112].

Recurrent and recursive networks for sequence modeling tasks have two major drawbacks: exploding/vanishing gradients and high resource consumption. Recent works combine the low-level spatio-temporal features extraction using CNN with the classification of high-level temporal information using RNNs ([106, 17, 20]). Bai, Kolter, and Koltun [9] argue that convolutional networks are best suited for modeling sequential data, obtaining good performance using TCNs. They use multiple layers of exponentially increasing dilated convolutions to cover a wider range of inputs with fewer resources. The convolution blocks are followed by normalization, nonlinear activation, and a dropout layer for regularization, forming residual blocks (two identical sub-blocks of dilated convolutions and a residual connection).

Various domains, including indoor localization, have benefited from the remarkable results of DNNs, but their large computational requirements limit their applicability on resource-constrained devices and real-time scenarios. Several compression techniques, such as quantization and pruning [71], have been proposed to reduce

the network size and complexity, but they often compromise the network accuracy. A more recent technique, known as KD, was introduced [50, 40] to transfer the knowledge from a large network (teacher) to a smaller one (student), mimicking a teacher-student relationship. Most of the existing KD methods focus on classification problems [67, 92, 121], where the softened logits of the teacher provide useful information about the class relationships, which is missing in the one-hot encoded ground-truth labels. However, this advantage does not exist in regression problems, where the ground-truth predictions are continuous and unbounded.

A study on how to combine the teacher, student, and ground-truth losses in a visual odometry regression problem [101] showed that the most effective formulation is attentive imitation loss, which considers the uncertainty of the teacher predictions and adjusts the weight of the distillation loss accordingly.

Chapter 3

Noise Rejection Front-End for Long-Range Capacitive Sensing

At long sensing ranges (e.g., 10 times the plate diameter), small capacitive sensors find applications for inconspicuous low power low cost indoor person localization or identification [10, 117, 51] for home automation or assisted living [18, 73]. Nonetheless, over extended ranges, maintaining the integrity of the measurement field against drift and noise becomes challenging. As evidenced in previous research, the stability and accuracy of measurements are susceptible to fluctuations induced by diverse environmental factors. This vulnerability is particularly pronounced due to the minute capacitance variations of interest, measuring below 0.01 % at these extensive sensing distances [117, 56].

In protracted monitoring applications involving stationary or slowly evolving scenarios, the mitigation of slow measurement drift proves to be a challenging task, often constraining the efficacy of long-range capacitive sensing methodologies. This drift may arise from actual changes in capacitance, such as those attributable to variable air humidity, which can be compensated for. Furthermore, other potential causes, may include variations in environmental conditions that influence the operation of the sensor frontend, independent of any changes in sensor capacitance.

Period modulation frontends represent a common approach for sensing the capacitance of sensor plates. Tan et al. [113] introduce self-calibration techniques and several trade-offs aimed at significantly reducing the power consumption of such frontends for prolonged monitoring via wireless sensor networks. In a separate study,

De Marcellis, Ferri, and Mantenuto [32] propose an astable multivibrator circuit employing a Schmitt trigger with adjustable thresholds for wide-range capacitance measurements. A period modulation frontend tailored for capacitive sensors utilized in sub-nanometer displacement measurement is presented in [2]. The design incorporates chopping, self-calibration, and active shielding to mitigate errors.In our prior research [95], we employed a straightforward period modulation frontend relying on a 555 timer, which exhibited heightened vulnerability to environmental noise, for the purpose of gathering capacitive sensor data geared towards person localization through machine learning techniques. Furthermore, we explored alternative approaches, including differential measurement methods. A direct interface, using only passive components, is proposed in [97] with 1 % full scale accuracy, and a method based on dual ramp integrator is proposed in [82] with errors below 0.2 %.

This chapter focuses on the initial objective of our study (Fig. 3.1), which

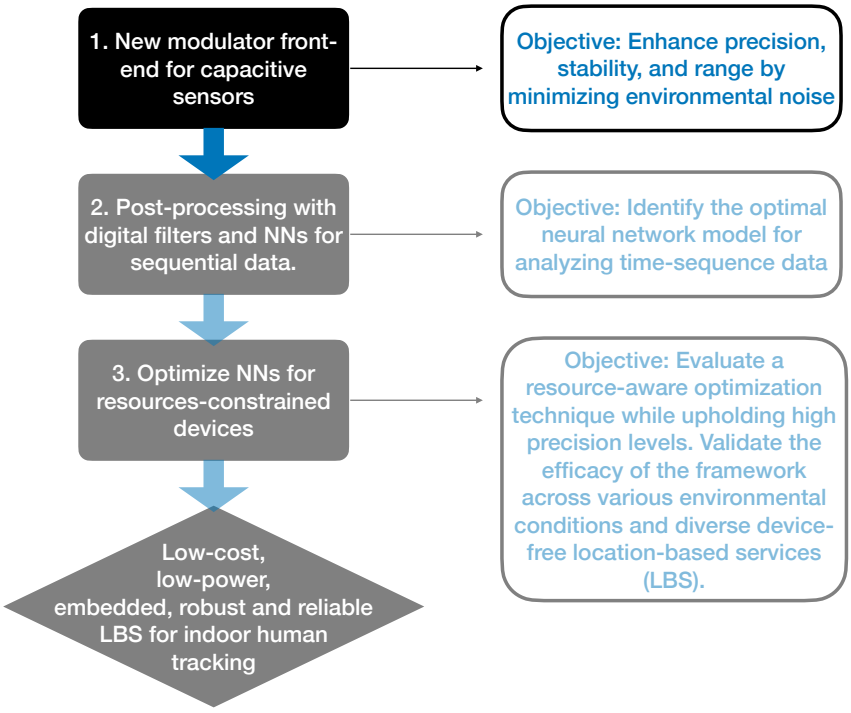


Fig. 3.1 Flowchart illustrating the organization of work and outlining objectives.

involves enhancing the precision, stability, and sensitivity range of a capacitive sensor operating in load mode. This technology aligns well with attributes such as cost-effectiveness, low power consumption, unobtrusiveness, and privacy consciousness. The chapter commences with an extensive analysis of the noise induced by variable

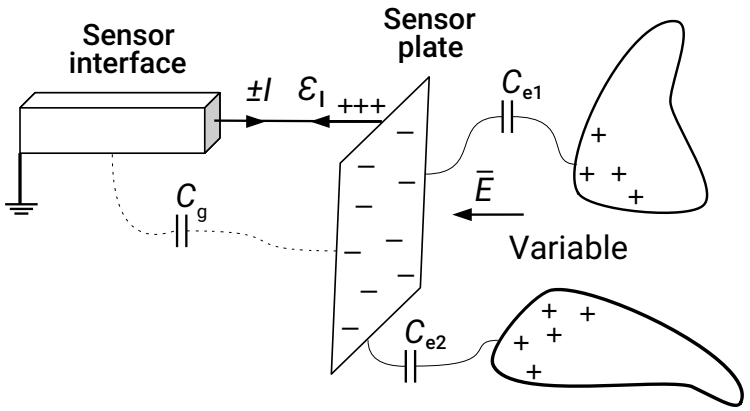


Fig. 3.2 Single-plate capacitive sensor model with measurement frontend, main capacitances, drive current I and drift current ε_I from charge induction in variable environmental electromagnetic fields

environmental electric fields on period modulation frontends. Subsequently, a novel frontend is proposed, leveraging slope modulation and differential measurements. This innovative approach effectively mitigates noise stemming from gradual drifts, while preserving the sensitivity required for precise capacitance measurements.

3.1 Sensor Drift Problem

The electrical capacitance C of an object is defined as the ratio between the variation of its electrical charge, ΔQ, and the corresponding variation of its electrical potential, ΔV_C

$$C = \frac{\Delta Q}{\Delta V_C}. \tag{3.1}$$

While the capacitance is fully defined by the electric and dielectric properties and geometries [11], the measurement frontends can instead be susceptible to several other environmental influences, e.g., from electromagnetic fields.

Fig. 3.2 shows the operating principle of a period modulation frontend for single-plate capacitance sensors. The sensor plate forms capacitances with bodies in the environment (C_{e_1}, C_{e_2}, ...) and with nearby objects, such as the frontend circuits and the ground (C_g). Period modulation frontends, with the block schematic shown in Fig. 3.3, cyclically charge and discharge the sensor plate with a constant current

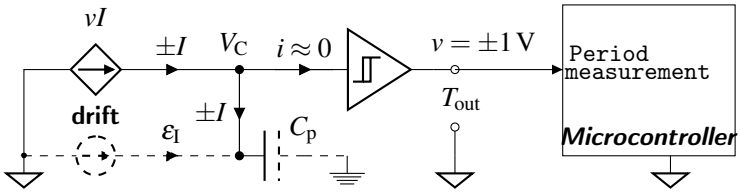


Fig. 3.3 Period modulation frontend block schematic with the drift current ε_{I} modeling the charge induction of the sensor plate mutual capacitance C_{p} with surrounding objects of different electrical potentials

I measuring the variation of its potential V_{C} against a fixed reference. The plate potential is defined by the electric charge density σ on the plate surface, S_{plate}, which balances the electric flux Φ_E from the environmental electric field $\mathbf{E}$ on the same surface

$$\sigma = \Phi_E, \qquad \Phi_E = \oiint_{S_{\mathrm{plate}}} \mathbf{E} \cdot \mathrm{d}\mathbf{S}. \tag{3.2}$$

$\mathbf{E}$ variations change σ, hence induce currents ε_{I} [11] that add algebraically to the frontend driver current I. Here we focus mostly on the slow, quasi-constant $\mathbf{E}$ variations, e.g., created during the dispersion of environmental charge buildups, and on measuring frontend rejection of the noise from the induced drift currents, ε_{I}. Significant charge can accumulate during common industrial or home activities, e.g., walking on carpeted floors can generate electrostatic voltages up to 15 kV [57].

Period modulation constant-current frontends (see Fig. 3.3) repeatedly charge-discharge the mutual plate capacitance C_{p} (made with the surrounding objects) with a current $\pm I$ from a constant current source, or an exponential current from a voltage source through a resistor. Under constant current I, the plate voltage V_{C} changes linearly (see the dashed plot in Fig. 3.4)

$$C_{\mathrm{p}} = \frac{I\Delta t}{\Delta V_{\mathrm{C}}}, \qquad \Delta V_{\mathrm{C}} = \frac{I}{C_{\mathrm{p}}}\Delta t. \tag{3.3}$$

When V_{C} reaches the thresholds of a hysteresis comparator (V_{TL}, V_{TH}), the sign of the current I changes to satisfy $V_{\mathrm{TL}} \le V_{\mathrm{C}} \le V_{\mathrm{TH}}$. With constant C_{p} and no noise, the charge and discharge times are identical

$$C_{\mathrm{p}}\frac{V_{\mathrm{TH}} - V_{\mathrm{TL}}}{I} = C_{\mathrm{p}}\frac{V_{\mathrm{TL}} - V_{\mathrm{TH}}}{-I} = \frac{T_{\mathrm{N}}}{2}, \tag{3.4}$$

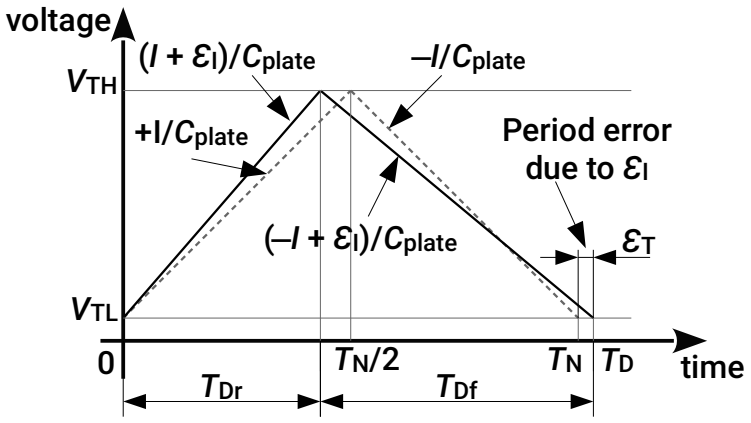


Fig. 3.4 Period modulation frontend errors due to the drift current, ε_I

C_p can be calculated from the oscillation period T_N

$$C_p = \underbrace{\frac{I}{2\,(V_{TH} - V_{TL})}}_{\text{constant}} T_N. \tag{3.5}$$

and the sensitivity of the output frequency to changes of C_p is

$$\frac{\partial f_N}{\partial C_p} = -\frac{I}{2\,(V_{TH} - V_{TL})} \cdot \frac{1}{C_p^2}, \qquad f_N = \frac{1}{T_N}. \tag{3.6}$$

However, as shown in Fig. 3.4, an external quasi-constant drift current ε_I unbalances the durations of the rising T_{D_R} and falling T_{D_F} ramps. From (3.4) we have

$$T_{D_R} = C_p \frac{V_{TH} - V_{TL}}{I + \varepsilon_I}, \quad T_{D_F} = C_p \frac{V_{TL} - V_{TH}}{-I + \varepsilon_I} \tag{3.7}$$

and the oscillation period T_D becomes

$$T_D = T_{D_R} + T_{D_F} = 2\,(V_{TH} - V_{TL}) \underbrace{\frac{I}{I^2 - \varepsilon_I^2}}_{\text{not constant}} C_p. \tag{3.8}$$

The relative period error is obtained from (3.5) and (3.8)

$$\varepsilon_T = \frac{T_D - T_N}{T_N} = \frac{\varepsilon_I^2}{I^2 - \varepsilon_I^2}. \tag{3.9}$$

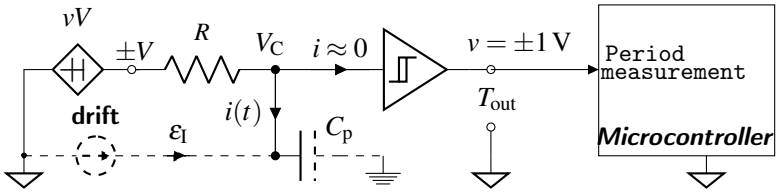


Fig. 3.5 RC period modulation frontend block schematic with the drift current ε_I modeling the charge induction of the sensor plate mutual capacitance C_p with surrounding objects of different electrical potentials

Hence, a constant drift current ε_I always increases the oscillation period T_D of the frontend, leading to measurement errors when calculating C_p using (3.5).

The oscillation period of period modulating frontends using RC astable multivibrators, rather than based on constant currents, is affected by similar errors. The block schematic in Fig. 3.5 shows how the frontend charges and discharges the plate capacitance C_p between the voltage thresholds of a hysteresis comparator through a resistor R from two constant dc voltages, $\pm V$. The charging current $i(t)$ is

$$i(t) = \frac{V - V_\mathrm{C}(t)}{R}, \quad \frac{\mathrm{d}V_\mathrm{C}(t)}{\mathrm{d}t} = \frac{i(t)}{C_\mathrm{p}} = \frac{V - V_\mathrm{C}(t)}{RC_\mathrm{p}}. \tag{3.10}$$

Solving for V_C we obtain the well-known RC charge expression

$$V_\mathrm{C}(t) = (V_\mathrm{TL} - V)\, e^{-\frac{t}{\tau}} + V, \qquad \tau = RC_\mathrm{p} \tag{3.11}$$

where V_TL is the voltage of the lowest threshold of the hysteresis comparator and V is the source (charging) voltage.

Assuming distinct dc voltages for charging, V_H, and discharging, V_L, the charging, T_Nr, and the discharging, T_Nf, times are

$$T_\mathrm{Nr} = -\tau \ln \frac{V_\mathrm{TH} - V_\mathrm{H}}{V_\mathrm{TL} - V_\mathrm{H}} \tag{3.12}$$

$$T_\mathrm{Nf} = -\tau \ln \frac{V_\mathrm{TL} - V_\mathrm{L}}{V_\mathrm{TH} - V_\mathrm{L}} \tag{3.13}$$

and the noise-free oscillation period is

$$\begin{aligned} T_{\mathrm{N}} = T_{\mathrm{Nr}} + T_{\mathrm{Nf}} &= -\tau \ln \frac{(V_{\mathrm{TH}} - V_{\mathrm{H}}) \cdot (V_{\mathrm{TL}} - V_{\mathrm{L}})}{(V_{\mathrm{TL}} - V_{\mathrm{H}}) \cdot (V_{\mathrm{TH}} - V_{\mathrm{L}})} \\ &= -\tau \ln \frac{K_{\mathrm{a}}}{K_{\mathrm{b}}} \qquad K_{\mathrm{a}} < 0, \quad K_{\mathrm{b}} < 0. \end{aligned} \tag{3.14}$$

An induced quasi-constant drift current ε_{I} (see Fig. 3.5) algebraically adds to $i(t)$ in (3.10). During charging we have

$$\frac{\mathrm{d}V_{\mathrm{C}}(t)}{\mathrm{d}t} = \frac{\frac{V_{\mathrm{H}} - V_{\mathrm{C}}(t)}{R} + \varepsilon_{\mathrm{I}}}{C_{\mathrm{p}}} = \frac{V_{\mathrm{H}} - V_{\mathrm{C}}(t) + \varepsilon_{\mathrm{V}}}{RC_{\mathrm{p}}}, \quad \varepsilon_{\mathrm{V}} = R\,\varepsilon_{\mathrm{I}} \tag{3.15}$$

and solving for V_{C}

$$V_{\mathrm{C}}(t) = (V_{\mathrm{TL}} - V_{\mathrm{H}} - \varepsilon_{\mathrm{V}})\, e^{-\frac{t}{\tau}} + V_{\mathrm{H}} + \varepsilon_{\mathrm{V}}. \tag{3.16}$$

Thus, the charging, $T_{\mathrm{D_R}}$, and the discharging, $T_{\mathrm{D_F}}$, times considering a quasi-constant drift current ε_{I} are

$$T_{\mathrm{D_R}} = -\tau \ln \frac{V_{\mathrm{TH}} - V_{\mathrm{H}} - \varepsilon_{\mathrm{V}}}{V_{\mathrm{TL}} - V_{\mathrm{H}} - \varepsilon_{\mathrm{V}}}, \tag{3.17}$$

$$T_{\mathrm{D_F}} = -\tau \ln \frac{V_{\mathrm{TL}} - V_{\mathrm{L}} - \varepsilon_{\mathrm{V}}}{V_{\mathrm{TH}} - V_{\mathrm{L}} - \varepsilon_{\mathrm{V}}}, \tag{3.18}$$

the frontend oscillation period is

$$\begin{aligned} T_{\mathrm{D}} = T_{\mathrm{D_R}} + T_{\mathrm{D_F}} &= \\ &= -\tau \ln \left(\frac{V_{\mathrm{TH}} - V_{\mathrm{H}} - \varepsilon_{\mathrm{V}}}{V_{\mathrm{TL}} - V_{\mathrm{H}} - \varepsilon_{\mathrm{V}}} \cdot \frac{V_{\mathrm{TL}} - V_{\mathrm{L}} - \varepsilon_{\mathrm{V}}}{V_{\mathrm{TH}} - V_{\mathrm{L}} - \varepsilon_{\mathrm{V}}} \right) = \\ &= -\tau \ln \frac{|K_{\mathrm{a}}| - \varepsilon_{\mathrm{V}}^2}{|K_{\mathrm{b}}| - \varepsilon_{\mathrm{V}}^2}, \end{aligned} \tag{3.19}$$

and the relative oscillation period error due to drift current ε_{I}

$$\varepsilon_{\mathrm{T}} = \frac{T_{\mathrm{D}} - T_{\mathrm{N}}}{T_{\mathrm{N}}} = \frac{\ln \frac{|K_{\mathrm{a}}| - \varepsilon_{\mathrm{V}}^2}{|K_{\mathrm{b}}| - \varepsilon_{\mathrm{V}}^2}}{\ln \frac{|K_{\mathrm{a}}|}{|K_{\mathrm{b}}|}} - 1 = \frac{\overbrace{\ln \frac{1 - \frac{\varepsilon_{\mathrm{V}}^2}{|K_{\mathrm{a}}|}}{1 - \frac{\varepsilon_{\mathrm{V}}^2}{|K_{\mathrm{b}}|}}}^{\text{not constant}}}{\ln \frac{|K_{\mathrm{a}}|}{|K_{\mathrm{b}}|}}. \tag{3.20}$$

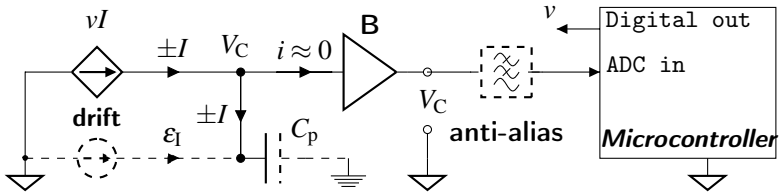


Fig. 3.6 Slope modulation frontend block schematic with the drift current ε_{I} modeling the charge induction of the sensor plate mutual capacitance C_{p} with surrounding objects of different electrical potentials

In a first approximation, $\varepsilon_{\mathrm{T}} \propto \ln\left(1-\varepsilon_{\mathrm{I}}^2\right)$ (since $\varepsilon_{\mathrm{V}} = R\,\varepsilon_{\mathrm{I}}$), higher than for constant current period modulation frontends.

3.2 Drift Rejection Method

The operation of the proposed slope modulation frontend shown in Fig. 3.6 is similar to the period modulation frontend shown in Fig. 3.3. The timing of the charge-discharge cycles is constant because we use the ramp slopes for C_{p} calculation, which allows to better reject drift noise.

The V_{C} ramp slope S is inversely proportional to C_{p} in (3.3)

$$S = \frac{\Delta V_{\mathrm{C}}}{\Delta t} = \frac{I}{C_{\mathrm{p}}}. \tag{3.21}$$

and a constant drift current ε_{I} changes both the rising S_{r} and the falling S_{f} slopes

$$S_{\mathrm{r}} = \frac{I+\varepsilon_{\mathrm{I}}}{C_{\mathrm{p}}}, \qquad S_{\mathrm{f}} = \frac{-I+\varepsilon_{\mathrm{I}}}{C_{\mathrm{p}}}. \tag{3.22}$$

But the average of the slope magnitudes, S_{a}, is invariant to ε_{I}

$$S_{\mathrm{a}} = \frac{|S_{\mathrm{r}}|+|S_{\mathrm{f}}|}{2} = \frac{1}{2}\left(\frac{I+\varepsilon_{\mathrm{I}}}{C_{\mathrm{p}}} - \frac{-I+\varepsilon_{\mathrm{I}}}{C_{\mathrm{p}}}\right) = \frac{I}{C_{\mathrm{p}}}. \tag{3.23}$$

Moreover, the measurement sensitivity to C_{p} is comparable with the period modulation frontend in (3.6)

$$\frac{\partial S_{\mathrm{a}}}{\partial C_{\mathrm{p}}} = -\frac{I}{C_{\mathrm{p}}^2}. \tag{3.24}$$

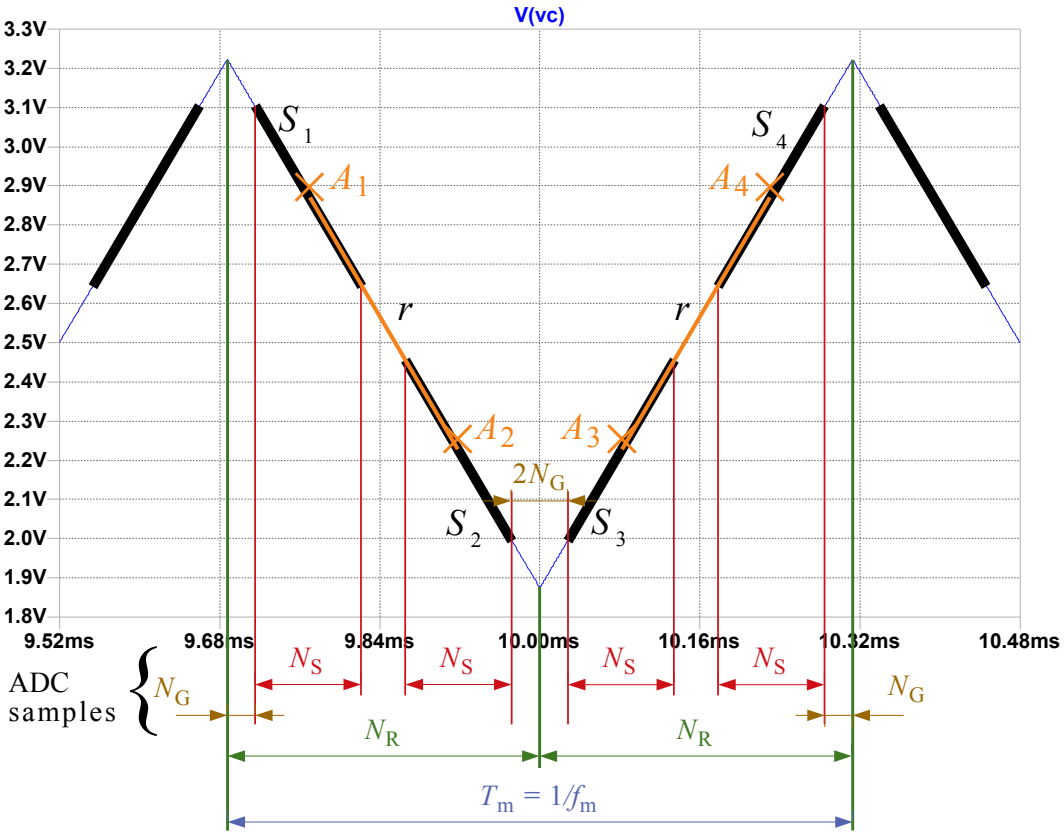


Fig. 3.7 Slope measurement elements and timings

Hence, by measuring the slope of two adjacent charge-discharge ramps we can calculate C_p using (3.23) and reject as common mode signals the errors from quasi-constant drift currents, ε_I, as from other drift sources, e.g., flicker $(1/f)$ noise.

