



UNIONE EUROPEA
Fondo Sociale Europeo



UNIVERSITA' DELLA CALABRIA

Dipartimento di Ingegneria Informatica, Modellistica, Elettronica e Sistemistica

Dottorato di Ricerca in

Tecnologie dell'informazione e della comunicazione

Con il contributo di

PON R&I 2014-2020 and ICAR-CNR

XXXVII CICLO

TESI DI DOTTORATO

Novel Approaches for Occupancy Prediction and Activity Recognition in Cognitive Buildings

Settore Scientifico Disciplinare: ING-INF/05

Coordinatore: Ch.mo Prof. Giancarlo Fortino
Firma _____

Supervisore: Ch.mo Prof. Antonio Guerrieri
Firma _____

Dottorando: Dott. Irfanullah Khan

Firma _____

Dedicated to my family:

*My beloved parents, my lovely wife, my son, and all my siblings.
for their prayer, love, and support.*

Acknowledgments

First and foremost, I would like to express my deepest gratitude to my late supervisor, Prof. Giandomenico Spezzano. His guidance and mentorship have been instrumental throughout this journey, and I will always be thankful for his support. My heartfelt thanks go to Dr. Antonio Guerrieri, whom I consider an angel during this PhD journey. He not only guided me in my research but also supported me in every aspect of life during my stay in Italy. His encouragement and assistance extended far beyond the academic realm, and for that, I will always be deeply grateful. I would also like to thank Dr. Edoardo Serra, my secondment institute supervisor, whose valuable insights and guidance made this work possible.

I am sincerely thankful to the Institute for High-Performance Computing and Networking of the National Research Council of Italy - (ICAR-CNR) and the Department of Informatics, Modeling, Electronics, and Systems Engineering - (DIMES) at the University of Calabria, Italy, for their financial support, which made this journey possible. I would also like to acknowledge the crucial assistance of the SCAI Lab, whose industry collaboration greatly benefited my research.

Finally, I am forever grateful to my family—my great parents, my lovely wife, son, siblings, and relatives—whose unwavering love, patience, and encouragement have been my pillars of strength throughout this journey. Thank you to my friends and colleagues for your motivation, encouragement, and camaraderie throughout this journey.

Abstract

In today’s world, energy efficiency in buildings has become a top priority due to the significant energy waste caused by the operation of inefficient electrical appliances or to the inefficient use of them. Conventional methods of reducing energy waste cause discomfort for occupants inside a building. One promising way to optimize energy consumption in buildings is to synchronize appliance operation with the dynamic behavior of occupants in these buildings. Internet of Things (IoT) technologies, which allow for widespread data collection and execution of Machine Learning (ML) algorithms, enabled the creation of so-called Cognitive Buildings (CBs). CBs can learn patterns from these behaviors residing in and adjust electrical appliance operations accordingly. Understanding these behaviors includes how the occupants use the space, including if they are present, how many they are, and which activity they are performing. By doing so, these CBs could reduce energy waste, enhance resource efficiency, and consequently reduce CO₂ gas emissions. Furthermore, they could improve the overall comfort of the living environment and help with sustainability initiatives.

On the one hand, this thesis contributes to predicting occupancy inside a building, which is crucial for energy management inside CBs. We first propose an approach using federated learning (FL) and long-short-term memory (LSTM) neural networks to predict binary occupancy across several rooms inside the building. Furthermore, the proposed approach shows how FL could aid in binary occupancy predictions for rooms without specific model training. Through the use of our proposed approach, we attain an overall accuracy of approximately 94.5% on the considered data. We extend binary occupancy prediction to multi-occupancy prediction, forecasting the number of occupants in a space using a traditional FL technique. We do a comparison analysis by examining several FL configurations derived from different combinations of high-level rounds (HLRs) and epochs. HLRs represent instances of model aggregation at a high level, typically in the cloud, whereas epochs pertain to training rounds involving clients at the network’s edge. We employ a multi-layer hierarchical FL architecture, which can reduce the total communication energy with respect to the traditional FL by adding additional aggregation layers. The proposed approach also presents a design template for developing real distributed systems in CBs. We validate our

approach by implementing a preliminary prototype at ICAR-CNR, Rende, Italy. Additionally, a performance study demonstrates an accuracy of 84.5% in predicting a multi-occupancy environment.

On the other hand, this thesis also considers the recognition of occupants' complex activities inside these CBs, which are crucial for the occupant's comfort as different activities demand, for example, different temperatures and light intensities. We develop a novel hybrid deep learning model that combines LSTM neural networks with one-dimensional convolutional neural networks (1D-CNN) and recognizes these complex occupants' activities. LSTM explores the comprehensive temporal dimension, thoroughly examining time-related data, whereas 1D-CNN focuses on the spatial dimension, carefully handling deep features to reduce spatial characteristics. We deploy a real-time case study by collecting data from 22 healthy participants involved in 10 diverse activities through the use of Ultra Wideband (UWB) radar technology at the headquarters of ICAR-CNR in the IoT Laboratory, Italy. Moreover, we conduct a comprehensive benchmark of our approach against various statistical techniques and other deep learning models recently introduced in the literature. Results show that our proposed approach outperforms conventional methods and achieves an impressive accuracy of 98.42% for complex activity recognition.

Contents

Contents	vii
List of Figures	xi
List of Labels	xiii
List of Abbreviations	xv
1 Introduction	1
1.1 Contributions to the research	5
1.2 Organization of the thesis	6
2 Background	9
2.1 Occupants' Behaviors in Buildings	9
2.1.1 Occupancy detection, estimation and prediction in build- ings	9
2.1.2 Occupants Activity Recognition in Buildings	10
2.2 Data Collection Technologies	11
2.2.1 Sensor used for Occupancy Prediction in Buildings	11

2.2.2	UWB Radar Technology for Activity Recognition in Buildings	12
2.3	Machine Learning	15
2.3.1	Artificial Neural Network (ANN)	15
2.3.2	Convolutional Neural Network (CNN)	16
2.3.3	Recurrent Neural Network (RNN)	16
2.3.4	Long Short-Term Memory (LSTM)	16
2.4	Edge Computing	17
2.5	Federated Learning	18
3	State of the Art on Occupancy Prediction and Activity Recognition in Buildings	19
3.1	Occupancy Prediction in Buildings	19
3.1.1	Binary occupancy detection and prediction in buildings	19
3.1.2	Multi occupancy estimation and prediction in buildings	23
3.2	Activity Recognition in Buildings	26
4	Novel Approaches for Occupancy Prediction in Cognitive Buildings	31
4.1	Binary occupancy prediction in buildings	31
4.1.1	Reference architecture for binary occupancy prediction in buildings	31
4.1.2	Case study for binary occupancy prediction in buildings	34
4.2	Multi occupancy prediction in buildings with single-layer FL	36
4.2.1	Proposed architecture for multi-occupancy prediction with single layer FL	37
4.2.2	Experiments for multi-occupancy prediction with single layer FL	38
4.3	Multi occupancy prediction in buildings with multi-layer FL .	44

4.3.1	The Reference architecture for multi occupancy prediction with multi layer FL	44
4.3.2	The Design Template	47
4.3.3	Case Study Implementation	50
4.3.4	Experimental Evaluation	56
5	A Novel Approach for Activity Recognition in Cognitive Buildings	65
5.1	UWB Radar-based Activity Recognition in Buildings	65
5.1.1	The Proposed Approach	66
5.1.2	Case study for activity recognition in buildings	70
5.1.3	Discussion and results for activity recognition	76
6	Conclusion, Publications, and Future Work	87
6.1	Conclusion	87
6.2	Publication Related to this Thesis	88
6.2.1	Journal Publications	88
6.2.2	Book Chapter	90
6.2.3	Conferences	90
6.2.4	Invited talk at International Conference	91
6.3	Future Work	92
	Bibliography	93

List of Figures

1.1	Visual Representation of Occupant Behaviour Inside CBs. . .	7
2.1	Ultra-wideband SLMX4 sensor.	13
2.2	Ultra-wideband scatter matrix.	14
4.1	The reference architecture of the proposed approach for binary occupancy prediction.	32
4.2	The reference flowchart of the proposed approach for binary occupancy prediction.	34
4.3	Sensor data with actual and predicted binary occupancy. . . .	35
4.4	The reference architecture considered in the proposed approach for multi-occupancy prediction with single layer FL. . .	38
4.5	The devices involved in the implemented case study for multi-occupancy prediction.	40
4.6	Evaluation metrics curves computed against the test set on increasing number of epochs.	43
4.7	Actual vs Predicted multi-occupancy for the considered case study.	44

4.8 The reference architecture proposed for multi-layer hierarchy Federated Learning.	45
4.9 The proposed Design Template.	49
4.10 The main components implemented for the case study.	51
4.11 The touchscreen connected to the EDGE ₁ Raspberry Pi.	54
4.12 The realized Node-Red application.	55
4.13 Correlation matrix of the sensors deployed in the IoT Labo- ratory.	58
4.14 The Testset used in the implemented scenario.	59
4.15 Actual vs Predicted occupancy for the considered case study.	61
5.1 HDL4AR model architecture.	70
5.2 The IoT laboratory in ICAR-CNR, Italy. In the foreground, the used UWB radar is shown in its working position.	71
5.3 Graphical visualization of the Accuracy, Precision, Recall, and F1 Score for the compared approaches.	78
5.4 Confusion Matrix of UMAP-LSTM based Classification.	81
5.5 Confusion Matrix of EDA-LSTM based Classification.	81
5.6 Confusion Matrix of AGRU-1D CNN based Classification.	82
5.7 Confusion Matrix of ALSTM-1D CNN based Classification	82
5.8 Confusion Matrix of Transformer-based Classification.	83
5.9 Confusion Matrix of our approach.	83
5.10 Accuracy graph of the training and validation sets.	84
5.11 Loss graph of the training and validation sets.	84
5.12 3D scattering visualisation of each activity for our proposed approach.	84
5.13 The IoT laboratory in ICAR-CNR with the new configuration.	86

List of Tables

4.1	Confusion Matrix of the test set.	36
4.2	Evaluation metrics computed for different configurations and comparison with the fully centralized solution. All the metrics are calculated by using a Macro-Averaging scheme. Bold values highlight the best configuration for a given measure.	42
4.3	List of the sensors exploited within the IoT layer for the presented case study.	53
4.4	Performance of the proposed model	62
4.5	Evaluation matrix table for the considered case study	63
4.6	Evaluation metrics computed for the comparison analysis of different methods. Bold values highlight the best method for a given measure.	63
5.1	Engagement of all participants in the data acquisition process.	73
5.2	Evaluation metrics computed for different approaches and comparison with our approach.	78
5.3	Accuracy's score of each activity for all the compared classifiers.	80

5.4 Computational complexities are computed for different approaches and comparisons with our approach. 85

List of Abbreviations

1D-CNN	One-Dimensional convolutional Neural Networks
AI	Artificial Intelligence
ANN	Artificial Neural Network
CB	Cognitive Building
CL	Centralized Learning
CNN	Convolutional Neural Network
CO	Carbon Monoxide
CO ₂	Carbon Dioxide
DL	Deep Learning
DT	Decision Trees
EC	Edge Computing
EU	European Union
FL	Federated Learning
HAR	Human Activity Recognition
HDL4AR	Hybrid Deep Learning model for Activity Recognition
HLRs	High-Level Rounds
HMM	Hidden Markov Model
HVAC	Heating, Ventilation and Air conditioning
ICT	Information and Communication Technologies
IoT	Internet of Things
LDA	Linear Discriminant Analysis
LSTM	Long Short-Term Memory
ML	Machine Learning

LIST OF ABBREVIATIONS

PCA	Principal Component Analysis
PIR	Passive Infrared
RF	Random Forest
RMSE	Root Mean Square Error
RNN	Recurrent Neural Network
SVM	Support Vector Machine
TS	Temporal-Sequential
UWB	Ultra Wideband

Chapter 1

Introduction

Cognitive Buildings (CBs) represent a transformative evolution in built environments, integrating sensors, actuators, and advanced technologies such as IoT, artificial intelligence (AI), and machine learning to create spaces that are not only responsive but also intelligent. These buildings go beyond traditional automation by employing AI-driven algorithms to analyze vast amounts of data from sensors and user interactions. This allows CBs to learn from historical and real-time data, reason about optimal actions, and adapt dynamically to changing conditions, enabling highly personalized and efficient management of resources and spaces [1]. Unlike smart buildings, which primarily rely on predefined rules and basic automation, CBs leverage the full power of AI and cognitive computing to achieve adaptability, reasoning, and continuous learning. A CB has to perform many complex tasks to improve, among others, the efficiency and maintenance of the building itself and the security and comfort of its inhabitants while assisting them in their daily life activities [2]. Since it has been estimated that, currently, buildings' energy consumption is, in the European Union, 40% of the total energy consumption¹, CBs have to take care of energy optimization issues [3]. Optimizing buildings' energy utilization can significantly contribute to energy conservation and sustainability [4].

The importance of energy optimization in buildings can be observed from various perspectives [5, 6]. CBs can reduce the usage of important resources and, in this case, reduce energy wastage, which in turn leads to lower energy bills and cost savings. CBs can help reduce greenhouse gas emissions and mitigate the effects of climate change. The building sector accounts for a significant amount of carbon emissions, and reducing energy consumption in buildings can help decrease the sector's impact on the environment. CBs can create a more sustainable and resilient built environment. These buildings,

¹In focus: Energy efficiency in buildings - European Commission - https://commission.europa.eu/news/focus-energy-efficiency-buildings-2020-02-17_en

with the smartness they must demonstrate, can help generate more comfortable and healthy indoor environments for occupants. This is especially important in global health crises, such as the COVID-19 pandemic, as CBs can help ensure better indoor air quality and minimize the risk of infectious disease transmission. CBs can help create green jobs and support the development of a low-carbon economy. By investing in these buildings, building owners and operators can create new job opportunities and contribute to the growth of the green economy.

Numerous approaches for optimizing energy usage in buildings are available, each particularly designed to address a specific part of the building's operation [7]. These approaches may include installing energy-efficient lighting, upgrading intelligent Heating, Ventilation, and Air conditioning (HVAC) systems, implementing high-performance insulation, employing dynamic control building equipment, and integrating renewable energy sources. Implementing these approaches can be beneficial, but they require substantial initial investments, resources, and time to upgrade them. Thanks to the Internet of Things (IoT) technology, a cheap means of gathering data within a space can be used to monitor the environmental parameters or occupancy activity patterns in order to determine if the occupant is present, how many occupants are present, and what activity they are performing in the buildings [8]. Utilizing IoT data to train Artificial Intelligence (AI) algorithms can predict occupancy patterns, enabling the synchronization of electrical appliances, including HVAC and lighting systems. This approach has significant potential in reducing energy consumption and enhancing occupant comfort [9].

Recent research has demonstrated that the utilization of occupancy data for HVAC management may result in considerable energy savings, with savings ranging from 10-40% [10]. Similarly, lighting systems have the potential to produce significant energy reductions of up to 75% [11]. These results show that binary occupancy detection has achieved significant energy savings. However, energy efficiency and occupant comfort can be further optimized by obtaining detailed information about the exact number of occupants in a space. Let's envision a scenario where we adjust space to a comfortable 20°C temperature to accommodate five people. The heat from their bodies will raise the ambient temperature when more people enter the area. Nevertheless, the heating system continues to function under the assumption that there are only five occupants, leading to a needless elevation in temperature beyond the recommended degree of comfort. Thus, the system may persist in providing heat, resulting in excessive temporary heat and discomfort for the occupants. This not only undermines comfort but also squanders electricity while the heating system functions needlessly. In this situation, implementing a multi-occupancy detection (or better prediction) system would enable the heating system to adaptively regulate the HVAC power according to the real-time number of people present, maxi-

mizing comfort and energy efficiency. Multi-occupancy prediction systems help set a desirable temperature and lighting intensity based on the number of occupants. However, recognize that individual activities have more potential to optimize the temperature and light intensity, as different activities have prioritized different environment settings. This can be further compounded depending on the inhabitants' priorities and comfort preferences. To address these challenges, a recent study in [12] proposed an approach for recognizing and classifying activities in smart buildings. These projects (predicting binary/multi-occupancy and recognizing occupants' activities) hold great potential for optimizing energy and comfort optimization in buildings. However, achieving these outcomes is challenging and can be possible through the effective use of advanced digital technologies, such as information and communication technologies (ICT) and AI algorithms.

CBs solutions depend on a variety of digital technologies of ICT, including IoT devices, wireless sensor networks, and edge/cloud computing. IoT is a discrete physical device network that communicates data using defined communication protocols. IoT-based systems use several sensors, actuators, radars, and smart objects in general to monitor numerous parameters of a building's interior. Then, the cloud or powerful computing entities receive these vast amounts of data, where analytical or AI approaches scour, process, and act to extract valuable insights on user context and building health, potentially informing CB management design. Previous studies have mainly employed traditional techniques, i.e., signal processing or stochastic approaches, to detect and predict occupant behaviors [13]. Though these techniques show promising performances, they either make it complex or difficult to implement in real time. Hence, such drawbacks identified in previous solutions motivated the investigation of the adoption of machine learning (ML), which has recently begun to be explored in the context of detecting and forecasting occupants' behaviors. ML techniques can provide higher accuracy with minimum human interaction in comparison with traditional approaches. The popularity of data-driven approaches, including ML and Deep Learning (DL) for detecting and predicting occupants' behaviors, is growing due to their ease of implementation and ability to achieve high accuracy. Both ML and DL algorithms typically process data collected by IoT sensors/devices and/or cameras and send it to the cloud. Though the data is collected by IoT devices, privacy concerns may still arise during the processing of the collected data in the cloud. It is important to take appropriate measures to ensure the security and privacy of the data.

All the mentioned technologies, including IoT, which provide a cost-effective and efficient means of collecting environmental data, and the use of AI techniques to analyze and uncover patterns in occupant behaviors [14] facilitate the prediction of various aspects of occupancy, including binary occupancy prediction (will be occupied or not), multi-occupancy scenarios (the number of occupants), and even complex occupant activity recognition

(what activities are performed in a building). By leveraging data such as occupants' behavior patterns, ML and DL models can uncover patterns and insights that contribute to creating smarter, adaptive, and more efficient cognitive building systems. Such knowledge can be leveraged to enhance the operational efficiency of electrical appliances, ensuring they align with the comfort and needs of occupants. For instance, appliances can be programmed to automatically switch on or off in anticipation of occupants' arrival or departure, minimizing unnecessary energy use. Additionally, this information enables real-time adjustments to key environmental factors such as temperature and light intensity, tailoring them to the specific activities occupants are engaged in. This approach not only improves occupant comfort but also contributes to significant energy savings by dynamically aligning resource usage with actual requirements.

Statistical data in [15] show that occupants spend an average of 87% of their time in enclosed buildings and around 6% in vehicles. These numbers highlight the urgent need to design indoor environments that should be resource-efficient and where residents stay comfortable and friendly to health. These buildings account for around 40% of worldwide energy use and nearly 36% of greenhouse gas emissions [16]. These significant portions of energy use and greenhouse emissions highlight the need to develop energy-efficient and sustainable construction methods to address the challenges of energy waste while considering the wellness and comfort of the occupants.

Different studies have recently focused on optimizing energy efficiency using different ML techniques [17, 18, 19]. These studies aim to enhance energy efficiency using advanced algorithms for detecting/predicting binary occupancy detection/prediction. Though they are useful in predicting occupants' behaviors, they are unable to explain the privacy of the data they have used for ML algorithms. It is essential to prioritize the privacy and security of resident data, especially when processing and storing it in the cloud. Additionally, human presence significantly impacts indoor environmental conditions, as our bodies generate heat and contribute to humidity levels. Accurate multi-occupancy prediction is essential for maintaining consistent indoor temperatures, ensuring comfort, and optimizing energy usage. By dynamically adjusting HVAC systems based on the number of occupants, CBs can respond effectively to fluctuations in heat and humidity, creating a more sustainable and comfortable indoor environment. Finally, Human Activity Recognition (HAR) plays a vital role in ensuring personalized comfort within these CBs, as different activities require varying levels of temperature and light intensity. For example, activities like reading or working may demand brighter lighting and moderate temperatures, while activities like relaxing or sleeping benefit from dimmer lighting and cool environments. By accurately identifying occupant activities, CBs can dynamically adjust environmental parameters, providing tailored comfort while optimizing energy consumption. This capability enhances user satisfaction and promotes

energy-efficient building operations. This highlights the need for novel and more accurate approaches to understanding and predicting human behaviors in buildings, enabling the creation of smarter, more adaptive, and energy-efficient environments.

The contributions of this research are elaborated in detail in the following subsection.

1.1 Contributions to the research

The primary contribution of this research is to investigate the impact of occupant behaviour on indoor environments within CBs. Specifically, our study focuses on several aspects of occupants' behaviors, including whether the space is occupied by any occupant, their exact number, and the activities they are performing. Figure 1.1 illustrates a visual representation of occupants' behaviors in a comprehensive 3D framework, highlighting these key dimensions. More specifically, the figure illustrates the progression from an unoccupied building environment to one occupied by a single individual or multiple people and further to an environment where occupants are engaged in specific activities.

In this research, we have developed a novel approach for binary occupancy prediction, emphasizing data collected by IoT devices and using LSTM for model training at the Edge [20, 21]. Recent studies employed centralized learning techniques, in which they train models on central servers, leading to privacy concerns. To resolve this, we have implemented a novel decentralized method for occupancy prediction based on FL to train the model on the edge and send just model parameters to higher computing layers (e.g., servers, cloud). In this case, instead of sharing our original data with the cloud, we only share the model's parameters (in our case, model weights). Upon the reception of the model parameters from the edge devices, the cloud aggregates them and sends the global updated model to the edge devices. By applying our approach, we reached an impressive overall accuracy of about 94.5% on the considered data and also solve the issue of privacy. We collected data using IoT devices installed in our laboratory, leveraging their ability to capture detailed time-series information about occupants behaviours. Given the temporal nature of the data, we employed an LSTM model for training, as it is well-suited for handling sequential data. Additionally, we experimented with various other models, but the LSTM consistently delivered the best performance. Binary occupancy prediction is crucial for the dynamic control of electrical appliances such as HVAC and lighting. However, relying only on the binary occupancy prediction impedes the efficient management of increasingly complex building systems.

We extended binary occupancy prediction to multi-occupancy prediction by introducing a novel approach based on the LSTM model trained on tradi-

tional and multi-layer hierarchical FL [22, 23]. We have also developed and validated our model through a real-time case study by collecting data for multi-occupancy prediction at the ICAR-CNR headquarters in Rende, Italy, to provide practical insights into occupancy patterns in multi-occupant settings. Following the data collection from IoT devices in real-time, ML, edge computing, and FL techniques have been used to forecast multi-occupancy in buildings. We performed a comprehensive comparative investigation of several FL configurations through changing combinations of High-Level Rounds (HLRs) and epochs. HLRs represent instances where locally trained models are aggregated at a higher level, often in the cloud, while epochs refer to training iterations performed on edge devices (clients). These variables are very important for the training of the LSTM model because they identify how many times a model is merged (calculated) at low, and high levels. These variables also define the number of communications that the algorithm will do at the Edge and towards the Cloud. Furthermore, we implemented a multi-layer hierarchical FL methodology that aims to reduce communication energy between the cloud and the edge nodes. The main aim of adding another layer to the traditional FL is to reduce the repeated communications between the edge nodes and the cloud. Our approach also introduces a versatile design template for developing real distributed systems for occupancy prediction. We evaluated the proposed approach against conventional federated and centralized learning models, demonstrating enhanced performance. In this case, the proposed approach achieved a global accuracy of 84.5% in forecasting ten minutes-ahead room multi-occupancy while preserving data privacy.

We have also developed an innovative hybrid deep-learning model for the occupants' activity recognition for CBs [24]. More specifically, our proposed model combines Long Short-Term Memory (LSTM) networks with one-dimensional convolutional neural networks (1D-CNN). The LSTM effectively captures long-term temporal dependencies in occupant behavior, whereas the 1D-CNN adeptly handles spatial aspects. We conducted a real-time case study at the ICAR-CNR IoT Laboratory, where 22 healthy individuals participated in 10 complex activities to validate this model. We evaluated the hybrid model against conventional statistical methods and contemporary deep learning models, showcasing its efficacy. Our results show that our proposed approach outperformed conventional methods and achieved an impressive accuracy of 98.42%.