Importantly, drift rejection depends on how well the constant ε_I (and noise in general) is matched on two adjacent ramps, and on the component nonlinearities and value mismatches.

3.2.1 Measurement Method

Fig. 3.7 shows the main elements used to measure the slopes of the charge-discharge ramps to calculate C_p using (3.23). The measurement period T_m is accurately set by a microcontroller (signal v in Fig. 3.6), and should be at most half of the shortest sensor signal period according to the Nyquist–Shannon sampling theorem [104]. T_m is synchronized with an ADC that acquires N_R samples of V_C on each ramp. The

voltages V_{A} of two A points must be determined to calculate the slope of each ramp. Each A is the midpoint of an N_{S} sample segment S. To lower the quantization noise through oversampling and decimation [14], we optimize N_{S} to a power of four

$$V_{\mathrm{A}_i} = \frac{1}{\frac{N_{\mathrm{S}}}{2}} \sum_{j=1}^{N_{\mathrm{S}}} V_{\mathrm{S}_{i,j}},\ i \in \{1,2,3,4\},\ N_{\mathrm{S}} = 4^n,\ n \in \mathbb{N} \tag{3.25}$$

where V_{S_i} are the samples of the N_{S} ramp in the segment S_i.

Note that the frontend in Fig. 3.6 controls only the amplitude of V_{C}, through the current I and the period T_{m}, but not its absolute value. Circuit asymmetries or environmental effects may accumulate over time and bring V_{C} partly into the frontend nonlinearity region, which may distort its extremes. Anti-aliasing filters (see Fig. 3.6) may also distort the V_{C} ramp ends. Hence, for better measurement accuracy the segments S must be long and well spaced, but also far enough from the ramp ends to skip distortions (N_{G} skipped samples in Fig. 3.7).

The time between the two A points on each ramp is

$$T_{\mathrm{A}_{1,2}} = T_{\mathrm{A}_{3,4}} = \frac{T_{\mathrm{m}}}{2N_{\mathrm{R}}} \left[N_{\mathrm{R}} - 2\left(N_{\mathrm{G}} + \frac{N_{\mathrm{S}}}{2} \right) \right]. \tag{3.26}$$

Eq. (3.26), (3.21) give the rising, S_{r}, and falling, S_{f}, ramp slopes

$$S_{\mathrm{r}} = \frac{2N_{\mathrm{R}} \left(V_{\mathrm{A}_4} - V_{\mathrm{A}_3} \right)}{T_{\mathrm{m}} \left(N_{\mathrm{R}} - 2N_{\mathrm{G}} - N_{\mathrm{S}} \right)},\ S_{\mathrm{f}} = \frac{2N_{\mathrm{R}} \left(V_{\mathrm{A}_2} - V_{\mathrm{A}_1} \right)}{T_{\mathrm{m}} \left(N_{\mathrm{R}} - 2N_{\mathrm{G}} - N_{\mathrm{S}} \right)} \tag{3.27}$$

then we obtain C_{p} from (3.23)

$$C_{\mathrm{p}} = \frac{I\, T_{\mathrm{m}}}{V_{\mathrm{A}_4} - V_{\mathrm{A}_3} - \left(V_{\mathrm{A}_2} - V_{\mathrm{A}_1} \right)} \cdot \frac{N_{\mathrm{R}} - 2N_{\mathrm{G}} - N_{\mathrm{S}}}{N_{\mathrm{R}}}. \tag{3.28}$$

Hence, from the frontend design parameters I, T_{m}, N_{R}, N_{S}, N_{G} and the voltages V_{A} of midpoints A of the acquisition segments S we can calculate the plate capacitance C_{p}.

3.3 Simulation and Experimental Testing

Plate capacitance slow measurement drifts may indicate actual capacitance variations (e.g., due to changes of air humidity), or measurement interface errors upon

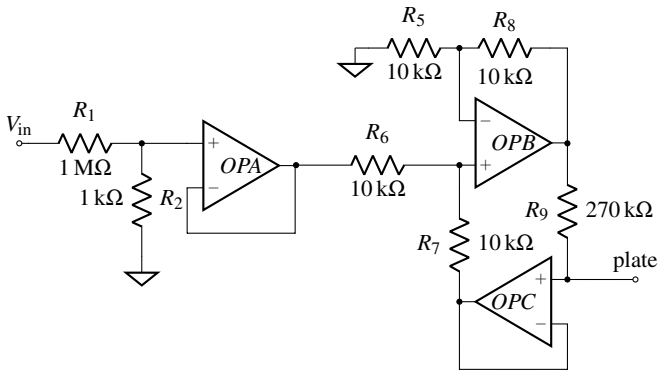


Fig. 3.8 Improved Howland current source modeling the noise current

changes of measurement conditions (e.g., charge induction currents from variable environmental electric fields from charge buildup or leakage). While the former variations can be usually independently measured and compensated, the latter can hardly be detected and the measurement interface should be intrinsically insensitive.

In Section 3.1 the slope modulation measurement interface rejection capability of the low frequency drift currents is analytically demonstrated. Here analytic results are validated through simulations and laboratory experiments for drift currents at different frequencies.

For grounded capacitors [75], changes of electrical potential of the grounded plate (surrounding objects in our case) often determine charge induction currents. Because mutual plate capacitance and relatively large voltage variations are hard to produce and control, the charge induction current is injected directly into the sensor plate node using a voltage-controlled current source (see Fig. 3.8).

The slope modulation frontend (see Fig. 3.6 and Fig. 3.9) uses a high input impedance differential amplifier made of operational amplifiers `OPB` and `OPD` to measure the voltage across the current sensing resistor R_9. The result drives the `OPA` inverting input, while its non-inverting input is set to the constant reference voltage driven by a microcontroller through the R_1–R_2 voltage divider. Thus, the current through R_9 (hence through C_p) is

$$I = \frac{(V_{in} - 1.65\,\mathrm{V})\frac{R_2}{R_1+R_2} + 1.65\,\mathrm{V}}{R_9}, \tag{3.29}$$

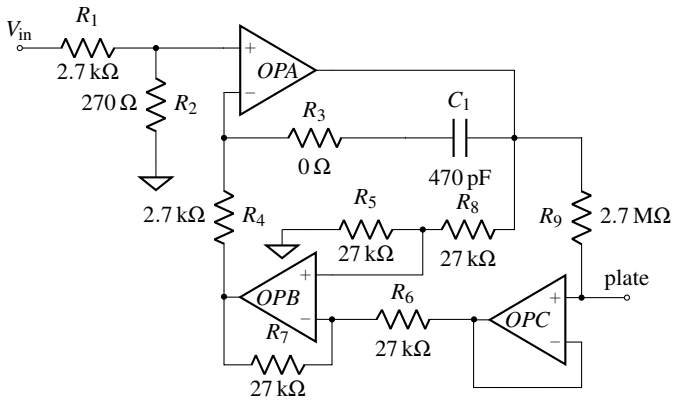


Fig. 3.9 Schematic of the slope modulation frontend

where V_{in} is a digital output assuming either 0 V (logic low) or supply voltage, 3.3 V (logic high). The same current source is used also for the constant current period modulation frontend.

The RC period modulation frontend uses an exponential current limited by a resistor R. Moreover, for the typically small C_{p}, practical R values increase the average current well beyond the constant current of the other two frontends.

3.3.1 Testing Parameters

The sensor plate capacitance is modeled with a constant $C_{\text{p}} = 10\,\text{pF}$ capacitor connected to the ground, comparable to the $16\,\text{cm} \times 16\,\text{cm}$ plate self-capacitance of the prototype.

The parameters of the slope modulation front-end (see Fig. 3.9) are set as discussed in Section 3.2.1. The measurement period T_{m} (see Fig. 3.7) must be long for the ADC to acquire many N_{R} samples per ramp (to increase accuracy and reduce quantization noise), yet short enough to keep V_{C} swings within `OPA` input limits. V_{C} swings also depend on the charge/discharge current magnitude, I. Thus, low C_{p} requires low I. But since I is set mainly by (see Fig. 3.6) the voltage divider R_1, R_2 and by R_9 value (3.29), the output voltage of the divider must be much higher than the `OPA` input voltage offset for low errors, and R_9 value is capped by practical values and to keep the driving impedance of the `plate` node relatively low, less susceptible to noise. Yet R_9 cannot be too low, because higher I values tend to decrease measurement sensitivity (3.24), as shown in Fig. 3.10.

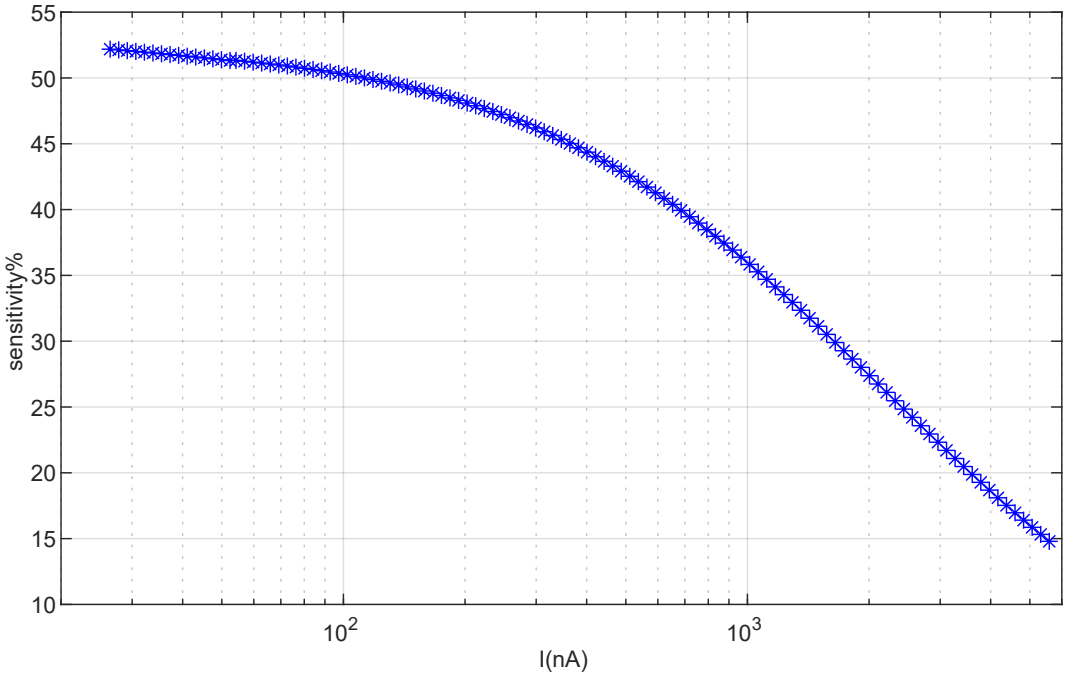


Fig. 3.10 Slope modulator sensitivity with the magnitude of the constant current I which charges/discharges the sensor plate

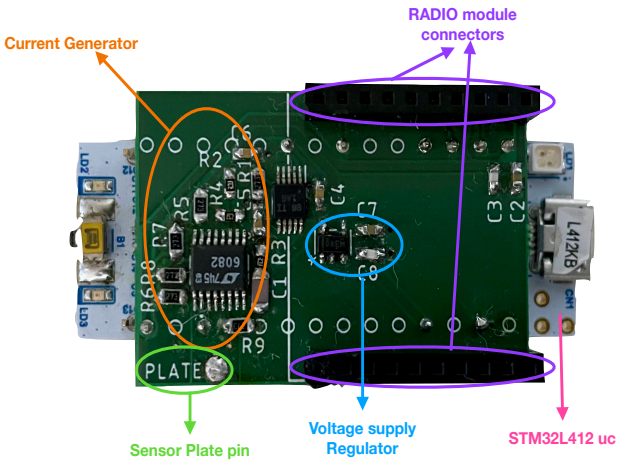


Fig. 3.11 Printed circuit board assembly of the slope modulation interface (top) with the STM32L412 microcontroller board (bottom)

The STM32L412 [31] low-power low-cost general purpose microcontroller board is selected to drive the prototype frontend. It provides sufficient onboard memory and processing power, a 5 Msample/s ADC, and a regulated 3.3 V supply. It plugs into the frontend (see Fig. 3.11) keeping the high sensitivity analog circuits distant and shielded from the noisy digital parts.

ADC acquisition is set at 2 Msample/s, more available on low-end microcontrollers. Table 3.1 lists, with the considerations above and in Section 3.2.1, the T_m periods using at least half of the ADC samples, the resolution bits n gained from oversampling, and the peak-to-peak V_C for $I = 40\,\text{nA}$. An integer number of ADC samples per ramp (see Fig. 3.9) is split in two $N_S = 4^n$ segments for quantization noise reduction through oversampling, two N_G samples covering non-linear regions at ramp ends, and N_R samples between the N_S segments to improve the slope measurement accuracy. T_m is chosen with $n = 4\,\text{bit}$ from oversampling and a high $V_C = 2\,V_{pp}$ to reduce quantization noise. Other T_m may require higher I values, which may lower the frontend sensitivity (see Fig. 3.10). The parameters should be optimized when the plate capacitance changes significantly, e.g., for different plate dimensions.

For better comparison, the constant current period modulation frontend uses the same current, $I = 40\,\text{nA}$, and measurement period, $T_m = 1\,\text{ms}$. However, the RC period modulation frontend must use higher average currents (and oscillation frequencies around $f_m \approx 2200\,\text{Hz}$) for practical R values.

The drift noise in simulation and experiments is modeled using a Howland [28] (see Fig. 3.8) voltage controlled current generator. The errors induced on capacitance measurement by frontends are measured for 3 decade frequencies, 100 Hz to 0.1 Hz (the low end relevant for drift rejection analysis), and amplitudes 1 nA (the lowest reliably distinguishable from noise) and 2 nA.

Table 3.1 Slope modulation frontend possible measurement frequencies ($1/T_m$) for 2 Msample/s sampling rate, oversampling bits (n), unused samples per ramp ($2N_G + r$) and their percentage of total, and plate capacitance voltage swing (V_C) for $I = 40\,\text{nA}$

Measurement frequency (Hz)	**Over-sampling** (bit)	**Unused samples per ramp**	**Ramp coverage** (%)	V_C **amplitude** ($I = 40\,\text{nA}$) (V)
1600	4	113	82	1.25
6250	3	32	80	0.32
5000	3	72	64	0.4
1250	4	288	64	1.6
4000	3	122	52	0.5
1000	4	488	52	2.0

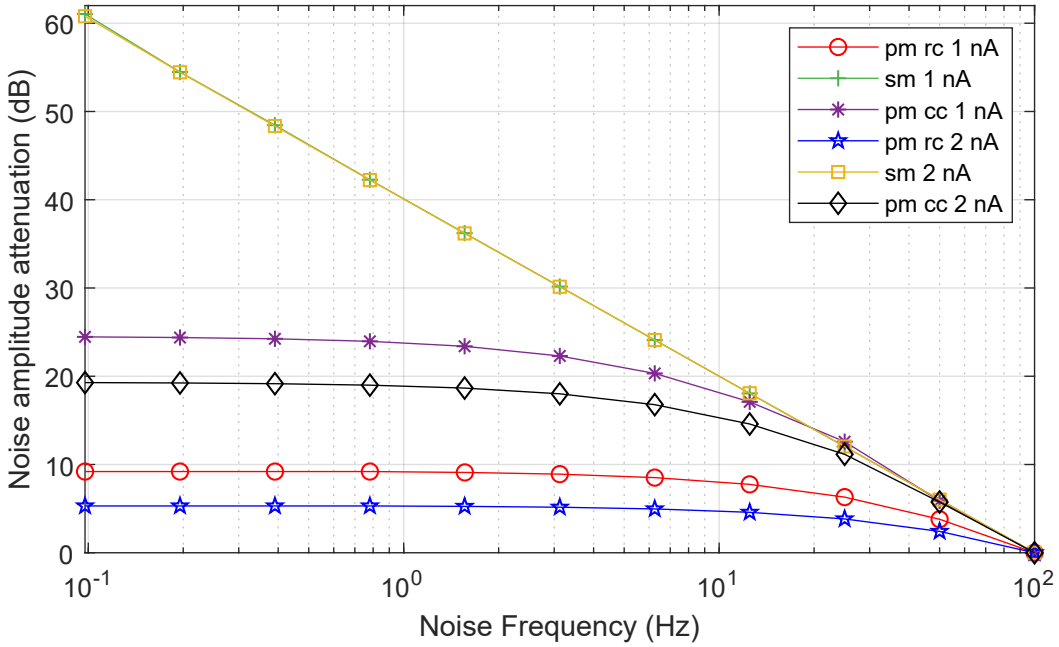


Fig. 3.12 Drift noise attenuation by slope modulation (sm), constant current period modulation (pm cc), and RC period modulation (pm rc) frontends for frequencies 0.1 Hz to 100 Hz and amplitudes 1 nA and 2 nA

3.3.2 Simulation Results

Using the parameters discussed in Section 3.3.1, drift noise rejection of all frontends in LTspice [74] simulations are analyzed and compared.

Fig. 3.12 comparatively shows the noise rejection for all three frontends. The attenuation at low frequencies is plotted against a reference set at a much higher frequency, 100 Hz. Note that for the slope modulation frontend the quantization error from the 12 hardware bits of the ADC and the 4 bits obtained from oversampling and decimation (see Section 3.3.1) is also implemented in simulation. For the period modulation frontends we use instead the double precision numerical representation of the simulator.

Above a certain frequency, the rejection of all frontends decreases linearly with the increase of the noise frequency. Below that, the rejection of both period modulation frontends has an inflection (at higher frequencies for the RC period modulation frontend), beyond which the noise rejection flattens. This is because these frontends have limited attenuation for constant (dc) drift currents (as discussed in Section 3.1, it is lower for the RC period modulation), which becomes dominant. However, noise

attenuation of the slope modulation frontend remains inversely proportional to frequency, according to the theoretically infinite attenuation of constant (dc) noise (see Section 3.1).

It can be also noticed that low frequency attenuation of period modulation frontends (in the flattened zone) depends on noise amplitude. This is because the attenuation of constant (dc) noise depends on the intensity of the noise current (ε_I in Section 3.1).

3.3.3 Experimental Results

The analytical and simulation results are tested experimentally using implementations of the frontends and noise generator.

The stability of the current source is important for both the slope modulator and constant current period modulator frontends. The circuit in Fig. 3.9 is used for both. It operates in uncontrolled environmental conditions (temperature, humidity, ...) without compensations, with the sensor plate replaced by a constant 10 pF capacitor, supplied at 3.3 V by an `REF2033AIDDCR` [96] accurate voltage regulator providing both the supply and the half-reference voltages (see Fig. 3.11). The measurement is stable over extended time periods as shown in Fig. 3.13.

After checking the stability, the frontends are first characterized using a lumped capacitor instead of the sensor plate to reduce the uncontrolled environmental noise. For practical reasons, the frequency range is limited from 25 Hz down to the lowest frequency where the injected noise can be distinguished from the total measurement noise (low frequencies are the most interesting for analyzing drift noise rejection of the frontends).

Fig. 3.14 shows comparatively both the experimental (dashed line) and the simulation (solid line) results for all three frontends, for a drift noise amplitude of 1 nA. As mentioned above, the new reference is the attenuation at 25 Hz instead of the attenuation at 100 Hz that is used in the simulation results in Fig. 3.12. Albeit noisier, the experimental results follow closely the simulation results. They reproduce the inflection and flattening behavior for the period modulation frontends (see Section 3.3.2), and the proportional increase of the drift noise attenuation with the decrease of the noise frequency (up to 1.56 Hz) of the slope modulation frontend.

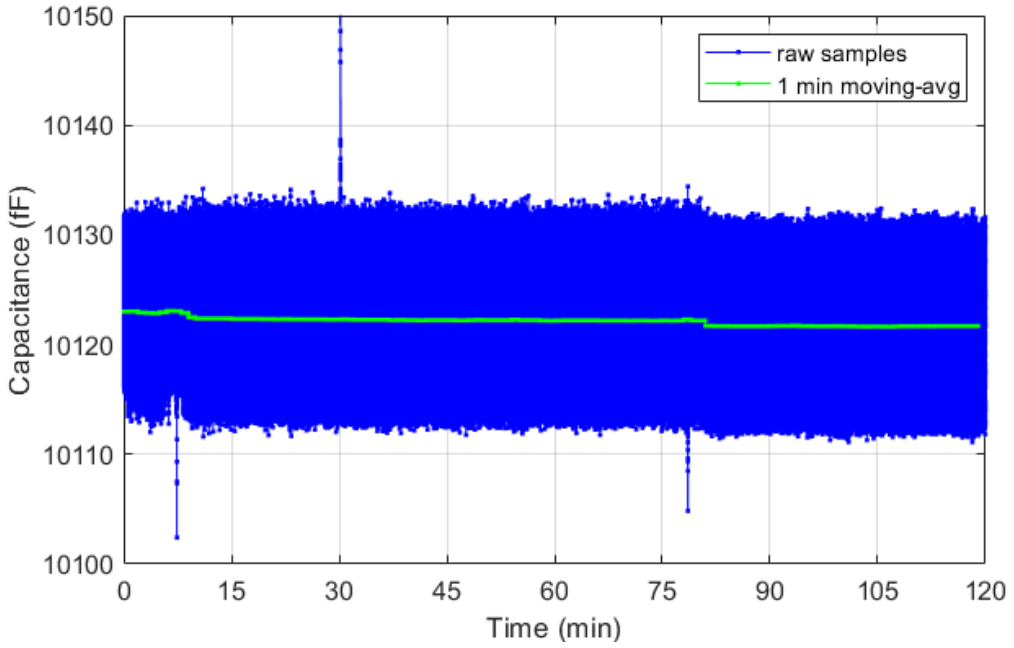


Fig. 3.13 Stability of the mean value of capacitance for two hours of acquisition of the slope modulator frontend with a fixed capacitor of 10 pF

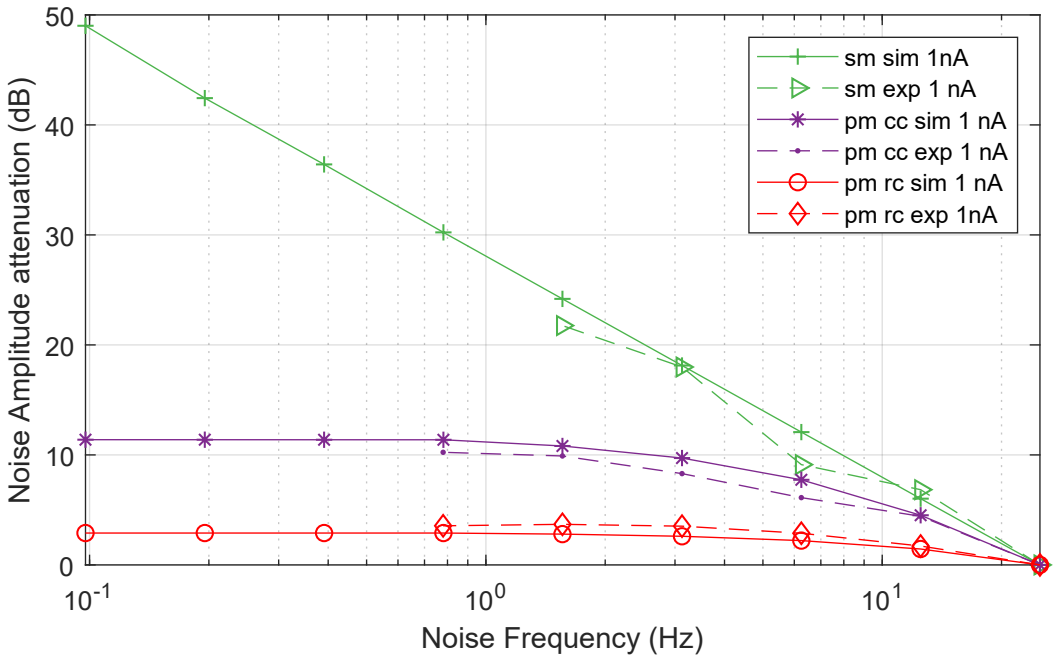


Fig. 3.14 Simulation and experimental results for the rejection of 1 nA amplitude sinusoidal noise with frequency

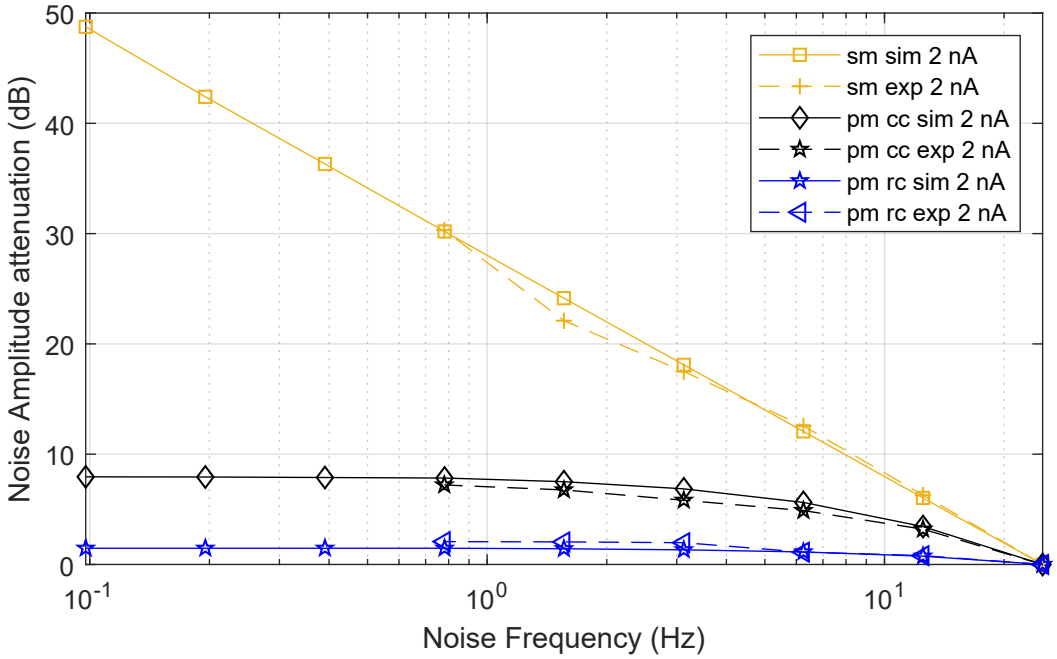


Fig. 3.15 Simulation and experimental results for the rejection of 2 nA amplitude sinusoidal noise with frequency

Fig. 3.15 shows the same comparison of the simulation (solid line) and experimental (dashed line) results for all three frontends for a higher drift noise amplitude, of 2 nA. The effects can be measured with better accuracy, thus the experimental data match better the simulation results for all frontends, which validates the analytic analysis and the simulation results.

Then the rejection of uncontrolled environmental noise (from sensor circuitry and the environment, electromagnetic, temperature, humidity, etc.) is measured for all frontends. A 16 cm $\times$ 16 cm sensing plate is connected to each frontend and measurements for 12 h (from 6:30 PM to 6:30 AM) are collected in the laboratory room. The frequency analysis results are shown in Fig. 3.16. The expected inflection point below which the 1/f noise becomes dominant can be seen for all frontends. The RC period modulation frontend is the noisiest over the whole spectrum, it has the inflection point at the lowest frequency (around 0.05 Hz), and the highest noise power below that. The constant current period modulation frontend has lower noise power at higher frequencies, the inflection point at a higher frequency (around 0.25 Hz), and shows an improved noise attenuation around 0.01 Hz and below. The slope modulation frontend has comparable noise envelope at higher frequencies and

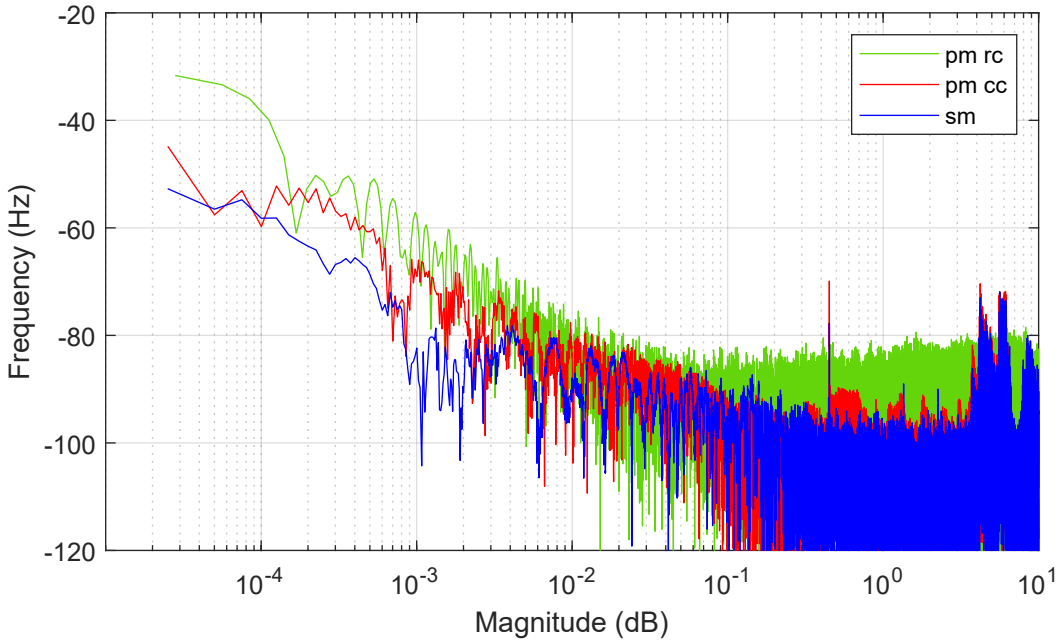


Fig. 3.16 Frequency analysis of 12 h measurements collected in uncontrolled laboratory conditions by slope modulation (sm), RC period modulation (pm rc), and constant current period modulation (pm cc) frontends

an inflection point roughly at the same frequency, yet the best noise attenuation at lower frequencies.

Although this is beyond the scope of the comparison of the frontend intrinsic performance, it can be noticed that unlike for the period modulation frontends, for the slope modulation frontend an antialiasing filter can be included before the ADC to further improve noise rejection, as shown in Fig. 3.6.

3.4 Sensitivity Characterization

Sensor noise is usually directly correlated to sensor sensitivity. Thus, it must be checked if a higher sensitivity is in fact the cause of the higher noise of the period modulation frontends.

The frontend noise is evaluated through two main parameters. First, the level below which the person movements produce no discernible changes of the frontend readings. Second, the stability of the frontend readings while the person remains still at various distances in front of the sensor.

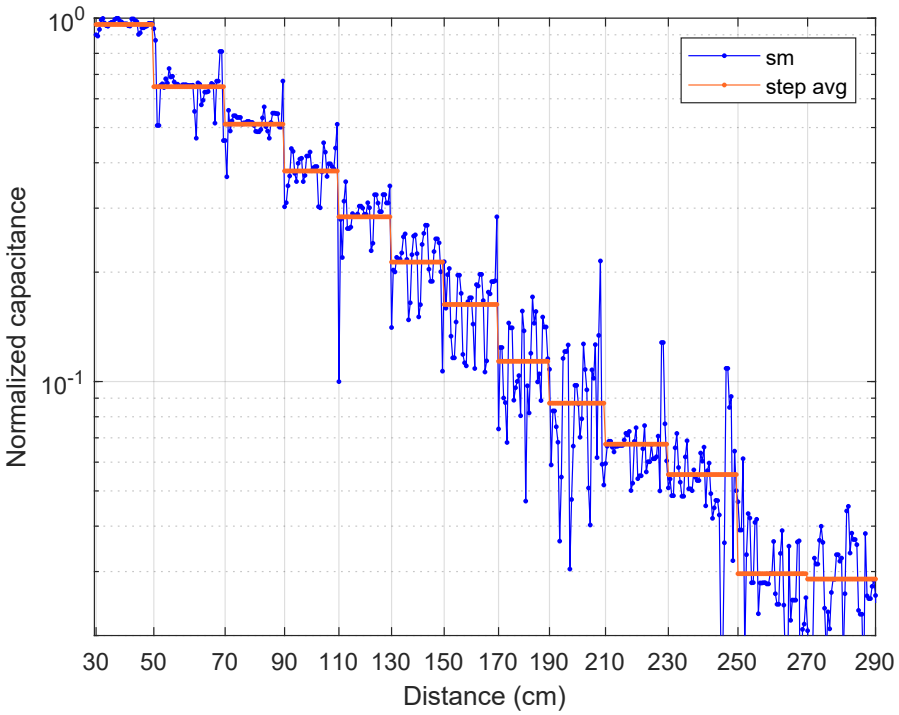


Fig. 3.17 Plate capacitance measured 4 times/s with the slope modulation frontend for body-plate distances changing by 20 cm every 10 s

For these experiments, the input of each frontend is connected to a 16 cm × 16 cm copper plate, mounted at chest level and several meters away from the surrounding objects. The person stations 10 s in each position, spaced 20 cm along a straight path orthogonal to the plate, at distances from 290 cm to 30 cm. The three frontends are characterized in sequence, in the same conditions, all sampling at 4 sample/s, which is suitable for monitoring indoor person movements [33].

Fig. 3.17 shows the sensitivity characterization for the slope modulation frontend, as the logarithm of the normalized capacitance on the Y axis function of the distance between the person and the sensor plate on the X axis. Capacitance is normalized for better comparison between frontends, because the absolute values depend on uncontrolled stray capacitances from PCB and components. The average of the readings is also drawn for each stationary position of the person to reduce the influence of the noise. The top of the scale is set at 0.1 % higher than the highest average level, and the bottom of the scale at 0.1 % lower than the lowest average level.

First, it can be observed that the sensitivity of the slope modulation frontend extends up to 230 cm, about 10 times the diagonal of the sensor plate. Second, the

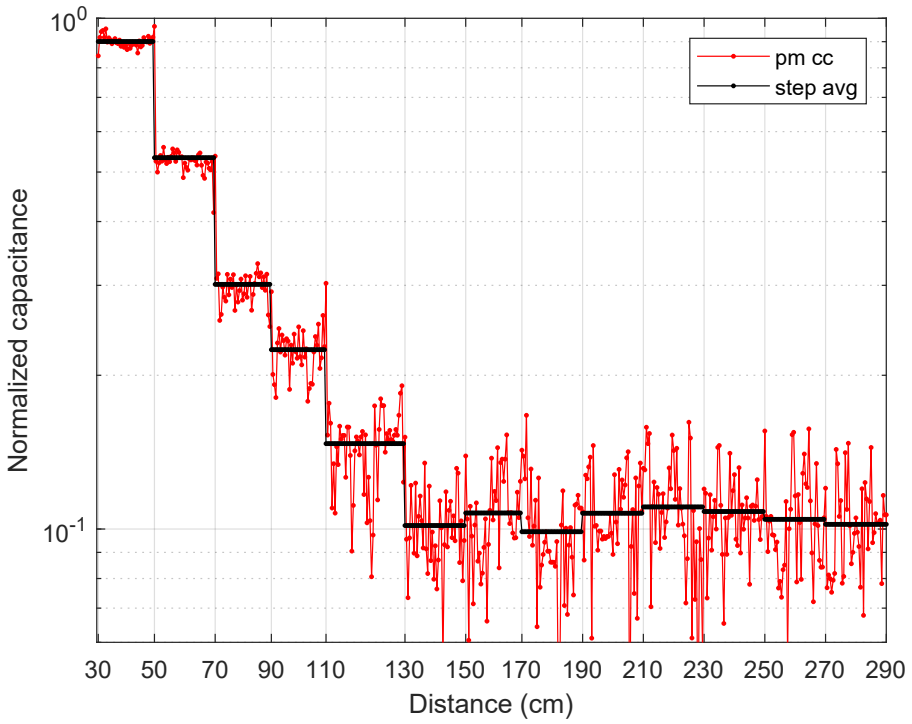


Fig. 3.18 Sensor capacitance measured 4 times/s with the constant current period modulation frontend for body-sensor distances changing 20 cm every 10 s

plate capacitance decreases linearly (on logarithmic scale) with the distance for the whole sensitivity range. This indicates a power dependence of capacitance on the distance to the person, in line with other empirical results in the literature [123]. Third, barring spikes when the person changes position, the measurement noise markedly increases beyond 130 cm, and around 250 cm it exceeds the signal.

Fig. 3.18 shows the results for the constant current period modulation frontend. For better comparison, it uses similar settings and considerations as Fig. 3.17. A quasi-linear capacitance-distance dependence (on logarithmic scale) is characterizing up to 110 cm–130 cm. Noise visibly afflicts measurements beyond 110 cm and shortly after it limits the sensitivity.

Fig. 3.19 shows the results for the experimental evaluation of the sensitivity of the RC period modulation frontend, using similar settings and considerations as Fig. 3.17 and Fig. 3.18. The linear capacitance-distance dependence (on logarithmic scale) is visible also here up to 90 cm–110 cm. Noise markedly afflicts the readings beyond 90 cm, then quickly limits the sensitivity.

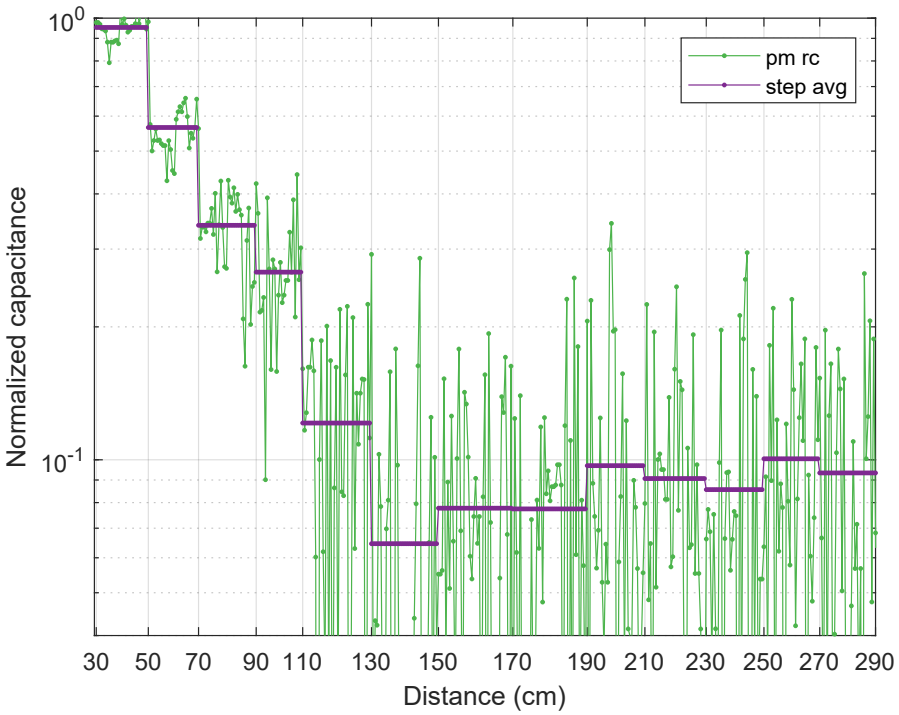


Fig. 3.19 Sensor plate capacitance measured 4 times/s with the RC period modulation frontend for body-sensor distances changing by 20 cm every 10 s

The experimental sensitivity characterizations show that the slope modulation frontend is the least afflicted by noise, thus extending the sensing range to become roughly twice as large as the frontends based on period modulation. The latter are more afflicted by noise, which limits early their sensing ranges as well as the accuracy at closer distances, especially for the RC period modulation frontend.

Fig. 3.20 comparatively shows the frontend sensitivities calculated as the absolute value of the capacitance reading variation between two adjacent person positions. For all frontends, the sensitivity decreases exponentially with the distance (linearly on logarithmic scale) until it abruptly drops at noise level. The period modulation frontend sensitivities decrease faster and are limited earlier by the higher noise level. Instead, the sensitivity of the slope modulation frontend decreases more slowly and is limited by noise at a much longer sensing distance.

Overall, the experimental results show that the slope modulation frontend is less afflicted by noise and has higher sensitivity, thus longer sensing range than the period modulation frontends.

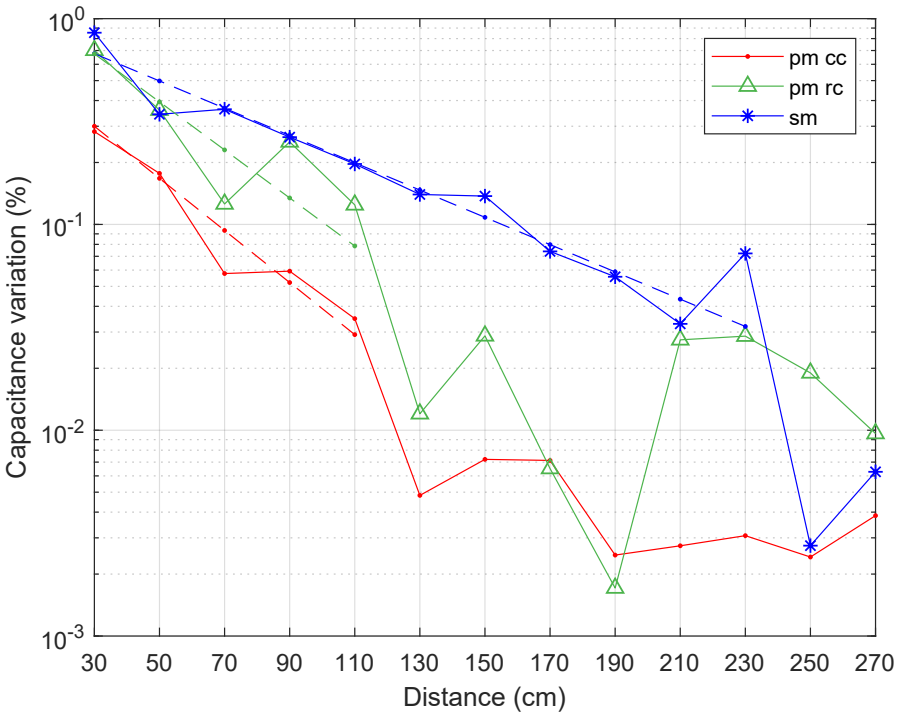


Fig. 3.20 Frontend sensitivity function of person distance for slope modulation (sm) and period modulation [using RC (pm rc) and constant current (pm cc)]

Chapter 4

Neural Network Processing for Human Indoor Tracking

Raw sensor data is often contaminated by various forms of noise, including random fluctuations, interference, and measurement errors. Post-processing techniques, such as signal filtering and statistical averaging, are instrumental in mitigating these noise sources, resulting in cleaner and more reliable estimates of location.

In addition to noise, sensors may exhibit inherent inaccuracies or biases that can adversely impact the accuracy of location estimates. Post-processing methods, such as calibration and error modeling, play a crucial role in rectifying these imperfections and enhancing the overall accuracy of the localization system.

Post-processing techniques assume a critical role in refining raw sensor data, addressing errors, integrating information from multiple sources, adapting to dynamic environmental conditions, enhancing system robustness, and optimizing performance metrics within indoor localization systems.

Specifically, ML models have the capability to learn from indirect signals or environmental features, thereby improving localization accuracy in non-line-of-sight scenarios, and they can capture complex relationships between sensor data and location, leading to more precise localization estimates. Furthermore, ML techniques can effectively filter out noise, thereby enhancing the reliability and robustness of the localization system.

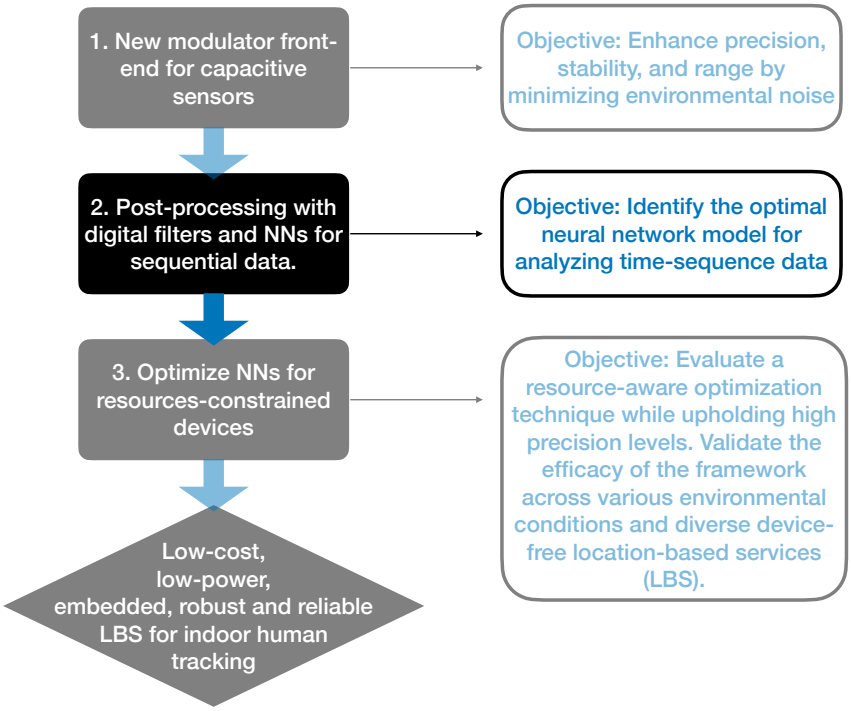


Fig. 4.1 Flowchart illustrating the organization of work and outlining objectives.

Tariq, Lazarescu, and Lavagno [115] evaluate several NN architectures for indoor person localization and tracking with data collected from a tagless localization system with four single-plate long-range capacitive sensors operating in load mode [41], mounted in the middle of the four sides of a 3 x 3 experimental space. The sensors are noisy and have a pronounced nonlinear characteristic, exposing the NN abilities to denoise, infer, and generalize both the location, for which the 1DCNN excels, and the human motion dynamics, for which the LSTM gives the best results. However, the pooling operations used in CNNs are known to lose important relations [4].

We explore two NNs designed for or adapted to sequential data analysis without pooling, the TCN [9, 45, 66] and the CAPS [59, 112]. Ye et al. [126] have used CAPS for indoor localization using a Wi-Fi network spread over 3 rooms with an average error of 0.68 m, outperforming ML based on CNN, SVM, CNN with stacked autoencoders, and KNN. Jia et al. [53] used TCN for indoor localization with Wi-Fi fingerprints in a reading area of a library (965.6 m^2) with an average error of 3.73 m, outperforming SVM, KNN, decision tree (DT), RF, and an 8-layer MLP.

This chapter is dedicated to the pursuit of the second objective outlined in our research (Fig. 4.1), focusing on the exploration of optimal NN models capable

of inferring the x and y positions of individuals within indoor environments in continuous mode. Initially, we delve into a detailed explanation of TCN and CAPS networks. Then, we discuss the utilization of NAS for conducting design space exploration aimed at optimizing the performance of NNs, with the obtained results compared to those presented in [115].

4.1 Background

4.1.1 Temporal Convolutional Networks

The state of the art suggests using recurrent and recursive networks for sequence modeling tasks. However, there are two major drawbacks: exploding/vanishing gradient and high resource demand. Recent works combine the low-level spatio-temporal features extraction using CNN with the classification of high-level temporal information using RNN ([106, 20, 17]). Bai, Kolter, and Koltun [8] persuade the audience to consider convolutional network as the primary candidate for modeling sequential data, due to the good performance achievement of the temporal convolutional model.

TCN is a type of neural network architecture designed for sequential data processing tasks, such as time series forecasting, natural language processing, and speech recognition. TCNs are particularly effective at capturing temporal dependencies within sequences while maintaining computational efficiency.

The main properties of a TCN are the following:

- *Dilated convolution.* In a normal convolution the kernel takes n contiguous input elements, whereas the dilated one inserts holes between two adjacent taps, achieving a wider range of inputs in the window with fewer resources.
- *Multiple Dilated convolutional layers.* In a forecasting model, the output should ideally depend on the full input window. In a normal convolutional network, this can be achieved at the expense of having a very deep network or very large filters. TCN stacks multiple dilated layers on top of each other, with a dilation value that increases exponentially (Fig. 4.2).
- *Layer output sequence has the same length as the input sequence.* This requires the use of zero-padding.

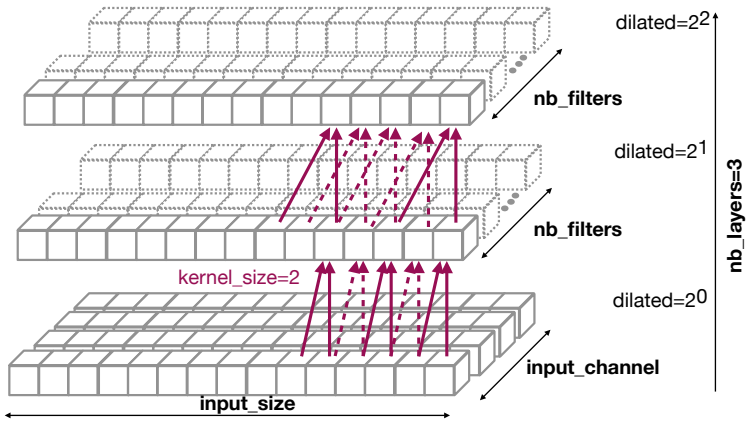


Fig. 4.2 Dilated causal convolutional blocks of a temporal convolutional network (TCN) have an input tensor of `input_size` length repeated `input_channel` times. Each dilated convolution block has `nb_layers`, each with `nb_filters` of `kernel_size` (purple arrows). The dilation factor (hole size between convolved elements) increases exponentially along the hidden layers.

- *No pooling layers.*
- *Residual blocks.* In the improved version of TCN instead of a single block of dilated layers, a residual block is implemented (Fig. 4.3). It consists of 2 identical blocks of dilated convolutions and a residual connection. The convolution blocks are followed by a normalization, a nonlinear activation, and a dropout layer for regularization.
- *Receptive field.* It is the maximum number of steps back in time that a filter can hit to predict the T element. The receptive field can be computed as:

$$R_{\text{field}} = 1 + 2 \cdot (K_{\text{size}} - 1) \cdot N_{\text{res\_block}} \cdot \sum_i d_i \tag{4.1}$$

  where $N_{\text{res\_block}}$ is the number of stacked residual blocks, d is a vector containing the dilations along the hidden layers, and K_{size} is the kernel size.

Fig. 4.2)

4.1.2 Capsule Networks

CAPSs overcome the CNN major limitations due to information loss using dynamic routing instead of pooling operations. They decompose a complex novel object into

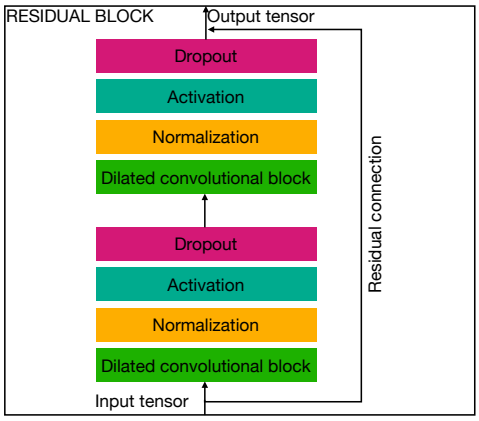


Fig. 4.3 Residual block of a temporal convolutional network. It consists of two identical sub-blocks of a dilation convolution block followed by normalization, activation and dropout layers for regularization. A residual connection is introduced between the input and output tensor.

a hierarchical representation of previously learned patterns. While a CNN neuron outputs a scalar that signals only if a feature is recognized, without relative object relationships, CAPSs explicitly model these relationships in the vector-form of the neuron outputs. Here, the length (modulus) of the vector signifies the detection probability, while the direction (position in space) conveys the characteristics of an entity (instantiation parameters e.g., pose, deformation, velocity). In contrast to conventional neural networks, which predominantly focus on isolated features, capsules in CAPS consider part-whole relationships. During the training of a CAPS, the network learns to adjust the instantiation parameters of each capsule, and the routing process aids capsules in reaching a consensus regarding the presence of higher-level features. To delve deeper into the conceptual underpinnings of CAPS, let's consider a common example often used for illustration: the recognition of a face. In this scenario, a face typically consists of recognizable components such as the face oval, two eyes, a nose, and a mouth. For a CNN, the mere presence of these individual components can serve as strong indicators for detecting a face within an image. However, CNNs may not inherently prioritize the orientational or relative relationships between these components.