1.2 Organization of the thesis

This thesis is organized as follows: Chapter 2 provides the in-depth foundational knowledge required to understand the technical background of this research. More specifically, we explore occupant behavior in indoor environ-



Figure 1.1: Visual Representation of Occupant Behaviour Inside CBs.

ments in more detail, focusing on occupancy prediction and activity recognition. We have also focused on the technologies essential to collect the data from the indoor environment and which machine-learning techniques can be used to make decisions for occupancy prediction and activity recognition. Finally, we have highlighted the importance of recently developed techniques such as EC and FL and how they help to use them. Chapter 3 explores the recently developed relevant literature, providing an overview of recent research, more particularly in binary occupancy prediction, multi-occupation forecasting, and activity recognition in CB environments. It highlights the current state of the art in occupants' behaviors and also outlines the gaps our research aims to address. Chapter 4 will explain in detail the proposed approaches we developed for occupancy prediction. Additionally, we have subdivided the occupancy prediction into binary occupancy prediction, multi-occupancy prediction with single-layer FL, and multi-occupancy prediction with multi-layer hierarchical FL. This chapter also examines the case studies we have developed in real-time and the simulation results we have achieved. Chapter 5 will explain in detail the proposed approach we developed for activity recognition in CBs. It also examines the case studies developed at the headquarters of our IoT laboratory at the ICAR-CNR, Rende, Italy, and the simulation results we have achieved for complex activity recognition. Additionally, we have benchmarked our proposed approaches with the recently developed approach and shown their effectiveness and superiority through a detailed evaluation matrix comparison. Finally, Chapter 6 will present the thesis as a whole summary. Additionally, we have also listed the list of publications we have published during the Ph.D. period and explained the possible future direction that has the potential to improve this research.

Background

This chapter provides a comprehensive explanation of the foundational knowledge necessary to understand this thesis. It begins with an introduction to occupant behavior, a core concept in this research. Next, it reviews the key technologies commonly used in the literature for data collection in this field. The chapter then explores the role of AI algorithms in analyzing the collected data, extracting meaningful insights, and improving prediction accuracy. Finally, it highlights recent advancements, such as EC and FL, which have significantly enhanced data processing at the edge while addressing privacy concerns in occupant behavior modeling.

2.1 Occupants' Behaviors in Buildings

Occupants' behavior in buildings refers to the residents' habits, preferences, social interaction, and their relationship with buildings [25]. These behaviors encompass knowing about the occupancy status of space if the status is occupied, the number of individuals, and which activity they are performing. Understanding and evaluating these behaviours are critical for efficient building administration, optimizing energy use, increasing occupant satisfaction, and enhancing overall building efficiency [26]. The following are significant aspects of residents' behavior that are the scope of this thesis:

2.1.1 Occupancy detection, estimation and prediction in buildings

Occupancy detection is the process of determining whether a space, room, or building is occupied. The main objective of occupancy detection is to activate or deactivate electrical appliance operations, such as switching on/off lighting, HVAC systems, or other building services, depending on the occupancy state, to optimize occupant comfort and energy consumption. Occupancy estimation measures the number of people present in a

particular area or building at a given time. The objective of an occupancy estimate is to provide more intricate data on occupancy levels, enabling more accurate regulation of building systems and the allocation of resources. Occupancy prediction is the process of predicting the future status or number of occupants in a building or place by analyzing past data, patterns, or other pertinent criteria. Predicting occupancy proved useful for optimizing building management methods, such as scheduling HVAC operations using scheduling algorithms [27], controlling lighting, or planning space utilization to match projected occupancy levels.

Understanding occupancy status and determining the exact number of occupants can be useful for automating tasks such as activating or deactivating electrical appliances and adjusting temperature and lighting to optimal levels. However, this approach is limited in scope, as it does not account for recognizing or responding to the specific activities being performed by the occupants. It is important to recognize these activities since occupants' needs for temperature and lighting are different for each activity. This can be further compounded depending on the inhabitants' priorities and comfort preferences. To address these challenges, a recent study in [12] proposed an approach for recognizing and classifying activities in smart buildings.

2.1.2 Occupants Activity Recognition in Buildings

Occupant activity recognition refers to the process of identifying and comprehending the activities that inhabitants are performing in a space. It is crucial for energy management systems in buildings since occupants need, for example, different temperatures and light intensities for different activities [28]. Building systems can optimize energy efficiency and occupant comfort by correctly identifying occupant activities.

Research has demonstrated the utilization of various technologies for monitoring human activities in indoor environments to enhance building comfort, security, and energy efficiency [29]. These technologies include sensor fusion approaches that combine data from many sources, including PIR, temperature, light, and CO₂ sensors [30, 31]. Another technology often used to recognize human activities in the literature relies on closed-circuit television system cameras to monitor an environment [32]. Some studies have used cellphone accelerometers and gyroscopes to recognize and monitor the activities of humans carrying such devices inside a building [33]. Though they are effective, these technologies carry some limitations. Ambient sensors may deliver unreliable outcomes, and while cameras can produce good outcomes, people frequently decline their use due to privacy concerns. Conversely, wearable devices and mobile phones necessitate frequent recharging and maintenance, and not everyone feels comfortable carrying them constantly. Furthermore, misplacing or forgetting these devices can lead to gaps in data collection and potentially produce inaccurate results. As we

understand the basic concept of occupants' behaviors, we will now focus on which data collection technologies can be used to monitor the surrounding environment of CBs.

2.2 Data Collection Technologies

This section will present the technologies used in this thesis to gather data involving occupancy prediction and activity recognition. Various sensors and radars were used to collect data for occupancy prediction and activity recognition. Such sensors and radar are introduced in the following.

2.2.1 Sensor used for Occupancy Prediction in Buildings

Recently, current literature has focused on the combination of a variety of sensors to gather data for occupancy prediction; the most popular ones used are CO₂, PIR, light, temperature, CO, and humidity sensors [34].

The subsequent subsections will provide a detailed explanation of these sensors.

CO₂ Sensor

The presence of people inside a building has a significant impact on the concentration of CO₂, and a CO₂ sensor can precisely measure this. As people exhale CO₂, the concentration of carbon dioxide in a room increases with the number of individuals present. By analyzing the CO₂ levels in the air, a CO₂ sensor can be employed to estimate occupants and their numbers in a building/room.

PIR sensor

A PIR sensor can identify human body heat within its field of view by detecting changes in infrared radiation emitted by objects within its range. CBs can use PIR sensors to estimate room occupancy [35]. Placing PIR sensors in different locations within a room or a building allows for detecting changes in the motion of people present or when people enter or leave an area. A central control system receives sensor data to infer occupancy patterns. To increase accuracy and reduce false positives, this system can be combined with other types of sensors, such as CO₂ or light sensors.

Light Sensor

A light sensor, often referred to as a lux sensor, measures the intensity of light within its environment. These sensors are commonly used in occupancy detection systems to estimate the presence of occupants by monitoring

fluctuations in natural or artificial light levels. For example, when an individual enters or exits a room, changes in light intensity caused by movement or shadowing can be detected. Light sensors, in combination with other sensors like PIR or CO₂ sensors, can enhance the accuracy of occupancy predictions, reducing false detections and improving energy management in smart buildings.

2.2.2 UWB Radar Technology for Activity Recognition in Buildings

UWB radar is an advanced technology that can offer cutting-edge services (such as precise positioning, real-time location measurement, or movement detection) thanks to its high-time resolution and short pulses. UWB can coexist with other wireless technologies, operates at low power, and is secure due to its low power density [36]. UWB radar consists of two antennas: a transmitter and a receiver. The transmitter antenna transmits the pulse, and after it is transmitted, it turns off, and the receiver antenna listens for the reflection. It operates at frequencies between 3.1 and 10.6 GHz [36]. Antenna beamwidth (Θ) determines the angular coverage of the radar beam and can be calculated for a given antenna as:

$$\Theta = \frac{2\lambda}{D} \quad (2.1)$$

Where D is the effective aperture or diameter of the antenna and λ is the wavelength of the transmitted signal. The time of flight is defined as the time it takes for a signal emitted from an antenna to travel to an object and reflect back. Using the formula below, we can calculate the time of flight (t):

$$t = \frac{2d}{c} \quad (2.2)$$

where d represents the distance traveled, and c is the speed of light, which is approximately 3×10^8 m/s. The above formula allows us to calculate the distance between an object and the radar. Using the following mathematical notion, we can determine the distance:

$$d = \frac{c \cdot \Delta t}{2} \quad (2.3)$$

In our experiments, we have used UWB radar, namely the SLMX4¹ UWB radar, which is developed by Novelda and shown in Fig. 2.1. It operates at a central frequency of 7.29 GHz.

Our radar is connected to a Personal Computer (PC) through a USB port to facilitate data transmission. To get data from the radar, we used

¹SLMX4 - UWB Radar Module <https://github.com/SensorLogicInc/modules?tab=readme-ov-file>. last seen: 25/07/2024



Figure 2.1: Ultra-wideband SLMX4 sensor.

MATLAB with the radar’s software interface. This radar generates data in a matrix format, which consists of rows and columns. The rows represent a specific time point or event at which the radar data is collected. Each row corresponds to the radar signal at a particular instant, recording the variations in the signal over consecutive readings. The columns in the radar data indicate distinct characteristics or data points recorded at each specific time interval. These features may include several attributes of the radar signal, such as distance bins obtained from the radar reflections. A UWB radar frame contains 184 bins, each corresponding to a distance of 5.14 cm [37]. The columns collect data from these bins to quantify the target distance at each time interval. Here is how to calculate the maximum distance the UWB radar can cover within one frame:

$$\text{Distance (d)} = \text{Number of Bins} \times \text{Distance per Bin} \quad (2.4)$$

That, in our case, means:

$$d = 184 \text{ bins} \times 5.14 \text{ cm/bin} = 946.96 \text{ cm} (\approx 9.47m) \quad (2.5)$$

For example, the 50th bin encapsulates information about objects or human movements occurring within a range of approximately 2.55 to 2.60 meters.

Figure 2.2 depicts the data obtained from the UWB radar in graphical form. The term “fast time” refers to the short duration of time that a radar pulse generates, which measures the rapid variation in the radar signal as it bounces off the objects. Data analysis visualizes this time horizontally,

$x_{1,1}$	$x_{1,2}$	$x_{1,3}$...	$x_{1,i}$...	$x_{1,S}$
$x_{2,1}$	$x_{2,2}$	$x_{2,3}$...	$x_{2,i}$...	$x_{2,S}$
$x_{3,1}$	$x_{3,2}$	$x_{3,3}$...	$x_{3,i}$...	$x_{3,S}$
$x_{4,1}$	$x_{4,2}$	$x_{4,3}$...	$x_{4,i}$...	$x_{4,S}$
$x_{5,1}$	$x_{5,2}$	$x_{5,3}$...	$x_{5,i}$...	$x_{5,S}$
$x_{6,1}$	$x_{6,2}$	$x_{6,3}$...	$x_{6,i}$...	$x_{6,S}$
$x_{7,1}$	$x_{7,2}$	$x_{7,3}$...	$x_{7,i}$...	$x_{7,S}$
$x_{8,1}$	$x_{8,2}$	$x_{8,3}$...	$x_{8,i}$...	$x_{8,S}$
$x_{9,1}$	$x_{9,2}$	$x_{9,3}$...	$x_{9,i}$...	$x_{9,S}$
$x_{10,1}$	$x_{10,2}$	$x_{10,3}$...	$x_{10,i}$...	$x_{10,S}$
\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots
$x_{j,1}$	$x_{j,2}$	$x_{j,3}$...	$x_{j,i}$...	$x_{j,S}$
\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots
$x_{N,1}$	$x_{N,2}$	$x_{N,3}$...	$x_{N,i}$...	$x_{N,S}$

Figure 2.2: Ultra-wideband scatter matrix.

providing instant information about object distances and their characteristics, with each data point corresponding to a specific distance bin [38]. On the other hand, the term "slow time" refers to the longer duration over multiple radar pulses. It records the gradual changes in the radar signal over a period of time, enabling an analysis of patterns of movement. The data processing displays this time vertically, aiding in the identification of slower variations. The combination of these dimensions enables the radar to construct a comprehensive representation of both the spatial configuration and temporal fluctuations in the perceived surroundings.

Scattering Matrix

As already introduced, a UWB radar emits signals when used, and such signals return to the radar itself, which represents the received signals as scattering matrices, characterized by rows and columns. This scatter matrix can be written as X_{raw} in mathematical notation.

$$x_{j,i} = X_{raw} \in \mathbb{R}^{N \times S}, \text{ for } j = 1, 2, \dots, N, i = 1, 2, \dots, S$$

Since our raw data is presented in matrix format, each column represents a feature, and each row represents a timestamp.

- x_{ji} indicates the amplitude value related to bin j in frame i .

- N represents the total population of frames; while S denotes the total number of features in a dataset.

The subsequent section will give an explanation of which ML algorithms can be used to learn and extract useful information from the data collected by these data collection technologies.

2.3 Machine Learning

With the advancement of technology, the integration of IoT devices is becoming increasingly prevalent in our daily routines, allowing us to monitor and track our surroundings. This has led to the accumulation of vast amounts of data that can be exploited in several application domains. One very relevant usage of IoT data is for the realization of smart building management systems. Building managers can gather lots of real-time data on occupant behaviors through IoT sensors. However, manually examining such large volumes of data is not only time-consuming but also prone to errors. As a result, ML is being increasingly utilized to analyze this data and provide accurate information for building management [39]. By employing ML algorithms, the massive amounts of data collected from sensors, radars, and other devices can be analyzed and used to gain a deeper understanding of building health and user context [40]. This includes identifying patterns in building usage, predicting occupancy levels, and analyzing energy consumption patterns.

One of the most significant applications of ML in CBs management is the understanding of occupant behaviors, including their status, the number of occupants, and their activities [41]. ML models can accurately predict the status of individuals, whether the space is occupied or not, and the number of occupants in a building or a specific building area and their activities by analyzing data collected from various sensors (e.g., passive infrared, CO₂, temperature, humidity, sound, camera) or radar. This information can be used to optimize energy consumption and improve occupant comfort by adjusting temperature, lighting, and other building systems. The subsequent subsections will provide a detailed explanation of popular ML or DL techniques that can be used for the above-mentioned tasks.

2.3.1 Artificial Neural Network (ANN)

ANN is a field of ML that deals with algorithms and is motivated by the structure and function of the brain. An ANN typically consists of one input layer, numerous hidden layers, and one output layer. Each layer is made up of several neurons. ANNs present some hyper-parameters, including the number of hidden layers and the number of neurons in each hidden layer. Each neuron in one layer communicates with neurons in neighboring layers

according to weights and biases that can be set through a training process [42, 43]. In the following, some hints will be given about some common types of ANN.

2.3.2 Convolutional Neural Network (CNN)

CNN is a type of ANN that outperforms as compared to the other conventional neural networks when dealing with pictures, speech, or audio signal inputs [44]. CNN is formed by three types of layers (or building blocks): convolution, pooling, and fully connected layers. The first two (convolution and pooling) extract features, while the third is fully linked and transforms the extracted features into final outputs. By augmenting the layers of a CNN, its complexity grows along with its capacity to recognize objects. The first layers emphasize simple elements like colors and edges. As the image data progresses through the CNN layers, the neural network detects various aspects or shapes of the image until it eventually recognizes the desired objects.

2.3.3 Recurrent Neural Network (RNN)

RNN can build a chain of neurons that can link to another neuron. Neurons in this kind of network may form loops and/or connect to neurons from a previous level [45]. Most recurrent networks consist of links that point backward or to the same level. Recurrence inherently includes network memory; therefore, this property makes this branch of neural networks particularly interesting. An RNN faces the problem of vanishing gradients. The parameter updates become negligible when gradients are too small. As a result, learning large sequences of data is difficult. The LSTM neural networks (described below) were developed to address such an issue.

2.3.4 Long Short-Term Memory (LSTM)

LSTM is indeed the machine learning algorithm belonging to the ANN (or, more precisely, RNN) family capable of learning and remembering long-term dependencies by default [46]. It is beneficial for addressing sequence classification issues and forecasting time series. The basic unit of the LSTM system is the cell, which provides long-term memory. The internal state of an LSTM cell can be broadly represented with two vectors: one defines the short-term state, while another describes the long-term state. An LSTM cell consists of three gates, each adding or removing information from or to the cell state and computing the output value. These gates are the input gate, the forget gate, and the output gate. The input gate decides which information from the current input state should be kept and which information should be discarded to calculate the current state. The forget gate determines which information from the previous state should be kept and which

information should be discarded, and the last gate determines both the output value and the current state. After collecting the data from the sounding environment of occupants' behaviors, some recently developed techniques, including EC and FL, helped us to process our data close to the IoT devices and care for the privacy of our data, which can be explained in the following section.

2.4 Edge Computing

Edge Computing (EC) uses distributed computing to bring computing power closer to end devices [47, 48, 49]. This technology uses a distributed computing paradigm that can be exploited for various applications like the ones regarding data processing, caching/storage, IoT management, security, and privacy. These applications can execute tasks on edge networks efficiently and effectively to achieve high bandwidth and low latency. This paradigm simplifies the management of end-to-end connections and resources. With an edge-based architecture, data can be produced and consumed closer to the edge itself, reducing the need to transfer data to the center of the network (e.g., Cloud) to augment security, reliability, and privacy. Combining Artificial Intelligence algorithms with EC allows it to reduce latency and network load in the network core [50].

For example, the EC paradigm can be utilized to estimate room occupancy by strategically deploying sensors and cameras to collect data on human movement and activity levels within a given space [51]. The collected data can then be processed and analyzed locally on edge devices, reducing the need for centralized data centers and decreasing the amount of time and resources required to estimate occupancy. For example, the study in [52] introduces a cost-effective edge-based platform for improving energy efficiency in buildings, capable of sensing, processing, and analyzing energy data within residential buildings. The paper describes how the EC approach has been employed to reduce delay and cost. Similarly, authors in [53] employed Edge Computing as an intermediary layer between the IoT layer and the cloud aggregation layer: the models used were trained on the edge devices themselves, ensuring an enhancement of the data security. Following the local training of models on the edge devices, the cloud aggregator analyzes all the trained models and selects the best one. The integration of edge computing with AI has recently gained momentum in industry and academia. Such integration gave rise to the paradigm known as Edge Intelligence, which leverages the incorporation of machine learning techniques in edge devices. This paradigm is capable of unleashing the real potential of the big data produced by sensor-instrumented environments. One of the techniques that takes advantage of the processing power available at the edge is federated learning.

2.5 Federated Learning

ML can be effective in understanding the pattern inside data, but it requires a substantial amount of training data. Data can be collected, for example, from various IoT devices in a building and transmitted to the Cloud to train ML models. Collecting data from different IoT devices and training ML models on a cloud-based server is known as the centralized learning approach. Although a centralized learning approach can provide high accuracy for ML algorithms, it raises privacy concerns when transferring data from IoT devices to the central server. FL is a viable alternative that permits ML models to be trained on decentralized data without transmitting it to a central server [54, 55]. The model's training is done locally on every device, and the updated models are then sent to a central server for aggregation, ensuring data privacy and security.

In Centralized Learning (CL), client data is transmitted to a central server for training, which may raise privacy concerns. In contrast, FL ensures data privacy by storing data locally on clients' devices and training models there. Edge devices are selected to train their models on local data, and their model weights are then sent to the server for aggregation into a global model. Clients use their private datasets to optimize the global model in each training round, and their updated models are sent back to the server. The server merges the updates following specific rules to create a new global model for the next round of training.

An FL algorithm typically follows the steps below.

- Step 1: The server delivers his global model (i.e., weights) to edge devices, instructing them to train their local model accordingly;
- Step 2: The edge devices receive the global model from the server and use their local data to train the models;
- Step 3: Edge devices send back an updated model (i.e., weights) based on the local training.
- Step 4: Upon the reception of the models from the edge devices, the server aggregates them and updates the global model towards the edge devices.

Such steps are usually repeated for several *rounds* or until the model converges.

Chapter 3

State of the Art on Occupancy Prediction and Activity Recognition in Buildings

This chapter provides a comprehensive review of recent literature on occupant behavior in smart buildings, with a particular focus on occupancy prediction and activity recognition. Understanding these behaviors is crucial for optimizing the control of electrical appliances, leading to improved energy efficiency, occupant comfort, and overall well-being within these environments. The following subsections will explore these topics in greater depth, focusing on recent advancements in occupancy prediction and activity recognition.

3.1 Occupancy Prediction in Buildings

The indoor environment is significantly influenced by occupancy prediction, which we have categorized into two subtopics: binary occupancy prediction and multi-occupancy prediction. The following subsections provide a detailed analysis of the recent developments in the literature in both areas.

3.1.1 Binary occupancy detection and prediction in buildings

Binary occupancy prediction is a basic concept for controlling the appliances the operation of electrical appliances. Vafeiadis et al. [56] demonstrated occupancy prediction (occupied / unoccupied) in a domestic environment. To forecast occupancy status, the authors employed Decision Trees (DT), SVM, back-propagation networks, and RF, as well as the AdaBoost method. The data for these ML models came from water and electricity

use. In terms of accuracy, it was discovered that RF and DT outperformed other approaches. In [57], the authors present a system for season-based occupancy prediction in residential buildings in which they also compare different ML models. The study in [58] also compared three occupancy prediction systems based on three popular ML algorithms, namely k-nearest neighbors, support vector machines, and artificial neural networks. In [59] a data-driven model is proposed for the prediction of occupancy using a long-short-term memory neural network. From experiments, such a solution results are effective for the prediction of future occupancy states. The paper in [60] introduces a weakly supervised occupancy prediction framework based on office sensor readings and occupancy estimations derived from an interactive learning approach. The results reveal that the proposed *Light-GBM*, a machine learning model, is a better fit for short-term predictions than other recursive neural networks known in the literature. The authors of [61] present an occupancy prediction model in smart buildings using environmental sensor data based on LSTM. They also introduce a transfer learning approach to enhance the occupancy prediction accuracy when the amounts of historical training data are limited.

Jin et al. [62] present a novel approach for accurately forecasting building occupancy. More specifically, the proposed approach combines temporal-sequential (TS) analysis and machine learning techniques to forecast occupancy. The researchers leverage hourly occupant data from 16 different buildings to demonstrate the effectiveness of their approach. The TS analysis takes into account the temporal and sequential nature of building occupancy, including seasonal patterns, by incorporating a one-week seasonal period. Additionally, an artificial neural network structure is employed (referred to as TS-week-ANN) to improve predictive performance. Through their experiments, the authors compare the proposed TS-week-ANN approach with other baseline methods. The results reveal that the TS-week-ANN method outperforms the baselines in terms of accuracy and predictive performance. Ryu et al. in [63] develop models driven by indoor environmental data, effectively detecting and predicting occupancy states. The researchers conducted experiments in the Building Integrated Control Test-bed at Dankook University, where they gathered accurate occupancy profiles, as well as data on indoor and outdoor CO₂ concentrations and electricity consumption from lighting systems and appliances. The findings indicate that the proposed models, employing decision tree and Hidden Markov Model (HMM) algorithms and driven by indoor environmental data, exhibit effectiveness in detecting current occupancy states and predicting future occupancy states, respectively. Peng et al. [64] explore energy efficiency by considering occupants' behavior, utilizing machine learning to optimize HVAC settings and achieve significant energy savings. The study focuses on stochastic behavior patterns of occupants in an office building and proposes a demand-driven control strategy that adjusts HVAC settings in real time. The learned oc-

cupancy information is then used to optimize temperature setpoints for the cooling system, resulting in significant energy savings. The proposed strategy was tested in various office spaces, demonstrating energy savings ranging from 7% to 52% compared to conventional cooling systems. By reducing the need for human intervention, this approach has the potential to enhance HVAC system efficiency and reduce energy consumption. Similarly, Wang et al. [65] compare machine learning algorithms highlighting the enhanced robustness of occupancy prediction with fused data. They compare three popular machine learning algorithms (k-nearest neighbors, support vector machine, and artificial neural network) and three data sources (environmental data, Wi-Fi data, and fused data) to optimize occupancy prediction models in various scenarios. Evaluation metrics such as mean average error, mean average percentage error, and root mean squared error are used to assess model accuracy. Through an on-site experiment, it was found that the artificial neural network model with fused data performs best, while the support vector machine model is suitable for Wi-Fi data. The study also reveals that the fused data set enhances robustness in occupancy prediction, even though it may not significantly improve accuracy compared to independent data sources.

In the study at [66], the authors have designed and tested various ML techniques, such as SVM and a mix of classification and regression algorithms, to predict classroom occupancy. They utilized a dense wireless network to collect data from a sizeable university campus and achieved satisfactory results. Similarly, the article in [67] focused on precise forecasting on three distinct topics: identifying indoor occupancy, predicting occupancy density, and determining the exact headcount, utilizing the gathered data and applying different ML algorithms, among which the SVM. The study examined how to identify the best feature set for indoor context by properly integrating static and dynamic context with Indoor Air Quality, resulting in satisfactory outcomes for both Occupancy detection and prediction. Akbar et al. [68] used pattern recognition to detect occupancy state without involving intrusive methods. To illustrate the importance of selecting the right kernel function, that study compared the performance of SVM with three different kernel functions (linear, polynomial, and radial basis functions). The accuracy of the results was between 55.37% and 79.12%, based on electricity data collected in a research center. Alternatively, Kleiminger et al. [69, 70] investigated SVM and electricity consumption data to estimate household occupancy. It was demonstrated in these works that the selection of characteristics used in learning processes is important. Authors validated their models using the public ECO dataset1 [71] and found an occupancy detection accuracy ranging from 68 to 94 percent. The research in [72] comprehensively evaluated five distinct ML techniques for occupancy detection, utilizing data from five sensor streams that were highly correlated with building occupancy. A model prototype was created and subjected

to training and testing to evaluate its performance. The results showed that the model performed well in predicting occupancy and showed satisfactory results. The authors introduce a novel, cost-effective, and eco-friendly method for detecting occupants in enclosed spaces using passive cognitive radio [73]. The proposed solution utilizes a reconfigurable software-defined radio system and adaptive spectrum sensing technology. Experimental results demonstrate that the CRhadora system can effectively and accurately detect human occupancy in indoor spaces. Though the ML algorithm has performed well for occupancy prediction, the DL algorithm has more potential for predicting occupancy inside CBs.