Due to the vector-representation all the subsequent neuron steps are modified as follows (Fig. 4.4):

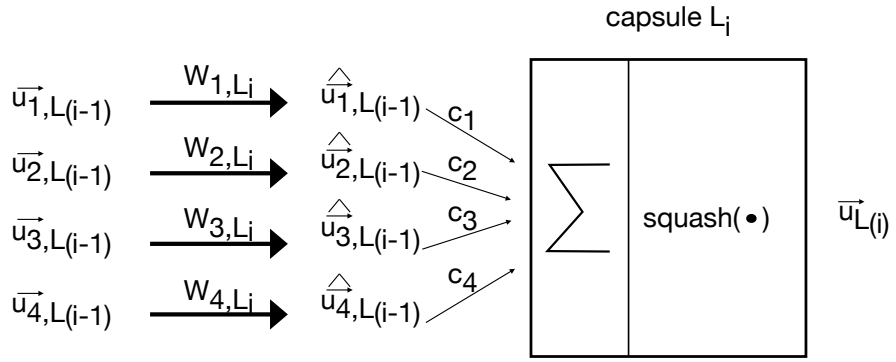


Fig. 4.4 Capsule network operations. Conversely to the normal neuron output, which is a scalar value, the vector-form can encode the state of the detected feature/object and the probability. At layer L_i the incoming vectors are multiplied by matrices W, then, the transformed vectors are weighted and summed. Finally, the activation function squash is applied to keep the vector length in 0 to 1, without altering the direction.

- *Matrix multiplication.* At layer L_i, the input vectors (outputs of the capsules at layer $L_{(i-1)}$) are multiplied by matrices W that encode the relationship between the features in $L_{(i-1)}$ and L_i. This step is called *affine transformation*.
- *Weighting and adding.* This is very similar to what happens in a normal CNN, but the scalar weights are learned through *dynamic routing* and not during backpropagation. These weights represent the probability distribution that a lower level feature belongs to a higher level feature
- *Vector-to-Vector non-linear function.* This function takes a vector in input and forces it to have a length < 1 without changing the direction.

The backbone implementation in [112] (see Fig. 4.5) is composed of two `1D-Conv` layers with `ReLU` activation to extract basic features. Then a `PrimaryCaps` layer does feature combination and encapsulation. Each nb_caps1 capsule applies dim_caps1 $\times$ kernel_size $\times$ nb_filters convolutional kernels. Then `ClassCaps` has nb_class represented as dim_caps2 vectors, with length encoding the detection probability, and di-

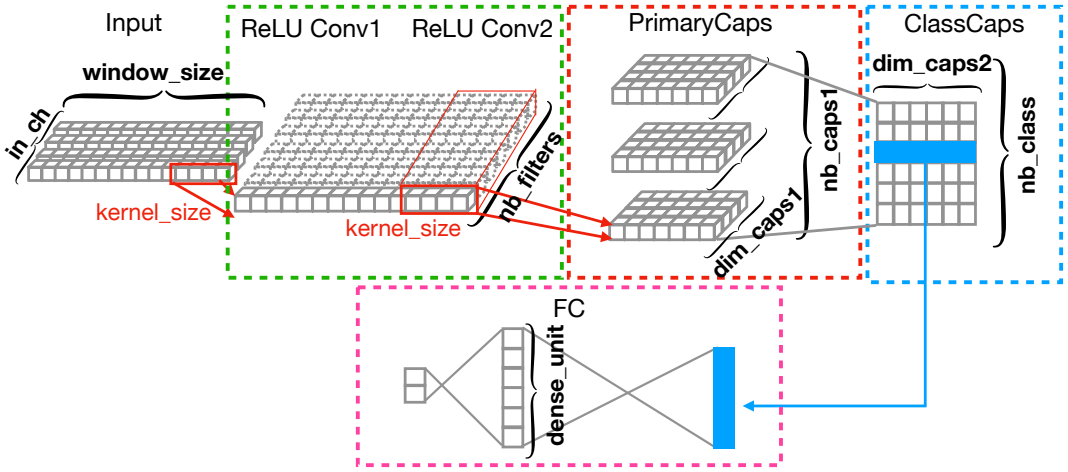


Fig. 4.5 Capsule network (CAPS) architecture has two `1D-convolutional` layers, a `PrimaryCaps` layer where encapsulation takes place, a `ClassCaps` high-level feature capsule layer where low-level features converge, and the vector with the highest detection probability is fed to a fully connected layer.

rection encoding the state of the recognized class/feature. Finally, a fully connected layers which take as input the vector with the highest detection probability output the final prediction.

4.2 Methodology and Experimental Setup

Our main goal is to improve the performance and the resource consumption of the NNs used for indoor human tracking with low-end noisy capacitive sensors [115].

4.2.1 Input Data

To compare the results with Tariq, Lazarescu, and Lavagno [115], the same sensor data and preprocessing are used. The input data come from four single-plate load-mode capacitive sensors mounted in the center of the virtual walls in the 3 x 3 laboratory experimental space, labeled with the person's coordinates, sampled at 3 Hz, and preprocessed using a median filter with a 50 s input window followed by a low-pass filter with a transition band within 0.3 Hz to 0.4 Hz. The experiment is conducted by a single person and lasts about 540 s. The data set is divided into 60 % for training, 20 % for validation, and 20 % for testing, each in time order (see Fig. 5.5).

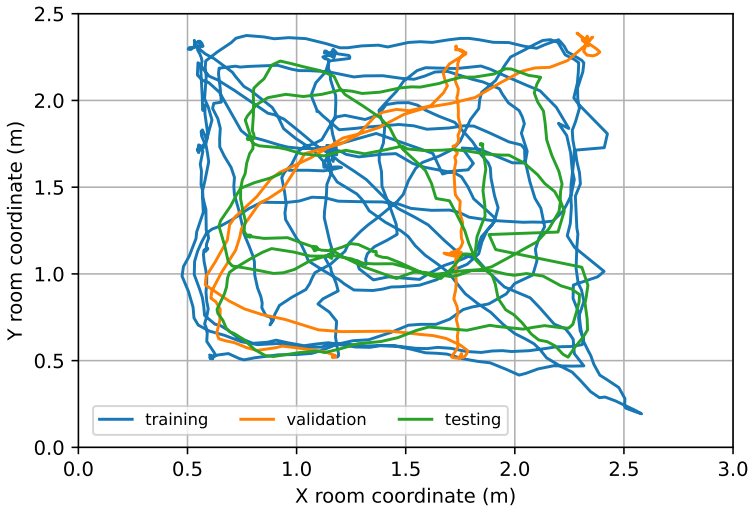


Fig. 4.6 Full trajectory (9 min) of the single person in the experiment 3 x 3 space divided into segments: 60 % for training, 20 % for validation, and 20 % for testing.

4.2.2 Neural Architecture Search

NAS aims to automate the design of NN to a level equal to or better than hand-designed architectures. It can optimize the NN architecture, and estimate or test its performance. NAS is a subfield of automated machine learning (AutoML), closely related to hyperparameter optimization and meta-learning.

As NAS was used AutoKeras [55], an AutoML system based on Keras [24] using a controller for generating NN architectures with a predefined grammar and encoding scheme, a searcher for evaluating the architectures with criteria such as accuracy, complexity, and resource consumption, and a trainer for training and validating the architectures.

The tuned TCN hyperparameters (see Fig. 4.2) and their values are chosen to ensure both that the receptive field (i.e., the number of time steps accessible to the filters for predicting the element at time step t (4.1)) is at least equal to the input sequence length, and to avoid excessive growth of the network size [110]. The number of dilated convolution filters in each hidden layer of the dilated convolution block is nb_filters $= [8, 16, 32]$, which are typical DSE values [115, 116]. The number of input tuples to convolve is kernel_size $= [2, 3, 5]$, where a kernel size of 2 captures localized patterns and nuanced fluctuations, while a kernel size of 5 is a good balance between input sequence length, broader patterns, and computational resources. The number of hidden layers in the dilated convolution block is nb_layers $= [2, 3, 5]$,

with upper bound 5 ensures that both computational resources and receptive field length remain within reasonable bounds, even under the most challenging parameter combinations. The size of a dense layer before the dense output layer of 2 used to perform the prediction of (x,y) position, is dense_unit $= [0, 8, 16, 32]$, adding an extra dense layer could boost the model's capacity to comprehend intricate patterns and relationships, however, it concurrently heightens the susceptibility to overfitting, thus providing the possibility of its removal (0) as an option. Given the limited length of the input window, the number of residual blocks is constrained to 1. Typically, multiple residual blocks are employed for longer sequences (hundreds of thousands of time steps) or in NLP tasks [8]. The dilation base is set to 2 to sufficiently expand the receptive field without exceeding the input sequence length for all parameter combinations.

Also CAPS hyperparameters are tuned (see Fig. 4.5) by NAS. The number of convolution filters in the layers `1DConv1` and `1DConv2`, nb_filters $= [0, 8, 16, 32]$ which are used to extract features. The number of convolved tuples, kernel_size $= [2, 3, 5]$, are typical DSE values [115, 116] considering the length of the input sequence. The number of capsules in the `PrimaryCaps` layer where feature combination and encapsulation take place, nb_caps1 $= [3, 5, 7, 10, 12]$ and the number of high-level classes in the `ClassCaps` layer, nb_class $= [3, 5, 7, 10, 12]$, both selected to stay within the limits of the computational resources of the baseline [115, 116], and the range 3 to 12 allows for flexibility in capturing different levels of abstraction and complexity in extracting movement patterns. The dimension of the capsules in the `PrimaryCaps` layer, dim_caps1 $= [3, 5]$. The dimension of the class vectors in the `ClassCaps` layer, dim_caps2 $= [3, 5]$, where the lower bound 3 ensures meaningful vector representations and minimum movement characteristic representation (e.g., positional displacement), while the upper bound 5 allows for a richer covering of movement aspects (e.g., positional displacement, velocity, acceleration). The size of two dense layers before the dense output layer of 2 used to perform the prediction of (x,y) position, is dense_unit $= [0, 8, 16, 32]$. These final dense layers receive the learned representations of movement features and translate them into actionable predictions of next positions. The CAPSs require at least one routing iteration to update the capsule weights, but excessive iterations can reduce convergence and increase overfitting. Therefore, the routing iteration is set to 3, consistent with the state-of-the-art, and the convolution step of `PrimaryCaps` is set to 2 [86, 120, 132].

Table 4.1 Our TCN and CAPS NNs optimized using neural architecture search (NAS) compared to [115] (highlighted) through the number of parameters, mean squared error (MSE), and average euclidean distance error (ADE)

Model	**Number of parameters**	**MSE** (m^2)	**ADE** (m)
1DCNN (2 layers)	14530	0.078	0.343
1DCNN (4 layers)	7618	**0.063**	**0.307**
1DCNN (6 layers)	8018	0.078	0.328
LSTM (bidirectional)	**2754**	0.079	0.326
CAPS (NAS)	5996	**0.063**	**0.303**
TCN (NAS)	**2034**	0.065	0.309

The NAS is repeated 3 times for each NN type. The AutoKeras tuner tries 50 different parameter combinations, retraining each 10 times for 800 epochs using the Adamax optimizer tuned autonomously by AutoKeras. We force the NAS to return the most compact network architecture whose MSE deviates by no more than 5 % from the optimal performance achieved by the leading 1DCNN as reported in the study by [115].

4.3 Experimental Results

As Tariq, Lazarescu, and Lavagno [115], the performance of the NNs is evaluated using the MSE, average euclidean distance error (ADE), speed, acceleration (smoothness), and resource consumption (number of parameters).

Table 4.1 shows the best results of our optimized networks, TCN and CAPS, compared to [115] (grey background). 1DCNN uses (2, 4, 6) convolutional layers followed by an average pooling layer (tested to perform better than max pooling) and two fixed dense layers with 64 neurons each.

The inference accuracy of the CAPS optimized by NAS (MSE $= 0.063\,\mathrm{m}^2$ and ADE $= 0.303\,\mathrm{m}$) matches the best in [115], but consumes less resources (5996 parameters) when it is configured with nb_filters $= 16$, kernel_size $= 2$, nb_caps1 $=$ 10, dim_caps1 $= 3$, nb_class $= 3$, dim_caps2 $= 5$, 2 dense layers with dense_unit $=$ 32, and trained with Adamax with learning rate 0.001.

The TCN optimized by NAS has inference accuracy (MSE $= 0.065\,\mathrm{m}^2$, ADE $=$ $0.309\,\mathrm{m}$), slightly higher than the best in [115] ($0.063\,\mathrm{m}^2$), but it achieves these

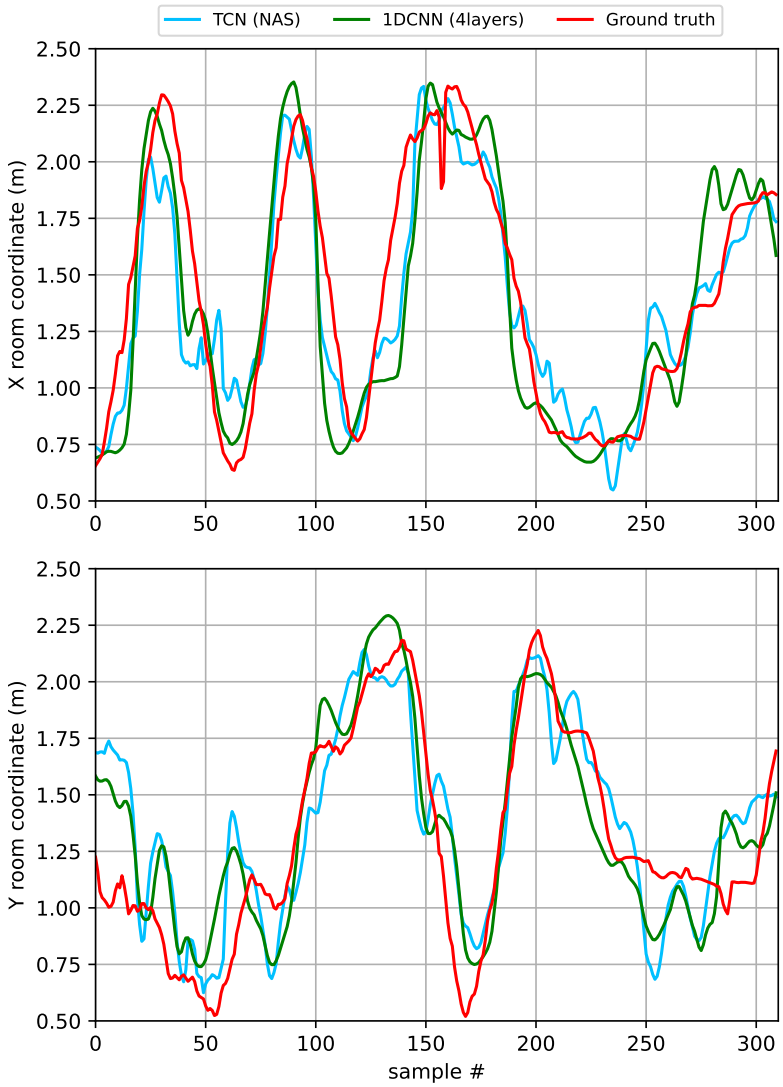


Fig. 4.7 Ground truth and inference of trajectory coordinates X (top) and Y (bottom) by the best neural network in [115], 1D convolutional neural network (1DCNN) (4 layers), and our temporal convolutional network (TCN) optimized using neural architecture search (NAS).

results with much fewer resources (2034 parameters vs. 7618) when configured with nb_filters $= 8$, kernel_size $= 5$, nb_layers $= 3$, dense_unit $= 8$, and trained with Adamax with learning rate 0.0001.

Fig. 4.7 shows the inference of the X and Y coordinates of the person in the room by the TCN optimized by NAS, the inference of the best 1DCNN (also the best NN) in [115], and the ground truth for reference. For both coordinates, the TCN inference shows more susceptibility to noise (more oscillations) than the 1DCNN, and it also

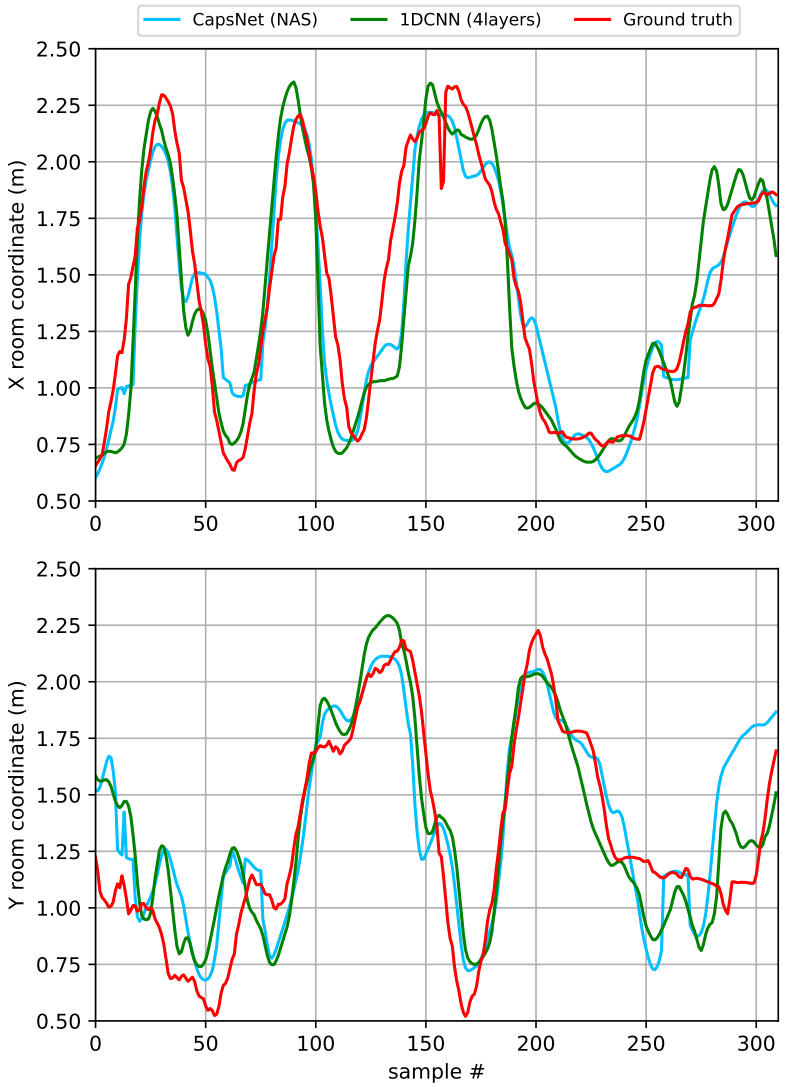


Fig. 4.8 Ground truth and inference of trajectory coordinates X (top) and Y (bottom) by the best neural network in [115], 1D convolutional neural network (1DCNN) (4 layers), and our capsule network (CAPS) optimized using neural architecture search (NAS).

seems to infer less accurately the extremes of either coordinate. Fig. 4.8 allows us to evaluate similarly the CAPS optimized by NAS. Its inference appears smoother than the TCN, less afflicted by noise, often comparable and occasionally exceeding the quality of the inference of the 1DCNN in [115], e.g., around the extremes of the coordinates.

Table 4.2 compares the NN inferences with the ground truth using several metrics: trajectory correlation, and RMSs of speed and acceleration. The speed and ac-

Table 4.2 Movement inference quality of our TCN and CAPS compared to [115] (highlighted) as ground truth correlation, and root mean square of speed and acceleration

Model	**Correlation** (%)	**RMS speed** (m/s)	**RMS acc.** (m/s^2)
Ground truth	100.0	0.180	0.333
1DCNN (2 layers)	83.3	0.157	0.172
1DCNN (4 layers)	**87.5**	0.162	0.187
1DCNN (6 layers)	84.5	**0.176**	0.259
LSTM (bidirectional)	84.0	0.133	0.129
TCN (NAS)	86.0	**0.180**	**0.347**
CAPS (NAS)	**87.1**	0.164	0.384

celeration RMS are calculated as the square root of the mean square of the first and the second derivatives of the inferred locations, respectively. The correlations of both the TCN and CAPS are slightly lower than that of the 1DCNN (87.5 %), but while the TCN RMSs of speed and acceleration agree very well with the ground truth, the CAPS inference appears to be too smooth.

Chapter 5

Neural Network Minimization for Continuous Indoor Human Tracking

Today, deep learning architectures are widely used that can operate on raw sensor data, extract valuable features, and identify spatial and temporal patterns (Chapter 4). However, they require massive training datasets and many resources for inference, which are difficult to provide on resource-constrained IoT.

Several compression techniques, such as quantization and pruning [71], have been proposed to reduce the network size and complexity, but they often compromise the network accuracy. A more recent technique, known as knowledge distillation, was introduced [50, 40] to transfer the knowledge from a large network (teacher) to a smaller one (student), mimicking a teacher-student relationship.

Knowledge distillation encompasses a wide variety of knowledge types, including response, feature, and relationship-based knowledge, as well as different methodologies such as offline, online, and self-distillation. Additionally, KD involves the utilization of various algorithms, including adversarial, attention-based, similarity-based, and ensemble-based approaches. The efficacy of KD hinges predominantly upon several critical factors, notably the quality of the teacher's knowledge, the chosen distillation strategy, and the formulation of the teacher-student architecture. This formulation encompasses architectural design considerations and interactive elements [40].

Most of the existing KD methods focus on classification problems [67, 92, 121], where the softened logits of the teacher provide useful information about the class relationships, which is missing in the one-hot encoded ground-truth labels.

However, this advantage does not exist in regression problems, where the ground-truth predictions are continuous and unbounded.

NN optimization using KD-based training has previously been applied to indoor person location *classification* [76, 78, 94], which allows only rough tracking of the person's activities.

Many proposed KDs address classification involving either multiple teachers or stages (different depths of the student network) or multi-step. The latter can use intermediaries (assistants) of the same architecture but different sizes for communication between teacher and student [77, 109]. Zhao et al. [131] suggested using two teachers to train a smaller student, all of the same type, to classify images: an expert teacher trained offline would focus the student on critical regions, while a scratch teacher trained alongside the student would provide incremental assistance. Liu, Zhang, and Wang [72] use heterogeneous teachers to guide a student learning image classification through an adapter that dynamically learns relevant instance-specific teacher weights from soft-targets and formulates the standard knowledge distillation loss, similar to [129]. Classification of human location indoors is relatively coarse and poorly suited to more advanced LBS applications that require continuous fine-grained tracking. In addition, most existing work focuses on coarser surveillance of larger indoor areas and typically does not address much sought-after LBS features such as low cost or privacy.

We propose a multi-step KD combined with network optimization specifically for *regression*-based indoor human tracking, providing fine-grained continuous tracking of a person's position.

We assess the efficacy of the proposed methodology through several key metrics, including the reduction of NN resource requirements, tracking accuracy, and generalization accuracy across unrelated experimental data, encompassing different movements and days. The methodology is systematically tested across various types of NN architectures to ascertain its effectiveness, irrespective of sensor characteristics or data format. Furthermore, we integrate the NNs into three distinct tagless LBSs, each employing disparate human body sensing techniques: capacitive, radar, and infrared sensors; enabling person tracking within a 3 x 3 virtual room in the laboratory.

This chapter addresses the concluding aspect of our study (Fig. 5.1), which entails the investigation of a resource-conscious optimization approach for ML models which maintains high precision levels. Having low-resource models is essential

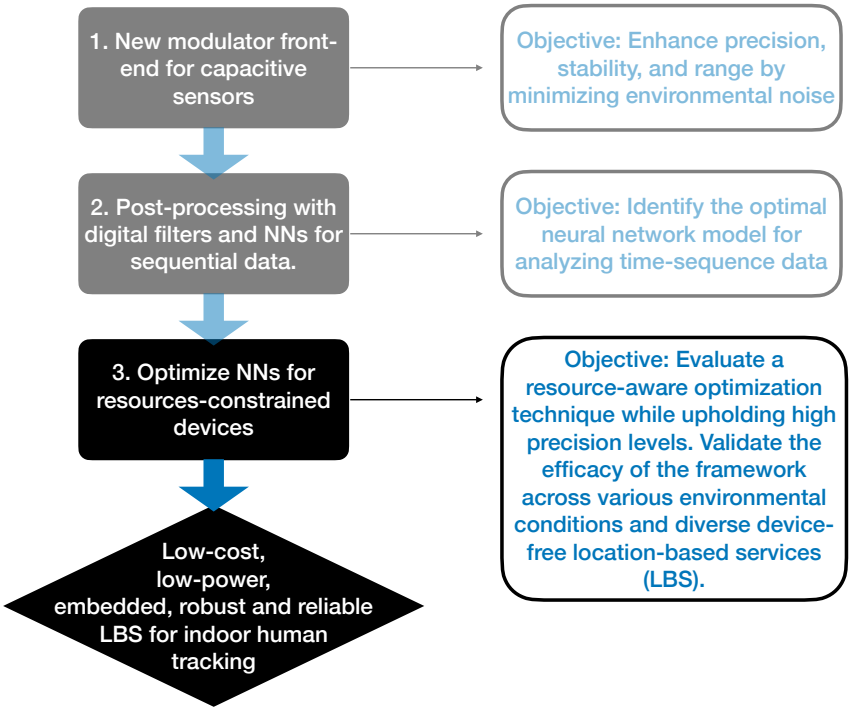


Fig. 5.1 Flowchart illustrating the organization of work and outlining objectives.

for the deployment into IoT devices. The development of low-resource models is imperative for their successful deployment within Internet of Things (IoT) devices. The initial segment of the chapter delineates the tools, environmental parameters, and the methodology employed. Subsequently, the experimental outcomes pertaining to diverse multi-step and multi-configuration KD optimization flows for three distinct LBSs, namely capacitive, radar, and infrared, are expounded upon. Finally, the chapter culminates with a demonstration of the generalization capability of the most effective multi-level KD optimization flow. This demonstration involves assessing its performance using data from various days and movements distinct from those encompassed during the training phase.