The paper in [74] presents a new deep-learning method called CNN-XGBoost for occupancy prediction based on indoor climate data. The study compares the new method with various supervised and unsupervised machine learning and artificial neural network algorithms. Indoor climate data collected from a residential building in Copenhagen over 13 days in December 2021, including CO₂, relative humidity, and temperature, are used in the study. The CNN-XGBoost method comprises a CNN and a scalable end-to-end tree-boosting classifier. The study evaluates the algorithms using mean absolute error, confusion matrix, and F1 score. The results show that CNN-XGBoost outperforms other algorithms in predicting occupancy levels in all rooms of the test building. The study in [75] introduces a method for predicting occupancy in smart homes, which relies on environmental factors like CO₂ levels, noise, and temperature, and employs a ML method and a forecasting strategy. The proposed algorithms aim to improve the energy management system by optimizing the use of the electric heating system. The study utilizes the LSTM neural network. Moreover, The authors demonstrate that FL can be utilized to improve occupancy predictions when training a specific model is not feasible. The study highlights how this approach can be applied to rooms where the occupancy patterns are unknown, and therefore, a custom model cannot be trained. Using the concept of occupancy detection without covering the whole room with sensors was proposed by Husnain and Choe in [76]. A decision module predicts Human presence patterns using LSTM for the sensor's off-range region. It reduces the installation costs for occupancy detection systems. On the same tangent, Pešić et al. [77] propose a technique for detecting occupancy utilizing a fusion of WiFi and Bluetooth data and a set of data analytics functions to examine occupancy data across logical and physical boundaries. An LSTM neural network was studied for occupancy forecasting, and the same data analytic features were used to present and predict occupancy statistics. For workdays, they achieved 75.45% similarity on real (EDR) signals. Chang et al. in [78] used six forecasting models to analyze the same dataset: gaussian process regression, regression by least squares, regression by backpropagation, regression by general regression, and regression by LSTM. The numerical results demonstrate that LSTM networks are superior to other models

in estimating hotel accuracy rates across three data repositories. Regarding root mean square error (RMSE), the model reached a value of 13.31%. Hitimana et al. in [79] use multivariate time series to predict occupancy patterns in regression forecasting. An empirical evaluation showed that the designed solution effectively collected, processed, and stored environmental data. LSTM was used to model the acquired data and then compared to various ML techniques to show good performance. In [80], the authors propose a solution for occupancy prediction in a large exhibition hall. The authors propose dividing the hall into sections and using a RNN with LSTM units to make accurate occupancy predictions. The effectiveness of the RNN model is evaluated by predicting short-term and long-term occupancy based on historical patterns of occupancy changes in different zones of the hall. Another paper in [79] introduces an IoT framework for capturing indoor environmental parameters to collect occupancy multivariate time-series data. The focus is on utilizing the LSTM network to detect the presence of individuals based on the collected data. An experiment is conducted in an office room to demonstrate the successful implementation of the system for acquiring, processing, and storing environmental information. The obtained data is then compared with other machine learning algorithms such as Support Vector Machine, Naïve Bayes Network, and Multilayer Perceptron Feed-Forward Network. The results indicate that LSTM outperforms the other algorithms in the proposed application based on parametric calibrations. In [20], the authors propose a unique method for predicting occupancy in multiple rooms of a smart building that combines FL and LSTM neural networks. The study highlights the importance of FL in predicting occupancy for rooms where training a specific model is not feasible. Through simulations on an existing dataset, the results demonstrate the promising performance of the proposed approach, achieving an accuracy of approximately 0.945.

3.1.2 Multi occupancy estimation and prediction in buildings

Though many studies have used binary occupancy prediction for energy and comfort management in CB. However, its application is limited to just activating or deactivating electrical appliances. Multi-occupancy estimation and prediction have more potential for energy and comfort optimisation inside these buildings. Yang et al. [81] utilize Convolutional Neural Network-based density estimation methods to extract occupancy information, improving accuracy in building energy simulation. Starting from this information, the study uses four AI algorithms, i.e., K-nearest neighbor, support vector machine, random forest, and artificial neural network for occupancy prediction, and conducts a case study in educational buildings to assess performance. The results indicate high accuracy and low error rates in occupancy prediction. The study successfully reduces the deviation between predicted and actual occupancy schedules, leading to improved ac-

curacy in building energy simulation. Arief et al. [82] propose a domain adaptation model for Carbon Dioxide - Human Occupancy Counter Double Plus to accurately predict the number of people in a room without adequate labeled data. The proposed semi-supervised domain adaptation method is compared with two baseline methods: Support Vector Regression and Seasonal Decomposition for Human Occupancy Counting. Domain adaptation is useful when historical data is unavailable, and the previous working model can be adapted to the new environment. The proposed method is evaluated using a real-world dataset and is compared with state-of-the-art techniques. Zou et al. [83] estimate the number of occupants. An evaluation of domain-invariant kernels was conducted using transfer kernel learning. In experiments, WiFree achieved an accuracy of 92.8 percent when counting occupancy. The study in [84] involved comparing multiple methods for predicting the number of indoor users, with and without using SB state variables. The experiment collected data on indoor temperature and CO₂, as well as devices' log data, in a SB. The study evaluated the effectiveness of five ML techniques, including KNN, GP, RF, BR, and MLP. The study in [85] aimed to compare two prominent time series prediction methods, LSTM and ARIMA models, for forecasting the number of occupants in smart building environments using Wi-Fi networks. The results showed that the combined LSTM approach outperformed individual models at different time scales, reducing the required number of neurons by 67.13% - 74.48%. Furthermore, LSTM forecasts exhibited higher accuracy compared to traditional ARIMA models, with an RMSE reduction of 80.9% - 93.4%. In a recent study [23], the authors proposed an approach for predicting multi-occupancy in CBs through the use of IoT technology. The proposed approach used a variety of sensors, including light, CO₂, and PIR sensors, to collect extensive data on occupancy. In addition, a study also considered a virtual sensor that utilizes a Raspberry Pi-based touch screen to enable precise data labelling for counting occupancy. The authors use a multi-layer federated learning technique to ensure decentralized model training, protecting the occupants' privacy. The study produced encouraging outcomes, with an accuracy of 84.5% in predicting the number of multiple occupants in CBs. Chen et al. [86] introduces a novel, non-intrusive environmental sensor and a cost-effective approach for estimating building occupancy. A conventional deep bi-directional long-short-term memory network was employed that can learn and memorize long-term relationships among the features and can learn autonomously from the data. The proposed approach accurately finds occupied spaces by combining convolutional networks with deep structures. This captures temporal dependencies in the data and considers both the past and future. According to the experimental evaluation, the proposed method is better than other methods at telling the difference between occupancy ranges (zero, low, medium, and high). Chidural et al. [87] examine three thermal imaging sensors that differ in resolution. The study focuses on char-

acterizing the sensors, evaluating the occupancy estimation algorithms, and comparing their performance. The study presents a pipeline of unified processing algorithms for occupancy estimation, along with algorithms for data pre-processing, feature extraction, and fine-tuning. The results show that occupancy estimation is feasible for smart building applications because of its high accuracy rate of approximately 99%. Longo et al. [88] explore estimating occupancy in buildings. This method leverages management frames transmitted by users' devices over Bluetooth, WiFi, and Bluetooth Low Energy. Choosing to depart from traditional methods, the suggested system offers a cost-effective and accurate solution implemented on minimally expensive hardware. By employing a supervised learning model, the system demonstrates the ability to adjust to different environments and provides estimated occupancy data to a web-based interface. Experimental findings obtained in uncontrolled outdoor and indoor environments substantiate the efficacy and dependability of the suggested solution. Through precise prediction of future occupancy levels, HVAC and illumination systems possess the capability to adapt dynamically to satisfy the requirements of occupants in real-time. Dino et al. [89] utilizes a camera to estimate room occupancy using a deep vision-based technique. This research has two objectives: first, to assess the effectiveness of deep vision-based occupancy counting, and second, to implement a deep vision-based occupant-centric control strategy in real-world settings [90]. The study evaluates the performance of the occupancy counting method in six offices and implements a deep vision-based energy recovery ventilator control strategy in a small office. The performance of this strategy is then compared with traditional control methods, considering indoor air quality and energy consumption. The study in [91] proposes a new population-based approach (PopAp), inspired by agent-based transportation models, to model occupants' forecasting. Comparing PopAp with traditional deterministic and stochastic methods, the study findings reveal notable differences in maximum occupant numbers and hourly energy demands, especially in education buildings. These results underscore the importance of detailed occupant modeling for accurate energy system planning, particularly at the hourly scale, to effectively predict peak demand and size technologies. Dutta et al. in [67] focused on precise forecasting on three distinct topics: identifying indoor occupancy, predicting occupancy density, and determining the exact headcount, utilizing the gathered data and applying different ML algorithms, among which the SVM. The study examined how to identify the best feature set for indoor context by properly integrating static and dynamic context with Indoor Air Quality, resulting in satisfactory outcomes for both Occupancy detection and prediction.

3.2 Activity Recognition in Buildings

Recognizing complex occupants' activities inside CB is a modern demand, as different activities need different levels of temperature and light intensities. Fauzi et al. [92] introduce the Group Activity Recognition technique for buildings. This system utilizes multi-image analysis from cameras to identify and track people based on their seating postures and facial features. The technique underwent testing in several circumstances, including meetings, seminars, and classroom settings. The proposed approach used a neural network algorithm that demonstrated a learning phase accuracy of 93.33% and a testing phase accuracy of 63%, with an error rate of 37%. Ranieri et al. [93] focus on the identification of everyday activities in residential settings by using data from movies, wearable devices, and environmental sensors. The Heriot-Watt University/University of Sao Paulo (HWU-USP) activities dataset serves as the presented dataset. The datasets are available online¹. It contains periodic and long-term activity patterns recorded using a humanoid robot's camera and wearable and smart home sensors. Furthermore, the study employed a deep learning architecture that identifies activities through various modes of input. The HWU-USP and University of Texas at Dallas Multimodal Human Activities Dataset (UTD-MHAD) datasets have tested the framework, revealing that the inclusion of ambient sensor data significantly enhances identification accuracy. Ganesh et al. [94] introduces a technique for capturing and examining gym exercises, including push-ups, squats, planks, forward lunges, and sit-ups, using an RGB camera. The films' features were categorized using support vector machines, decision trees, K-nearest neighbours, and random forest classifiers. The models were assessed using metrics such as accuracy, balanced accuracy, precision, recall, and F1 score. The Random Forest classifier had the greatest level of accuracy, reaching 98.98%. All of this research produced very high accuracies by using vision-based techniques and deploying cameras for video or image-processing-based activity recognition. However, they encounter substantial privacy apprehensions since people are more concerned about the confidentiality of their data in the contemporary digital era.

Prasad et al. [95] utilized cell-phone accelerometers to recognize different human activities and employed a convolutional neural network. The proposed approach divides the data into nodes and then separates it into three primary CNN layers before the result is predicted in the output layer. The research assesses the performance of the two-dimensional CNN model in the training and testing stages, resulting in an average accuracy of 89.67%. The study emphasizes the promise of constructing efficient models using smartphone-generated information while highlighting this field's current hurdles. Peppas et al. [96] utilized a deep neural network model for real-time

¹Data Sets available at - <https://doi.org/10.5061/dryad.v6wwpzgsj>

human activity identification utilizing tri-axial accelerometer data from mobile devices. It has two-layer convolutional neural networks and 40 statistical features that feed into a fully connected layer. The model combines feature extraction with convolutional layers. It achieves 94.18% accuracy on the WISDM dataset² and 79.12% on the Actitracker dataset, consuming 5-8 times less storage space and providing more than twice the throughput of existing state-of-the-art approaches. While cell-phone-based activity recognition provides privacy benefits, it has drawbacks. One key difficulty is sometimes forgetting to bring phone, or the old people don't often use their phone. Furthermore, cell phones' limited sensor coverage and background noise sensitivity might reduce activity identification algorithms' accuracy. These issues, together with concerns about battery depletion and computing power restrictions. Bouchard et al. [12] investigate the use of ultra-wideband (UWB) radar to recognize activities of daily living (ADLs) in smart homes, addressing the aforementioned problems. A dataset of 15 distinct activities was performed by 10 people in a 40-square-meter flat. With an F1-Score, the Random Forest algorithm achieved 80% accuracy. Using UWB technology for activity recognition is still in its early stages and needs further research and development. Its ability to correctly identify human activities for energy optimization and occupant comfort in buildings is promising.

A vision-based deep learning approach is employed by Tien et al. [97] for detecting and recognizing human activities in an office environment. The proposed approach used a convolutional neural network that classified occupancy activities in real-time with an average detection accuracy of 98.65%. Such an approach optimizes HVAC based on dynamic occupancy changes, reducing errors in estimating heating gains. Integrating with HVAC controls, thermostat set points can be adjusted to suit building space requirements, enhancing comfort and reducing energy consumption. Similarly, a video-based analysis has been introduced in [98] for recognizing daily activities in ambient assisted living. With the help of an affordable camera, the system was able to predict when a subject would act and move, enabling the detection of abnormal activities. The proposed approach uses a discriminative HMM to recognize action sequences. For training and evaluating the HMM, the study introduces the RADiaL dataset, which contains RGB-D images and 3D person positions. The proposed research aims to contribute to monitoring health and providing elder care services. While camera-based HAR has demonstrated significant accuracy and efficacy in many environments, it presents challenges like privacy issues. Due to these challenges, researchers are investigating alternate approaches to tackle these concerns while maintaining the same level of accuracy and effectiveness in HAR.

In recent decades, smartphone-based HAR has captivated interest across academics and industries, promising huge improvements in how we interact

²Data Sets available at - <https://www.cis.fordham.edu/wisdm/dataset.php>

with our devices and the environment. Shi et al. [99] explored smartphone technology to propose a lightweight HAR system. The proposed approach involved a deep-learning algorithm to enhance HAR. Furthermore, the study introduced a novel pre-classification strategy based on clustering centers to reduce the frequency with which deep learning models are used. Cheng et al. [100] combine cellphone technology with wearable devices for HAR so to take advantage of both technologies. This paper proposes a computationally efficient CNN using conditionally parametrized convolutions for real-time HAR on mobile and wearable devices. The proposed approach was benchmarked on different public datasets, including the WISDM, the PAMAP2, the UNIMIB-SHAR, and the OPPORUNITY datasets. Authors demonstrate that, compared to the single cell phone-based technology for HAR, their study has achieved better results. The cellphones and wearables-based prototypes for HAR perform very well, but anyway, they have some limitations. More in particular, these technologies are subject to user compliance and require individuals to always carry their phones or wear these devices, which can sometimes make them uncomfortable. In addition, the accuracy of phone-based sensors may vary depending on the phone's positioning and the user's behavior, resulting in possible data inconsistencies.

Cook et al. [101] introduced the concept of unsupervised learning for HAR based on the CASAS dataset, which is composed of a series of sensors' acquisitions that can trigger specific activities in an environment (e.g., if some appliances are used in the kitchen, people are cooking). This methodology does not rely on established classes but rather analyzes data without predefined labels. By modeling and tracking the recurrence of these identified patterns in conjunction with predetermined actions, the approach improves the accuracy of detecting these predetermined activities. The study outlines methodologies for HAR, discovering new activities, and integrating discovery-based boosting to enhance activity identification. The authors evaluate the proposed approach by analyzing sensor data from three smart apartments where people perform their regular activities. Gomez et al. [102] developed a real-time HAR system for older people who are living alone. The authors employed unsupervised learning that can distinguish sixteen different activities. The proposed approach used the public dataset SDHAR-HOME, which collects data from non-intrusive sensors and Bluetooth beacons to monitor and observe HAR. The system utilizes a fusion of two concealed Markov models to determine the whereabouts and actions of the inhabitants accurately. The proposed approach demonstrated accuracy rates of 86.78%.

The aforementioned works utilize sensors to monitor interactions with devices to understand activities. While this approach is highly effective for activities involving specific devices, it has significant limitations for activities that do not involve furniture or appliances. For example, it would be very difficult to understand people making gymnasts in a room or cleaning

around with a mop. Maitre et al. [103] use three UWB radars for recognizing fifteen different activities in an SB environment. The authors employ different deep learning models, including CNN and LSTM, to distinguish each activity from the others. The proposed approach achieved an accuracy of 90% for the classification of 15 different activities. Hamalainen et al. [104] explore the use of UWB radar technology to monitor predetermined movement patterns in a residential area of a building. The authors have employed fault tolerance and reliability analysis, significantly contributing to the study. The study utilized various machine learning algorithms, including KNN, to classify diverse activities. Beaulieu et al. [105] used three UWB radars to investigate a deep learning model that uses transfer learning, data fusion, and minimal pre-processing to recognize activities and movements. It combines LSTM with an improved EfficientNet model. For activity recognition, studies with 15 activities made by 10 participants in a furnished apartment revealed a Top-1 accuracy gain of 18.63% over earlier work, with Leave-One-Subject-Out cross-validation obtaining 65.59%. A single subject executed four unique arm motions, resulting in a Top-1 accuracy of 73%. The model correctly identified large and fine-grained motions but struggled with medium-sized movements due to insufficient variance.

Noori et al. [106] used UWB radar on a mobile robot to track subjects within a range of 1.5-2.0m, and train an LSTM model for HAR based on data from five activities. The proposed approach encompasses a discriminant analysis that combines PCA and LDA, which achieved 99.6% accuracy and demonstrated superior performance compared to conventional machine learning techniques. The technique's robustness was demonstrated through cross-validation and public dataset testing, highlighting its potential application in human-robot interaction, such as in eldercare. The study found that UWB-enabled systems accurately identified daily activities and optimized healthcare services for the elderly. Imbeault et al. [107] use three UWB radars to detect activities in a prototype apartment. The SelectKbest algorithm was used to reduce the dimension of the data from the radar. The proposed approach employed a Random Forest to recognize and classify each activity. The proposed approach has been compared with the author's previous work and found to have 13% higher accuracy. Bouchard et al. [12] investigate the feasibility of using UWB Doppler radars to identify activities. A small set of UWB radars recognized activities in a realistic 40-square-meter apartment environment using a simple configuration and classical feature engineering. A Random Forest algorithm classifies 15 different activities performed by 10 individuals with an accuracy of 80%, an F1-score of 79%, and a Kappa coefficient of 77%. These results demonstrate that UWB radars have potential applications in home automation for detecting human activities. Li et al. [108] introduce a novel approach to monitoring vital signs and detecting falls of elderly individuals using UWB radar technology. The system provides tailored telehealth services using edge

computing and cloud infrastructure. Attention-based 2D-CNN and LSTM were used to train the model and extract features from the raw data. The proposed study accurately identifies three types of falls: standing-to-fall, bowing-to-fall, and squatting-to-fall, and recognizes people's sitting-down status in an environment. The study in [109] explored machine learning and deep learning models for detecting indoor human activities using the UWB radar. The study proposes a self-attention-based CNN that enhances activity monitoring performance by efficiently extracting information from multiple orientations. The experimental findings indicate that the Concat self-attention CNN attained 94% accuracy across two datasets, Parallel Spatial Attention Ensemble and Cascade. The study claims their proposed model has fewer parameters than others, making it more suitable for real-time applications. The study in [110] presents the Squeeze-and-Excitation Residual Concatenate Network (SE-RCNet), a fall behavior recognition network that improves recognition performance by including residual connections and Squeeze-and-Excitation modules. The SE-RCNet achieved average F1 scores of 94.0%, 94.3%, and 95.4% when it examined three types of radar maps, outperforming other deep learning models. The study developed an adaptive weighted fusion method that changes component weights based on the quality of each radar map. This makes recognition more accurate and goes beyond the limits of traditional decision-fusion methods. Using SE-RCNet and the adaptive weighted fusion method to improve an F1 score on a custom dataset helped the study reduce false positives and false negatives in fall recognition. The paper in [111] proposes Fusion Map-MobileNetV3 and ShuffleNetV2-Neural Networks(FuM-MS-Net), a radar-based HAR method. This method integrates multiple data representations into a fusion map, including a time-doppler map, time-range map, and cadence velocity diagram. The approach achieves a recognition accuracy of 96.88% on a public dataset by efficiently extracting features using lightweight networks, MobileNetV3 and ShuffleNetV2. By utilizing data and feature-level fusion, this approach enhances performance over traditional methods, demonstrating the system's efficacy in privacy-sensitive and low-light conditions.

Even though the works for HAR in the state of the art are promising, they still present some limitations. Most of the studies, including the one in [108]) have just considered three to five activities for activity recognition.

Novel Approaches for Occupancy Prediction in Cognitive Buildings

This chapter provides a brief overview of the proposed approaches we developed for occupancy prediction, along with the real-time case studies we conducted for them. Furthermore, we detail the experimental configuration and give the simulation outcomes derived from these initiatives. We will explain the broad topic of occupant behavior, which in this case includes occupancy prediction such as binary and multi-occupancy.

4.1 Binary occupancy prediction in buildings

This subsection briefly explains the use of our proposed architecture for binary occupancy prediction, which employs FL and LSTM neural networks to predict occupancy in various rooms inside a building. Moreover, the proposed approach showed how FL could help in the occupancy predictions for those rooms where the training of a specific model cannot be performed. Simulations made on the dataset show that the approach is promising and can be effectively used to train a preliminary occupancy model for all the building rooms. In particular, we reached an accuracy of about 94.5% on the considered data by applying our approach. The following subsections present a comprehensive analysis of our proposed approach.

4.1.1 Reference architecture for binary occupancy prediction in buildings

The reference architecture of our approach is depicted in Fig. 4.1, where three layers are highlighted. Such layers comprehend:

- a *Cloud/Edge Aggregator Layer* where an *Aggregator Node* (AN) is hosted. Such AN can be deployed in the cloud or at the edge and has

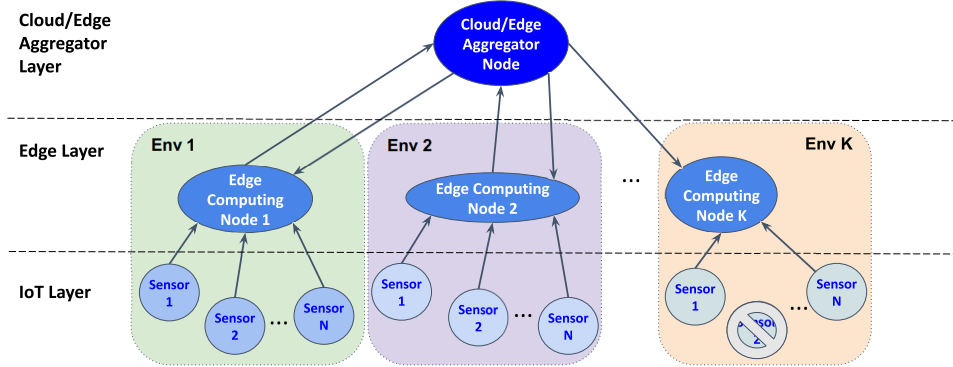


Figure 4.1: The reference architecture of the proposed approach for binary occupancy prediction.

the aim to merge all the models from the underlying layer and send the aggregated model to the nodes in the layer below;

- an *Edge Layer* in which some Edge Computing Nodes (ECNs) are hosted. These nodes are deployed at the edge to control their environment (*Env*), close to the sensors from which they have to collect data to calculate their model. It is essential to have these nodes on edge so to (i) limit the communications towards the cloud or a higher edge layer, (ii) decrease the energy consumption in the communication itself, (iii) increase the privacy of the acquired IoT data by keeping it as close as possible to the data producers, and (iv) tolerate disconnections from the Internet or higher-level networks.

ECNs also run the local calculated model which, during the execution of the federated learning algorithm, is updated according to the new data it gathers and the global model is given by the AN. Some ECNs can be incapable to train their own model due to, e.g., missing computing resources or lack of sensors for measuring the real occupancy of a space. In this case, these ECNs will only receive and exploit an already trained model from the AN;

- an *IoT Layer* hosting sensors scattered in an environment and collecting useful data for both the training and the execution of the model to be computed.