5.1 Tools

We assess the proposed methodology by employing TCN [9, 45, 66] and CAPS [59, 112] models. These models are selected based on the comprehensive exploration of NNs for indoor human tracking outlined in Chapter 4. Notably, both models circumvent the utilization of pooling operations specific to CNNs, as such operations

are recognized to potentially compromise the preservation of crucial data relationships. A comprehensive elucidation of the architecture and implementation details is provided in Section 4.1.1, Section 4.1.2, and Section 4.2.2.

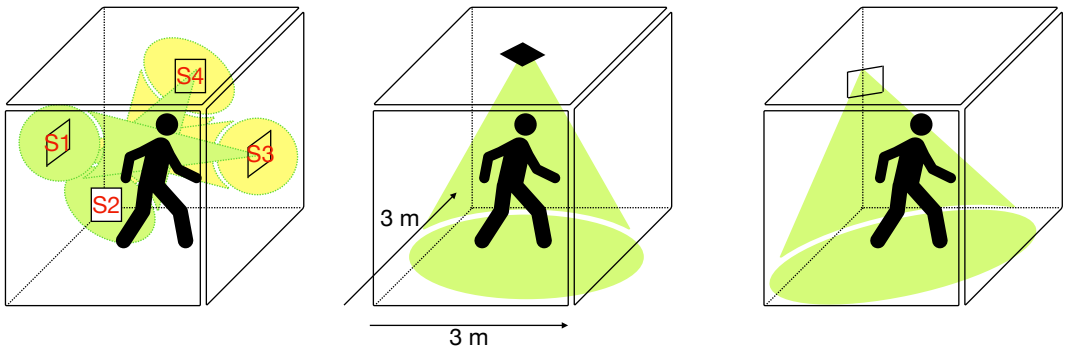
NAS is used to optimize NN sizes and architectures and is based on the widely used AutoKeras framework [55] with the Bayesian tuner [39]. The objective function includes the teacher and KD attentive imitation loss [101]. NAS is used with default values for parameters, except for the maximum number of different parameter combinations, which is limited to 50; this choice represents a reasonable balance between processing time and space exploration. Additionally, the number of independent trainings per trial is limited to 10, a tradeoff between improving robustness and minimizing optimization time. The training cycle uses early stopping and Adamax optimizer tuned autonomously by AutoKeras.

Offline KD involves the transfer of knowledge from pre-trained larger neural networks (referred to as teachers, optimized for accuracy) to smaller ones (referred to as students, optimized for size) with the aim of enhancing their accuracy. Larger models typically possess a greater knowledge capacity and are adept at learning compact representations of knowledge. Intuitively, through KD, these representations are imparted to student models that may struggle to acquire them independently.

Although KD for regression problems has not been extensively researched because it lacks exploitable hidden knowledge such as the class labels incorporated in softened logits that can be used for correlations, posing a significant risk of misguiding the student model, teacher predictions can still benefit regression problems. Notably, the reliability of such predictions may be estimated based on teacher loss. Saputra et al. [101] outline different formulations of blending teacher, student and imitation loss:

- Imitation loss regularization (I_{loss}), implies that the student uses also labels sampled from the teacher distribution, as regularization, in addition to the ground-truth labels.

- Attentive imitation loss adaptively adjusts the weighting of imitation loss based on the reliability of teacher predictions, effectively mitigating its influence in instances where such predictions demonstrate diminished reliability. This adjustment is informed by the empirical error derived from the teacher loss.



(a) Capacitive (b) Infrared (c) Radar

Fig. 5.2 Experimental setting in a 3 x 3 laboratory space for (a) capacitive, (b) infrared radiation, and (c) radar location-based systems

- Teacher loss as upper bound, which implies that the loss function used to train a student to imitate a teacher should not exceed the loss function used to train the teacher itself, which is assumed to be more accurate and reliable.

Attentive imitation loss formulation was selected due to its demonstrated superiority in achieving optimal results for regression problems [101]. The specific expression for the *attentive imitation loss* is

$$L_{\text{reg}} = \frac{1}{n}\sum_{i=1}^{n} \alpha \left| p_{\text{S}} - p_{\text{gt}} \right|_i^2 + (1-\alpha)\,\Phi_i \left| p_{\text{S}} - p_{\text{T}} \right|_i^2 \tag{5.1}$$

where p_{S} is the student prediction, p_{gt} is the ground truth, p_{T} is the teacher prediction, and Φ is the normalized teacher loss for each sample, it is the *attentive* term in charge of putting different relative importance to the imitation loss. $\alpha \in [0,1]$ is a hyperparameter tuned by NAS to balance the weights of student and distillation losses.

5.2 Location-based systems

The capacitive LBS [115] (see Fig. 5.2a) consists of four single-plate load-mode capacitive sensors mounted in the center of the virtual walls. The sensors are sensitive to electromagnetic noise and have a highly nonlinear distance-capacitance characteristic. They are sampled at 3 Hz and their data is pre-processed by a median filter

with a window size of 5 s, followed by a low-pass filter with a transition band from 0.3 Hz to 0.4 Hz [115].

The IR LBS (see Fig. 5.2b) uses a single 4 x 4pixel Omron D6T-44L-06 thermopile infrared sensor with a resolution of 0.06 °C installed in the center of the virtual room ceiling at a height of 3.05 m covering a field of view of 2.48 x 2.57 at floor level. The 16 thermal sensor readings are sampled at 5 Hz. The IR sensor output is less noisy than that of the capacitive sensor and has a linear relationship between the thermal readings and their location, so we do not pre-process it.

The radar LBS (see Fig. 5.2c) uses the 60 GHz Infineon BGT60TR13C radar [13]. Its output data is made of range-angle-magnitude point clouds. The sensor is placed in a corner of the virtual room pointing towards its center with a 45° downward inclination, and situated at a height of 1.30 m above the floor. Its setup is: 2.5 MHz ADC sample rate, chirp frequency from 61.8 GHz to 63.5 GHz, 64 chirps per frame, 128 samples per chirp, 0.3 ms chirp repetition time. These parameters are configured using the Infineon framework tool to meet a sensing range requirement of approximately 5 m (roughly equivalent to our diagonal) and a 4 Hz sampling rate. The intentional avoidance of sophisticated settings and complex point cloud extraction algorithms is deliberate, aiming to increase the workload for the NNs. The output seems to be less sensitive to noise than the capacitive sensor, but more sensitive than the IR sensor.

5.3 Methodology

As outlined in the introductory section of the chapter, our aim is to assess the efficacy of the suggested methodology for continuous indoor person tracking from various perspectives and under diverse experimental conditions. This includes testing with different types of NNs, employing training datasets collected under varying conditions, and integrating the methodology into different LBSs.

The experimental data is collected with the LBSs described in Section 5.2 in similar but distinct experiments (denoted `exp1`, `exp2`, `exp3` see Appendix A). These experiments were carried out on different days by the same individual and each lasted for a duration of 9 minutes. This variability is useful to test the generalization ability of the NN optimization and training in the proposed methodology for continuous

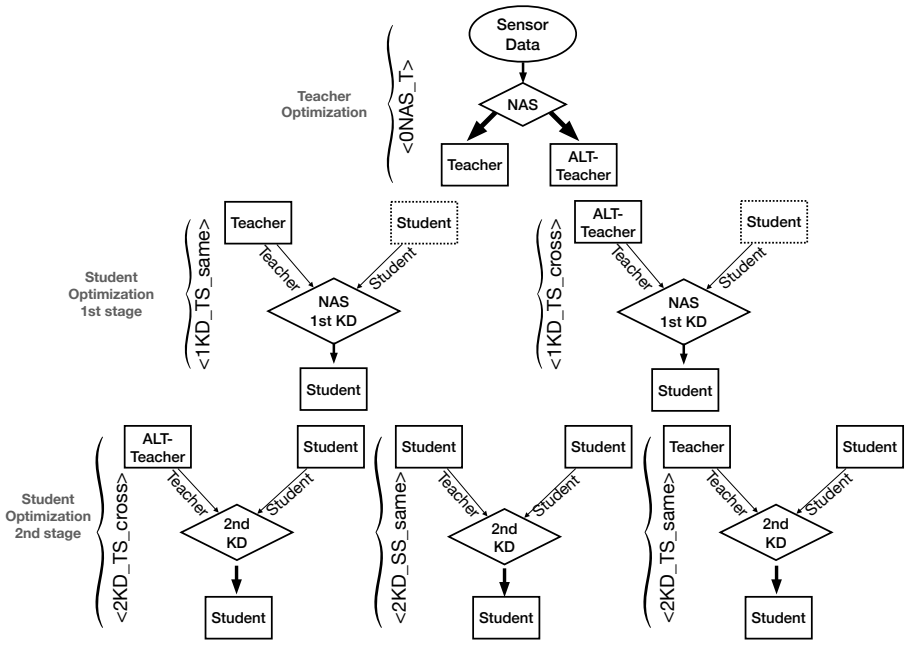


Fig. 5.3 Flowchart of multi-level optimization and two-level knowledge distillation (KD) with multiple optimization branches

indoor human tracking. To do this, NNs are trained on data collected during an experiment, then tested both on an unseen segment of the same experiment and on data from distinct experiments, (see Section 5.4 and Section 5.5 for details). The comparison of these test results indicates the generalization ability of the NNs, i.e. how well they have captured the essential features that lead to robust accurate tracking in unrelated experimental conditions. The training and testing process outlined above is reiterated using a different experiment as the training dataset. This repetition aims to evaluate the robustness and effectiveness of the framework under varied conditions.

KD can be used multiple times [40] and in the proposed methodology it is used twice in the combinations shown in Fig. 5.3, while the main phases are summarized in Table 5.1. For each combination and experiment, the student NN is compared to teacher NN in terms of size (a proxy for resource requirements), inference accuracy, and generalization ability.

The sensor data is used for training the teacher NNs, while the student NNs are trained by KD based on the sensor data and teacher's knowledge.

The following processing is divided into three main stages. In the first stage, *Teacher Optimization* (top of Fig. 5.3), the sensor data of the LBS is used to train

Table 5.1 Algorithmic table outlining the steps of the NN optimization procedure that incorporates KD and NAS

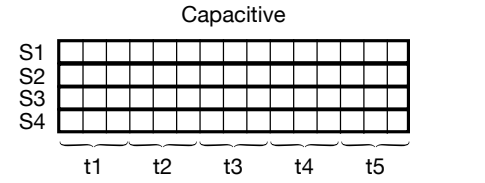
Stage	**Action**	**Description**
0NAS_T	NAS opt.	NAS optimizes the TCN teacher.
0NAS_T	NAS opt.	NAS optimizes the CAPS teacher.
1KD_TS_same	NAS for 1st stage KD	NAS optimizes: TCN student with KD from *same type* (TCN) teacher, or CAPS student with KD from *same type* (CAPS) teacher.
1KD_TS_cross	NAS for 1st stage KD	NAS optimizes: TCN student with KD from *different type* (CAPS) teacher, or CAPS student with KD from *different type* (TCN) teacher.
2KD_TS_same, 2KD_TS_cross	2nd stage KD	Continue KD-based training of students from 1KD_TS_cross or 1KD_TS_same, respectively, with teachers of a different type than those trained in the first stage.
2KD_SS_same	2nd stage KD	Students trained in 1KD_TS_same now act as teachers to continue the training of a copy of them through KD.

and optimize the hyperparameters of all teachers NNs using NAS (see Section 5.1) to maximize their accuracy. The NAS is based on AutoKeras [55] and driven by a Bayesian optimization tuner.

In the second stage, *Student Optimization 1st stage* (middle of Fig. 5.3), KD-aware NAS minimizes the size and errors of student NNs using the knowledge of the pre-trained teacher NNs. The teacher and student NNs can be of the same type (left flow, 1KD_TS_same) or different (right flow, 1KD_TS_cross).

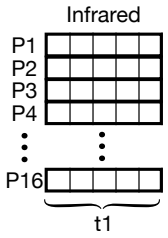
In the third stage, *Student Optimization 2nd stage* (bottom of Fig. 5.3), continues the training of the student NNs. In the left (2KD_TS_cross) and right (2KD_TS_same) flows, the student NNs from 1KD_TS_same and 1KD_TS_cross, respectively, are trained by teacher NNs of a different type, to ensure that student NNs are always trained by teachers of different types. The concept entails presenting the student with an additional perspective on the given task, thereby facilitating ongoing refinement and enhancement of their understanding and performance.

The middle flow in the third stage, 2KD_SS_same, freezes a student NN after the first stage and uses it as teacher in the second stage to further train as teacher a



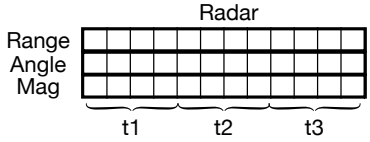
5 s/win · 3 tuple/s · 4 sampl/tuple = 60 sampl/win

(a) Capacitive



1 s/win · 5 tuple/s · 16 sampl/tuple = 80 sampl/win

(b) Infrared



3 s/win · 4 tuple/s · 3 sampl/tuple = 36 sampl/win

(c) Radar

Fig. 5.4 Input layer organization

copy of itself. Here the idea is that of enabling students to acquire new knowledge while retaining the foundational understanding obtained in prior stages of learning.

5.4 Input Data and Dataset Organization

Capacitive LBS data has 4 sample tuples (one sample per sensor, see Fig. 5.2a), processed using a 5 s sliding window [115] (see Fig. 5.4a).

IR LBS data has 16 sample tuples (one sample per pixel of the IR sensor), processed using a 1 s sliding window. The window is smaller to limit the size of NN input layer, since IR LBS tuples are nearly four times larger than those of capacitive LBS (see Fig. 5.4b).

Radar LBS data has 3 sample tuples (range, angle, magnitude), processed using a 3 s sliding window (see Fig. 5.4c), because in our experiments 1 s windows give poor accuracy, while there is little difference between 3 s and 5 s windows.

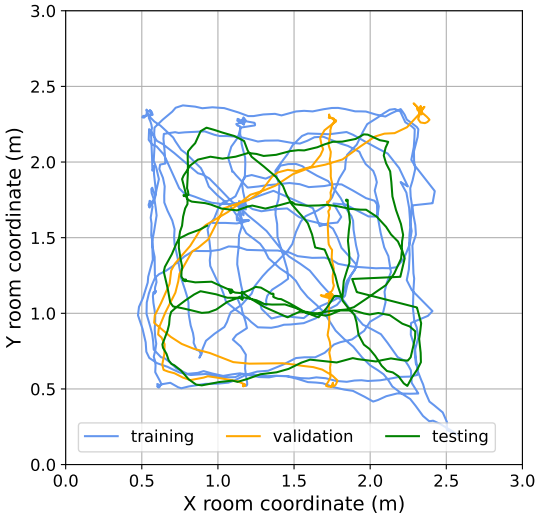
For meaningful comparisons, the tracking data with each LBS was collected in different but similar experiments, 9 min long, with trajectories of similar characteristics, segmented into 60 % for training, 20 % for validation, and 20 % for testing, as shown in Fig. 5.5a for capacitive LBS, Fig. 5.5b for infrared radiation LBS, and Fig. 5.5c for radar LBS. In each experiment was also collected a ± 2 cm accurate ground truth location using an ultrasound-based Marvelmind Starter Set HW v4.9 localization system (see Section 5.2).

5.5 Experimental Results

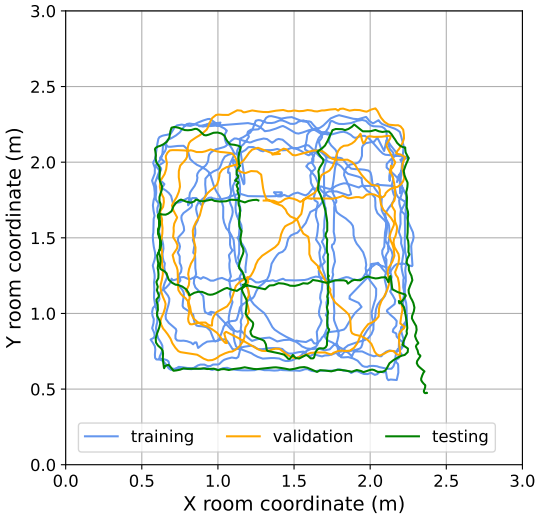
Several similar but different experiments (`exp1`, `exp2`, `exp3` see Appendix A) for each LBS, lasting 9 minutes each, were performed on different days and using different LBSs as described in Sections 5.2 to 5.4.

The experimental data is used to train and optimize the teacher NNs using NAS, and the student NNs with the additional teacher's knowledge using NAS in the first stage, and without NAS in the second stage, as discussed in Section 5.3 and shown in Fig. 5.3, *Teacher Optimization*, *Student Optimization 1st stage*, *Student Optimization 2nd stage*. NAS tunes the hyperparameters of the two NNs of interest, TCN and CAPS, to minimize their inference MSE. As a first approximation, NN complexity is evaluated based on its number of NN parameters, since computing effort, measured as floating point operations per second (sFLOPs), is closely related. Complexity is limited by underfitting at the low end, and by the complexity of the well-optimized NNs in the state-of-the-art [115, 116] at the high end.

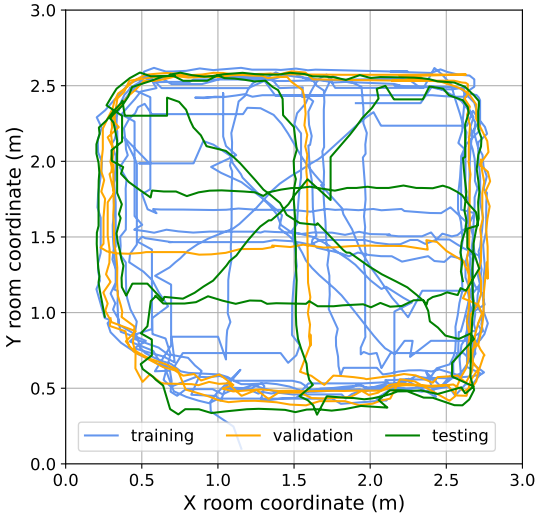
The validity and robustness of the proposed methodology are evaluated through the following procedures: 1) The methodology is subjected to rigorous testing by repeating its steps three times, each time utilizing a distinct experiment (`exp1`, `exp2`, `exp3`) as the training dataset. The other two experiments are then employed to assess the robustness of the optimized student (NNs) in terms of their ability to generalize. The performance of these optimized NNs is compared against that of the larger teacher NNs, particularly in terms of tracking accuracy, using data from experiments



(a) Experimental path `exp1` for capacitive location-based system



(b) Experimental path `exp1` for infrared radiation location-based system



(c) Experimental path `exp1` for radar location-based system

Fig. 5.5 Experimental path `exp1` in a 3 x 3 virtual room for (a) capacitive, (b) infrared radiation, and (c) radar location-based systems split into 60 % for training, 20 % for validation, and 20 % for testing the neural networks

Table 5.2 Mean squared errors (sMSEs) for best temporal convolutional network (TCN) and capsule network (CAPS) teachers for capacitive location-based system

Experiment	NN model	Parameters	MSE (m^2)
exp1	TCN	7242	0.059
	CAPS	6734	0.061
exp2	TCN	16826	0.071
	CAPS	11479	0.085
exp3	TCN	16826	0.066
	CAPS	10323	0.067

that were not used during training. 2) Additionally, the methodology is tested using LBSs that are based on different operating principles, outputs, and noise levels.

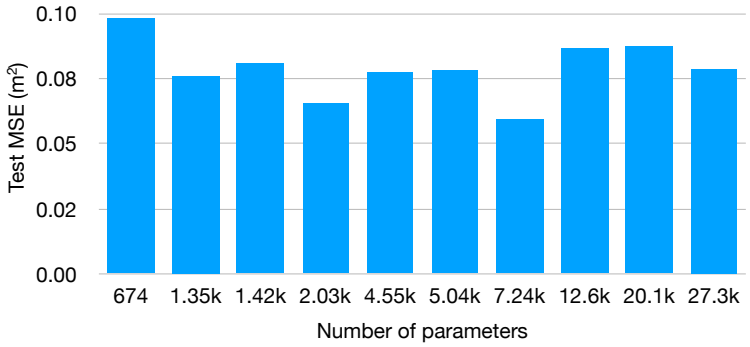
5.5.1 Training and Optimization

Teacher NNs are optimized by NAS by varying various parameters, discussed in Section 5.1. For this purpose, the NN complexity of the two evaluated teacher network architectures is divided into intervals that double in size until they reach the maximum complexity of the best state-of-the-art implementations [115, 116]. For each of these intervals, NAS is used to obtain the best teacher network of each type.

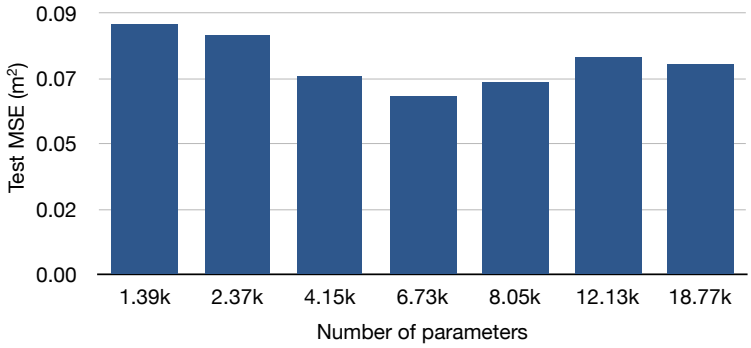
Three different experiments, exp1, exp2, and exp3 were used for training. For exp1, the complexity of the networks and their inference MSEs are shown in Fig. 5.6, while Table 5.2 summarizes the best configurations for all experiments.

Although within each experiment the teachers NNs have comparable accuracies (TCN slightly better), the number of parameters can vary significantly depending on the quality (e.g., noise level) of the experimental data, with TCN always being larger than CAPS. Note that NNs size is a measure of both memory and computational cost, FLOP (see Section 5.5).

The typical "V" shape of the inference MSE function of network complexity in Fig. 5.6 is largely due to the tendency to underfit of small networks and to overfit of larger networks [7]. Also, large networks like in exp2 are prone to overfitting



(a) Temporal convolutional network (TCN) neural network model



(b) Capsule network (CAPS) neural network model

Fig. 5.6 Neural architecture search-optimized mean squared error (MSE) accuracy and number of parameters for (a) temporal convolutional network (TCN) and (b) capsule network (CAPS) teachers for `exp1`

because the training dataset is relatively small, 972 tuples (60 % of 9 min of the experiment duration sampled 3 times/s).

The teacher networks are then used both as a source of knowledge for the KD-based optimization stages of the proposed methodology, and as a baseline for evaluating the performance gains of the optimizations, characterized in terms of resource requirements, inference accuracy, and generalization ability.

5.5.2 Results for Capacitive location-based system

Optimization of NNs in experiments with capacitive LBS is more difficult, resulting in larger and less accurate NNs due to higher noise and strongly nonlinear character-

Table 5.3 TCN student with KD from TCN first teacher for capacitive LBS (gains from best exp1 TCN teacher, Table 5.2)

	KD stage (Fig. 5.3a, b, Table 5.1)	**Size**	**ΔSize** (%)	**MSE** (m^2)	**ΔMSE** (%)
exp1	1KD_TS_same	**986**	**-86.4**	0.067	13.6
		1346	−81.4	0.059	0
		2034	-71.9	**0.055**	**-6.78**
		3210	−55.7	0.073	23.7
		4586	−36.7	0.065	10.2
		5010	−30.8	0.063	6.78
	2KD_TS_cross	2034	-71.9	**0.05**	**-15.3**
	2KD_SS_same	2034	−71.9	0.052	−11.9

istic of sensors [115]. Thus, for this LBS are analyzed in detail all the steps of the proposed methodology. Then, for the sake of brevity, only the most effective optimization procedures from the analysis of the capacitive LBS will be analyzed.

As shown in Figs. 5.3 and 5.7, KD has two stages, each using a different type of teacher NN, optimized before using sensor data and NAS, as discussed in Section 5.5.1.

In the first KD stage, the student NN is trained by a teacher NN of either the same architecture (1KD_TS_same, Fig. 5.7a) or a different architecture (1KD_TS_cross, Fig. 5.7c). This is followed by a second KD stage where the same student NN continues training with a teacher NN of a different architecture than in the first KD stage: 1KD_TS_same → 2KD_TS_cross (Fig. 5.7a) and 1KD_TS_cross → 2KD_TS_same (Fig. 5.7c). For the second KD stage, a third option is also explored, namely to freeze the student NN after the first stage and continue training a copy of it in the second stage, using the frozen student NN as the teacher (Fig. 5.7b).

For `exp1`, the results of all multi-stage KD optimizations flows shown in Figs. 5.3 and 5.7a are reported in Table 5.3 for TCN students and in Table 5.4 for CAPS students. In the first KD stage, 1KD_TS_same, the performance of the student NNs was optimized using NAS in six intervals of the number of parameters, up to that of the teacher TCN (see `exp1` in Table 5.2). The best TCN student NN has 2034 parameters (1KD_TS_same in Table 5.3), while the best CAPS student NN has 4151, more than double (1KD_TS_same in Table 5.4).