The proposed approach is formalized in Algorithm 1 which represents the traditional FL approach with the extension related to our proposed case, namely the dissemination of the computed LSTM model to ECNs (e.g., the Edge Computing Node K in Fig. 4.1) that do not compute their models

because they cannot count on historical data from their specific room or cannot gather the ground truth about the occupancy status of their rooms. This approach, purposely designed for our case, can be extended to exploit any neural network model. More in detail, in Algorithm 1, an initial model is sent from the AN to all the ECNs. Among such nodes, those that are able to make the training phase collect IoT data from the underlying layer and update their internal models (these two operations can be repeated per *epoch* times, in the case in which the ECNs want to repeat their training on some other collected data). All of this can be repeated for *Rounds* times together with the migration of the models computed on the ECNs towards the AN, their merge, and the final dissemination of such a merged model to all the ECNs. It is worth noting that in this way, an ECN, for example, the ECN K in Fig. 4.1, can benefit from an LSTM (or any other model) that has been already trained. Fig. 4.2 shows a flow chart that illustrates how, in the proposed approach, all the layers presented above work in parallel. In particular, it highlights how the IoT layer gathers data through sensors and passes them to the edge layer. The edge layer stores data and updates the local model as soon as the training is performed. Additionally, the cloud aggregator sends/receives updated parameters to/from the edge layer to aggregate all the models. It's worth noting that all of this is made by all the environments collaborating in the presented training phase.

Algorithm 1 The Proposed Approach based on the Federated Learning Algorithm

```

1: procedure LSTM_FL(Rounds)
2:   currR = 0
3:   sendModel(aggregatorLayer, edgeLayer) ▷ the initial model is sent
   to the edge
4:   while currR < Rounds do
5:     collectIoTData(edgeLayer_training) ▷ data is collected from
   the IoT Layer
6:     updateInternalModel(edgeLayer_training) ▷ the current
   internal model is updated based on the data collected
7:     sendModel(edgeLayer_training, aggregatorLayer) ▷ the
   updated model is sent to the aggregator layer
8:     aggregateModel(aggregatorLayer) ▷ the updated models are
   aggregated in the aggregator layer
9:     sendModel(aggregatorLayer, edgeLayer) ▷ the aggregated
   model is sent again to the edge
10:    currR ++
11:  end while
12: end procedure

```

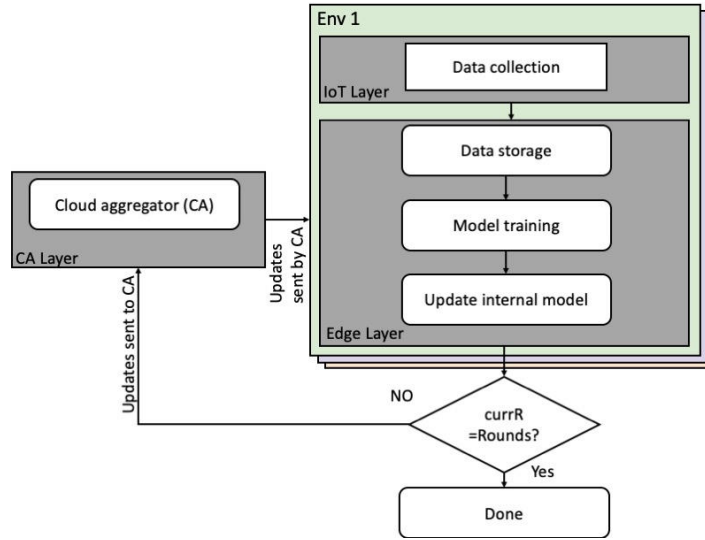


Figure 4.2: The reference flowchart of the proposed approach for binary occupancy prediction.

4.1.2 Case study for binary occupancy prediction in buildings

The approach introduced in subsection *Reference Architecture* has been executed in a simulation case study that will be described in the following. In particular, subsection *Setup* will present the setup of the system, and subsection *Results* will show some simulation results.

Setup

In this project, we conducted the experiments by using *Google Colab*¹. Such experiments are coded in *Python*² (Release 3.7.0) and rely on *TensorFlow*³ (version 2.8.2), *TensorFlow Federated*⁴ (version 0.16.1), and *pandas* (version 1.3.5)⁵ libraries. The code uses the dataset that is analyzed and explained here [112] and that can be found on GitHub⁶. In particular, such a dataset comprehends sensor data about temperature, humidity, CO₂, and Light for a room in a building. Moreover, it shows the actual occupancy value.

To execute our simulations, we assume that we can have a building with eleven rooms (with their ECNs) that are used approximately in the same way. Ten of them have an IoT layer underneath collecting data for training,

¹Google Colaboratory. <https://research.google.com/colaboratory/>

²Python 3.7.0. <https://www.python.org/downloads/release/python-370/>

³TensorFlow Homepage. <https://www.tensorflow.org/>

⁴TensorFlow Federated. <https://www.tensorflow.org/federated>

⁵pandas - Python Data Analysis Library. <https://pandas.pydata.org/>

⁶<https://github.com/LuisM78/Occupancy-detection-data>

while the last one cannot do training as it misses the IoT sensors required for assessing the truth occupancy value. By applying the FL algorithm reported in subsection *Reference Architecture*, models are calculated in parallel in all the rooms, merged, and then disseminated to all the eleven ECNs. According to this, we divided the dataset into 12 parts: 10 parts (of about 1600 elements each) represent the data from the ten different rooms and are used as the training set for the models calculated at the edge of the rooms themselves; one part (of about 1800 elements) is used as a validation set, and one part is used as the test set of the merged model. This last test set is simulated as executed in the eleventh room. All the results shown in the following are related to this last room.

The ANN used for the occupancy prediction comprehends one LSTM layer of 32 units, a Dropout layer, one LSTM layer of 16 units, one Dropout layer, and, finally, a Dense layer with just one output. On each edge node we executed, for the training, five epochs and the learning processes were repeated per 25 rounds. The occupancy prediction is trained considering as input the previous 60 minutes of the real data and as output the occupancy state at the next minute.

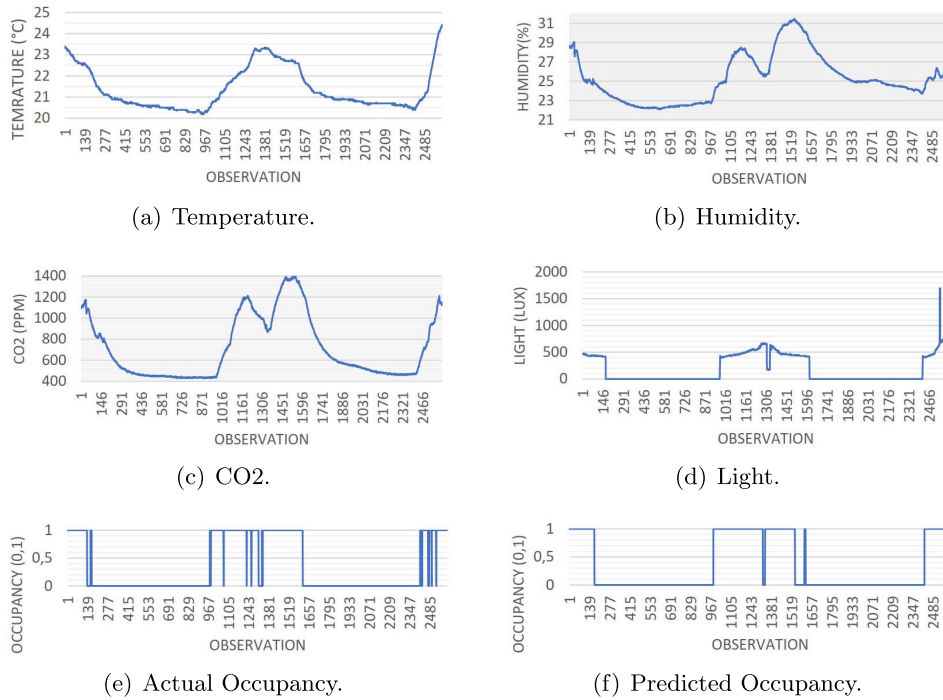


Figure 4.3: Sensor data with actual and predicted binary occupancy.

		Predicted Occupancy	
		Predicted Positive (PP)	Predicted Negative (PN)
Actual Occupancy	Positive (P)	(TP) 834	(FN) 77
	Negative (N)	(FP) 67	(TN) 1626

Table 4.1: Confusion Matrix of the test set.

Results

This section highlights the simulation results that we got. In particular, Fig. 4.3 shows the values of temperature, humidity, CO₂, Light, and actual occupancy taken from the test set we used, which in our example corresponds to the data of the eleventh room, as introduced above. Moreover, Fig. 4.3 also depicts the occupancy predicted with our simulations. As can be seen, the accuracy of the trained prediction model is very high. In this specific test, we have above 94.5% accuracy. Table 4.1 shows the confusion matrix for the test set considered. Starting from this table, we also calculate the precision, recall and F1 scores [59] that reached the values of 0.925, 0.915, and 0.920, respectively.

4.2 Multi occupancy prediction in buildings with single-layer FL

This subsection will introduce a data-driven approach combining Internet of Things sensors, Machine Learning, Edge computing, and Federated Learning to predict multi-occupancy in buildings. We will explain how our proposed approach is used in real-time data collection from the ICAR-CNR IoT Laboratory to extract insights into occupancy patterns within a multi-occupant environment. Such data are gathered through a purposely developed IoT-based network. Moreover, we conduct a comparative study by analyzing several FL configurations, which are obtained by varying different combinations of High-Level Rounds (HLRs) and epochs. HLRs denote the instances when a model is aggregated at a high level, often in the Cloud, while epochs refer training rounds involving clients at the network's edge. The following subsections present an in-depth analysis of our proposed approach.

4.2.1 Proposed architecture for multi-occupancy prediction with single layer FL

This paper aims to build upon the approach introduced in [53] by extending its capabilities with a more sophisticated FL solution. Specifically, the added enhancement allows the possibility to explore different configurations in the pair High-Level-Rounds/epochs (*HLRs/epochs*). Moreover, the approach is designed for considering multi-occupancy forecasting, combining FL with LSTM networks, and is based on a three-layer architecture, as illustrated in Figure 4.4, where:

- the *IoT Layer* comprises intelligent IoT devices such as sensors, actuators, and smart objects distributed across specific rooms in a building. These devices play a crucial role in collecting valuable data necessary for training and executing the LSTM model;
- the *Edge Layer* is the intermediate layer and hosts Edge Computing Nodes (ECNs), e.g., users' notebooks, mobiles, or cheap single-board computers. These nodes are strategically deployed close to the sensors at the edge to collect data and calculate their respective models efficiently. This positioning at the edge brings several significant advantages, including limiting communications towards higher edge layers or the Cloud, reducing energy consumption during communication, enhancing data privacy by keeping it close to data producers and ensuring resilience to potential disconnections from the Internet or higher-level networks. Additionally, the ECNs autonomously run their local calculated models, which are updated during the FL process based on newly gathered data. It is important to note that some ECNs may not be capable of training their own models due to various constraints, such as limited computing resources or the absence of sensors to measure real occupancy. In such cases, these ECNs can receive and utilize pre-trained models from the Cloud/Edge Aggregator Layer;
- the *Cloud/Edge Aggregator Layer*: is the upper layer of our architecture and can be deployed both in the Cloud and at the edge, in a node capable of interacting with all the other edge nodes and with sufficient processing capabilities. This layer hosts a Cloud/Edge Aggregator (CEA) which receives models from the Edge Layer, aggregates them, and distributes the merged model to the underlying nodes.

The literature offers several algorithms for merging the models in FL. FedAvg [113] is the first and mostly used algorithm, which merges locally trained models at each HLR. Among the others, FedProx [114] enhances FedAvg with a proximal term and FedNova [115] ensures objective consistency and rapid error convergence through normalized averaging. This paper adopts a weighted strategy based on the most popular FedAvg algorithm.

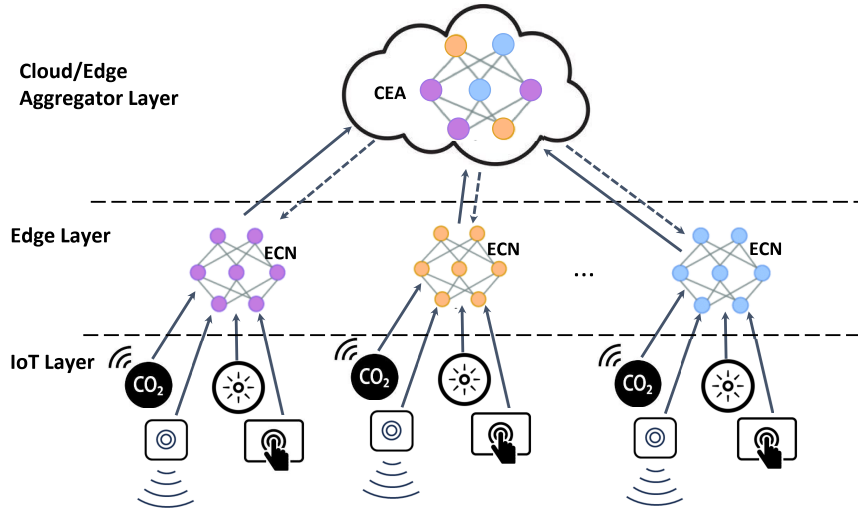


Figure 4.4: The reference architecture considered in the proposed approach for multi-occupancy prediction with single layer FL.

To better describe the approach and its capabilities to accommodate various configurations (*HLRs/epochs*) for FL, we introduce Algorithm 2. In particular, such an algorithm initiates by sending an initial model from the CEA to all the ECNs. Among the ECNs, those capable of participating in the training phase, since they have sufficient data from the IoT Layer and availability of computation resources, actively collect data from IoT devices and update their internal models. This process is repeated for *Epochs* times, allowing ECNs to refine their models. Then, ECNs send their computed models to the CEA, which merges them and shares the result with the ECNs whenever the newly merged model's F1 score (macro) is greater than the last one's F1 score (macro). All the steps, apart from the initial model sharing phase, are repeated for *HLRs* times so as to make several iterations in the local model training and high-level merge.

This collaborative training approach ensures that all the ECNs, including nodes that can't calculate their own model, can benefit from LSTM (or other) models that have already been trained in other nodes.

4.2.2 Experiments for multi-occupancy prediction with single layer FL

The approach presented in subsection *Proposed Architecture* has been tested in a real case study, which will be discussed in the following subsections. In more detail, subsection *Case Study Setup* provides an overview of our case study and details the deployment used and the setup of the system, while Subsection *Numerical Results* showcases the obtained simulation

4.2. MULTI OCCUPANCY PREDICTION IN BUILDINGS WITH SINGLE-LAYER FL

Algorithm 2 The FL Approach used with variable *HLRs* and *Epoch* parameters.

```
1: procedure LEARN_FL_MODEL(HLRs, Epochs)
2:   currH = 0, currE = 0
3:   sendModel(CEA, ECNs)
4:    $\triangleright$  the initial model is sent to the ECN nodes at the edge
5:   while currH < HLRs do
6:     while currE < Epochs do
7:       gatherIoTData(ECNs)
8:        $\triangleright$  the ECNs gather data from the IoT devices
9:       updateEdgeModel(ECNs)
10:       $\triangleright$  the ECNs update their internal model
11:      currE ++
12:    end while
13:    sendModel(ECNs, CEA)
14:     $\triangleright$  ECNs send their updated models to the CEA
15:    mergeModel(CEA)
16:     $\triangleright$  CEA merges the updated models from the ECNs
17:    if mergedModel(CEA).F1Score > lastmodel
18:    then
19:      sendModel(CEA, ECNs)
20:       $\triangleright$  CEA sends the merged model to the ECNs only if the
21:       $\triangleright$  F1 Score of the new Model is greater than the last one
22:    end if
23:    currH ++
24:  end while
25: end procedure
```

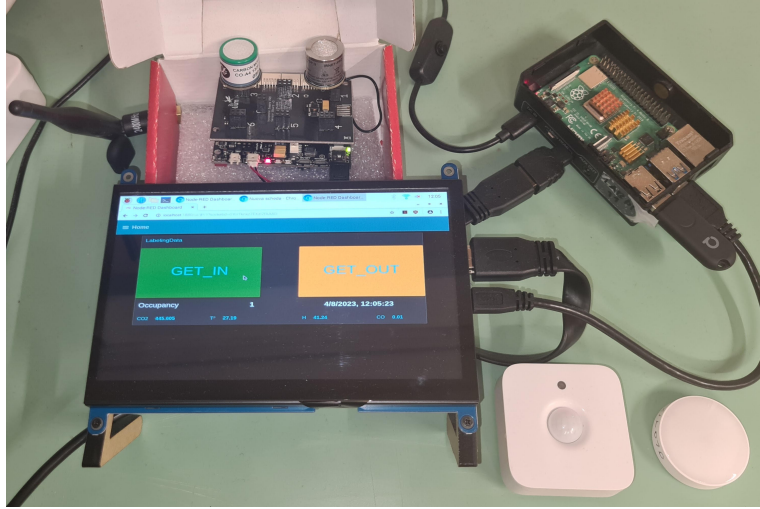


Figure 4.5: The devices involved in the implemented case study for multi-occupancy prediction.

results.

Case Study Setup

The case study focused on creating an occupancy prediction system for the ICAR-CNR Institute building located in Rende, Italy. In such a building, we deployed several sensors in all the rooms so to understand occupancy and forecast it for all the users. Anyway, in this case study, we will focus only on the real data collected in the Internet of Things Laboratory, where we developed a system comprehending sensors and actuators of different technologies and several Raspberry Pi 4s, acting as ECNs and responsible for collecting and processing data. In particular, for our case study, we used a ZigBee light sensor (Xiaomi Mijia Smart Light Sensor), a ZigBee passive infrared (PIR) sensor (Philips Hue Motion Sensor), and a WiFi CO₂ sensor (in the Gases Sensor Board Pro of the Wasmote WiFi Pro⁷). In order to complement the devices required for the experiments, we also implemented a Virtual Sensor, using a touchscreen connected to one of the ECNs, to obtain accurate data on occupancy. During the data collection periods, occupants in the laboratory used this touchscreen to indicate their presence. This touchscreen was also used to visualize the most recent data in the laboratory through a Node-Red application. All the devices used for this case study are shown in Figure 4.5.

Starting from this system, two datasets were created with data from

⁷Wasmote Gases Sensor Board Pro. https://development.libelium.com/gases_pro_sensor_guide/gases-pro-sensor-board-calibrated

4.2. MULTI OCCUPANCY PREDICTION IN BUILDINGS WITH SINGLE-LAYER FL

light, CO₂, PIR, and virtual touchscreen sensors: the first one comprehends 17 days gathered in the month of November 2022, and the second one contains 5 days taken in October 2022. Both datasets present data collected during working hours with a timestep of one minute between two tuples, and the data collected is limited to represent zero, one, or two people. It is worth noting that the approach proposed is general and would also be applied to data showing more occupants in the considered environment. In order to use the approach reported in subsection *Proposed Architecture*, we emulated to have, in the first dataset, data from ten different rooms with similar patterns of use. So, this dataset has been divided into eleven segments. Among these, ten segments are used for training models at distinct emulated ECNs. The remaining segment is utilized as a validation set encompassing all ECNs. It is worth noting that, prior to using the data for forecasting, some data preprocessing (e.g., handling missing values, data normalization, and data transformation to 3D structure) is done to prepare the raw data stored in the datasets, making it suitable for the LSTM network training.

We have implemented the approach using *Google Colaboratory*⁸, *Python*⁹ (Release 3.7.0) and libraries like *TensorFlow*¹⁰ (version 2.8.2), *TensorFlow Federated*¹¹ (version 0.16.1), and *Pandas*¹² (version 1.3.5). The proposed model incorporates specialized LSTM with one layer having 32 units and another having 16 units. These LSTM layers effectively capture temporal dependencies, enabling the model to learn long-term patterns. To prevent overfitting, two dropout layers with a 0.10 dropout rate are utilized, encouraging robust learning. The final dense layer generates continuous numerical predictions. The activation function used for LSTM is the hyperbolic tangent (tanh), facilitating efficient information flow and preventing issues like exploding gradients. The model has been trained to predict the occupancy 10 timesteps ahead (occupancy at the tenth minute in the future) using the previous 60 observations (sixty minutes in the past) as input in each case. The (*HLRs/epochs*) couples compared in the following, and hereafter represented as "FL-Model(*HLRs*, *epochs*)," are: FL-Model(50, 2), FL-Model(20, 5), FL-Model(10, 10), FL-Model(5, 20), and FL-Model(2, 50). Moreover, all the cases above will be compared with a centralized case, in which there is no training at the ECNs in the edge, but all the data for training the only centralized model will always be sent to a centralized node. It is worth noting that the initial centralized model consists of the LSTM introduced above.

⁸Google Colaboratory, <https://research.google.com/colaboratory/>

⁹Python 3.7.0. <https://www.python.org/downloads/release/python-370/>

¹⁰TensorFlow. <https://www.tensorflow.org/>

¹¹TensorFlow Federated. <https://www.tensorflow.org/federated>

¹²Pandas. <https://pandas.pydata.org/>

Numerical Results

This section is devoted to illustrating the quantitative results achieved in terms of the evaluation measures used to demonstrate the quality of the proposed approach. Specifically, the predictive performances of the models were assessed using the following well-known metrics: *Accuracy*, *Precision*, *Recall*, *AUC* (Area Under the ROC Curve), *F1 Score* (a harmonic mean of Precision and Recall), the *Mean Absolute Error* (MAE), *Precision*, and the *Recall*. Except for Accuracy and MAE, all metrics were computed using a Macro-Averaging scheme, furnishing a per-class average for each measure.

Table 4.2 reports the results obtained by varying configurations of the FL-Model, particularly by considering different values of *HLR* and epochs. Additionally, we present the performance of the model trained in a centralized way against the complete dataset. This serves as a benchmark, representing an ideal yet unrealistically centralized scenario where all data are available in a single node for analysis. While FL-Model(20,5) strikes an optimal balance between precision and recall, FL-Model(10,10) delivers competitive results while demanding reduced communication with the coordinator node. Notably, at each HLR iteration of the FL process, the performances of the current model are compared with the ones of the model trained in the previous iteration. If the new model exhibits a higher F1-Score on the validation set, then the current model is updated. The learning algorithm proceeds without any model updates if the new model doesn't improve. This behavior influences the results on the test set; specifically, when the model is not updated, its accuracy remains constant over the epochs.

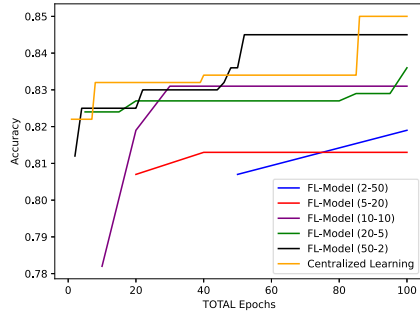
Table 4.2: Evaluation metrics computed for different configurations and comparison with the fully centralized solution. All the metrics are calculated by using a Macro-Averaging scheme. Bold values highlight the best configuration for a given measure.

Configuration	MAE	Accuracy	Macro Precision	Macro Recall	Macro F1	Macro AUC
FL-Model(50,2)	0.160	0.850	0.860	0.820	0.840	0.850
FL-Model(20,5)	0.174	0.836	0.860	0.791	0.816	0.854
FL-Model(10,10)	0.176	0.831	0.865	0.786	0.814	0.854
FL-Model(5,20)	0.201	0.813	0.809	0.763	0.781	0.818
FL-Model(2,50)	0.192	0.819	0.831	0.766	0.789	0.829
Centralized Model	0.173	0.850	0.845	0.820	0.842	0.854

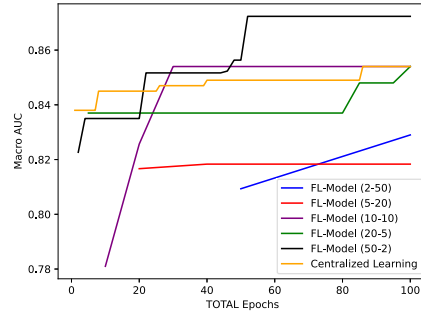
Moreover, we illustrate how the values of these metrics change as the number of epochs increases, as depicted in Figures 4.6, where we print a point per merge in the CEA (per each HLR). As expected, the number of epochs plays a key role; indeed, with the increasing number of iterations, we also observe an improvement in the predictive performances of each model.

Finally, in Figure 4.7, we show in detail the real occupancy as in the testset and compare it with the predicted ones according to all the cases

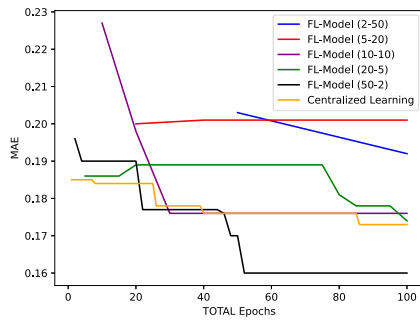
4.2. MULTI OCCUPANCY PREDICTION IN BUILDINGS WITH SINGLE-LAYER FL



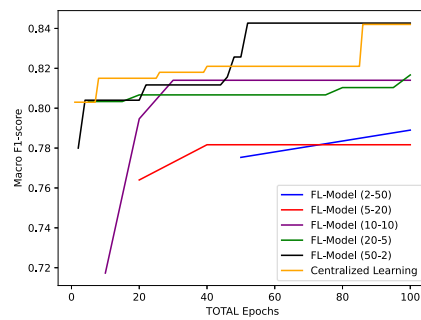
(a) Accuracy.



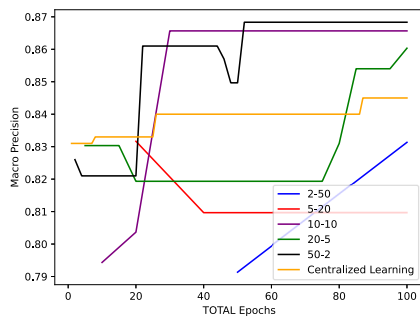
(b) Area Under the Curve (AUC).



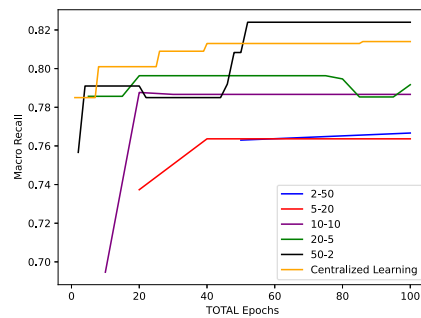
(c) Mean Absolute Error (MAE).



(d) F1-Score.



(e) Precision.



(f) Recall.

Figure 4.6: Evaluation metrics curves computed against the test set on increasing number of epochs.

introduced above.

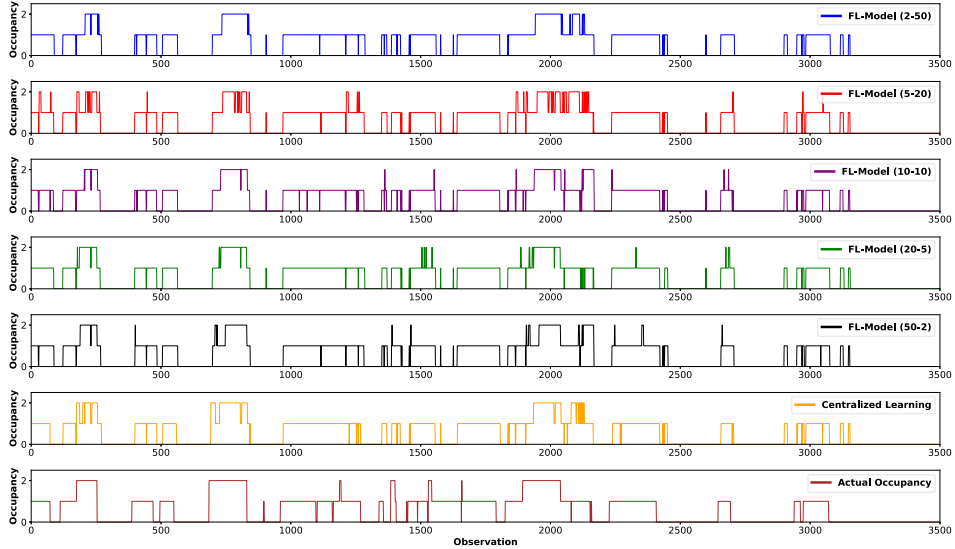


Figure 4.7: Actual vs Predicted multi-occupancy for the considered case study.

4.3 Multi occupancy prediction in buildings with multi-layer FL

This project proposes an innovative approach for multi-occupancy prediction in Cognitive Buildings, incorporating a multi-layer hierarchy for Federated Learning, utilizing IoT devices at the Edge, implementing long short-term memory neural network models, and exploiting Edge Computing. We also introduce a versatile design template for developing real distributed systems for occupancy prediction. The proposed approach uses a distributed paradigm to safeguard data privacy so that the collected data is used to train separate local deep learning models, which are then merged into the Cloud. The paper validates the approach by providing a preliminary prototype realized at ICAR-CNR, Rende, Italy, and presents a performance analysis, which shows that the occupancy is predicted with an 84.5% accuracy. The following subsections will describe the proposed approach.

4.3.1 The Reference architecture for multi occupancy prediction with multi layer FL

The reference architecture of the proposal is shown in Figure 4.8. It is simplified through a tree-shaped network in which the IoT devices are located in the bottom layer, the Edge devices are in the intermediate layer,

4.3. MULTI OCCUPANCY PREDICTION IN BUILDINGS WITH MULTI-LAYER FL

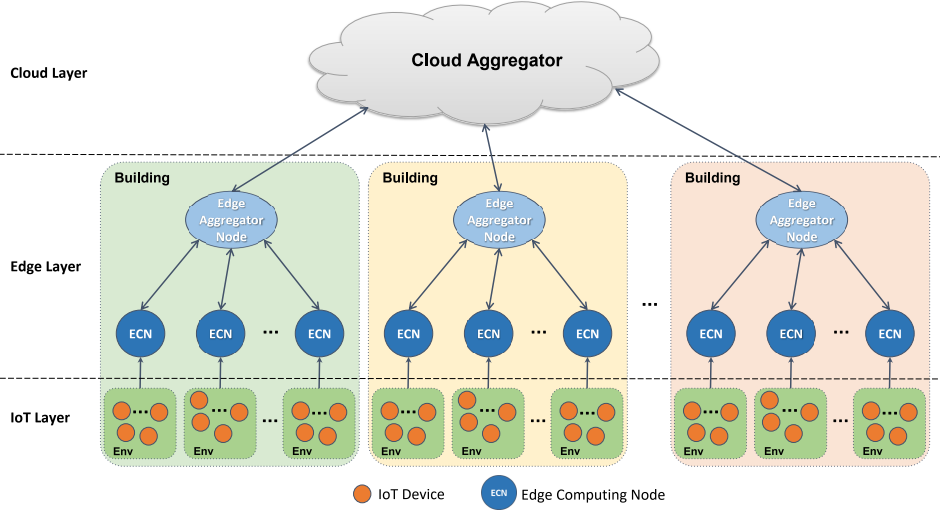


Figure 4.8: The reference architecture proposed for multi-layer hierarchy Federated Learning.

the Cloud is the top layer, and the links among them are the communication channels. More specifically, the three layers are:

- The *IoT layer*. It hosts IoT devices including sensors/actuators/smart objects. These devices are scattered throughout their own environments (Env), which are specific rooms of a building, and collect useful data for training and executing the LSTM model. They send the data, possibly after a pre-elaboration, to the Edge Layer.
- The *Edge Layer*. It is divided into two sub-layers, namely, the *Low-Level Edge Layer* and the *High-Level Edge Layer*.

The *Low-Level Edge Layer* hosts the Edge Computing Nodes (ECNs) deployed at the network's Edge to control the building's rooms. Usually, they are installed as close as possible, or inside, the specific room that they manage. ECNs accomplish the following tasks: (i) they collect and store data gathered from the IoT Layer; (ii) they calculate and periodically update their LSTM model based on the data collected from the IoT devices to which they are connected; (iii) they execute the model to forecast the presence of people in the room they control; (iv) they send the computed model to the higher levels; and (v) they update their models according to the models received from the upper layers. It is worth noting that some ECNs may be unable to train their models due to missing computing resources or a lack of data regarding real occupancy. In this case, the ECNs will only exploit the trained models received from the High-Level Edge Layer.

The *High-Level Edge Layer* hosts some Edge Aggregator Nodes (EANs). These nodes are also deployed at the network’s Edge but they usually have higher capacities in terms of computation with respect to the ECNs. According to our architecture, in each building, one EAN is required to be capable of gathering all the models from the ECNs in its building. The work of the EAN is to: (i) aggregate such models and send the merged one to the ECNs below and (ii) send the merged model to the Cloud and wait for a Cloud-aggregated model to be still forwarded to the ECNs. In some situations, an EAN can also be represented by an ECN capable of aggregating all the models from the other EANs and communicating with the Cloud layer.

The presence of both ECNs and EANs in the Edge layer is very important to: (i) prevent too many communications towards the Cloud, (ii) reduce energy consumption during the communication process, (iii) ensure privacy for the acquired IoT data by keeping it close to the data producers, and (iv) tolerate disconnections from the Internet. Regarding the latter point, it is worth noting that, in the case of Internet connection loss, ECNs and EANs continue to operate and interact thanks to local networks (e.g., WiFi or Ethernet), so allowing the data collection from the IoT Layer and the model exploitation. The only effect of temporary Internet disconnections is a delay in the process of model aggregation.

- The *Cloud Layer*. It hosts the *Cloud Aggregator* (CA), which has the objective of receiving all the models from the Edge Layer, aggregating them, and distributing the aggregated model to the Edge Aggregator nodes.

The introduced Federated Learning architecture is based on a multi-layer Hierarchical Federated Learning (HFL) model [116]. Specifically, the multi-layer HFL conceptualizes an Edge part including several layers to make aggregations of trained models at different levels, thus reducing the communications towards the Cloud and preserving the privacy of the data collected from the IoT devices.

Our work is based on the algorithm 3, reported below, which takes three variables as inputs, namely, *LowLevelRounds*, *MidLevelRounds*, and *HighLevelRounds*. These variables are very important for the training of the LSTM model because they identify how many times a model is merged (calculated) at low, mid, and high levels. These variables also define the number of communications that the algorithm will do at the Edge and towards the Cloud. In particular, the algorithm first sends the initial model from the CA to the EANs and then from the EANs to the ECNs. Once the ECNs have the initial model, they start to collect data from the IoT devices below them and update their model according to this data. When

the low-level rounds, also referred to “epochs” in the literature, end (i.e., $currLLR = LowLevelRounds$), the models trained in the ECNs are sent to the related EANs, in which they are aggregated and then sent back to the ECNs. These operations are repeated for $MidLevelRounds$ times. Successively, the resulting model is sent to the CA, which aggregates the models from all the EANs and sends the merged one back to the EANs and, consequently, to the ECNs. All the operations described here are repeated for $HighLevelRounds$ times.

Some interesting considerations can be made. In particular:

- the explained approach, which has been thought specifically for LSTMs, can also be used with any kind of neural network model;
- the dissemination of the computed LSTM model to the ECNs at the Edge level can also be very useful for the ECNs that do not compute their specific models, either because they cannot count on historical data from their specific rooms or because they cannot gather the ground truth about the occupancy of their rooms.
- as anticipated, the algorithm also considers that, at each layer, there is the possibility to execute several rounds before sending the merged LSTM model to the upper layers. These rounds can be useful to reduce the communications towards the Cloud further and, consequently, reduce the energy needed for communications [117]. Although very interesting, the evaluation of this specific aspect is postponed to future works.

4.3.2 The Design Template

The diagram in Figure 4.9 depicts a design template for the development of an Edge-Based distributed system that applies Machine Learning and Federated Learning for multi-occupancy prediction. It is worth noting that this design template could also be very useful for implementing many kinds of distributed ML applications, and that each component of the model can be placed in a different node.

The template provides some basic items that require to be specialized to develop any specific application. The goal of the proposed design template is to promote separation of concerns and allow the designers to focus on recurrent cross-cutting problems independently. In the following, its composing items are explained in detail.

- **IoT Sensors and Actuators.** They are the real devices of the IoT Layer that can sense the environment around them and act based on commands received from the controlled environment. These devices are fundamental for the specific system to be implemented.

Algorithm 3 The Proposed Algorithm based on the Multi-Layer Hierarchical Federated Learning.

```

1: procedure LSTM_ML_HFL(LowLevelRounds, MidLevelRounds,
   HighLevelRounds)
2:   currLLR = 0, currMLR = 0, currHLR = 0
3:   sendModel(CA, EANs) ▷ the initial model is sent from the Cloud to
   all the EANs
4:   sendModel(EANs, ECNs) ▷ the initial model is sent from the EANs to
   all the ECNs
5:   while currHLR < HighLevelRounds do
6:     while currMLR < MidLevelRounds do
7:       while currLLR < LowLevelRounds do
8:         collectIoTData(ECNs) ▷ data is collected by the ECNs
9:         updateInternalModel(ECNs) ▷ the models are updated in the
   ECNs based on the data collected
10:        currLLR ++
11:      end while
12:      sendModel(ECNs, EANs) ▷ the updated models are sent to the
   respective EANs
13:      aggregateModel(EANs) ▷ EANs merge the updated models from
   the related ECNs
14:      sendModel(EANs, ECNs) ▷ ECNs receive the merged models
   from the related EANs
15:      currMLR ++
16:    end while
17:    sendModel(EANs, CA) ▷ EANs send the aggregated model to the CA
18:    aggregateModel(CA) ▷ CA merge the models from all the EANs
19:    sendModel(CA, EANs) ▷ at each high-level round, all the nodes are
   updated with the last merged model
20:    sendModel(EANs, ECNs)
21:    currLLR = 0, currMLR = 0
22:    currHLR ++
23:  end while
24: end procedure

```

4.3. MULTI OCCUPANCY PREDICTION IN BUILDINGS WITH MULTI-LAYER FL

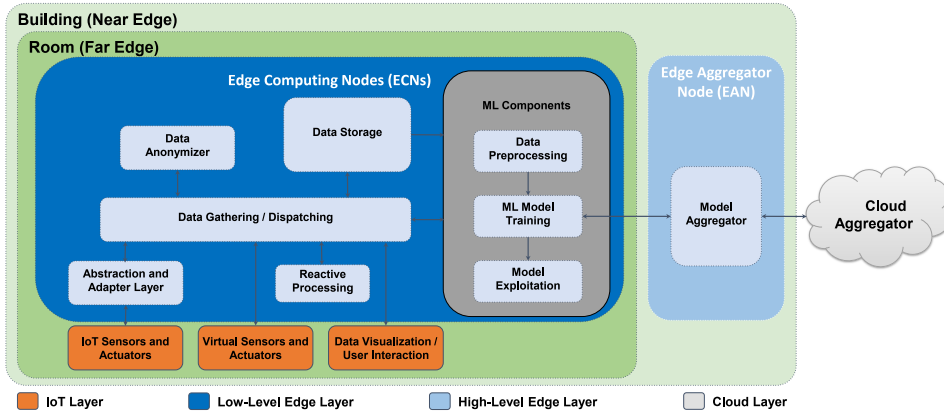


Figure 4.9: The proposed Design Template.

- Virtual Sensors and Actuators.** These virtual devices can produce simulated, historical, and user-inserted data or information taken from external sources (e.g., weather forecasting services). This item can be used for modeling auxiliary entities purposely suited to add further information to the data from real sensors to favor important activities like user data labeling. These entities can also be used to emulate specific sensors that are not physically deployed in the considered environment. It is worth noting that the labeling activity can be essential for supervised ML-learning tasks.
- Data Visualization / User Interaction.** This item models the entity that provides an interaction with the users, including data visualization, which helps the user be aware of the system's state.
- Abstraction and Adapter Layer.** This component abstracts the IoT Devices' details in order to adapt the produced data to the specific system needs. This abstraction enables the replaceability of IoT Sensors and Actuators without affecting the upper layers.
- Reactive Processing.** This item involves simple data processing activities, allowing basic elaborations on the data flowing at the Edge. Such processing can include data filtering, aggregation, and threshold management for reactive actuation.
- Data Gathering/Dispatching.** This is the entity that takes care of the data flow at the Edge: it takes information from all the Edge components and, when needed, forwards it to the needed ones. Different types of dispatching policies can be transparently applied, such as publish/subscribe or broadcast. The main goal of this item is to decouple data producers from data consumers.

- **Data Anonymizer.** This item has the purpose of properly anonymizing data from the IoT Layer in order to preserve privacy without affecting the elaboration of the data itself. For instance, this item can be used to hide faces in photos or video streams.
- **Data Storage.** The Data Storage takes care of saving all the required data on persistent storage like databases or specific files. It is also very useful to provide historical data to the ML Components or, through the Data Gathering/Dispatching component, to a visualizer.
- **ML Components.** This item comprehends three sub-items, namely “Data Preprocessing,” “ML Model Training,” and “Model Exploitation,” that realize the multi-occupancy prediction task. In particular, these items oversee the activities of data preprocessing, model training, and exploitation. It is important to note that the “ML Model Training” sub-item directly interacts with the *Near Edge* “Model Aggregator” so as to share its own model that has been trained at the Edge and receive the new aggregated one.
- **Model Aggregator.** The Model Aggregator receives several ML models from the *Far Edge* rooms of a building and aggregates them, according to specific aggregation algorithms, with the goal of providing enhanced models to the Far Edge computing nodes. This component also interacts with the “Cloud Aggregator”, which is the highest-level model aggregator, to achieve further improved models.
- **Cloud Aggregator.** This aggregator produces the highest-level ML model, which takes into account all the buildings of the developed system.

It is worth noting that, with the exception of the “Data Gathering/Dispatching” component, all the Edge components can be considered as *optional* once a system based on this design template is developed.

4.3.3 Case Study Implementation

This section presents the design and implementation of a case study for the forecasting of multi-occupancy, exploiting the approach presented in the paper. The case study has been implemented in the ICAR-CNR Institute in Rende, Italy, where ten rooms (five on the ground floor and five on the first floor) have been equipped with several sensors (the full set of them is described below) and Edge nodes, i.e., Raspberry Pi. In this section, we present the implemented components and show the deployment of the room equipped with the widest variety of sensors, namely, the IoT Laboratory. Each floor of the ICAR-CNR institute has its own Edge Aggregator node

4.3. MULTI OCCUPANCY PREDICTION IN BUILDINGS WITH MULTI-LAYER FL

(see Section *The Reference Architecture*), so it could be logically considered to belong to a different building.

The collection and exchange of data inherently give rise to privacy issues, which were addressed carefully. On the one hand, the Multi-Layer Hierarchical Federated Learning approach ensures that most data is processed and transmitted locally, while only high-level models are exchanged with the Cloud. On the other hand, before deploying the system, we signed an agreement with all the involved people stating that the data must be anonymized and can be used only for scientific purposes, in accordance with EU General Data Protection Regulation (GDPR) rules [118].

Figure 4.10 shows the main components implemented in each room and building and the information flows among the components, according to the design model introduced in the previous section.

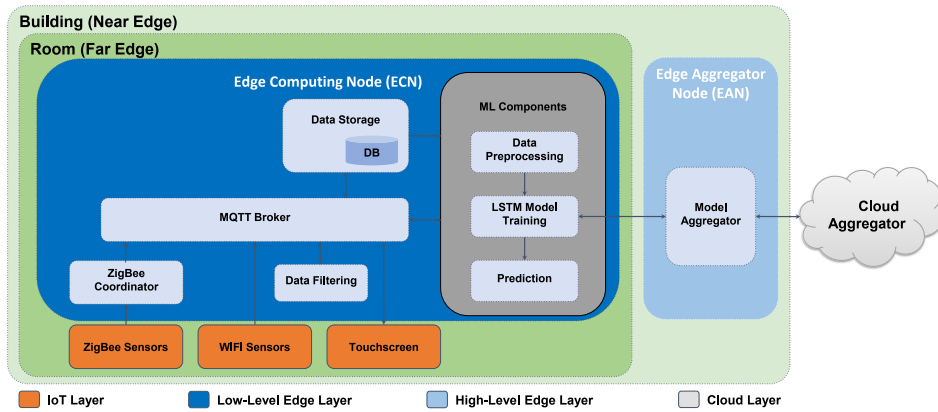


Figure 4.10: The main components implemented for the case study.

In order to implement our case study, we developed a set of components for integrating and historicizing information gathered from sensors connected with different protocols. Moreover, in order to collect ground true data about occupancy, we also implemented a virtual sensor relying on a touchscreen, which can be used by the people in the rooms to notify their presence. All the components are briefly introduced in the following.

- *ZigBee Coordinator and Sensors.* The ZigBee Coordinator acts as an Abstraction and Adapter Layer item and is the primary control center or “hub” for a group of ZigBee IoT Sensors deployed in the room. The coordinator is in charge of transmitting data to the MQTT broker (corresponding to the Data Gathering / Dispatching item) and controlling communications among the ZigBee devices.
- *WiFi Sensors.* We employed WiFi-enabled Waspnote IoT Sensors¹³

¹³Waspnote: <https://development.libelium.com/waspnote-technical-guide/waspnote-kit>

to gather environmental data and transmit it to an MQTT broker.

- *Touchscreen.* We developed a Node-Red-based application that uses a touchscreen video as Virtual Sensor and Data Visualization item to (i) capture human-inserted information about the number of people in a room and (ii) visualize data from all the sensors in the environment.
- *Data Filtering.* It is a Reactive Processing item used to filter not valid data coming from the IoT Layer (e.g., null values or values outside a given interval).
- *MQTT broker.* We used an MQTT broker as a Data Gathering / Dispatching item to enable communication among the different components of the system. The broker uses the publish/subscribe paradigm to collect and deliver messages to the intended destinations.
- *Data Storage and DB.* We implemented a Data Storage item to give persistence to the data coming from the MQTT broker. It receives the sensors' data, transforms it into a database-friendly format, and stores it in the database using a database connector. Once stored, the data can be queried, retrieved, and analyzed by different items and services that can interface with the Data Storage.
- *Data Preprocessing.* This component has been implemented to take the raw data stored in the database and preprocess it to make it ready for the LSTM Model Training.
- *LSTM Model Training.* Once the data has been preprocessed by the Data Preprocessing item, it is used by this component to train the LSTM model. Once the model is trained, it is passed to the Prediction component. The LSTM Model Training also takes care of (i) sending the trained model to the Model Aggregator after executing *LowLevelRounds* and (ii) receiving the updated model from the Edge Aggregator node and from the Cloud.
- *Prediction.* This component, which acts as a Model Exploitation sub-item, makes predictions based on a trained LSTM model received from the LSTM Model Training component. This model can be either a model computed on the Edge Computing Node or a model coming from outside the node.
- *Model Aggregator.* The Model Aggregator, implemented at the Building level, is the item taking care of receiving and aggregating the models from all the rooms of a building. It also takes care of sending, every *MidLevelRound* times, the merged model to the Cloud Aggregator component, receiving the updated model from the Cloud, and

4.3. MULTI OCCUPANCY PREDICTION IN BUILDINGS WITH MULTI-LAYER FL

Table 4.3: List of the sensors exploited within the IoT layer for the presented case study.

Device	Technology	Power	Measurements	Note
Heiman HS1HT-E	Zigbee 3 HA	battery	Temperature [°C]	range: from -20° to 60 °C.
			Relative humidity [%]	range: from 0 to 100%.
Philips Hue Motion Sensor	Zigbee 3 HA	battery	Movement [boolean]	detection range <5m, angle 100°.
Xiaomi Mijia Smart Light Sensor	Zigbee 3 HA	battery	Light [lux]	range: from 0 to 83000 lux.
Waspote WiFi Pro with Gases Sensor Board Pro	WiFi	main	Carbon Monoxide (CO) [ppm]	range: from 0 to 25 ppm.
			Carbon Dioxide (CO2) [ppm]	range: from 0 to 5000 ppm.
Occupancy Counter	Virtual Sensor	main	Number of occupants	Virtual Sensor, measurement available through touchscreen interface and NodeRed application. See description in Section 4.3.3.

disseminating it to the LSTM Model Training components in all the rooms.

- *Cloud Aggregator.* The Cloud Aggregator takes care of receiving and aggregating the models from all the Buildings involved in the system. Once the models, calculated at the Building level, are merged, the Cloud Aggregator sends the models back to all the Model Aggregators. In accordance with the algorithm explained, the Cloud Aggregator repeats its actions *HighLevelRound* times.

It is worth noting that, at each room, all the components in the Low-Level Edge Layer are hosted by a Raspberry Pi 4 mod B with 8GB of RAM. The Model Aggregation component runs, at each floor of the ICAR-CNR institute, on a Desktop computer, and the Cloud Aggregator is implemented as a Virtual Machine in the ICAR-CNR cluster.

Some of the components introduced above are further detailed in the following subsections.

ZigBee Coordinator As we mentioned earlier, the ZigBee coordinator serves as a central hub that receives data from multiple sensors that are deployed to monitor the environment. The coordinator is responsible for managing the communication between the sensors and the MQTT broker and ensuring that the data is transmitted reliably and efficiently. We have incorporated several ZigBee sensors to monitor the environmental parameters within the rooms. The Zigbee sensors' specifications are summarized in Table 4.3.

The ZigBee coordinator is responsible for obtaining data from all these sensors and delivering it to the MQTT broker, which in turn sends it to the database.

Wifi Sensors In our case study, we used the Waspote WiFi Pro node equipped with the Waspote Gases Sensor Board PRO¹⁴. The board is

¹⁴Waspote Gases Sensor Board PRO. https://development.libelium.com/gases_pro_sensor_guide/gases-pro-sensor-board-calibrated



Figure 4.11: The touchscreen connected to the EDGE₁ Raspberry Pi.

customized with a Carbon Dioxide (CO₂) and a Carbon Monoxide (CO) gas sensor (see Table 4.3). As soon as the Waspnode node has new data from its sensors, it sends them to the MQTT broker, which forwards them to the components that request them, i.e., the Data Storage component.

Virtual Sensor and Data Visualization The ECN is connected to a small 7-inches touchscreen that is used to count the number of people in a room. Specifically, when someone enters/leaves the room and presses the “GET_IN” or “GET_OUT” button on the touchscreen, the count increases/decreases by one, and an MQTT message is sent to the MQTT Broker accordingly. The counting is displayed on the touchscreen in real-time, as shown in Figure 4.11. The touchscreen also displays real-time data coming from different sensors, including CO₂, temperature, humidity, and CO sensors. All this data is gathered by subscribing to the specific MQTT topic.