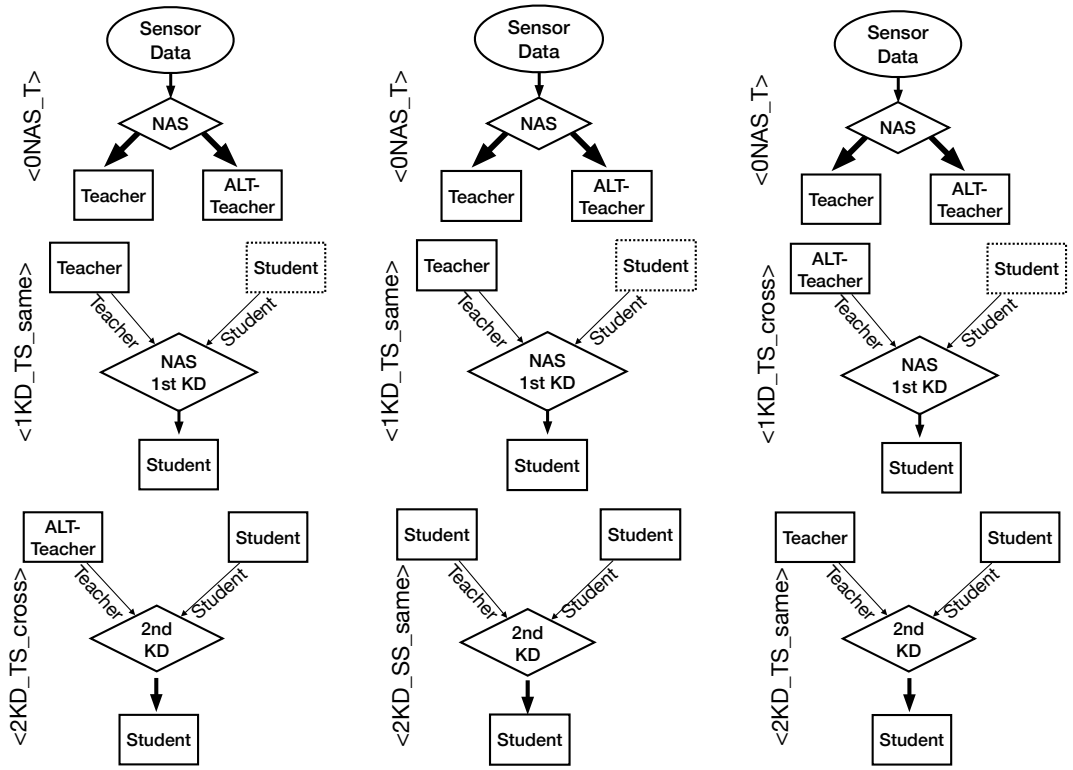


Fig. 5.7 Flowchart of optimization branches using neural architecture search (NAS) and two-level mixed knowledge distillation (KD)

Table 5.4 CPS student with KD from CAPS first teacher for capacitive LBS (gains from best exp1 CAPS teacher, Table 5.2)

	KD stage (Fig. 5.3a, b, Table 5.1)	Size	ΔSize (%)	MSE (m^2)	ΔMSE (%)
exp1		**1389**	**-79.4**	0.076	24.6
		2366	−64.9	0.064	4.59
	1KD_TS_same	3276	−51.4	0.062	2.62
		4151	-38.4	**0.061**	**0**
		5274	−21.7	0.064	4.92
	2KD_TS_cross	4151	-38.4	**0.057**	**-6.56**
	2KD_SS_same	4151	−38.4	0.061	0

Table 5.5 TCN student with KD from CAPS first teacher for capacitive LBS (gains from best exp1 TCN teacher, Table 5.2)

	KD stage (Fig. 5.3c, Table 5.1)	**Size**	**ΔSize** (%)	**MSE** (m^2)	**ΔMSE** (%)
exp1	1KD_TS_cross	**986**	**-86.4**	0.075	27.1
		1346	−81.4	0.062	5.08
		2122	−70.7	0.06	1.69
		3074	-57.6	**0.057**	**-3.39**
		4738	−34.6	0.063	6.78
		5609	−22.6	0.061	3.39
	2KD_TS_same	3074	-57.6	**0.054**	**-8.47**

In the second KD stage (2KD_TS_cross in Fig. 5.7a), the best students from the first KD stage (1KD_TS_same) are further trained by teacher NNs of a different type, because they can intuitively give the student a different view on the meaningful features, thus reinforcing the learning of the useful ones. The TCN student performs best (2KD_TS_cross in Table 5.3), reducing its parameters by 71.9 % and inference MSE by 15.3 % to 0.05 m^2 from the TCN teacher (`exp1` TCN in Table 5.2). The CAPS student (2KD_TS_cross in Table 5.4) improves accuracy much less from the CAPS teacher, by 3.39 % (`exp1` CAPS in Table 5.2), but still significantly reduces its number of parameters, by 38.4 %.

Figure 5.7b shows another option for the second KD stage, 2KD_SS_same, where the best student NN from the first stage is frozen and used as a teacher NN in the second stage to train a copy of itself. However, this flow yields fewer gains for both the TCN student (2KD_SS_same in Table 5.3) and the CAPS student (2KD_SS_same in Table 5.4). This is probably because the students that learn the essential teacher knowledge in the first KD stage do not acquire enough diversity to improve their learning in the second KD stage.

The flow of the third optimization branch shown in Fig. 5.7c reverses the flow of Fig. 5.7a: in the first KD stage, 1KD_TS_cross, the student NN is trained with the knowledge of a teacher NN with a *different* architecture, while in the second KD stage, 2KD_TS_same, its training continues with a teacher NN of the *same* architecture. Table 5.5 shows the results for the TCN student NNs, the best of which with 3074 parameters and 0.054 m^2 MSE is less performant than the one optimized

Table 5.6 CPS student with KD from TCN first teacher for capacitive LBS (gains from best CAPS teacher, Table 5.2)

	KD stage (Fig. 5.3, Table 5.1)	**Size**	**ΔSize** (%)	**MSE** (m^2)	**ΔMSE** (%)
exp1	1KD_TS_cross	**1733**	**-74.3**	0.081	32.8
		2375	−64.7	0.075	23
		3814	-43.4	**0.064**	**4.92**
		4139	−38.5	0.067	9.84
		5274	−21.7	0.066	8.2
	2KD_TS_same	3814	-43.4	**0.058**	**-4.92**

Table 5.7 Student NNs with KDs from first teacher of same type for capacitive LBS in exp2, exp3 (gains from best teacher, Table 5.2)

	KD stage (Fig. 5.7a)	**Network**	**Size**	**ΔSize** (%)	**MSE** (m^2)	**ΔMSE** (%)
exp2	1KD_TS_same	TCN	8274	−50.8	0.058	−18
		CAPS	5371	−53.2	0.085	0
	2KD_TS_cross	TCN	8274	−50.8	0.057	−18.6
		CAPS	5371	−53.2	0.073	−14.9
exp3	1KD_TS_same	TCN	7242	−56.96	0.059	−10
		CAPS	8877	−14.01	0.074	10.1
	2KD_TS_cross	TCN	7242	−56.96	0.057	−12.9
		CAPS	8877	−14.01	0.07	3.71

with the flow in Fig. 5.7a (Table 5.3), when the first teacher has the same architecture. Table 5.6 shows the results for CAPS student NNs, of which the best (highlighted) is less accurate than the optimization flow in Fig. 5.7a (Table 5.4), although it has slightly fewer parameters.

These results suggest that student NNs learn better if they are first trained by teacher NNs with affine architectures through the flow in Fig. 5.7a. Since the optimization using the KD flow in Fig. 5.7b was also less performant, only the KD in Fig. 5.7a flow will be used in the following.

Thus, the proposed methodology, limited to the flow in Fig. 5.7a, is then applied to capacitive LBS data from another experiment, `exp2`, to verify its validity. Table 5.7

shows the best results for the KD optimization flow in Fig. 5.7a for both the TCN and CAPS students. The results for the best teacher NNs are reported in Table 5.2, `exp2`.

As discussed in Section 5.5.1, capacitive LBSs can be affected by high variability and noise, and thus the best teacher and student NNs in `exp2` are much larger than for `exp1`. The best TCN teacher NN has 16 826 parameters and 0.071 m^2 MSE, while the best CAPS teacher NN with 11 479 parameters is smaller, but less accurate with 0.085 m^2 MSE. The best TCN student NN with 8274 parameters is about half the size of the best TCN teacher NN, and has also better accuracy by 18.6 %, 0.074 m^2 MSE. The best CAPS student NN also more than halves the best CAPS teacher NN size to 5371 parameters, but improves less the accuracy, by 14.9 % to 0.073 m^2 MSE. Note that the second KD stage, done with a TCN teacher NN, significantly improves the CAPS student accuracy.

In `exp3`, the best teacher NNs have sizes comparable to those in `exp2` (Table 5.2) and accuracies in the range of those in `exp1` and `exp2`. The best TCN student NN (Table 5.7, `exp3`) has about half the parameters of the best TCN teacher NN and 12.9 % better accuracy at 0.057 m^2 MSE. The CAPS student NN has 8877 parameters, less than the best CAPS teacher NN by 14 %, but with 3.71 % lower accuracy for 0.07 m^2 MSE from 0.067 m^2 MSE of the teacher NN.

Experimental results for capacitive LBS show that KD can significantly reduce the parameters of NNs, and a second KD, with a different teacher NN, very likely improves the accuracy, best when the first KD is from a teacher with the same architecture as the student.

5.5.3 Results for infrared radiation location-based system

The experiments and methodology for the IR LBS are similar to those for capacitive LBS discussed in Section 5.5.2. Again, only the KD in Fig. 5.7a flow is discussed that gives the best results, and it is tested on data from three experiments, `exp1`, `exp2`, and `exp3`.

The performance of the optimized teacher NNs are reported in Table 5.8. They typically require more parameters than capacitive LBS (Table 5.2), because the sensor tuples have 16 values for the 4 x 4pixel camera used, which is 4 times higher than for capacitive LBS. But they are much more accurate, in part because IR sensor data is much less affected by ambient noise.

Table 5.8 Mean squared errors (sMSEs) for best temporal convolutional network (TCN) and capsule network (CAPS) teachers for infrared radiation (IR) location-based system

Experiment	**NN model**	**Parameters**	**MSE** (m^2)
exp1	TCN	8658	0.0046
	CAPS	7726	0.005
exp2	TCN	19130	0.0137
	CAPS	12654	0.0132
exp3	TCN	9218	0.0075
	CAPS	7046	0.0078

Table 5.9 Student NNs with KD from first teacher of same type for infrared radiation LBS (gains from best teacher, Table 5.8)

	KD stage (Fig. 5.7a)	**Network**	**Size**	**ΔSize** (%)	**MSE** (m^2)	**ΔMSE** (%)
exp1	1KD_TS_same	TCN	2546	−70.6	0.0038	−17.4
		CAPS	2334	−69.7	0.0051	2.8
	2KD_TS_cross	TCN	2546	−70.6	0.0035	−23.9
		CAPS	2334	−69.7	0.0047	−5.8
exp2	1KD_TS_same	TCN	11962	−37.5	0.0182	32.8
		CAPS	5506	−56.5	0.0197	49.9
	2KD_TS_cross	TCN	11962	−37.5	0.0181	32.2
		CAPS	5506	−56.5	0.0195	48.3
exp3	1KD_TS_same	TCN	3570	−61.3	0.0074	−1.2
		CAPS	4946	−29.8	0.0081	3.8
	2KD_TS_cross	TCN	3570	−61.3	0.0074	−0.4
		CAPS	4946	−29.8	0.0078	0

Student NNs optimization following the flow shown in Fig. 5.7a, determined to be optimal in Section 5.5.2, yields the results shown in Table 5.9. In exp1, the best TCN student NN improves the accuracy of the best TCN teacher NN (exp1 in Table 5.8) by 17.4 % in the first KD stage and by 23.9 % in the second, to 0.0035 m^2, and also requires much fewer parameters, 2546 from teacher's 8658, 70.6 % less. But the best CAPS student NN improves the accuracy of the best CAPS teacher

(exp1 in Table 5.8) only in the second KD stage, by 5.81 % to 0.0047 m^2 MSE, also using much less parameters, 2334 from teacher's 7726, 69.7 % less.

In exp2, all NNs are larger. The best student NNs improve only the size of their respective teacher NNs (exp2 in Table 5.8), but by less than in exp1: TCN student by 37.5 %, to 11 962 parameters from teacher's 19 130, and CAPS student by 56.5 %, to 5506 parameters from teacher's 12 654. However, the lower resource requirements correspond to losses in accuracy, by 32.2 % (0.0181 m^2 MSE) for the best student TCN NN from TCN teacher's 0.0137 m^2 MSE, and by 48.3 % (0.0195 m^2 MSE) for the best student CAPS NN from CAPS teacher's 0.0132 m^2 MSE. This may be due to the greater variability in the experimental data of exp2, which, combined with the relatively limited size of the training set of 1620 tuples (60 % of 9 min of the experiment with the sensor sampled at 5 Hz) and a narrow input window of 1 s, may have caused the teacher NN to learn fewer or less useful features, potentially leading to overfitting, and thus guide the student less efficiently during KD.

NN dimensions in exp3 shown in Tables 5.8 and 5.9 are similar to exp1 for both teachers and students, but for smaller differences to their teachers: the best TCN student at 3570 parameters is a reduction of 61.3 % of the best TCN teacher 9218 for practically the same accuracy, while the best CAPS student achieves 29.8 % reduction of parameters to 4946 from the best CAPS teacher 7046 for the same accuracy.

Experimental results for IR LBS confirm that KD can very much reduce the parameters of NNs and improve the accuracy, but not always, e.g., when it is less likely that the teacher NNs capture the most important features from the sensor data. However, a second KD, with a different teacher NN, usually improves the accuracy.

5.5.4 Results for Radar location-based system

The radar LBS uses a completely different sensor and operating principle than the capacitive and IR LBSs, but the experiments and methodology remain similar to those discussed in Sections 5.5.2 and 5.5.3. The data from three experiments, exp1, exp2, and exp3, are processed using only the Fig. 5.7a flow that gives the best results.

Teacher NNs are optimized as for capacitive and IR LBSs. The results in Table 5.10 show that they have a number of parameters typically in the same range as the best teachers in the capacitive (Table 5.2) and IR (Table 5.8) LBSs. Because

Table 5.10 Mean squared errors (sMSEs) for best temporal convolutional network (TCN) and capsule network (CAPS) teachers for radar location-based system

Experiment	**NN model**	**Parameters**	**MSE** (m^2)
exp1	TCN	7754	0.0182
	CAPS	7589	0.0187
exp2	TCN	16634	0.116
	CAPS	16250	0.115
exp3	TCN	7970	0.053
	CAPS	3227	0.054

Table 5.11 Student NNs with KD from first teacher of same type for radar LBS (gains from best teacher, Table 5.10)

	KD stage (Fig. 5.7a)	**Network**	**Size**	**ΔSize** (%)	**MSE** (m^2)	**ΔMSE** (%)
exp1	1KD_TS_same	TCN	954	−87.7	0.0169	−7.14
		CAPS	2725	−64.1	0.0149	−20.3
	2KD_TS_cross	TCN	954	−87.7	0.0155	−14.8
		CAPS	2725	−64.1	0.0132	−29.4
exp2	1KD_TS_same	TCN	7906	−52.5	0.11	−5.32
		CAPS	4523	−72.2	0.114	−0.5
	2KD_TS_cross	TCN	7906	−52.5	0.107	−7.55
		CAPS	4523	−72.2	0.113	−2.01
exp3	1KD_TS_same	TCN	449	−43.7	0.0481	−9.42
		CAPS	1508	−53.3	0.0557	2.2
	2KD_TS_cross	TCN	449	−43.7	0.051	−3.95
		CAPS	1508	−53.3	0.0495	−9.17

occasionally the noise of the radar LBS can be comparable to that of capacitive LBS and thus higher than that of the IR LBSs, the accuracy of the best teacher NNs for the radar LBS fluctuates around that of the teachers for the capacitive LBS, but is always worse than that of the IR LBS.

The best results for both TCN and CAPS NN optimization through KD are shown in Table 5.11. In exp1, the best TCN student NN improves the best TCN teacher NN

accuracy (exp1 in Table 5.10) by 7.14 % (0.0169 m^2 MSE from teacher's 0.0182 m^2) after the first KD stage, and by 14.8 % (0.0155 m^2 MSE) after the second stage with 87.7 % fewer parameters (only 954 parameters from teacher's 8658). Also the CAPS student NN improves the accuracy of the best CAPS teacher NN by 20.3 % (0.0149 m^2 MSE from teacher's 0.0187 m^2) after the first KD stage, and by 29.4 % (to 0.0132 m^2 MSE) after the second stage, with 2725 parameters, 64.1 % fewer than the teacher's 7589, but much more than the best TCN NN with 954 parameters.

In exp2, all NNs, teacher and student, are larger and less accurate because of greater movement and speed variations. However, the best TCN student NN improves both the best TCN teacher NN size by 52.5 % (7906 parameters from teacher's 16634 in Table 5.10, exp2) and accuracy by 7.55 % (0.107 m^2 MSE from teacher's 0.116 m^2). The best CAPS NN with 4523 parameters is smaller than the best TCN and 72.2 % smaller than the best CAPS teacher NN with 16250 parameters. It also has a lower accuracy (0.113 m^2 MSE) than the best TCN NN, but higher than the best CAPS teacher NN's 0.115 m^2, although the differences are small, in the order of a few percent.

In exp3, the best TCN student NN has 43.7 % fewer parameters (4490) than the best TCN teacher NN (7970), and slightly improves the accuracy by 9.42 % in first KD stage and 3.95 % in the second, to 0.051 m^2 MSE from teacher's 0.053 m^2. The best CAPS student NN has 1508 parameters, 53.3 % less than the 3227 of the best CAPS teacher NN, also improving its accuracy after the second KD stage by 9.17 % to 0.0495 m^2 MSE from the teacher's 0.054 m^2.

Experimental results for radar LBS also confirm that KD can significantly reduce NN parameters, and typically also improve accuracy. Also this experiment shows that a second KD stage, with a different teacher NN, very often improves the accuracy.

5.5.5 Generalization Performance

For the NNs optimized by the best two-stage KD flow shown in Fig. 5.7a, the generalization ability is further evaluated by testing their localization accuracy compared to the best teacher NNs trained directly with the LBS sensor data. To do this, the NNs are tested on data from experiments that are unrelated to the experiments that provided the training data.

Generalization accuracy for the capacitive LBS is summarized in Table 5.12. As

Table 5.12 Generalization MSE for capacitive LBS of CAPS and TCN student NNs tested on unseen experiments

		Network_role	Size	ΔSize (%)	MSE (m^2)	ΔMSE (%)
Train exp1	Test exp2	TCN_T	7242	n/a	**0.316**	n/a
		TCN_S1	2034	−71.9	0.44	39.3
		TCN_S2	2034	−71.9	0.393	24.6
		CPS_T	6734	n/a	**0.393**	n/a
		CPS_S1	4151	−38.4	0.409	4.2
		CPS_S2	4151	−38.4	0.416	11.1
	Test exp3	TCN_T	7242	n/a	0.334	n/a
		TCN_S1	2034	−71.9	0.34	0.65
		TCN_S2	2034	−71.9	**0.278**	−17.8
		CPS_T	6734	n/a	0.328	n/a
		CPS_S1	4151	−36.4	**0.279**	−15
		CPS_S2	4151	−36.4	0.29	−11.5
Train exp2	Test exp1	TCN_T	16826	n/a	0.433	n/a
		TCN_S1	8274	−50.8	0.374	−13.8
		TCN_S2	8274	−50.8	**0.357**	−17.7
		CPS_T	11479	n/a	0.536	n/a
		CPS_S1	5371	−53.2	**0.409**	−23.7
		CPS_S2	5371	−53.2	0.493	−8.04
	Test exp3	TCN_T	16826	n/a	0.293	n/a
		TCN_S1	8274	−50.8	**0.263**	−10.2
		TCN_S2	8274	−50.8	0.293	0
		CPS_T	11479	n/a	**0.333**	n/a
		CPS_S1	5371	−53.2	0.385	15.6
		CPS_S2	5371	−53.2	0.346	3.9
Train exp3	Test exp1	TCN_T	16826	n/a	0.535	n/a
		TCN_S1	7242	−57	0.367	−31.5
		TCN_S2	7242	−57	**0.324**	−39.6
		CAPS_T	10323	n/a	**0.403**	n/a
		CAPS_S1	8877	−14	0.435	7.99
		CAPS_S2	8877	−14	0.439	8.88
	Test exp2	TCN_T	16826	n/a	0.773	n/a
		TCN_S1	7242	−57	0.846	9.49
		TCN_S2	7242	−57	**0.719**	−6.92
		CAPS_T	10323	n/a	**0.493**	n/a
		CAPS_S1	8877	−14	0.531	7.74
		CAPS_S2	8877	−14	0.53	7.48
Average		TCN_S1	5850	−59.9	0.438	−1.01
		TCN_S2	5850	−59.9	0.394	−9.57
		CAPS_S1	6133	−35.19	0.408	−0.53
		CAPS_S2	6133	−35.19	0.422	1.97

Notations: T→teacher NN, S1→student NN from KD stage 1, S2→student NN from KD stage 2

expected, the accuracy of all NNs, teacher and student, is lower (by up to an order of magnitude) from testing on data from the same experiment (Tables 5.2 to 5.6), from which are reported the number of parameters (size).

Optimized NNs usually generalize better than the teacher NNs, but not always. For TCN NNs after the first KD stage, it ranges from $+39.3\,\%$ to $-31.5\,\%$ with an average of $-1.01\,\%$, and after the second KD stage from $+24.6\,\%$ to $-39.6\,\%$ with an average of $-9.57\,\%$. For CAPS NNs after the first KD stage, it ranges from $+15.6\,\%$ to $-23.7\,\%$ with an average of $-0.53\,\%$, and after the second KD stage from $+11.1\,\%$ to $-11.5\,\%$ with an average of $+1.97\,\%$. Most of the averages are negative, indicating better generalization than the corresponding teacher NNs, but they have a wide spread due to the large variability of the capacitive sensors. It does not seem to be a correlation between the number of parameters of the NNs and their generalization performance.

Generalization accuracy for the IR LBS is summarized in Table 5.13. Again, the accuracy of all NNs, teacher and student, is several times lower (up to an order of magnitude) than when tested on data from the same experiment (Tables 5.8 and 5.9) from which the number of parameters (size) is reported.

Similar to the capacitive LBS, the generalization accuracy of the optimized NNs is usually better than that of the teacher, but not always. For TCN NNs after the first KD stage, it ranges from $+27.9\,\%$ to $-40.4\,\%$ with an average of $-9.4\,\%$, and after the second KD stage from $+15.3\,\%$ to $-43.5\,\%$ with an average of $-14.4\,\%$. For CAPS NNs after the first KD stage, it ranges from $+12.7\,\%$ to $-32.3\,\%$ with an average of $-8\,\%$, and after the second KD stage from $+9.59\,\%$ to $-37.4\,\%$ with an average of $-10.3\,\%$. Averages are negative, indicating better generalization than the corresponding teacher NNs, but with a relatively large spread, this time probably due to the poor feature extraction of the teacher NNs discussed in Section 5.5.3. Also for this LBS, there seems to be no discernible correlation between the number of parameters of the NNs and their generalization performance. In particular, larger networks do not seem to generalize better than smaller ones.

As future work, it is worth investigating the effects of reduced size of the training set combined with increased variability of the person movements on the size of the NNs and the extraction of meaningful features that can improve KD and generalization.

Finally, generalization accuracy for the radar LBS is summarized in Table 5.14.