All the functions provided by the touchscreen are implemented through a Node-red program, whose diagram is shown in Figure 4.12. This figure illustrates three distinct flows. The one at the top of the figure is responsible for displaying the current date and time on the touchscreen. The middle one displays the two “GET_IN”/“GET_OUT” buttons mentioned before. The data regarding the number of people is sent to the MQTT broker and subsequently displayed on the screen. The third and final flow receives data from the MQTT broker, via an MQTT receiver, about sensors such as CO₂ concentration, temperature, humidity, light intensity, and CO level, which are then displayed on the screen.

Data Preprocessing The data stored in the database are in a raw format and are not directly suitable for model training. Moreover, in our case study, different types of devices were used, including ZigBee, WiFi, and virtual sensors: the collection of heterogeneous data may pose several challenges, such as uncertainty, redundancy, and missing values, making it difficult for the model to provide accurate forecasts. For these reasons, the data, before being used for training the LSTM model, must go through a preprocessing stage to be cleaned and normalized. Hence, the Data Preprocessing com-

4.3. MULTI OCCUPANCY PREDICTION IN BUILDINGS WITH MULTI-LAYER FL

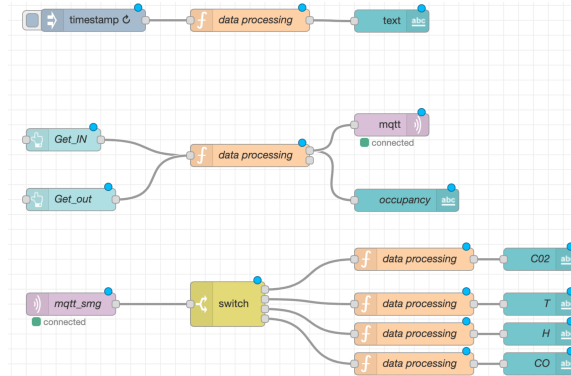


Figure 4.12: The realized Node-Red application.

ponent takes as input the raw data of the database and outputs a dataset that can be used for the model training. In our case study, the Data Pre-processing component has been purposely designed to deal with missing values, execute data normalization, and make data transformation. These operations are detailed in the following.

Missing values The data retrieved from the database can have missing values, which can have a significant impact on the accuracy of the model, since they can lead to biased or incomplete results during data analysis. Several techniques are available to deal with missing values, such as imputation, deletion, or modeling-based methods. In our case, the Last Observation Carried Forward (LOCF) technique [119] was implemented, which is a statistical method in which a missing value is replaced by the previously observed one.

Data Normalization Standardization is needed to enable consistent comparison among different data sources. In particular, data normalization techniques need to be used. Indeed, training an LSTM model with a normalized dataset leads to better results in terms of accuracy. In our case, the Standard Scalar Normalization [120] is the chosen technique, as it guarantees data consistency and enhances model accuracy. Standard scalar normalization scales the dataset’s features to have zero mean and unit variance.

Data Transformation Since we are dealing with a multivariate time series problem, visualizing the data can be challenging, particularly when there are numerous variables. To address this issue, this component transforms our dataset into a 3D structure [samples, observations, features] where:

- A *sample* is a row of data of the 3D structure;
- for each sample, there is a number of *observations* in the past, which

are used to forecast the target variable; the time interval between two consecutive observations is hereafter named as `Time_Step`.

- an *observation* consists of a number of *features*, which are the values measured by the sensors at the specific observation, according to the pre-processed data.

Now, the preprocessed dataset is ready for use and can be used to train the model. To perform forecasting for room occupancy estimation we exploited the LSTM model, which was specifically designed to work with time-series data.

LSTM Model Training and Occupancy Prediction The proposed approach involves the use of multivariate time series to evaluate the current occupancy number in a room and predict such occupancy for the future time steps. To train the LSTM model, sensor data was used as input, while the target output of the model is the occupancy counter N -observations-ahead, where, in this specific case, N has been chosen as 10. The LSTM Model Training component is trained to predict three classes, namely, no presence, presence of one person, and presence of more than one person. More details about the adopted LSTM model are provided in the next section.

The Occupancy Prediction component takes care of making occupancy predictions by using the previously trained LSTM model. In particular, occupancy prediction involves forecasting the number of people who are likely to be present in a given space in the future.

Model Aggregator This component, implemented at the Building level, is devoted to the aggregation of different trained LSTM models according to the *FedAvg* [113] algorithm. We chose FedAvg, rather than other algorithms for the merging of LSTM models, such as *FedProx* [121] and *FedPso* [122], since it is the most common in literature. The comparison among these algorithms is out of the scope of this paper, and we postpone it to future works. It is worth noting that also the Cloud Aggregator component uses the FedAvg algorithm for the model merging.

4.3.4 Experimental Evaluation

After the proposal of our approach in subsection *The Reference Architecture* and the introduction of our case study in subsection *Case Study Implementation*, this section shows some experimental results. Since the rooms are equipped with different sets of sensors, the first step of the experimentation aimed at understanding which is the best set of sensors to perform presence forecasting. To this purpose, we calculated the correlation matrix, reported in Figure 4.13, which involves all the sensors deployed in the IoT Laboratory, i.e., the room equipped with the largest variety of

4.3. MULTI OCCUPANCY PREDICTION IN BUILDINGS WITH MULTI-LAYER FL

sensors. A correlation matrix shows how strongly the pairs of variables are related in a dataset. If we have a set of variables X_1, X_2, \dots, X_n , the correlation matrix R is an $n \times n$ matrix where each entry r_{ij} is the correlation coefficient between X_i and X_j and is given by the following formula:

$$r_{ij} = \frac{\text{cov}(X_i, X_j)}{\sqrt{\text{var}(X_i) \cdot \text{var}(X_j)}} \quad (4.1)$$

Here, $\text{cov}(X_i, X_j)$ represents the covariance between X_i and X_j , and $\text{var}(X_i)$ and $\text{var}(X_j)$ are the variances of X_i and X_j respectively. The general form of the correlation matrix R is:

$$R = \begin{bmatrix} 1 & r_{12} & \dots & r_{1n} \\ r_{21} & 1 & \dots & r_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ r_{n1} & r_{n2} & \dots & 1 \end{bmatrix} \quad (4.2)$$

Each r_{ij} in the matrix represents the correlation coefficient between X_i and X_j , with $-1 \leq r_{ij} \leq 1$. The diagonal elements (r_{ii}) are always 1 because a variable is perfectly correlated with itself. When looking at the correlation matrix (Figure 4.13), it is clear that the most useful sensors to infer and predict the value of occupancy, set manually through the virtual sensor of Figure 4.10, and represented by the *Occupancy Counter* field in Figure 4.13, are the light (illuminance), the CO₂, and the PIR (passive infrared) sensors. Part of the collected dataset is available at https://sobigdata.d4science.org/catalogue-sobigdata?path=/dataset/multi-sensor_dataset_of_environmental_conditions_in_smart_office. For this reason, in the following, without losing the generality of the approach and for a faster development of a prototype of the proposed system, we will consider only the data coming from the IoT Laboratory, which is the only room containing all the needed sensors. Such data will be divided as to represent data coming from different rooms.

In the following subsections, we will show (i) the setup of the system realized for testing purposes, (ii) the evaluation metrics used, and (iii) a set of results for assessing the effectiveness of our approach.

Setup

The Multi-Layer Hierarchical FL training and prediction tasks have been realized by using *Python*¹⁵ (Release 3.7.0) and a set of libraries, among which *TensorFlow*¹⁶ (version 2.8.2), *TensorFlow Federated*¹⁷ (version 0.16.1), and *Pandas* (version 1.3.5)¹⁸. As explained above, the considered part of the

¹⁵Python 3.7.0. <https://www.python.org/downloads/release/python-370/>

¹⁶TensorFlow Homepage. <https://www.tensorflow.org/>

¹⁷TensorFlow Federated. <https://www.tensorflow.org/federated>

¹⁸pandas - Python Data Analysis Library. <https://pandas.pydata.org/>

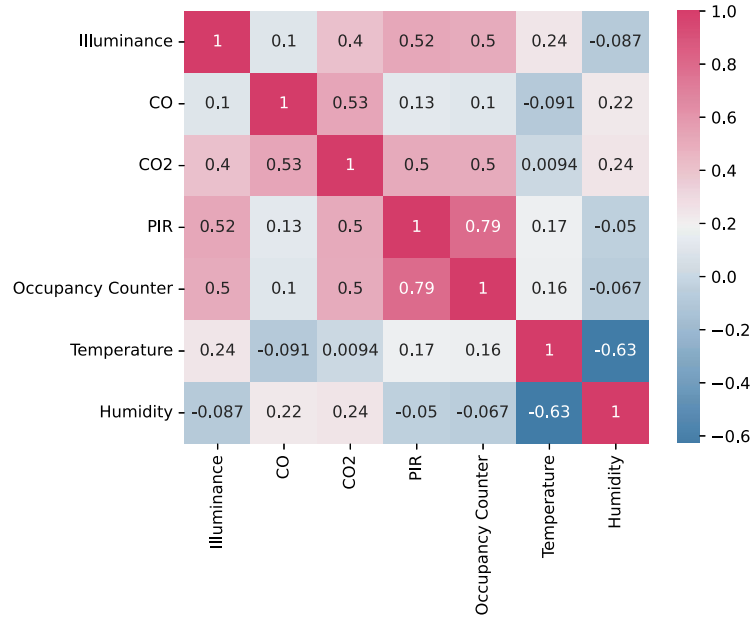


Figure 4.13: Correlation matrix of the sensors deployed in the IoT Laboratory.

dataset consists of the data coming from the light, the CO₂, and the PIR sensors of the IoT Laboratory.

In particular, we collected two datasets: the first has been obtained by gathering data from November 1st to November 10th and from November 21st to November 27th; the second one has been collected between October 1st and October 14th. These datasets present data collected only during the working hours and one row (observation) per minute, and thus the Time_step is set to sixty seconds. Hereafter, the second dataset will be used as the testset. Such a testset, together with information on the room’s actual occupancy (Occupancy Counter), is drawn in Figure 4.14.

Regarding the first dataset, it has been divided into 11 parts: 10 parts are used to represent data coming from ten different rooms (each one with its own ECN) of the considered building, supposed to be located as five rooms per floor, and are used as the training set for the models calculated at each ECN. The eleventh part is used in all the ECNs as validation set.

We separated each dataset into two parts: features and targets. For features, we chose CO₂, occupancy and illuminance data since, as explained before, they exhibit the highest correlation with our target variable, which is the occupancy number. Successively, we performed standard scalar normalization, which is a useful preprocessing technique that can enhance the performance and stability of models, and make their interpretation easier. Our LSTM model was trained using such normalized features. The model

4.3. MULTI OCCUPANCY PREDICTION IN BUILDINGS WITH MULTI-LAYER FL

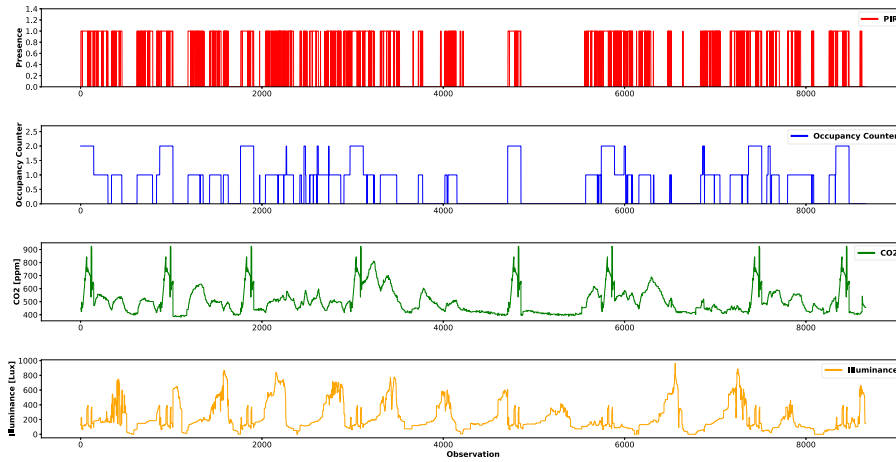


Figure 4.14: The Testset used in the implemented scenario.

consisted of an input LSTM layer of 32 units, one Dropout layer, another LSTM layer of 16 units, one more Dropout layer, and a final Dense layer with a single output. In defining the LSTM model, we used the tangent hyperbolic (\tanh) function [123] as the activation function for the LSTM memory cell, as well as for the output gate and the candidate state. This helps to ensure that the output of the model is within a bounded range and that the model can learn to distinguish between positive and negative inputs. Additionally, we have used the Root Mean Square Propagation (RMSprop) optimizer [124], which calculates an exponentially weighted moving average of the square of the gradients. This moving average is used to normalize the gradient during training, which helps to prevent the learning rate from oscillating too much and provides a smoother convergence towards the optimal solution. Regarding the Multi-Layer Hierarchical Federated Learning parameters, we used the following values: the number of *LowLevelRounds* (or *epochs*) was set to 10, the number of *MidLevelRounds* was set to 1, and the number of *HighLevelRounds* was also set to 10.

The model was trained to give as output the occupancy prediction 10 Time_steps ahead (i.e., 10 minutes ahead) by using the previous 60 observations of data as input (i.e., 60 minutes in the past). Since the output of our LSTM model is a real value, to furnish the final occupancy prediction values we used two thresholds, set to 0.5 and 1.5, and divided the predictions into three sub-classes: *occupancy equal to zero*, *to one*, and values *greater than one*, named respectively as *occupancy class 0*, *occupancy class 1*, and *multi-occupancy-class*.

Through the execution of the algorithm introduced in subsection *The Reference Architecture*, at each room, the ECN calculates the LSTM model for the room, and each EAN, one per floor, aggregates the models computed

in the rooms of its floor. Finally, the Cloud Aggregator (CA) merges the global model that will be disseminated to all the nodes of both building floors. In this way, even though a room cannot perform the training, as it misses the touchscreen required for assessing the true occupancy value, also this room will receive a model to run.

Evaluation Metrics

The proposed approach has been assessed by exploiting both quantitative and categorical metrics. In particular, to evaluate the error of predicting the exact number of people occupying a room, this study uses the *Mean Square Error (MSE)* and the *Mean Absolute Error (MAE)*. The study also evaluates the approach in predicting the previously defined three classes of occupancy, namely *occupancy class 0*, *occupancy class 1*, and *multi-occupancy-class*, by exploiting the well-known *Precision*, *Recall*, *F1 score*, and *accuracy* metrics. These metrics measure the degree of deviation between the predicted values and the actual data, providing an overview of the model's overall performance. In the following, we define them mathematically:

$$MSE = \frac{1}{n} \sum_{i=1}^n (x_i - y_i)^2 \quad (4.3)$$

$$MAE = \frac{1}{n} \sum_{i=1}^n |x_i - y_i| \quad (4.4)$$

$$Precision = \frac{TP}{TP + FP} \quad (4.5)$$

$$Recall = \frac{TP}{TP + FN} \quad (4.6)$$

$$F1 = \frac{2 * Precision * Recall}{Precision + Recall} \quad (4.7)$$

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (4.8)$$

where x_i and y_i represent the actual and predicted values at time i , respectively, and n denotes the total number of time steps. The acronyms TP , TN , FP , and FN stand for True Positive, True Negative, False Positive, and False Negative, respectively. Smaller MSE and MAE values indicate a smaller deviation between the predicted and actual values, indicating a more accurate prediction. *Precision*, *Recall*, *Accuracy*, and *F1 score* parameters closer to 1 indicate fewer errors and a more precise prediction. Finally, the study also considers *Macro Precision*, *Macro Recall*, and *Macro F1-score*, which are computed by averaging the Precision, Recall, and F1-score on the

4.3. MULTI OCCUPANCY PREDICTION IN BUILDINGS WITH MULTI-LAYER FL

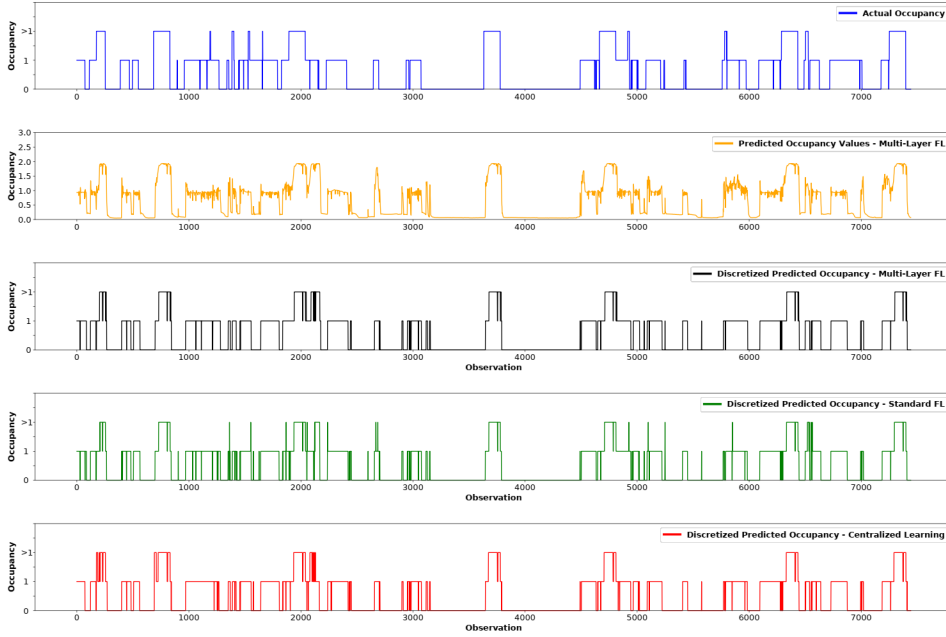


Figure 4.15: Actual vs Predicted occupancy for the considered case study.

prediction classes.

Results

In this section, we provide a discussion of the results gathered from the described data. In particular, the aim is twofold: (i) assessing the effectiveness of the proposed Multi-Layer Hierarchical FL approach in predicting occupancy in the considered use case; and (ii) comparing the performance of the proposed approach against a classical centralized approach and a standard, single layer, federated learning approach.

In Figure 4.15, the blue and the yellow lines represent, respectively, the actual occupancy data in the testset and the predicted value of occupancy obtained by our trained model. After applying the thresholds described above to the LSTM output (yellow line), we obtain the black line in Figure 4.15 that represents our model’s occupancy prediction, which includes all three categories - *Occupancy class 0*, *Occupancy class 1*, and *multi-occupancy class* - and closely matches the actual occupancy represented by the blue line.

As anticipated above, we evaluated the performance of the model using the *Mean Square Error* and the *Mean Absolute Error*. Our code was executed for 10 *HighLevelRounds*, and, for each *HighLevelRounds*, we executed 10 *LowLevelRounds*. We observed improvements on the results at each *HighLevelRound* iteration, as displayed in Table 4.4. Further iterations beyond

Table 4.4: Performance of the proposed model

	HighLevelRound									
	1	2	3	4	5	6	7	8	9	10
<i>Mean Square Error</i>	0.2413	0.2208	0.2179	0.2115	0.2031	0.1997	0.1978	0.1983	0.1970	0.1956
<i>Mean Absolute Error</i>	0.3460	0.3206	0.3135	0.3056	0.2981	0.2973	0.2935	0.2930	0.2895	0.2903

10 HighLevelRounds did not lead to significant additional improvements.

Our experimental results show that the proposed model achieves an overall *accuracy score* of 84.5% on our testset. To evaluate the performance of the final model on individual classes, we also computed the *F1 score*, together with *precision* and *recall*, as shown in Table 4.5. The *F1 score* for the *occupancy class 0*, the *occupancy class 1*, and the *multi-occupancy classes* reached the values 0.88 and 0.82 and 0.81, respectively. These values confirm that the model performed well for all three classes. In terms of *precision*, the model achieved a score of 0.88 for the *occupancy class 0*, 0.79 for the *occupancy class 1*, and 0.89 for the *multi-occupancy class*. The *recall* score for the *occupancy class 0* was 0.87, for the *occupancy class 1* was 0.84, and for the *multi-occupancy class* was 0.73. Finally, the last row of Table 4.5 shows the whole macro average values of the metrics introduced so far.

A comparative analysis was also performed considering other two methodologies, namely Discretized Predicted Occupancy - Standard FL (DPO - Standard FL) and Discretized Predicted Occupancy - Centralized Learning (DPO - Centralized Learning), which evaluate all the three classes of occupancy: *occupancy class 0*, *occupancy class 1*, and *multi-occupancy class*. DPO - Standard FL uses a single-layer approach to train the model across various clients, followed by aggregation to determine the optimal weights. In the DPO - Centralized Learning approach, the whole training set is available in the Cloud, where also the model training takes place. The results of these two approaches are shown in Figure 4.15 with green and red lines. The related evaluation metrics, including *Accuracy*, *Macro Precision*, *Macro Recall*, and *Macro F1-score*, are reported in Table 4.6. DPO - Centralized Learning achieved the highest *accuracy* of 0.850, followed by DPO - Multi-Layer FL 0.845, and finally, DPO - Standard FL 0.831. *Macro Precision* analysis favored DPO - Standard FL. *Macro Recall* measurement showed equal performances for DPO - Multi-Layer FL and DPO - Centralized Learning. Additionally, the *Macro F1-score* assessment indicates superior performance for DPO - Multi-Layer FL, followed by DPO - Centralized Learning and DPO - Standard FL.

4.3. MULTI OCCUPANCY PREDICTION IN BUILDINGS WITH
MULTI-LAYER FL

Table 4.5: Evaluation matrix table for the considered case study

Occupancy classes	<i>F1-score</i>	<i>Precision</i>	<i>Recall</i>	<i>Accuracy</i>
<i>Occupancy class 0</i>	0.88	0.88	0.87	0.88
<i>Occupancy class 1</i>	0.82	0.79	0.84	0.83
<i>Multi-occupancy class</i>	0.81	0.89	0.73	0.81
Macro avg	0.845	0.850	0.820	0.845

Table 4.6: Evaluation metrics computed for the comparison analysis of different methods. Bold values highlight the best method for a given measure.

Methods	<i>Accuracy</i>	<i>Macro Precision</i>	<i>Macro Recall</i>	<i>Macro F1 Score</i>
DPO - Centralized Learning	0.850	0.845	0.820	0.842
DPO - Standard FL	0.831	0.865	0.786	0.816
DPO - Multi-Layer FL	0.845	0.850	0.820	0.845

*CHAPTER 4. NOVEL APPROACHES FOR OCCUPANCY
PREDICTION IN COGNITIVE BUILDINGS*

Chapter 5

A Novel Approach for Activity Recognition in Cognitive Buildings

This chapter presents a detailed overview of the novel proposed approach for activity recognition, emphasizing preprocessing and novel methodology to recognize and classify occupant complex activities accurately. It also highlights the case studies conducted for validation, details the experimental setup, and summarizes the simulation outcomes that demonstrate the effectiveness of this approach.

5.1 UWB Radar-based Activity Recognition in Buildings

UWB radar technology is capable of monitoring, observing, and recognizing occupants' activities/movements even when they are not within direct line of sight or even on the other side of the wall. However, there are still challenges associated with this technology, i.e., it generates a vast quantity of raw data. This data may increase the complexity of data processing and present challenges in extracting and learning critical features for model training. Recent research has tried to address this problem by reducing the dimensionality of data using statistical techniques such as principal component analysis (PCA), linear discriminant analysis (LDA), and uniform manifold approximation and projection (UMAP) [106]. However, these statistical methods could not work effectively or might ignore some crucial features when it comes to complex activity recognition, especially when the activities involve movement or share a common space. Also, many studies in the literature process images generated from UWB radar data rather than directly using the raw data. This technique adds complexity to the data processing and leads to models concentrating on specific image areas,

ignoring crucial temporal and spatial patterns in the radar data.

To overcome the limitations mentioned above and advance the current literature, we propose a novel approach based on UWB radar that uses a Hybrid Deep Learning model for Activity Recognition (HDL4AR) in CBs environments. Our proposed approach combines LSTM with 1D-CNN, in which LSTM delves into the global temporal dimension while 1D-CNN concentrates on the spatial dimension, adeptly managing deep features to diminish spatial characteristics. This approach was conceived to improve the accuracy and reduce the computational complexity of the model with respect to the others in the literature. Specifically, the proposed approach can be summarized as follows:

5.1.1 The Proposed Approach

This section provides an overview of the proposed approach. Specifically, we will delve into (i) the way to preprocess the data and (ii) the proposed hybrid deep learning model, which includes deep feature extraction from raw data and time-related feature exploration. The proposed approach is explained in detail in the following subsections.

Data Preprocessing

Data preprocessing aims to prepare and clean raw data to make it suitable for analysis and modeling in machine learning applications. This crucial step in the data analysis pipeline helps improve data quality, addresses missing or inconsistent information, and enhances the performance and interpretability of the models. We have labeled each activity with the respective names and merged all individual activities into one CSV file.

Normalization After this, we have employed a two-step approach to normalize our dataset, which consists of two main steps aimed at scaling and standardizing the features of our dataset [12]. To normalize each value within a dataset's column, we divide each value by the maximum absolute value for that specific column. In the following, a mathematical representation of this step is reported:

$$x'_{j,i} = \frac{x_{j,i}}{\max |x_{j,i}|}, \quad \text{for } j = 1, 2, \dots, N; \quad i = 1, 2, \dots, S; \quad (5.1)$$

This normalization scales the values in each column to $[-1, 1]$. The second step involves further refinement of the dataset using a standard scalar normalization. It ensures that the distribution of our data is aligned with a standard normal distribution with a standard deviation of 1 and a mean of 0. Following is a mathematical representation of this step:

$$x''_{j,i} = \frac{x'_{j,i} - \mu_i}{\sigma_i}, \quad \text{for } j = 1, 2, \dots, N; \quad i = 1, 2, \dots, S; \quad (5.2)$$

where μ_i represents the mean and σ_i represents the standard deviation of the i^{th} column after the first normalization step, respectively. After applying the two-step normalization process, the dataset is transformed into a new normalized form, X_{norm} , where each original data point $x_{j,i}$ in X_{raw} has been replaced with its normalized counterpart $x''_{j,i}$. The normalized dataset can be represented as:

$$X_{\text{norm}} \in \mathbb{R}^{N \times S}, \text{ with } x''_{j,i}, j = 1, 2, \dots, N; \quad i = 1, 2, \dots, S; \quad (5.3)$$

Data augmentation We use a two-step data augmentation approach to bring variability and diversity to our input data and enhance the model's robustness [125]. For this, we introduced a Gaussian noise, also known as jittering, to our data. This noise originates from a normal distribution with a mean of zero and a specified standard deviation. After jittering, we applied a random time shift to our data, allowing for minor delays or shifts in recording, thereby ensuring the model's ability to accommodate misaligned input sequences. We shift the time series by a random number within a specified range, thereby maintaining the temporal structure.

The proposed model for activity recognition

When an individual participates in an activity, the radar consistently generates a signal that interacts with the surroundings and the people nearby. If the radar detects no movement, the signal remains constant and does not change. When individuals or objects are in motion, the radar signals that bounce back experience changes, leading to noticeable changes in the radar bins depending on the distance between the radar and the target. Typically, as objects approach the radar, the beginning bins show a higher amplitude of change; however, as they move further away, the bin's amplitude decreases. It is crucial to observe these particular variations in radar bins since they hold significant information. Minor bin variations may result from environmental noise, but these are frequently negligible. A fundamental challenge while utilizing this technology is the identification and extraction of the significantly changed bins that carry crucial information. To accomplish this, our proposed HDL4AR approach combines LSTM networks with a 1D-CNN. This approach can extract deep features from the dataset using its hybrid learning model, facilitating a comprehensive examination of individual activity recognition.

Temporal Analysis of Data Given an augmented data input sequence $X_{\text{aug}} \in \mathbb{R}^{N \times S}$, where N is the number of sequences (batch size) and S is the sequence

length, the LSTM processes this data, updating its cell states and generating the output. The steps that our data passes through to get to the result are shown below.

Forget Gate A forget gate can be represented by f_t , which tells us what information has to be discarded from the previous cell state C_{t-1} . It applies a sigmoid function to the weighted sum of the previous hidden state h_{t-1} and the current input data x_t . The forget gate is computed as follows:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \quad (5.4)$$

Input Gate and Candidate Cell State The input gate determines what information should be stored in the cell and can be represented by i_t . In this process, a sigmoid layer determines what values should be updated, and a hyperbolic tangent layer determines what new values should be added to the state. The following formulas are used to calculate them:

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad (5.5)$$

$$g_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_t) \quad (5.6)$$

Update Cell State In the LSTM, the update cell state C_t represents the memory. It is updated by using the forget gate output f_t and the input gate output i_t , which determine what to discard and what to add from the candidate values. Using the following formula, we can compute the updated cell's state:

$$C_t = f_t \times C_{t-1} + i_t \times g_t \quad (5.7)$$

Output Gate The output gate o_t decides what information from the cell state C_t should be exposed as the hidden state h_t . It is computed using a sigmoid function over the weighted sum of the previous hidden state h_{t-1} and the current input data z_t . It is calculated as follows:

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \quad (5.8)$$

$$H = h_t = o_t \times \tanh(C_t) \quad (5.9)$$

The variables f_t , i_t , and g_t represent the values of the forget gate, input gate, and candidate cell state at a specific time t in the input data sequence. The parameters W_f , W_i , and W_c are the weight matrices associated with the forget gate, input gate, and candidate cell state, respectively. Similarly, the parameters b_f , b_i , and b_c are the bias terms corresponding to the forget gate,

input gate, and candidate cell state. The function σ represents the sigmoid activation function, and \tanh represents the hyperbolic tangent activation function.

Spatial Analysis of Data The output of the LSTM layer is then forwarded to a 1D-CNN layer, which applies convolutional filters across the sequence of data, capturing local patterns within the high-level features produced by the LSTM.

For a given filter k in the 1D-CNN, the convolution operation at a position t in the sequence is:

$$c_t^k = f \left(\sum_{i=0}^{K-1} W_k[i] \cdot \mathbf{X}'_{[t+i]} + b_k \right) \quad (5.10)$$

Where:

- c_t^k is the feature produced by filter k at position t .
- f is a nonlinear activation function, and here it is ReLU.
- $W_k[i]$ are the weights of filter k at position i within its receptive field.
- K is the size of the convolutional kernel.
- b_k is the bias term for filter k .

This hybrid approach leverages LSTM's ability to capture long-term time dependencies across the sequence and 1D-CNN's efficiency in extracting critical features. The modified data is then forwarded to other layers, which will be explained in the next subsection.

Model Architecture

The overall presented hybrid deep learning model architecture has been depicted in Figure 5.1.

Our model starts with an LSTM layer using 64 memory units to capture and analyze sequential data patterns. This layer's primary role is to maintain temporal information, safeguarding crucial time-dependent patterns in the data. A convolutional layer with 256 filters and a kernel size of 3 processes the sequence after it has passed through the LSTM layer. This layer operates by analyzing local features. We utilize the rectified linear unit (ReLU) activation function to enable the extraction of intricate non-linear features. Next, we add a max-pooling layer to gradually reduce the data size, lowering its complexity and enhancing computational performance. Utilizing a pool size of 2, the pooling method helps reduce overfitting and enhance the model's robustness. We use a 128-unit fully connected layer with the

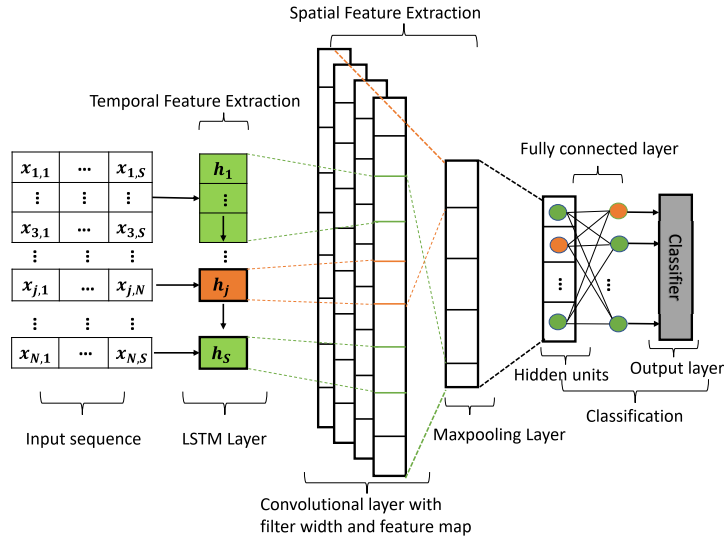


Figure 5.1: HDL4AR model architecture.

ReLU activation function to facilitate the extraction of complex hierarchical representations, thereby aiding in learning higher-level abstractions from the data. Finally, the output layer is equipped with the *softmax* activation function [126], facilitating the accurate classification of the input into one activity class. During the training process, the model optimizes its internal parameters using the Adam optimizer, a popular and efficient variant of stochastic gradient descent.

5.1.2 Case study for activity recognition in buildings

This subsection demonstrates the practical implementation of the case study we conducted at the IoT Laboratory in the ICAR-CNR headquarters in Italy. More in particular, we will explain how we deployed UWB radar technology to gather data from various individuals and the diverse activities we carried out. In addition, we will offer a detailed explanation of the simulation configuration we employed for activity recognition, ensuring a thorough comprehension of our experimental method.

Dataset and Building Description

As written above, we have installed the UWB radar and conducted all the experiments in the ICAR-CNR IoT laboratory. The laboratory is equipped with a variety of computers, an oval table, and four chairs, as shown in Figure 5.2. In order to maintain the secrecy of the objects kept in the laboratory, we intentionally blurred some parts of Figure 5.2. All the activities were performed by the participants around the table that are kept in the central



Figure 5.2: The IoT laboratory in ICAR-CNR, Italy. In the foreground, the used UWB radar is shown in its working position.

part of the IoT laboratory. We obtained data from a group of 22 individuals who were in excellent health status. Among them, 16 were male and 6 were female, covering a wide range of ages. Table 5.1 provides comprehensive explanations of the activities in which each participant was involved. Part of the collected dataset is available at https://sobigdata.d4science.org/catalogue-sobigdata?path=/dataset/uwb_radar_dataset_of_human_activity_detection_in_smart_office. To protect the confidentiality, we categorized individuals based on their age into three groups: 18–25, 26–35, and 36–45. The following is a list of the activities in which the participants took part:

- *Sitting at position 1:* Monitoring individuals sitting at the left of the table in the office with a laptop;
- *Sitting at position 2:* Monitoring individuals sitting at the right of the table in the office with a laptop;
- *Standing still:* Monitoring individuals standing still without engaging in any specific activity;
- *Walking:* Monitoring individuals walking in the office environment;
- *Cleaning:* Measuring the activity of office floor cleaning with a mop;

- *Presenting to screen*: Gathering data on individuals conducting presentations in front of the big screen on the left of the office;
- *Teaching to whiteboard*: Capturing data during writing or drawing activities on the whiteboard in the central part of the office;
- *Gun firing*: Monitoring an individual who has a gun in his hands and is moving inside the office;
- *Laying still on the floor*: Monitoring an individual laying on the office floor with no movements;
- *Laying on the floor with movements*: Measuring the activity of an individual who is laying on the office floor while moving.

We also collected data from the Empty room, gathering the data with no participant in the office. In the following (and also in Table 5.1) Empty room is considered as an activity.

All participants performed each activity for 90 seconds to ensure a uniform dataset. The experiments were conducted at a frame rate of ten frames per second. Each participant received detailed information about the activity they would perform, their role, and the data collection process. They signed the consent forms so that compliance and transparency could be maintained. Thus, the collected dataset was obtained in an ethically approved and carefully managed manner.

Setup

A UWB radar has been installed and connected through USB to a PC equipped with a 64-bit Intel Core i7 quad-core processor and 16 GB of RAM. The PC collected all the data from the UWB radar by using MATLAB¹, which is supporting our specific device, and stored all the data gathered into CSV files.

A data collection exercise was conducted in July 2023 during regular business hours. Following the collection phase, the data gathered was manually merged so creating 11 distinct CSV files, each representing a specific activity. The total dataset was so partitioned into training, validation, and test sets. In particular, 70% of the dataset was used for the training set, 20% for the test set, and another 10% for the validation set. Through this partitioning, we ensure data distribution is appropriate for comprehensive model training, testing, and performance evaluation.

To analyze our data, we used *Python*² (Release 3.7.0) with *TensorFlow*³ (version 2.8.2), and *Pandas* (version 1.3.5)⁴. We also employed the *scikit-*

¹<https://www.mathworks.com/solutions/data-acquisition.html>

²Python 3.7.0. <https://www.python.org/downloads/release/python-370/>

³TensorFlow Homepage. <https://www.tensorflow.org/>

⁴pandas - Python Data Analysis Library. <https://pandas.pydata.org/>

Table 5.1: Engagement of all participants in the data acquisition process.

Participant ID	M/F	Age range	Sitting position1	Sitting position2	Standing still	Walking	Cleaning	Gun firing	Presenting to screen	Teaching to whiteboard	Laying still on the floor	Laying on the floor with movements	Empty room
001	M	36-45	x	x	x	x	x	x	x	x	x	x	.
002	M	26-35	x	x	x	x	x	x	x	x	x	x	.
003	M	26-35	x	x	x	x	x	x	x	x	x	x	.
004	M	26-35	x	x	x	x	x	x	x	x	x	x	.
005	M	26-35	x	x	x	x	x	x	x	x	x	x	.
006	M	26-35	x	x	x	x	x	x	x	x	x	x	.
007	F	26-35	x	x	x	x	x	x	x	x	.	.	.
008	M	26-35	x	x	x	x	x	x	x	x	x	x	.
009	M	36-45	x	x	x	x	x	x	x	x	.	.	.
010	F	26-35	x	x	x	x	x	x	x	x	.	.	.
011	M	26-35	x	x	x	x	x	.	x	x	.	.	.
012	F	26-35	x	x	x	x	x	x	x	x	.	.	.
013	F	36-45	x	x	x	x	x	x	x	x	.	.	.
014	F	26-35	x	x	x	x	x	.	x	x	.	.	.
015	M	36-45	x	x	x	x	x	x	x	x	.	.	.
016	M	26-35	x	x	x	x	x	x	x	x	x	x	.
017	M	26-35	x	x	x	x	x	x	x	x	x	x	.
018	M	26-35	x	x	x	x	x	x	x	x	x	x	.
019	F	26-35	x	x	x	x	x	x	x	x	x	.	.
020	M	26-35	x	x	x	x	x	x	x	x	x	x	.
021	M	26-35	x	x	x	x	x	x	x	x	x	x	.
022	M	18-25	x	x	x	x	x	x	x	x	x	x	.
No participant	N/A	N/A	x

*learn*⁵ library (version 0.24.1), which provided a comprehensive suite of tools for data preprocessing, feature selection, and machine learning algorithms. Moreover, we also installed two resource-intensive libraries *Principal Component Analysis (PCA)*⁶ and *Uniform Manifold Approximation and Projection (UMAP)*⁷, to facilitate a comparison of our hybrid deep learning model with various statistical approaches, enabling us to identify the best-performing method.

According to the proposed approach introduced in subsection *The Proposed Approach*, we have preprocessed the training, validation, and test sets by doing data normalization. Subsequently, we built our proposed model by combining the introduced LSTM+1D-CNN and proceeded to train it. More specifically, the model underwent training for a total of 100 epochs. The model achieved its optimal performance after 38 epochs. We initially set the fine-tuning for both the LSTM and CNN models using standard practices and then refined them via a systematic tuning process. We begin with standard settings for important parameters, including the number of units in LSTM layers and the filter counts and kernel sizes in CNN layers. We used cross-validation to select the optimal parameters that enhanced our model's performance. To prevent overfitting, we closely observed the validation loss and accuracy during the training process.

Finally, the obtained model has been evaluated by applying the metrics introduced in the following subsection. The obtained results and a discussion about them are reported in subsection *Discussion and Results*.

Model Evaluation

To measure the effectiveness of our model, we have calculated the following metrics, which will be explained below and evaluated in the next section: accuracy, precision, recall, F1 score, and confusion matrix.

Accuracy Accuracy is a metric used to measure a model's overall correctness. In other words, it represents how many instances were correctly predicted (both positive and negative) relative to how many instances were in the dataset. Its mathematical representation is as follows:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

TP stands for True Positives. Using our particular multiclass approach, a true positive is when the model correctly predicts an observation as belonging to a specific class of activity, and it appears to be in that class. FP

⁵scikit-learn - Machine Learning, <https://scikit-learn.org/>

⁶PCA in scikit-learn, <https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.PCA.html>

⁷UMAP in scikit-learn, <https://pypi.org/project/umap-learn/>

denotes False Positive. It occurs when the model incorrectly predicts an observation as belonging to a particular category of activity when, in reality, it does not. A TN stands for a True Negative. A true negative occurs when the model correctly predicts the sample does not belong to a particular class, and it does not. FN stands for False Negatives. When a sample is incorrectly predicted as not belonging to a particular class but belongs to that class, the model produces a false negative.

Precision The precision of a classification task is an important metric, especially when the positive class (i.e., the class of interest) is of particular importance. It is the ratio of true positives to the sum of true positives and false positives. The precision of a model (P) can be calculated as follows:

$$P = \frac{TP}{TP + FP}$$

Recall In classification tasks, recall also called sensitivity or true positive rate, measures how well a model can identify all the relevant class instances. It is the ratio of true positives to the sum of true positives and false negatives.

In mathematical terms, recall (R) is calculated as:

$$R = \frac{TP}{TP + FN}$$

F1-Score The F1-Score is a metric that combines precision and recall into a single value. It provides a balance between these two metrics and is particularly useful when dealing with imbalanced classes or when both false positives and false negatives are important.

Mathematically, the F1-Score is calculated as the harmonic mean of precision (P) and recall (R):

$$F1 = 2 \times \frac{P \times R}{P + R}$$

Confusion Matrices A confusion matrix for a multi-class classification problem with N classes is represented as an $N \times N$ matrix. It is a tabular representation that helps visualize the performance of a classification model. Each row corresponds to the actual class, and each column corresponds to the predicted class. Given a confusion matrix C with N classes:

$$C = \begin{bmatrix} C_{11} & C_{12} & \cdots & C_{1N} \\ C_{21} & C_{22} & \cdots & C_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ C_{N1} & C_{N2} & \cdots & C_{NN} \end{bmatrix}$$

The elements in this matrix are defined as follows: C_{ij} is the number (or percentage) of instances of class i that were predicted as class j .

In this matrix, there is the possibility to see True Positives (TP), for class i , C_{ii} represents the number of instances correctly predicted as class i ; False Positives (FP), for class i , C_{ij} where $j \neq i$ represents the number of instances incorrectly predicted as class i ; True Negatives (TN), the sum of all the elements in the matrix except for row i and column i for all i represents the number of instances correctly predicted as not belonging to class i ; False Negatives (FN), the sum of all the elements in row i except for C_{ii} represents the number of instances belonging to class i but incorrectly predicted as not belonging to class i .

5.1.3 Discussion and results for activity recognition

This subsection aims to show the simulation results. In particular, we will evaluate our approach using the evaluation metrics introduced above, namely accuracy, precision, recall, F1 score, and confusion matrices. Moreover, we will compare our results with recent literature studies incorporating statistical and machine-learning approaches.

In particular, we will examine our approach compared to those in the following:

- *EDA-LSTM* based Classification: an Enhanced Discriminant Analysis-LSTM (EDA-LSTM) based classification approach is used by the authors in [106], where EDA is a combination of PCA and linear discriminant analysis (LDA) used for the dimension of data reduction, followed by LSTM for temporal analyses and classification of activities. We performed some experiments by using this approach on our dataset by varying some parameters, and we chose to use for our comparison the model with the best results.
- *UMAP-LSTM* based Classification: as described in [127], the authors used UMAP for reducing data dimensions. Using this method, we have reduced the dimensions of raw data generated by our UWB radar. Several parameters, already introduced in [127], were fine-tuned, and the configuration with the best results was chosen for comparison.
- Attention-based LSTM-CNN called *ALSTM-CNN* Classification: the article in [108] employs attention-based LSTM and CNN for monitoring elderly people's activities. For comparison purposes, we used the above-mentioned method on our dataset, combining 1D-CNNs with LSTM.
- Attention-based GRU-CNN called *AGRU-1D CNN* Classification: instead of the LSTM, the attention-based GRU-1D-CNN employs the

GRU (Gated Recurrent Unit) to handle time series data. For comparison purposes, we also used this approach on our dataset.

- *Transformer-based Classification*: Since the last two methods we chose for comparison are attention-based and they produce good results, we also employed transformers, which have multi-head self-attention mechanisms available, on our dataset for comparison purposes.

Table 5.2 and Figure 5.3 presents the evaluation metrics for each method, including UMAP-LSTM, EDA-LSTM, AGRU-1D CNN, ALSTM-1D CNN, and Transformer-based classification, along with our proposed approach, HDL4AR. Statistical methods, including UMAP-LSTM, exhibit lower performance because this approach reduces the dimension based on manifold methods and also reinforces the linearity relationship, which complicates the approach of lowering the data’s dimension as a whole. EDA-LSTM performs better because it combines the advantages of both PCA and LCA for data dimension reduction and classification, which contributes to its improved performance. Compared to the above statistical methods, attention-based deep learning algorithms perform better in understanding intricate, non-linear patterns within data and extracting critical features. However, they exhibit lower performance compared to our proposed approach because of the introduction of attention layers in their models, which add complexity to the models and make it difficult to capture intricate patterns within the data as a whole effectively. Furthermore, they have utilized the data in an image-based format that focuses on certain parts of the image, thereby ignoring less critical features within the data. Training attention-based models requires significantly more computational resources, consumes longer time on a simple PC, and sometimes requires a GPU to process it. Including attention layers in these models enables selecting some important features while ignoring less significant ones that still contribute to the overall outcome, although to a lesser degree. These features substantially affect evaluation metrics such as accuracy, precision, recall, and F1 score. Our proposed approach outperforms other methods by combining the advantages of LSTM and 1D-CNN models, in which LSTM analyzes time-related features while 1D-CNN manages deep features to reduce spatial characteristics.

Table 5.3 presents the accuracy scores for each activity in the respective cell, along with the corresponding approaches. Each approach demonstrates different accuracy scores. More specifically, static activities where no movement is involved exhibit excellent accuracy scores. UMAP-LSTM exhibits the lowest performance when it comes to activities that require physical movement. UMAP performs poorly in movement-based activities. Following UMAP-LSTM, EDA-LSTM has the second-lowest accuracy. AGRU-LSTM ranks third, with slightly higher accuracy for each activity than the statistical approaches. ALSTM-1D-CNN outperforms the other algorithms stated

Table 5.2: Evaluation metrics computed for different approaches and comparison with our approach.

Methods	Accuracy	Precision	Recall	F1 Score
UMAP-LSTM	0.8629	0.851	0.860	0.8620
EDA-LSTM	0.9209	0.9198	0.9209	0.9201
AGRU-1D CNN	0.9380	0.92	0.9200	0.9200
ALSTM-1D CNN	0.9461	0.9400	0.9391	0.9393
Transformer-based	0.9546	0.9433	0.9430	0.9426
Our approach	0.9842	0.9840	0.9850	0.9852

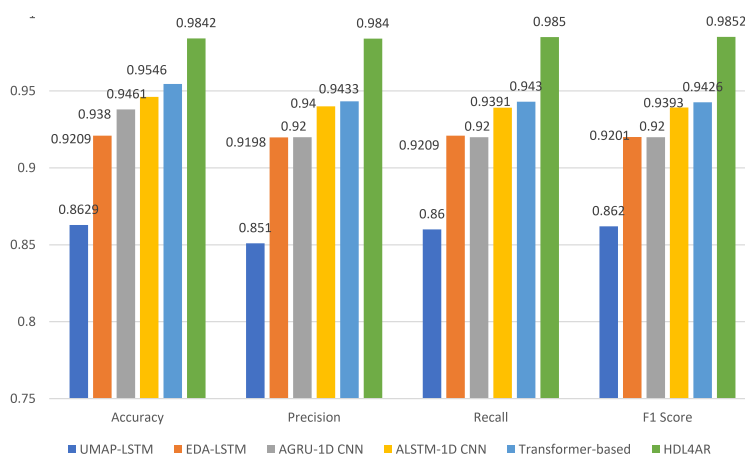


Figure 5.3: Graphical visualization of the Accuracy, Precision, Recall, and F1 Score for the compared approaches.

above in terms of accuracy. Transformer-based classification outperforms all other approaches in terms of accuracy for each activity. Upon analyzing the accuracy of each activity, it is obvious that our proposed model consistently obtains the highest accuracy scores.

We have also evaluated the confusion matrices for all the considered approaches. As we are recognizing multiple activities, we have considered measuring a multi-class confusion matrix to assess the effectiveness of each class by comparing the actual labels with the predicted labels. Figure 5.4 depicts the confusion matrix of UMAP-LSTM. This approach shows poor performance in most activities when comparing actual labels to predicted activities. The confusion matrix for EDA-LSTM can be seen in Figure 5.5, which exhibits marginally superior performance compared to UMAP-LSTM. The confusion matrices for deep learning models, such as AGRU-1D CNN, ALSTM-1D CNN, and transformer-based classification approaches, can be shown in Figure 5.6, Figure 5.7, and Figure 5.8 respectively. These approaches show better performance as compared to the statistical approaches, but they still require improvements. Figure 5.9 illustrates our proposed approach confusion matrix, which uses a hybrid model. LSTM focuses on the temporal dimension by carefully examining time-related data, while 1D-CNN focuses on the spatial dimension by effectively using deep features to reduce spatial characteristics. By incorporating the advantages of each model, this hybrid approach accurately recognizes the majority of activities, with true labels closely matching the predicted labels. Furthermore, Figures 5.10 and 5.11 show a graph illustrating the model's accuracy and loss for the training and validation phases, respectively. We trained the model for 100 epochs and observed that it reached convergence after 38 epochs. Finally, the 3D scatter plot shown in Figure 5.12 demonstrates the classification of each activity within our proposed approach, confirming the model's overall higher performance.