Table 5.13 Generalization MSE for infrared radiation LBS of CAPS and TCN student NNs tested on unseen experiments

		Network_role	Size	ΔSize (%)	MSE (m^2)	ΔMSE (%)
Train exp1	Test exp2	TCN_T	8658	n/a	0.131	n/a
		TCN_S1	2546	−70.6	**0.078**	−40.4
		TCN_S2	2546	−70.6	0.086	−34.2
		CAPS_T	7726	n/a	**0.159**	n/a
		CAPS_S1	2344	−69.7	0.179	12.7
		CAPS_S1	2344	−69.7	0.174	9.59
	Test exp3	TCN_T	8658	n/a	0.232	n/a
		TCN_S1	2546	−70.6	0.143	−38.4
		TCN_S2	2546	−70.6	**0.131**	−43.5
		CAPS_T	7726	n/a	0.224	n/a
		CAPS_S1	2344	−69.7	0.145	−32.3
		CAPS_S2	2344	−69.7	**0.14**	−37.4
Train exp2	Test exp1	TCN_T	19 130	n/a	0.028	n/a
		TCN_S1	11 962	−37.5	0.028	−0.22
		TCN_S2	11 962	−37.5	**0.025**	−13.2
		CAPS_T	12 654	n/a	**0.079**	n/a
		CAPS_S1	5506	−56.5	0.081	2.8
		CAPS_S2	5506	−56.5	0.081	2.44
	Test exp3	TCN_T	19 130	n/a	**0.053**	n/a
		TCN_S1	11 962	−37.5	0.068	27.9
		TCN_S2	11 962	−37.5	0.062	15.3
		CAPS_T	12 654	n/a	0.095	n/a
		CAPS_S1	5506	−56.5	**0.081**	−14.9
		CAPS_S2	5506	−56.5	0.082	−14
Train exp3	Test exp1	TCN_T	9218	n/a	0.051	n/a
		TCN_S1	3570	−61.3	0.042	−17.2
		TCN_S2	3570	−61.3	**0.04**	−21.1
		CAPS_T	7046	n/a	0.078	n/a
		CAPS_S1	4946	−42.5	0.065	−15.7
		CAPS_S2	4946	−42.5	**0.062**	−20.7
	Test exp2	TCN_T	9218	n/a	**0.067**	n/a
		TCN_S1	3570	−61.3	0.075	11.9
		TCN_S2	3570	−61.3	0.074	10.5
		CAPS_T	7046	n/a	0.082	n/a
		CAPS_S1	4946	−42.5	0.081	−0.6
		CAPS_S2	4946	−42.5	**0.08**	−1.6
Average		TCN_S1	6026	−56.44	0.072	−9.4
		TCN_S2	6026	−56.44	0.070	−14.4
		CAPS_S1	4265	−56.20	0.105	−8
		CAPS_S2	4265	−56.20	0.103	−10.3

Notations: T→teacher NN, S1→student NN from KD stage 1, S2→student NN from KD stage 2

Table 5.14 Generalization MSE for radar LBS of CAPS and TCN teacher and student NNs tested on unseen experiments

		Network_role	Size	ΔSize (%)	MSE (m^2)	ΔMSE (%)
Train exp1	Test exp2	TCN_T	7754	n/a	**0.295**	n/a
		TCN_S1	954	−87.7	0.302	2.42
		TCN_S2	954	−87.7	0.306	3.58
		CAPS_T	7589	n/a	0.289	n/a
		CAPS_S1	2725	−64.1	**0.278**	−3.79
		CAPS_S2	2725	−64.1	0.293	1.25
	Test exp3	TCN_T	7754	n/a	0.174	n/a
		TCN_S1	954	−87.7	**0.16**	−8.1
		TCN_S2	954	−87.7	**0.157**	−9.51
		CAPS_T	7589	n/a	0.168	n/a
		CAPS_S1	2725	−64.1	0.173	2.77
		CAPS_S2	2725	−64.1	**0.166**	−1.2
Train exp2	Test exp1	TCN_T	16634	n/a	0.124	n/a
		TCN_S1	7906	−52.5	0.125	1.42
		TCN_S2	7906	−52.5	**0.114**	−7.33
		CAPS_T	16250	n/a	0.1	n/a
		CAPS_S1	4523	−72.2	0.099	−0.247
		CAPS_S1	4523	−72.2	**0.098**	−1.66
	Test exp3	TCN_T	16634	n/a	0.468	n/a
		TCN_S1	7906	−52.5	0.469	0.1
		TCN_S2	7906	−52.5	**0.452**	−3.6
		CAPS_T	16250	n/a	0.446	n/a
		CAPS_S1	4523	−72.2	**0.441**	−1.12
		CAPS_S2	4523	−72.2	0.452	1.3
Train exp3	Test exp1	TCN_T	7970	n/a	0.085	n/a
		TCN_S1	4490	−43.7	0.079	−7.06
		TCN_S2	4490	−43.7	**0.077**	−9.65
		CAPS_T	3227	n/a	0.076	n/a
		CAPS_S1	1508	−53.3	**0.072**	−5.11
		CAPS_S1	1508	−53.3	**0.072**	−5.87
	Test exp2	TCN_T	7970	n/a	0.551	n/a
		TCN_S1	7906	−43.7	0.543	−1.44
		TCN_S2	7906	−43.7	**0.537**	−2.67
		CAPS_T	3227	n/a	0.526	n/a
		CAPS_S1	1508	−53.3	**0.518**	−1.42
		CAPS_S2	4523	−53.3	0.521	−0.924
Average		TCN_S1	4450	−61.28	0.280	−2.11
		TCN_S2	4450	−61.28	0.274	−4.86
		CAPS_S1	2918	−63.176	0.264	−1.49
		CAPS_S2	2918	−63.176	0.267	−1.18

Notations: T→teacher NN, S1→student NN from KD stage 1, S2→student NN from KD stage 2

As expected, the accuracy of all NNs, teacher and student, is several times lower (up to *two* orders of magnitude) than when tested on data from the same experiment (Tables 5.10 and 5.11) from which the number of parameters (size) is reported.

The generalization accuracy of the optimized NNs closely follows that of its teacher NNs, between $+3.58\,\%$ and $-9.65\,\%$ overall, with a relatively small spread. Moreover, the optimized NNs have relatively few parameters, often much less than for the capacitive and IR LBSs.

It is imperative to acknowledge that the observed levels of generalization exceed those achieved on the $20\,\%$ test split. It is widely recognized that the accuracy of NNs can experience significant degradation when confronted with data originating from disparate distributions. This degradation is contingent upon various factors, including but not limited to distributional shifts, NN architectures, and task complexities.

Our principal objective is to undertake a comparative analysis of the generalization capacity and robustness exhibited by distinct models, with a specific focus on delineating discrepancies between teacher and student models. Rather than directly assessing the quality of generalization per se, our emphasis lies in discerning disparities in performance across model variants.

While we acknowledge the potential efficacy of methodologies such as dynamic normalization, feature alignment, and drift detection and adaptation in bolstering the consistency and reliability of model predictions across diverse recording periods and datasets, the investigation of these techniques lies beyond the immediate scope of our current research endeavor. Such considerations, however, remain pertinent and shall be duly addressed in subsequent investigations.

Chapter 6

Conclusion and Future Work

6.1 Conclusion

The proliferation of IoT technologies has underscored the growing need for accurate real-time location tracking in various applications. Location identification involves assessing the spatial coordinates of targets using various methods, with pivotal factors including precision, accuracy, cost-effectiveness, reliability, scalability, energy efficiency, and robustness. The advent of the IoT has spurred increased interest in device-free location research, resulting in a variety of innovative applications. DFL plays a crucial role in enabling AAL in smart buildings. Its applications span across intrusion detection, fall detection, remote monitoring of the elderly, occupancy detection for energy-efficient HVAC and lighting, emergency occupancy counting, business analytics in retail, and accessibility aids for the visually impaired.

Among the device-free passive technologies, capacitive sensors working in load mode fulfil most of the requirements listed above. However, environmental drift noise can affect the measurement stability of single plate capacitive sensors for long-time long-range environmental monitoring. Based on a measurement principle similar to period modulation frontends and comparable processing complexity, a slope modulation frontend can effectively reject drift noise for single plate capacitive sensors without sensitivity loss. Moreover, the overall lower noise level notably extends the sensing range of the slope modulator frontends ($\approx 230\,\text{cm}$) compared to period modulation frontends (90 cm to 110 cm) using the same $16\,\text{cm} \times 16\,\text{cm}$ sensor

plate. Additionally, the slope modulation frontends are compatible with antialiasing filters, further reducing the noise.

These features make frontends based on slope modulation more suitable for long term indoor monitoring of persons using small size sensors that are easier to conceal to avoid discomfort and reduce cost, e.g., in assisted living applications.

Neural network architectures designed to infer the dynamics of the human body movements can improve the results due to the smoothing of the noise. 1DCNNs have been shown to excel in processing experimental time series data for indoor human localization and tracking from noisy long-range single-plate capacitive sensors [115]. However, NNs that do not use the CNN max/average pooling operations can exceed the accuracy of 1DCNNs with fewer resources. TCNs and CAPSs are designed to process time series data without using the CNN max pooling operations. Combining hyperparameter optimization with NAS can significantly improve both their accuracy and resource consumption. With NAS optimization, the CAPS was as accurate as the 1DCNN, but used only 78.7 % of the resources. Similarly, the TCN was almost as accurate as the 1DCNN after the NAS optimization, but used only a small fraction, 26.7 %, of the resources.

While deep learning models offer significant representational capabilities with minimal effort, they also require large training datasets and significant computational resources for inference. These requirements pose challenges, particularly in resource-constrained IoT systems where providing such abundant data and computing power is inherently difficult. The inherent conflict between the effectiveness of deep learning models and the practical limitations of IoT environments highlights the importance of devising innovative strategies to address these divergent requirements.

A method is proposed to optimize NNs for continuous indoor human tracking based on regression of the person's position. It combines two state-of-the-art optimization techniques: multi-level mixed-teacher KD and KD-aware NAS.

KD contributes to the proposed method an efficient transfer of knowledge from larger NNs (teachers) to smaller ones (students), helping to reduce their resource requirements and often also improving their accuracy and generalizability. Mixing the teacher NN types in a two-step KD helps student NNs focus on the relevant features that improve accuracy and generalizability by providing a different perspective on the features in the sensor data.

The NAS simultaneously optimizes both NN and KD parameters, such as NN architecture, size, and loss functions for teacher and student NNs, to achieve the best tradeoff between resource consumption and person tracking accuracy.

The test and validation of the proposed method is done using several LBSs, each using different sensing principles and data format, and affected by different noise types and levels: long-range capacitive, IRs, and radar. The experiments were also independent, conducted similarly but on different days and conditions, to help verify the robustness and applicability of the method for continuous indoor human tracking. For practical evaluation, two NNs optimized for sequential data analysis are used, TCN and CAPS.

The experimental results show that the method often significantly reduces the size of NNs and can improve their accuracy and generalizability compared to state-of-the-art models and teacher NNs. Most NNs optimized by two-stage KD generalize better, which indicates the effectiveness of the mixed two-stage KD adopted in the proposed methodology.

The proposed method proves effective and robust to minimize NN resource requirements for continuous indoor human tracking, and can be applied to other regression problems that require small and accurate NNs. Regression allows for finer-grained tracking of human activity indoors. This can improve the accuracy and overall quality of service of existing applications, as well as potentially extract more information from accurate human tracking, such as intent, mood, or other changes. They can support more tailored actions, responses, improved quality of service, or virtual or augmented reality applications such as games, guidance, or advertising.

6.2 Future Work

In the initial phase of our research, focusing on the capacitive sensor frontend, we did not account for the impact of body shape and attire on capacitive sensitivity. Exploring these factors could prove beneficial in discerning individual characteristics while respecting privacy boundaries. Moreover, all the experiments conducted in this study entail the tracking of a single individual within a stationary environment, thereby posing limitations on real-world deployment scenarios. In the future, it is envisioned that these localization techniques can be expanded to accommodate mul-

tiple individuals and larger experimental spaces, potentially encompassing furnished real rooms. Such extensions would impose greater demands on the system, thereby facilitating a more comprehensive evaluation of its capabilities.

In the post-processing phase, we will delve into a new class of NNs known as neuro-symbolic system (NSS). The evolution of artificial intelligence (AI) can be delineated into three distinct phases. The initial wave, spanning from the 1970s to the 1990s, was characterized by expert systems and symbolic, heuristic, and rule-based algorithms. Although proficient in reasoning tasks, these systems lacked the ability to learn. In contrast, the second wave emerged in the 2000s, marked by the advent of ML and neural networks. These systems exhibited remarkable learning and perception capabilities but were limited in their reasoning abilities. Recently, a third wave, signed by NSS, has emerged in the 2020s, bringing significant advancements in contextual adaptation. These systems demonstrate the capacity to comprehend context and meaning, allowing for adaptive responses [64].

Symbolic systems and NNs occupy opposite ends of the spectrum. Symbolic systems exhibit a high sensitivity to errors or inconsistencies in inputs or rules, potentially leading to unexpected or catastrophic outcomes. Additionally, they demand considerable human effort and significant computational resources, impeding their suitability for real-time applications or processing large datasets. In contrast, NNs, as discussed extensively in Chapter 5, pose challenges due to their resource-intensive nature for both training and execution, particularly for DNNs, rendering them energy-intensive, slow, and challenging to scale up [124]. Moreover, NNs lack comprehensive theoretical understanding and interpretability, hindering aspects such as their operational principles, optimal design and training, explanation, debugging, improvement, and ultimately trust [91]. Lastly, the absence of common sense and reasoning abilities poses a significant concern for NN, as it hampers their ability to comprehend contextual information and solve novel problems effectively.

Our intention is to employ this approach for indoor continuous human monitoring and tracking, with the primary objectives being to improve the system's reliability, reasoning capabilities, and, most importantly, its ability to adapt to new environments autonomously.

References

[1] Tarun Kumar Agrawal et al. “Low-Cost Intelligent Carpet System for Footstep Detection”. In: *IEEE Sensors Journal* 17.13 (2017), pp. 4239–4247. DOI: 10.1109/JSEN.2017.2703633.

[2] Arash Ahmadpour Bijargah et al. “An accurate and power-efficient period-modulator-based interface for grounded capacitive sensors”. In: *International Journal of Circuit Theory and Applications* (2019). DOI: 10.1002/cta.2642.

[3] 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* 8.6 (2021), pp. 4228–4249. DOI: 10.1109/JIOT.2020.3030174.

[4] Laith Alzubaidi et al. “Review of deep learning: Concepts, CNN architectures, challenges, applications, future directions”. In: *Journal of big Data* 8 (2021), pp. 1–74.

[5] Jack Andrews et al. “A Motion Induced Passive Infrared (PIR) Sensor for Stationary Human Occupancy Detection”. In: *2020 IEEE/ION Position, Location and Navigation Symposium (PLANS)*. 2020, pp. 1295–1304. DOI: 10.1109/PLANS46316.2020.9109909.

[6] Mihai Andries, Olivier Simonin, and François Charpillet. “Localization of Humans, Objects, and Robots Interacting on Load-Sensing Floors”. In: *IEEE Sensors Journal* 16 (July 2015). DOI: 10.1109/JSEN.2015.2493122.

[7] Devansh Arpit et al. “A Closer Look at Memorization in Deep Networks”. In: (2017). arXiv: 1706.05394 [stat.ML].

[8] Shaojie Bai, J Zico Kolter, and Vladlen Koltun. “An empirical evaluation of generic convolutional and recurrent networks for sequence modeling”. In: *arXiv 1803.01271* (2018). DOI: 10.48550/arXiv.1803.01271.

[9] Shaojie Bai, J. Zico Kolter, and Vladlen Koltun. “An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling”. In: (2018). DOI: 10.48550/ARXIV.1803.01271.

[10] Yang Bai et al. “Enhanced detection performance in electrosense through capacitive sensing”. In: *Bioinspiration & biomimetics* 11.5 (2016), p. 055001. DOI: 10.1088/1748-3190/11/5/055001.

[11] Larry K. Baxter. *Capacitive Sensors: Design and Applications*. Electronics Technology. Wiley-IEEE Press, Sept. 1996. ISBN: 978-0-7803-5351-0.

[12] Vladimir Bellavista-Parent, Joaquín Torres-Sospedra, and Antoni Pérez-Navarro. "Comprehensive Analysis of Applied Machine Learning in Indoor Positioning Based on Wi-Fi: An Extended Systematic Review". In: *Sensors* 22.12 (2022). ISSN: 1424-8220. DOI: 10.3390/s22124622.

[13] *BGT60TR13C Datasheet*. Rev. 2.4.6. Infineon Technologies. Jan. 2021.

[14] Miloš Bjelić et al. "Statistical properties of quantisation noise in analogue-to-digital converter with oversampling and decimation". In: *IET Circuits, Devices & Systems* 11.5 (2017), pp. 421–427. DOI: 10.1049/iet-cds.2016.0506.

[15] J Boehm, R Gerber, and NRC Kiley. "Sensors for magnetic bearings". In: 29.6 (1993), pp. 2962–2964. DOI: 10.1109/20.280903.

[16] Joan Bordoy et al. "Single transceiver device-free indoor localization using ultrasound body reflections and walls". In: *2015 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*. IEEE. 2015, pp. 1–7.

[17] James Bradbury et al. "Quasi-Recurrent Neural Networks". In: (2016). DOI: 10.48550/ARXIV.1611.01576.

[18] Andreas Braun, Silvia Rus, and Martin Majewski. "Invisible Human Sensing in Smart Living Environments Using Capacitive Sensors". In: *Ambient Assisted Living*. Springer, 2017, pp. 43–53. DOI: 10.1007/978-3-319-52322-4_3.

[19] William Buller and Brian Wilson. "Measuring the Capacitance of Electrical Wiring and Humans for Proximity Sensing with Existing Electrical Infrastructure". In: *2006 IEEE International Conference on Electro/Information Technology*. 2006, pp. 93–96.

[20] Shiyu Chang et al. "Dilated Recurrent Neural Networks". In: (2017). DOI: 10.48550/ARXIV.1710.02224.

[21] B.B. Chaudhuri and U. Bhattacharya. "Efficient training and improved performance of multilayer perceptron in pattern classification". In: *Neurocomputing* 34.1 (2000), pp. 11–27. ISSN: 0925-2312. DOI: https://doi.org/10.1016/S0925-2312(00)00305-2. URL: https://www.sciencedirect.com/science/article/pii/S0925231200003052.

[22] Zhangjie Chen, Ya Wang, and Hanwei Liu. "Unobtrusive Sensor-Based Occupancy Facing Direction Detection and Tracking Using Advanced Machine Learning Algorithms". In: *IEEE Sensors Journal* 18.15 (2018), pp. 6360–6368. DOI: 10.1109/JSEN.2018.2844252.

[23] Kenneth Cheung, Stephen Intille, and Kent Larson. "An Inexpensive Bluetooth-Based Indoor Positioning Hack". In: Jan. 2006.

[24] François Chollet et al. *Keras*. https://keras.io. 2015.

[25] Chien-Hao Chu et al. "High-accuracy indoor personnel tracking system with a ZigBee wireless sensor network". In: *2011 Seventh International Conference on Mobile Ad-hoc and Sensor Networks*. IEEE. 2011, pp. 398–402.

[26] Marco Cimdins et al. “Exploiting Ultra-Wideband Channel Impulse Responses for Device-Free Localization”. In: *Sensors* 22.16 (2022). ISSN: 1424-8220. DOI: 10.3390/s22166255. URL: https://www.mdpi.com/1424-8220/22/16/6255.

[27] Jose Clemente et al. “Smart Seismic Sensing for Indoor Fall Detection, Location, and Notification”. In: *IEEE Journal of Biomedical and Health Informatics* 24.2 (2020), pp. 524–532. DOI: 10.1109/JBHI.2019.2907498.

[28] *Comprehensive Study of the Howland Current Pump | Texas Instruments*. https://www.ti.com/lit/pdf/snoa474. Accessed: 2022-02-09.

[29] Mohamad Daher et al. “Elder Tracking and Fall Detection System Using Smart Tiles”. In: *IEEE Sensors Journal* 17.2 (2017), pp. 469–479. DOI: 10.1109/JSEN.2016.2625099.

[30] Jihan Dai et al. “A Survey of Latest Wi-Fi Assisted Indoor Positioning on Different Principles”. In: *Sensors* 23.18 (2023). ISSN: 1424-8220. DOI: 10.3390/s23187961. URL: https://www.mdpi.com/1424-8220/23/18/7961.

[31] *Datasheet STM32L412xx Ultra-low-power Arm® Cortex*. https://www.st.com/resource/en/datasheet/stm32l412kb.pdf. Accessed: 2022-02-09.

[32] Andrea De Marcellis, Giuseppe Ferri, and Paolo Mantenuto. “A CCII-based non-inverting Schmitt trigger and its application as astable multivibrator for capacitive sensor interfacing”. In: *Int. J. Circ. Theor. Appl.* 45.8 (2017), pp. 1060–1076. DOI: 10.1002/cta.2268.

[33] Estefania Munoz Diaz et al. “Optimal sampling frequency and bias error modeling for foot-mounted IMUs”. In: *International Conference on Indoor Positioning and Indoor Navigation*. IEEE. 2013, pp. 1–9.

[34] El Alaoui-Elfels, Omaima and Gadi, Taoufiq. “From Auto-encoders to Capsule Networks: A Survey”. In: *E3S Web Conf.* 229 (2021), p. 01048. DOI: 10.1051/e3sconf/202122901048.

[35] Zahid Farid, Rosdiadee Nordin, and Mahamod Ismail. “Recent Advances in Wireless Indoor Localization Techniques and System”. In: *Journal of Computer Networks and Communications* 2013 (Jan. 2013). DOI: 10.1155/2013/185138.

[36] Nathaniel Faulkner et al. “Device Free Localization with Capacitive Sensing Floor”. In: *2020 IEEE Sensors Applications Symposium (SAS)*. 2020, pp. 1–5. DOI: 10.1109/SAS48726.2020.9220042.

[37] N Faulkner et al. “Watchers on the Wall: Passive Visible Light-Based Positioning and Tracking with Embedded Light-Sensors on the Wall”. In: *IEEE Transactions on Instrumentation and Measurement* 69.5 (2020), pp. 2522–2532.

[38] Jana Fortes and Dusan Kocur. “Solutions of mutual shadowing effect between people tracked by UWB radar”. In: *2013 IEEE International Conference on Microwaves, Communications, Antennas and Electronic Systems (COMCAS 2013)*. IEEE. 2013, pp. 1–5.

[39] Peter I. Frazier. "A Tutorial on Bayesian Optimization". In: (2018). arXiv: 1807.02811 [stat.ML].

[40] Jianping Gou et al. "Knowledge Distillation: A Survey". In: *International Journal of Computer Vision* 129.6 (Mar. 2021), pp. 1789–1819. ISSN: 1573-1405. DOI: 10.1007/s11263-021-01453-z.

[41] Tobias Grosse-Puppendahl et al. "Finding Common Ground: A Survey of Capacitive Sensing in Human-Computer Interaction". In: *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. CHI '17. Denver, Colorado, USA: Association for Computing Machinery, 2017, pp. 3293–3315.

[42] Tobias Grosse-Puppendahl et al. "Platypus: Indoor Localization and Identification through Sensing of Electric Potential Changes in Human Bodies". In: *Proceedings of the 14th Annual International Conference on Mobile Systems, Applications, and Services*. MobiSys '16. Singapore, Singapore: Association for Computing Machinery, 2016, pp. 17–30. ISBN: 9781450342698. DOI: 10.1145/2906388.2906402. URL: https://doi.org/10.1145/2906388.2906402.

[43] 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)*. 2020, pp. 1926–1931. DOI: 10.1109/ICIEA48937.2020.9248333.

[44] Tianbo Gu et al. "mmSense: Multi-Person Detection and Identification via mmWave Sensing". In: *Proceedings of the 3rd ACM Workshop on Millimeter-Wave Networks and Sensing Systems*. mmNets'19. Los Cabos, Mexico: Association for Computing Machinery, 2019, pp. 45–50. ISBN: 9781450369329. DOI: 10.1145/3349624.3356765. URL: https://doi.org/10.1145/3349624.3356765.

[45] Karim Guirguis et al. "SELD-TCN: Sound Event Localization & Detection via Temporal Convolutional Networks". In: *2020 28th European Signal Processing Conference (EUSIPCO)*. IEEE, Jan. 2021. DOI: 10.23919/eusipco47968.2020.9287716.

[46] Gongde Guo et al. "KNN Model-Based Approach in Classification". In: *On The Move to Meaningful Internet Systems 2003: CoopIS, DOA, and ODBASE*. Ed. by Robert Meersman, Zahir Tari, and Douglas C. Schmidt. Berlin, Heidelberg: Springer Berlin Heidelberg, 2003, pp. 986–996. ISBN: 978-3-540-39964-3.

[47] Qi Hao, Fei Hu, and Jiang Lu. "Distributed multiple human tracking with wireless binary pyroelectric infrared (PIR) sensor networks". In: *2010 IEEE Sensors* (2010), pp. 946–950. URL: https://api.semanticscholar.org/CorpusID:22219780.

[48] S.J. Hayward et al. "A Survey of Indoor Location Technologies, Techniques and Applications in Industry". In: *Internet of Things* 20 (2022), p. 100608. ISSN: 2542-6605. DOI: https://doi.org/10.1016/j.iot.2022.100608. URL: https://www.sciencedirect.com/science/article/pii/S2542660522000907.

[49] 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*. UbiComp ’14. Seattle, Washington: Association for Computing Machinery, 2014, pp. 141–145. ISBN: 9781450329682. DOI: 10.1145/2632048.2636084. URL: https://doi.org/10.1145/2632048.2636084.

[50] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. “Distilling the Knowledge in a Neural Network”. In: (2015). arXiv: 1503.02531 [stat.ML].

[51] Javed Iqbal et al. “Capacitive Sensor for Tagless Remote Human Identification Using Body Frequency Absorption Signatures”. In: 67.4 (2018), pp. 789–797. DOI: 10.1109/TIM.2017.2789078.

[52] Hirunima Jayasekara et al. *TimeCaps: Capturing Time Series Data With Capsule Networks*. 2022. arXiv: 1911.11800 [cs.LG].

[53] Bing Jia et al. “TTSL: An indoor localization method based on Temporal Convolutional Network using time-series RSSI”. In: *Computer Communications* 193 (2022), pp. 293–301. ISSN: 0140-3664. DOI: https://doi.org/10.1016/j.comcom.2022.07.003.

[54] Feng Jin et al. “Multiple Patients Behavior Detection in Real-time using mmWave Radar and Deep CNNs”. In: *2019 IEEE Radar Conference (RadarConf)*. 2019, pp. 1–6. DOI: 10.1109/RADAR.2019.8835656.

[55] Haifeng Jin et al. “AutoKeras: An AutoML Library for Deep Learning”. In: *Journal of Machine Learning Research* 24.6 (2023), pp. 1–6.

[56] Daesil Kang, Wongoo Lee, and Wonkyu Moon. “A technique for drift compensation of an area-varying capacitive displacement sensor for nanometrology”. In: *Procedia Engineering* 5 (2010), pp. 412–415. DOI: 10.1016/j.proeng.2010.09.134.

[57] M.A. Kelly, G.E. Servais, and T.V. Pfaffenbach. “An investigation of human body electrostatic discharge”. In: *International Symposium for Testing and Failure Analysis*. American Technical Publishers ltd. Nov. 1993, pp. 167–167.

[58] Kang-Wook Kim et al. “Accurate Indoor Location Tracking Exploiting Ultrasonic Reflections”. In: *IEEE Sensors Journal* 16.24 (2016), pp. 9075–9088. DOI: 10.1109/JSEN.2016.2617398.

[59] Youngjoo Kim et al. “A Capsule Network for Traffic Speed Prediction in Complex Road Networks”. In: *2018 Sensor Data Fusion: Trends, Solutions, Applications (SDF)*. 2018, pp. 1–6. DOI: 10.1109/SDF.2018.8547068.

[60] Youngwook Kim, Ibrahim Alnujaim, and Daegun Oh. “Human Activity Classification Based on Point Clouds Measured by Millimeter Wave MIMO Radar With Deep Recurrent Neural Networks”. In: *IEEE Sensors Journal* 21.12 (2021), pp. 13522–13529. DOI: 10.1109/JSEN.2021.3068388.

[61] Youngwook Kim, Sungjae Ha, and Jihoon Kwon. “Human detection using Doppler radar based on physical characteristics of targets”. In: *IEEE Geoscience and Remote Sensing Letters* 12.2 (2014), pp. 289–293.

[62] Christian Kowalski. et al. “Multi Low-resolution Infrared Sensor Setup for Privacy-preserving Unobtrusive Indoor Localization”. In: *Proceedings of the 5th International Conference on Information and Communication Technologies for Ageing Well and e-Health - ICT4AWE*. INSTICC. SciTePress, 2019, pp. 183–188. ISBN: 978-989-758-368-1. DOI: 10.5220/0007694601830188.

[63] 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)*. 2012, pp. 2042–2047. DOI: 10.1109/ICSMC.2012.6378039.

[64] John Launchbury. *A DARPA Perspective on Artificial Intelligence*. Technica Curiosa. Accessed: Dec 2, 2023. Mar. 2017. URL: https://machinelearning.technicacuriosa.com/2017/03/19/a-darpa-perspective-on-artificial-intelligence/.

[65] Christl Lauterbach, Axel Steinhage, and Axel Techmer. “Large-area wireless sensor system based on smart textiles”. In: *International Multi-Conference on Systems, Signals & Devices*. 2012, pp. 1–2.

[66] Colin Lea et al. “Temporal Convolutional Networks for Action Segmentation and Detection”. In: (2016). DOI: 10.48550/ARXIV.1611.05267.

[67] Seung Hyun Lee, Dae Ha Kim, and Byung Cheol Song. “Self-supervised Knowledge Distillation Using Singular Value Decomposition”. In: (2018). arXiv: 1807.06819 [cs.LG].

[68] Samuel G. Leitch et al. “On Indoor Localization Using WiFi, BLE, UWB, and IMU Technologies”. In: *Sensors* 23.20 (2023). ISSN: 1424-8220. DOI: 10.3390/s23208598. URL: https://www.mdpi.com/1424-8220/23/20/8598.

[69] Tianxing Li, Qiang Liu, and Xia Zhou. “Practical Human Sensing in the Light”. In: *Proceedings of the 14th Annual International Conference on Mobile Systems, Applications, and Services*. MobiSys ’16. Singapore, Singapore: Association for Computing Machinery, 2016, pp. 71–84. ISBN: 9781450342698. DOI: 10.1145/2906388.2906401. URL: https://doi.org/10.1145/2906388.2906401.

[70] Zewen Li et al. “A Survey of Convolutional Neural Networks: Analysis, Applications, and Prospects”. In: *IEEE Transactions on Neural Networks and Learning Systems* 33.12 (2022), pp. 6999–7019. DOI: 10.1109/TNNLS.2021.3084827.

[71] Tailin Liang et al. “Pruning and Quantization for Deep Neural Network Acceleration: A Survey”. In: (2021). arXiv: 2101.09671 [cs.CV].

[72] Yuang Liu, Wei Zhang, and Jun Wang. “Adaptive multi-teacher multi-level knowledge distillation”. In: *Neurocomputing* 415 (2020), pp. 106–113.

[73] Shane A Lowe and Gearóid ÓLaighin. “Monitoring human health behaviour in one’s living environment: a technological review”. In: *Medical engineering & physics* 36.2 (2014), pp. 147–168. DOI: 10.1016/j.medengphy.2013.11.010.

[74] *LTspice Simulator | Analog Devices*. https://www.analog.com/en/design-center/design-tools-and-calculators/ltspice-simulator.html. Accessed: 2022-02-09.

[75] D Marioli, E Sardini, and A Taroni. "High-accuracy measurement techniques for capacitance transducers". In: *Measurement Science and Technology* 4.3 (Mar. 1993), pp. 337–343. DOI: 10.1088/0957-0233/4/3/012. URL: https://doi.org/10.1088/0957-0233/4/3/012.

[76] Aqilah Binti Mazlan, Yin Hoe Ng, and Chee Keong Tan. "A Fast Indoor Positioning Using a Knowledge-Distilled Convolutional Neural Network (KD-CNN)". In: *IEEE Access* 10 (2022), pp. 65326–65338. DOI: 10.1109/ACCESS.2022.3183113.

[77] Aqilah Binti Mazlan, Yin Hoe Ng, and Chee Keong Tan. "A fast indoor positioning using a Knowledge-Distilled Convolutional Neural Network (KD-CNN)". In: *IEEE Access* 10 (2022), pp. 65326–65338. DOI: 10.1109/ACCESS.2022.3183113.

[78] Aqilah Binti Mazlan, Yin Hoe Ng, and Chee Keong Tan. "Teacher-Assistant Knowledge Distillation Based Indoor Positioning System". In: *Sustainability* 14.21 (2022). ISSN: 2071-1050. DOI: 10.3390/su142114652.

[79] Zhen Meng et al. "Gait Recognition for Co-Existing Multiple People Using Millimeter Wave Sensing". In: *Proceedings of the AAAI Conference on Artificial Intelligence* 34.01 (Apr. 2020), pp. 849–856. DOI: 10.1609/aaai.v34i01.5430.

[80] Mostafa Mirshekari et al. "Occupant localization using footstep-induced structural vibration". In: *Mechanical Systems and Signal Processing* 112 (2018), pp. 77–97. ISSN: 0888-3270. DOI: https://doi.org/10.1016/j.ymssp.2018.04.026. URL: https://www.sciencedirect.com/science/article/pii/S0888327018302280.

[81] Mostafa Mirshekari et al. "Step-Level Occupant Detection across Different Structures through Footstep-Induced Floor Vibration Using Model Transfer". In: *Journal of Engineering Mechanics* 146 (Mar. 2020), p. 04019137. DOI: 10.1061/(ASCE)EM.1943-7889.0001719.

[82] N Madhu Mohan et al. "Digital converter for differential capacitive sensors". In: *IEEE Transactions on Instrumentation and Measurement* 57.11 (2008), pp. 2576–2581. DOI: 10.1109/TIM.2008.922109.

[83] Bodhibrata Mukhopadhyay et al. "Indoor localization using analog output of pyroelectric infrared sensors". In: Apr. 2018, pp. 1–6. DOI: 10.1109/WCNC.2018.8377063.

[84] Ibrahim Al-Naimi and Chi Biu Wong. "Indoor human detection and tracking using advanced smart floor". In: *2017 8th International Conference on Information and Communication Systems (ICICS)*. 2017, pp. 34–39. DOI: 10.1109/IACS.2017.7921942.

[85] Viet Nguyen et al. “Eyelight: Light-and-Shadow-Based Occupancy Estimation and Room Activity Recognition”. In: *IEEE INFOCOM 2018 - IEEE Conference on Computer Communications*. 2018, pp. 351–359. DOI: 10.1109/INFOCOM.2018.8485867.

[86] Inyoung Paik, Taeyeong Kwak, and Injung Kim. “Capsule networks need an improved routing algorithm”. In: *Asian Conference on Machine Learning*. PMLR. 2019, pp. 489–502.

[87] Aakash Parmar, Rakesh Katariya, and Vatsal Patel. “A Review on Random Forest: An Ensemble Classifier”. In: *International Conference on Intelligent Data Communication Technologies and Internet of Things (ICICI) 2018*. Ed. by Jude Hemanth et al. Cham: Springer International Publishing, 2019, pp. 758–763. ISBN: 978-3-030-03146-6.

[88] Jacopo Pegoraro, Francesca Meneghello, and Michele Rossi. “Multiperson Continuous Tracking and Identification From mm-Wave Micro-Doppler Signatures”. In: *IEEE Transactions on Geoscience and Remote Sensing* 59.4 (2021), pp. 2994–3009. DOI: 10.1109/TGRS.2020.3019915.

[89] Jacopo Pegoraro et al. “Deep Learning for Accurate Indoor Human Tracking with a mm-Wave Radar”. In: *2020 IEEE Radar Conference (RadarConf20)*. 2020, pp. 1–6. DOI: 10.1109/RadarConf2043947.2020.9266400.

[90] Derek A. Pisner and David M. Schnyer. “Chapter 6 - Support vector machine”. In: *Machine Learning*. Ed. by Andrea Mechelli and Sandra Vieira. Academic Press, 2020, pp. 101–121. ISBN: 978-0-12-815739-8. DOI: https://doi.org/10.1016/B978-0-12-815739-8.00006-7. URL: https://www.sciencedirect.com/science/article/pii/B9780128157398000067.

[91] Tomaso Poggio, Andrzej Banburski, and Qianli Liao. “Theoretical issues in deep networks”. In: *Proceedings of the National Academy of Sciences* 117.48 (2020), pp. 30039–30045.

[92] Antonio Polino, Razvan Pascanu, and Dan Alistarh. “Model compression via distillation and quantization”. In: (2018). arXiv: 1802.05668 [cs.NE].

[93] H Prance et al. “Position and movement sensing at metre standoff distances using ambient electric field”. In: *Measurement Science and Technology* 23 (Oct. 2012), p. 115101. DOI: 10.1088/0957-0233/23/11/115101.

[94] Aji Gautama Putrada et al. “Knowledge Distillation for a Lightweight Deep Learning-Based Indoor Positioning System on Edge Environments”. In: *2023 International Seminar on Intelligent Technology and Its Applications (ISITIA)*. 2023, pp. 370–375. DOI: 10.1109/ISITIA59021.2023.10220996.

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

[96] *REF2033 3.3-V Vref, low-drift, low-power, dual-output Vref & Vref/2 voltage reference*. https://www.ti.com/product/REF2033. Accessed: 2022-05-16.

[97] Ferran Reverter and Òscar Casas. “Interfacing differential capacitive sensors to microcontrollers: A direct approach”. In: *IEEE Transactions on Instrumentation and Measurement* 59.10 (2009), pp. 2763–2769. DOI: 10.1109/TIM.2009.2036500.

[98] Ferran Reverter, Xiujun Li, and Gerard CM Meijer. “Stability and accuracy of active shielding for grounded capacitive sensors”. In: *Measurement Science and Technology* 17.11 (2006), p. 2884. DOI: 10.1088/0957-0233/17/11/004.

[99] Priya Roy and Chandreyee Chowdhury. “A Survey of Machine Learning Techniques for Indoor Localization and Navigation Systems”. In: *Journal of Intelligent & Robotic Systems* 101 (Mar. 2021). DOI: 10.1007/s10846-021-01327-z.

[100] Hamidreza Sadreazami, Miodrag Bolic, and Sreeraman Rajan. “CapsFall: Fall Detection Using Ultra-Wideband Radar and Capsule Network”. In: *IEEE Access* 7 (2019), pp. 55336–55343. DOI: 10.1109/ACCESS.2019.2907925.

[101] Muhamad Risqi U. Saputra et al. “Distilling Knowledge From a Deep Pose Regressor Network”. In: (2019). arXiv: 1908.00858 [cs.CV].

[102] Arindam Sengupta et al. “mm-Pose: Real-Time Human Skeletal Posture Estimation Using mmWave Radars and CNNs”. In: *IEEE Sensors Journal* 20.17 (Sept. 2020), pp. 10032–10044. DOI: 10.1109/jsen.2020.2991741.

[103] Ahmadreza Sezavar, Randa Atta, and Mohammed Ghanbari. “DCapsNet: Deep capsule network for human activity and gait recognition with smartphone sensors”. In: *Pattern Recognition* 147 (2024), p. 110054. ISSN: 0031-3203. DOI: https://doi.org/10.1016/j.patcog.2023.110054. URL: https://www.sciencedirect.com/science/article/pii/S0031320323007513.

[104] Claude Elwood Shannon. “Communication in the presence of noise”. In: *Proceedings of the IRE* 37.1 (1949), pp. 10–21. DOI: 10.1109/JRPROC.1949.232969.

[105] Laixi Shi et al. “Device-free Multiple People Localization through Floor Vibration”. In: *Proceedings of the 1st ACM International Workshop on Device-Free Human Sensing*. DFHS’19. New York, NY, USA: Association for Computing Machinery, 2019, pp. 57–61. ISBN: 9781450370073. DOI: 10.1145/3360773.3360887. URL: https://doi.org/10.1145/3360773.3360887.

[106] Xingjian Shi et al. “Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting”. In: *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*. NIPS’15. Montreal, Canada: MIT Press, 2015, pp. 802–810.

[107] Akash Deep Singh et al. “RadHAR: Human Activity Recognition from Point Clouds Generated through a Millimeter-Wave Radar”. In: *Proceedings of the 3rd ACM Workshop on Millimeter-Wave Networks and Sensing Systems*. mmNets’19. Los Cabos, Mexico: Association for Computing Machinery, 2019, pp. 51–56. ISBN: 9781450369329. DOI: 10.1145/3349624.3356768.

[108] J. Smith et al. “Electric field sensing for graphical interfaces”. In: *IEEE Computer Graphics and Applications* 18.3 (1998), pp. 54–60. DOI: 10.1109/38.674972.

[109] Wonchul Son et al. "Densely Guided Knowledge Distillation using Multiple Teacher Assistants". In: *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*. IEEE. 2021, pp. 9375–9384. DOI: 10.1109/ICCV48922.2021.00926.

[110] Giorgia Subbicini, Luciano Lavagno, and Mihai T. Lazarescu. "Enhanced Exploration of Neural Network Models for Indoor Human Monitoring". In: *2023 9th International Workshop on Advances in Sensors and Interfaces (IWASI)*. 2023, pp. 109–114. DOI: 10.1109/IWASI58316.2023.10164436.

[111] Yongliang Sun, Xiaocheng Wang, and Xuzhao Zhang. "Deep Learning-Based Device-Free Localization Using ZigBee". In: *Communications, Signal Processing, and Systems*. Ed. by Qilian Liang et al. Singapore: Springer Singapore, 2020, pp. 2046–2049. ISBN: 978-981-13-9409-6.

[112] Karush Suri and Rinki Gupta. "Continuous sign language recognition from wearable IMUs using deep capsule networks and game theory". In: *Computers & Electrical Engineering* 78 (2019), pp. 493–503. ISSN: 0045-7906. DOI: https://doi.org/10.1016/j.compeleceng.2019.08.006.

[113] Zhichao Tan et al. "An energy-efficient 15-bit capacitive-sensor interface based on period modulation". In: 47.7 (2012), pp. 1703–1711. DOI: i10.1109/JSSC.2012.2191212.

[114] Xinyao Tang and Soumyajit Mandal. "Indoor Occupancy Awareness and Localization Using Passive Electric Field Sensing". In: *IEEE Transactions on Instrumentation and Measurement* 68.11 (2019), pp. 4535–4549. DOI: 10.1109/TIM.2018.2890319.

[115] Osama Bin Tariq, Mihai Teodor Lazarescu, and Luciano Lavagno. "Neural Networks for Indoor Human Activity Reconstructions". In: *IEEE Sensors Journal* 20.22 (2020), pp. 13571–13584. DOI: 10.1109/JSEN.2020.3006009.

[116] Osama Bin Tariq, Mihai Teodor Lazarescu, and Luciano Lavagno. "Neural networks for indoor person tracking with infrared sensors". In: *IEEE Sensors Letters* 5.1 (2021), pp. 1–4.

[117] Osama Bin Tariq et al. "Performance of machine learning classifiers for indoor person localization with capacitive sensors". In: *IEEE Access* 5 (2017), pp. 12913–12926. DOI: 10.1109/ACCESS.2017.2721538.

[118] Ahmed Tealab. "Time series forecasting using artificial neural networks methodologies: A systematic review". In: *Future Computing and Informatics Journal* 3.2 (2018), pp. 334–340. ISSN: 2314-7288. DOI: https://doi.org/10.1016/j.fcij.2018.10.003. URL: https://www.sciencedirect.com/science/article/pii/S2314728817300715.

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

[120] Sai Raam Venkataraman, S Balasubramanian, and R Raghunatha Sarma. "Iterative collaborative routing among equivariant capsules for transformation-robust capsule networks". In: *arXiv preprint arXiv:2210.11095* (2022).

[121] Hui Wang et al. “Progressive Blockwise Knowledge Distillation for Neural Network Acceleration”. In: *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*. International Joint Conferences on Artificial Intelligence Organization, July 2018, pp. 2769–2775. DOI: 10.24963/ijcai.2018/384.

[122] Zhenghuan Wang et al. “Towards robust and efficient device-free localization using UWB sensor network”. In: *Pervasive and Mobile Computing* 41 (2017), pp. 451–469. ISSN: 1574-1192. DOI: https://doi.org/10.1016/j.pmcj.2017.03.006. URL: https://www.sciencedirect.com/science/article/pii/S1574119217301608.

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

[124] Qizhe Xie et al. “Self-training with noisy student improves imagenet classification”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2020, pp. 10687–10698.

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

[126] Qianwen Ye et al. “CapsLoc: A Robust Indoor Localization System with WiFi Fingerprinting Using Capsule Networks”. In: *ICC 2020 - 2020 IEEE International Conference on Communications (ICC)*. 2020, pp. 1–6. DOI: 10.1109/ICC40277.2020.9148933.

[127] Yong Yu et al. “A Review of Recurrent Neural Networks: LSTM Cells and Network Architectures”. In: *Neural Computation* 31.7 (July 2019), pp. 1235–1270. ISSN: 0899-7667. DOI: 10.1162/neco_a_01199. eprint: https://direct.mit.edu/neco/article-pdf/31/7/1235/1053200/neco_a_01199.pdf. URL: https://doi.org/10.1162/neco%5C_a%5C_01199.

[128] 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. DOI: 10.1109/COMST.2019.2911558.

[129] Hailin Zhang, Defang Chen, and Can Wang. “Confidence-aware multi-teacher knowledge distillation”. In: *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2022, pp. 4498–4502. DOI: 10.1109/ICASSP43922.2022.9747534.

[130] Renyuan Zhang and Siyang Cao. “Real-Time Human Motion Behavior Detection via CNN Using mmWave Radar”. In: *IEEE Sensors Letters* 3.2 (2019), pp. 1–4. DOI: 10.1109/LSENS.2018.2889060.

[131] Haoran Zhao et al. “Highlight every step: Knowledge distillation via collaborative teaching”. In: *IEEE Transactions on Cybernetics* 52.4 (2020), pp. 2070–2081. DOI: 10.1109/TCYB.2020.3007506.

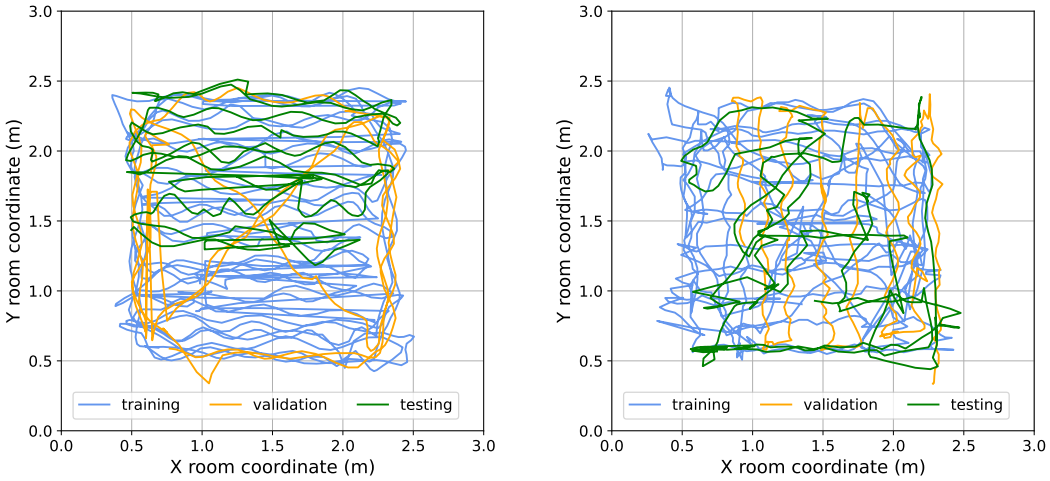
[132] Zhihao Zhao and Samuel Cheng. “Capsule networks with non-iterative cluster routing”. In: *Neural Networks* 143 (2021), pp. 690–697.

Appendix A

Generalization Experiments

Capacitive location-based system

In Fig. A.1 multiple experimental paths for capacitive LBS, related to `exp2` and `exp3`



(a) Experimental path `exp2` for capacitive location-based system

(b) Experimental path `exp3` for capacitive location-based system

Fig. A.1 Multiple experimental paths for capacitive location-based systems.

discussed in Chapter 5 are displayed. All the trajectories are conducted by a single young individual in a 3 x 3 virtual room and they are split into 60 % for training, 20 % for validation, and 20 % for testing the neural networks. In Table A.1

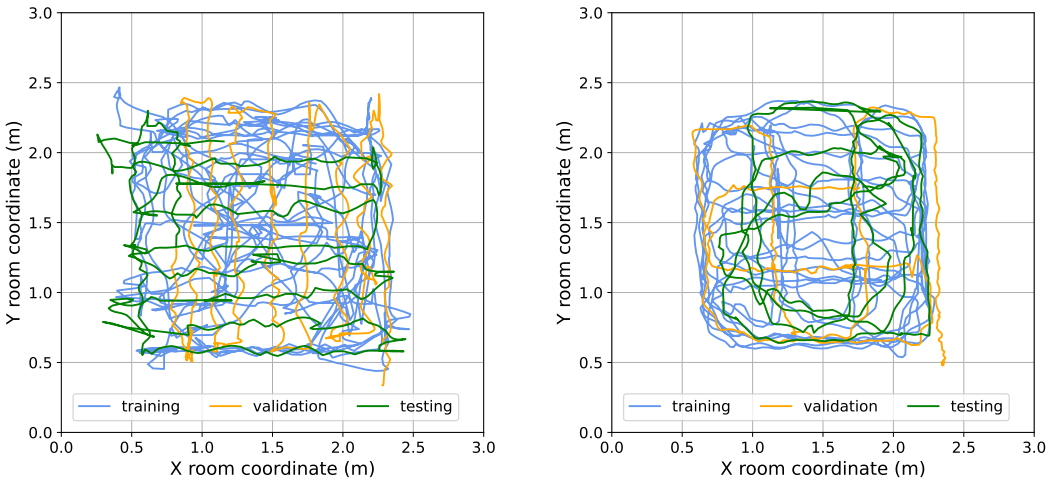
Table A.1 Details for `exp1` `exp2` `exp3` capacitive location-based system datasets

Exp	**Sampl.Freq.** (Hz)	**Duration** (minutes)	**Total Dist.** (meters)	**Avg.Speed** m/s
`exp1`	3	∼9	72	0.13
`exp2`	3	∼9	148	0.27
`exp3`	3	∼9	117	0.21

detailed characteristics of each dataset are reported.

Infrared location-based system

In Fig. A.2 multiple experimental paths for IR LBS, related to `exp2` and `exp3`



(a) Experimental path `exp2` for IR location-based system

(b) Experimental path `exp3` for IR location-based system

Fig. A.2 Multiple experimental paths for IR location-based systems

discussed in Chapter 5 are displayed. All the trajectories are conducted by a single young individual in a 3 x 3 virtual room and they are split into 60 % for training, 20 % for validation, and 20 % for testing the neural networks. In Table A.2

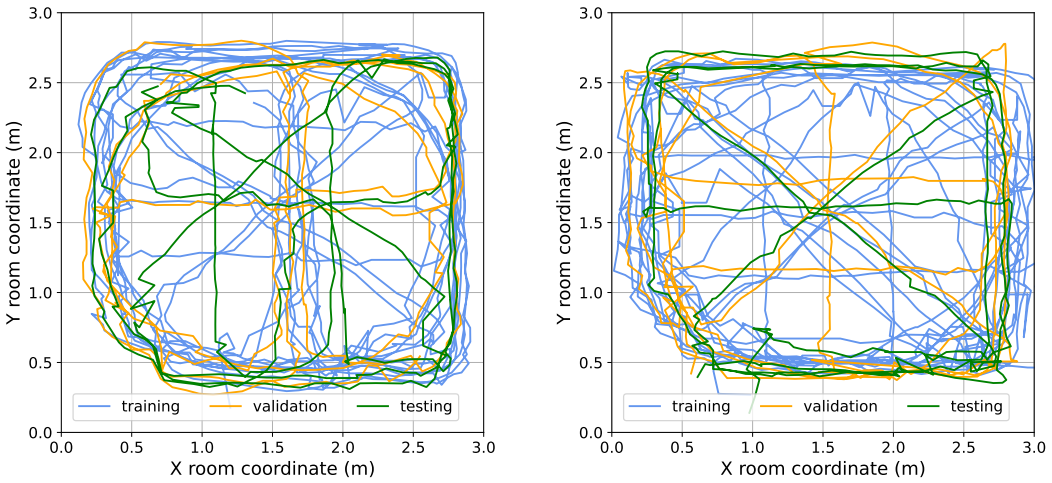
Table A.2 Details for `exp1` `exp2` `exp3` IR location-based system datasets

Exp	**Sampl.Freq** (Hz)	**Duration** (minutes)	**Total Dist.** (meters)	**Avg.Speed** m/s
`exp1`	5	∼9	78	0.13
`exp2`	5	∼9	142	0.23
`exp3`	5	∼9	106	0.18

detailed characteristics of each dataset are reported.

Radar location-based system

In Fig. A.3 multiple experimental paths for radar LBS, related to `exp2` and `exp3`



(a) Experimental path `exp2` for radar location-based system

(b) Experimental path `exp3` for radar location-based system

Fig. A.3 Multiple experimental paths for radar location-based systems.

discussed in Chapter 5 are displayed. All the trajectories are conducted by a single young individual in a 3 x 3 virtual room and they are split into 60 % for training, 20 % for validation, and 20 % for testing the neural networks. In Table A.2

Table A.3 Details for `exp1` `exp2` `exp3` radar location-based system datasets

Exp	**Sampl.Freq.** (Hz)	**Duration** (minutes)	**Total Dist.** (meters)	**Avg.Speed** m/s
`exp1`	4	∼9	154	0.31
`exp2`	4	∼9	207	0.41
`exp3`	4	∼9	224	0.44

detailed characteristics of each dataset are reported.