The computational complexity of each model is evaluated, as summarized in Table 5.4. We trained each model for varying numbers of epochs, as shown in the corresponding rows with the relevant methods; some methods required more epochs to achieve the same performance calculated in Table 5.2. The UMAP-LSTM model training was run for 400 epochs. This model concluded its training in a short window period of 615.66 seconds; however, it was the most computationally costly in terms of RAM memory consumption, with a total of 9491.1 MB, which is the highest RAM consumption among all the methods. The EDA-LSTM model was executed for the same number of epochs. This model is not computationally expensive in terms of training time and consumes less memory than all methods, taking only 5018.9 MB of RAM during training. All other hybrid deep learning models were trained for 100 epochs. AGRU-LSTM exhibited the most extended training duration among all methods, requiring 38,042.78 seconds and using 8,472.0 MB of RAM. ALSTM-1D CNN was the second-most time-consuming

Table 5.3: Accuracy’s score of each activity for all the compared classifiers.

Activities	Sitting position 1	Sitting position 2	Walking	Standing still	Cleaning	Presenting to screen	Teaching to whiteboard	Gun firing	Laying on the floor with movements	Laying still on the floor	NO Participant
UMAP-LSTM	1.0	0.9975	0.7007	0.9655	0.6730	0.8756	0.9022	0.7324	0.8683	0.9809	0.9796
EDA-LSTM	0.9814	0.9975	0.8204	0.9286	0.6991	0.9652	0.9340	0.7765	0.9854	0.9904	1.0
AGRU-1D CNN	1.0	0.9975	0.8554	0.9877	0.7678	0.9502	0.9682	0.9147	1.0	1.0	1.0
ALSTM-1D CNN	0.9973	0.9975	0.9202	0.9852	0.8081	0.9527	0.9584	0.8912	1.0	0.9952	1.0
Transformer-based	1.0	1.0	0.8953	0.9901	0.8697	0.9801	0.9756	0.8912	0.9902	0.9713	1.0
Our approach	1.0	1.0	0.9576	1.0	0.9526	0.9925	0.9804	0.9765	1.0	0.9952	1.0

5.1. UWB RADAR-BASED ACTIVITY RECOGNITION IN BUILDINGS

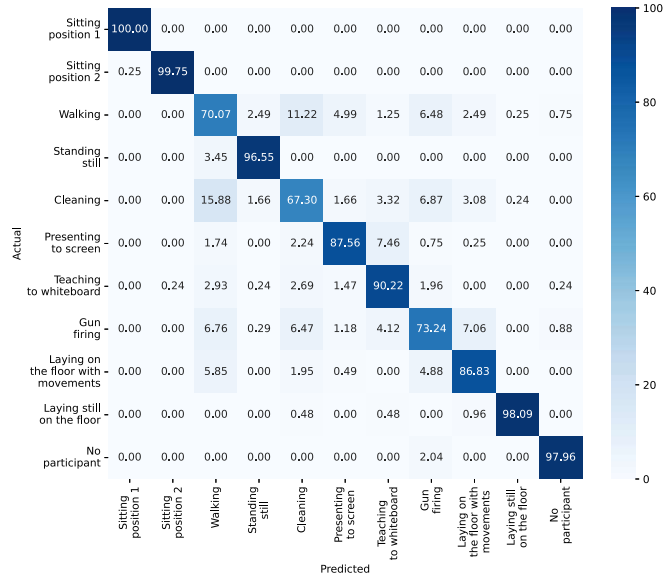


Figure 5.4: Confusion Matrix of UMAP-LSTM based Classification.

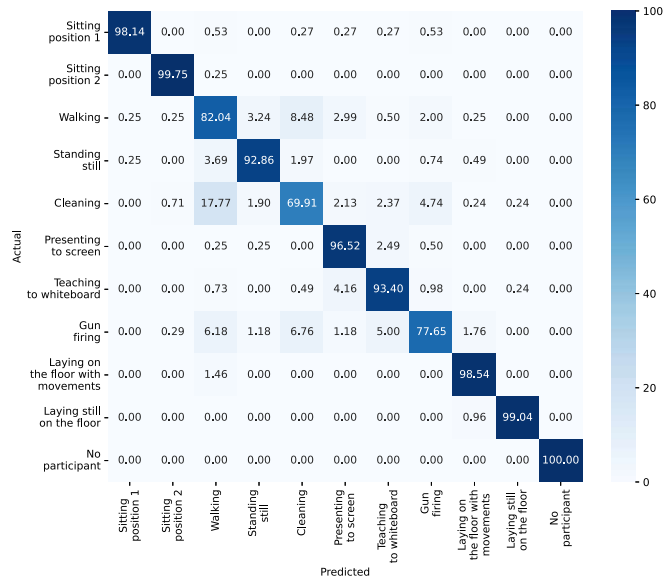


Figure 5.5: Confusion Matrix of EDA-LSTM based Classification.

CHAPTER 5. A NOVEL APPROACH FOR ACTIVITY RECOGNITION IN COGNITIVE BUILDINGS

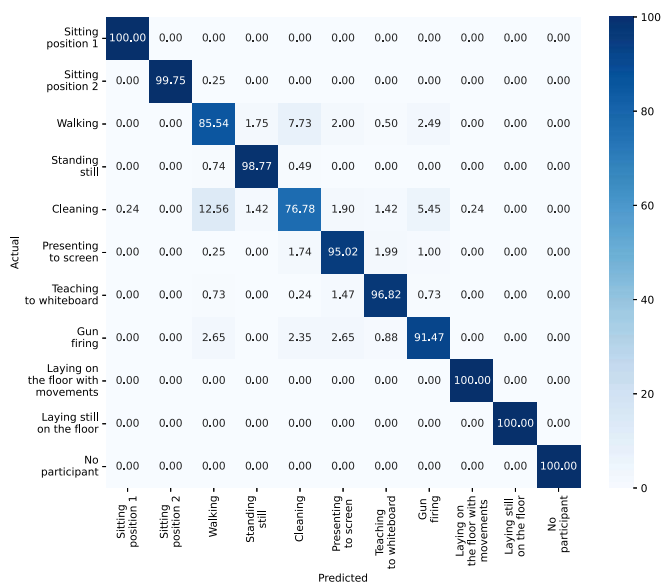


Figure 5.6: Confusion Matrix of AGRU-1D CNN based Classification.

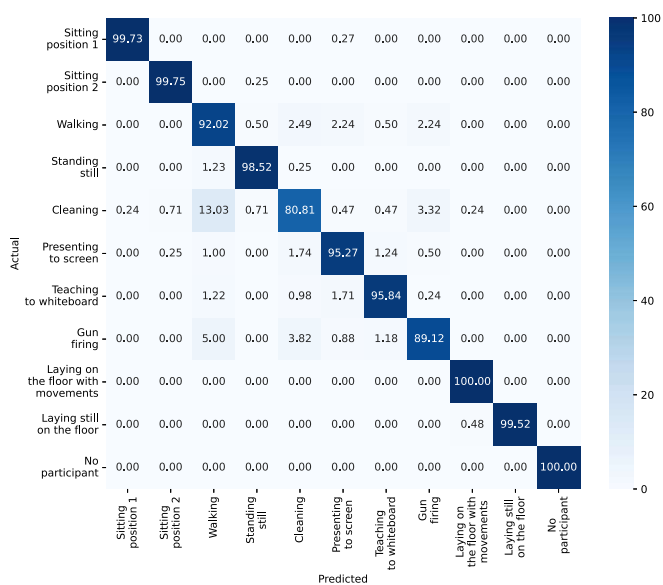


Figure 5.7: Confusion Matrix of ALSTM-1D CNN based Classification

5.1. UWB RADAR-BASED ACTIVITY RECOGNITION IN BUILDINGS

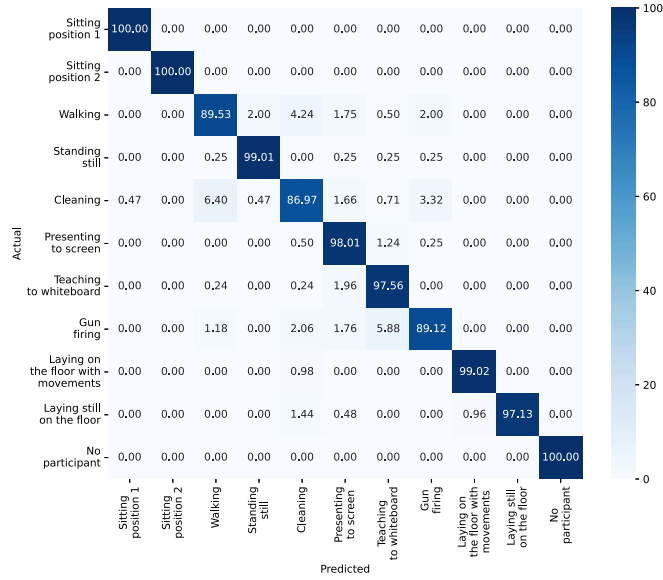


Figure 5.8: Confusion Matrix of Transformer-based Classification.

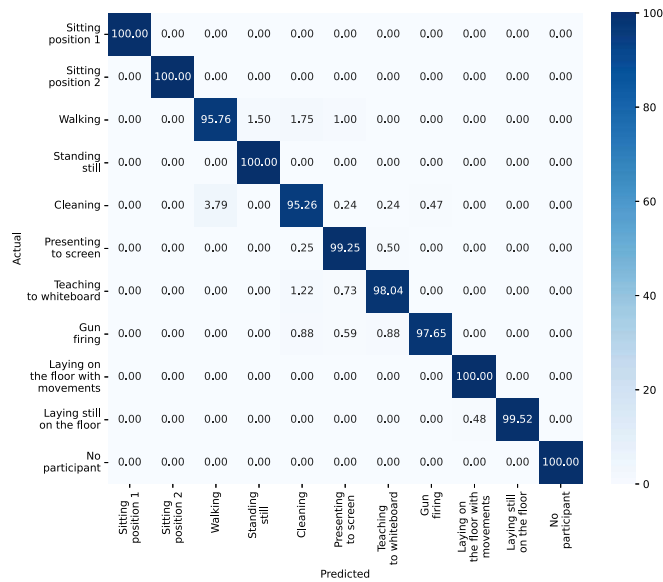


Figure 5.9: Confusion Matrix of our approach.

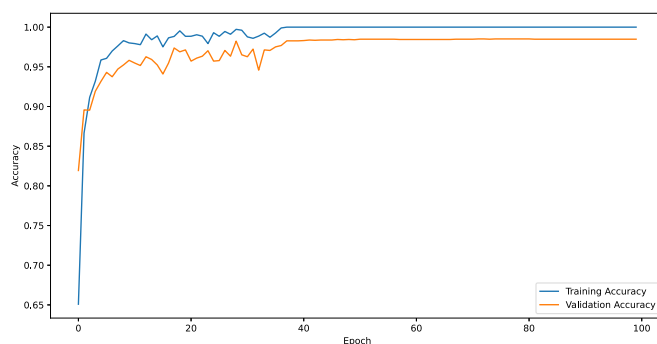


Figure 5.10: Accuracy graph of the training and validation sets.

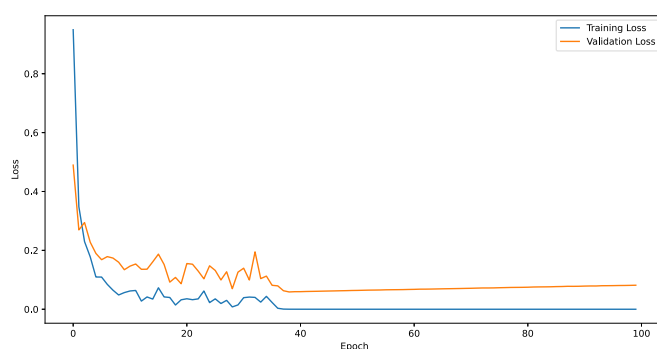


Figure 5.11: Loss graph of the training and validation sets.

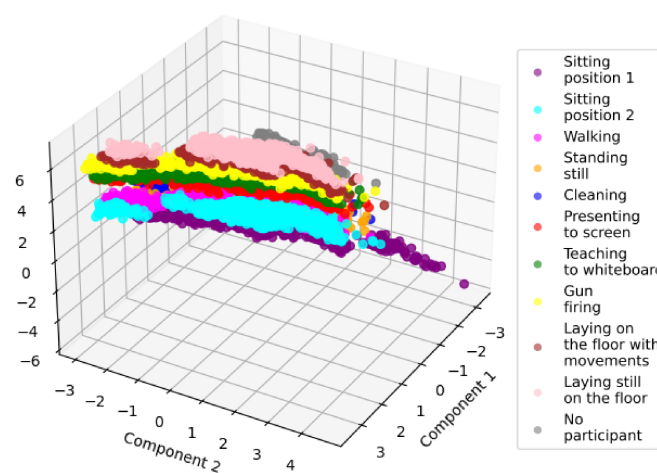


Figure 5.12: 3D scattering visualisation of each activity for our proposed approach.

5.1. UWB RADAR-BASED ACTIVITY RECOGNITION IN BUILDINGS

Table 5.4: Computational complexities are computed for different approaches and comparisons with our approach.

Methods	No. of epochs	Training time (seconds)	RAM memory usage during training (MB)
UMAP-LSTM	400	615.66	9491.1
EDA-LSTM	400	564.23	5018.9
AGRU-1D CNN	100	38042.78	8472.0
ALSTM-1D CNN	100	16070.10	8296.9
Transformer-based	100	2251.26	7672.2
HDL4AR	100	456.80	7336.0

model, requiring 16,070.1 seconds to complete its training. This model used 8296.9 MB of RAM during training. The Transformer-based model required 2,251.26 seconds for training and used 7,672.2 MB of RAM. HDL4AR had the lowest training duration among all models, completing it in only 456.80 seconds, using 7,336 MB of RAM.

Finally, we calculated the amount of memory needed to execute our trained model. In particular, we monitored the RAM occupied by the Python process on the computer running it for one hour during execution. We noted that the occupied memory was at its maximum 1218.4 megabytes. This means that the model can also be executed on edge devices [1] such as Raspberry Pi. This allows for dramatically improved privacy since the data used during the model execution can be acquired to the edge and deleted as soon as it is used.

Discussion on the robustness of our model

To understand the robustness of our model, a new data collection experiment was conducted, with the aim of creating a new test set to be evaluated on the already trained model. In such an experiment, we moved, replaced, and/or added some furniture in the smart laboratory to create a new configuration. An occupant participated and performed all the activities for 30 seconds to collect the new data. In this new configuration, shown in Figure 5.13, the row of desks that were previously attached to the wall (as seen on the left side of Figure 5.2) has been moved 70 centimeters away from the wall. We relocated the big screen that was on the desk on the left and put it on the stand on the right. We also moved the small desk that was on the right. A black metal and glass box has been added on the right to host servers. During all the data collection, one person was always sitting on the left of the room. Additional minor modifications to the smart laboratory have also been made on the back side of the UWB radar, which are not visible in the figure.

On this new test set, our trained model attained an overall accuracy of 65.9%. It's worth noting that by integrating part of the new data collected



Figure 5.13: The IoT laboratory in ICAR-CNR with the new configuration.

into the training dataset, we can retrain our model with an increased accuracy of about 90%. This methodology can be used to perform transfer learning activities [128], which will be better evaluated in future work.

Conclusion, Publications, and Future Work

6.1 Conclusion

This study introduced some novel approaches to understand and predict occupants' behaviors inside Cognitive Buildings. More specifically, these approaches focus on determining and predict whether a space is occupied and the number of people present. Moreover, they aim to understand what activities are performed. To prove the proposed approaches, some case studies have been implemented in a smart office environment, at ICAR-CNR in Rende, Italy.

Firstly, we proposed an approach based on FL and LSTM for binary occupancy prediction in several building rooms. Moreover, the proposed approach shows, through some simulation results, the importance of FL in the binary occupancy predictions for the spaces in which the training of a specific model was not performed. Some simulation experiments have shown the effectiveness of the proposed approach by achieving an overall accuracy of 94.5%. We worked on extending binary occupancy prediction to predict multi-occupancy as it has more abilities to optimize energy consumption in such buildings' environments. We introduced a data-driven approach combining IoT sensors, ML, Edge computing, and standard FL to predict multi-occupancy in buildings. The proposed approach is applied to real-time data collected at the ICAR-CNR headquarters (Rende, Italy) to gain practical insights, enabling a comprehensive understanding of the occupancy patterns and dynamics within a multi-occupant environment. Such data are gathered through a purposely developed IoT-based network. The approach also introduced a versatile design template for developing real distributed systems for occupancy prediction. Moreover, we conducted a comparative study by analyzing several FL configurations obtained by varying combi-

nations of HLRs and epochs. HLRs denote the instances when a model is aggregated at a high level, often in the Cloud, while epochs refer to training rounds involving clients at the network's edge. We have also employed the multi-layer approach proposed to reduce the energy consumed for communication by avoiding continuous data transmission to the cloud. The proposed approach achieved a global accuracy of 84.5% in forecasting 10-minute-ahead room multi-occupancy while preserving data privacy.

Secondly, we have developed an approach for the occupant's activity recognition inside CBs. The proposed approach utilized non-wearable UWB radar technology to detect occupants' activity inside smart buildings. This approach used hybrid learning models, a combination of LSTMs and 1D-CNNs, to analyze the temporal and spatial components of the data and extract deep features from it. The LSTM focuses on the temporal analysis of the data; it ensures an in-depth analysis of the time-related features, whereas the 1D CNN addresses the spatial elements, reducing the spatial characteristics and extracting deep features from them. A case study was conducted in real-time by collecting data from 22 healthy individuals involved in 10 different activities at the headquarters of a smart laboratory in ICAR-CNR, Italy. Furthermore, our study has been benchmarked against existing statistical approaches and other hybrid deep learning models that have recently been developed. As a result of the above comparison, we achieved the highest overall accuracy score of 98.53%.

6.2 Publication Related to this Thesis

The research work related to this thesis has resulted in the following publications: Some of them are published in peer-reviewed journals, book chapters, and international IEEE conferences.

In the following, a brief description of each publication is provided.

6.2.1 Journal Publications

- A Hybrid Deep Learning Model for UWB Radar-Based Occupants Activity Recognition in Energy-Efficient Buildings [24].

Irfanullah Khan, Edoardo Serra, Antonio Guerrieri, Giandomenico Spezzano, "A Hybrid Deep Learning Model for UWB Radar-Based Occupants Activity Recognition in Energy-Efficient Buildings. (Accepted to the Internet of Things Journal, Elsevier, on 26th, November 2024.)
<https://doi.org/10.1016/j.iot.2024.101458>

This paper proposes a novel approach that uses a hybrid deep-learning model to recognize complex human activities based on data collected from ultra-wideband radar technology. Our approach, called Hybrid Deep Learning Model for Activity Recognition (HDL4AR), includes long-short-term memory and a one-dimensional convolutional neural

network. We deploy a real-time case study by collecting data from 22 participants involved in 10 diverse activities at the headquarters of the ICAR-CNR in the IoT Laboratory, Italy. Moreover, we conducted a comprehensive benchmark of the HDL4AR approach against various statistical techniques and other deep learning models recently introduced in the literature. Results show that our proposed approach outperformed conventional methods and achieved an impressive accuracy of 98.42%.

- Leveraging distributed AI for multioccupancy prediction in cognitive buildings [23].

Irfanullah Khan, Franco Cicirelli, Emilio Greco, Antonio Guerrieri, Carlo Mastroianni, Luigi Scarcello, Giandomenico Spezzano, and Andrea Vinci. "Leveraging distributed AI for multioccupancy prediction in cognitive buildings." *Internet of Things (2024)*: 101181.

<https://doi.org/10.1016/j.iot.2024.101181>

This paper proposes an innovative approach for multi-occupancy prediction in cognitive buildings, incorporating a multi-layer hierarchy for Federated Learning, the utilization of IoT devices at the Edge, the implementation of long short-term memory neural network models, and the exploitation of Edge Computing. The approach also introduces a versatile design template for developing real distributed systems for occupancy prediction. The proposed approach uses a distributed paradigm to safeguard data privacy so that the collected data is used to train separate local deep learning models, which are then merged in the Cloud. The paper validates the approach by providing a preliminary prototype realized at ICAR-CNR, Rende, Italy, and presents a performance analysis, which shows that the occupancy is predicted with an 84.5% accuracy.

- Occupancy Prediction in IoT-Enabled Smart Buildings: Technologies, Methods, and Future Directions [14].

Irfanullah Khan, Ouarda Zedadra, Antonio Guerrieri, and Giandomenico Spezzano. "Occupancy Prediction in IoT-Enabled Smart Buildings: Technologies, Methods, and Future Directions." *Sensors 24, no. 11 (2024)*: 3276.

<https://doi.org/10.3390/s24113276>

This paper provides a comprehensive overview of recent advancements in Internet of Things (IoT) technologies that have been designed and used for the monitoring of indoor environmental conditions within buildings. Using these technologies is crucial to gathering data about the indoor environment and determining the number and presence of occupants. Furthermore, the paper critically examines both the strengths and limitations of each technology in predicting occupant

behavior. In addition, it explores different methods for processing this data and making future occupancy predictions. Moreover, we highlight some challenges, such as determining the optimal number and location of sensors and radars, providing a detailed explanation and detailed insight into these challenges. Furthermore, the paper explores possible future directions, including the security of occupants' data and the promotion of energy-efficient practices such as localizing occupants and monitoring their activities within the building.

6.2.2 Book Chapter

- Occupancy Prediction in Buildings: State of the Art and Future Directions [6].

Irfanullah Khan, Emilio Greco, Antonio Guerrieri, and Giandomenico Spezzano. "Occupancy Prediction in Buildings: State of the Art and Future Directions." *Device-Edge-Cloud Continuum: Paradigms, Architectures and Applications (2023)*: 203-229.

https://doi.org/10.1007/978-3-031-42194-5_12

This chapter aims to provide an overview of the prominent research conducted on occupancy forecasting in building environments. Initially, we have examined the key monitoring methods, based on Internet of Things technologies, employed for assessing presence in buildings. Subsequently, we have delved into some machine learning and deep learning algorithms utilized for predicting occupancy. Finally, we have explored the possible potential future directions in this field

6.2.3 Conferences

- Human-Centric Based Energy Efficiency and Comfort Optimization in Cognitive Buildings: A Review [21].

Irfanullah Khan, A. Stephen McGough, Antonio Guerrieri, Giandomenico Spezzano, "Human-Centric Based Energy Efficiency and Comfort Optimization in Cognitive Buildings: A Review (to appear in the proceedings of the 22nd IEEE International Conference on Pervasive Intelligence and Computing (PICom 2024), November 2024).

<https://doi.org/10.1109/PICom64201.2024.00020>

This paper provides a review of the most recent progress in using occupant behaviors to empower cognitive buildings to make autonomous choices, thereby enhancing security, energy efficiency, and comfort. As a novelty, it investigates how detailed information on building occupancy, including if the building is occupied, the number of people, their activity, and their precise positions within the space, could improve and optimize decision-making procedures. Furthermore, it also explores the challenges associated with the implementation of these

concepts and discusses the potential future research directions that could enhance these concepts.

- Occupancy Prediction in Multi-Occupant IoT Environments leveraging Federated Learning [22].

Irfanullah Khan, Flavia C. Delicato, Emilio Greco, Massimo Guarascio, Antonio Guerrieri, and Giandomenico Spezzano. "Occupancy Prediction in Multi-Occupant IoT Environments leveraging Federated Learning." In *2023 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress (DASC/PiCom/CBDCCom/CyberSciTech)*, pp. 0036-0043. IEEE, 2023.

<https://doi.org/10.1109/DASC/PiCom/CBDCCom/Cy59711.2023.10361520>

This paper introduces a data-driven approach combining Internet of Things sensors, Machine Learning, Edge computing, and Federated Learning to predict multi-occupancy in buildings. The proposed approach is used on real data from the ICAR-CNR IoT Laboratory in order to extract insights into occupancy patterns within a multi-occupant environment. Finally, a comparative analysis conducted by varying Federated Learning configurations demonstrates the robustness of the solution.

- Occupancy prediction in buildings: An approach leveraging LSTM and federated learning [53].

Irfanullah Khan, Antonio Guerrieri, Giandomenico Spezzano, and Andrea Vinci. "Occupancy prediction in buildings: An approach leveraging LSTM and federated learning." In *2022 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress (DASC/PiCom/CBDCCom/CyberSciTech)*, pp. 1-7. IEEE, 2022.

<https://doi.org/10.1109/DASC/PiCom/CBDCCom/Cy55231.2022.9927838>

This paper proposes an approach based on Federated Learning (FL) and Long Short-Term Memory neural networks for binary occupancy prediction in several rooms of a building. It also shows how FL helps in the occupancy predictions for the spaces in which the training of a specific model was not already performed. Some simulation experiments have shown the effectiveness of the proposed approach.

6.2.4 Invited talk at International Conference

- Dynamic Occupant Behaviors: Paving the Way for Energy-efficient and Privacy-Preserving Smart Buildings.

Irfanullah Khan, Antonio Guerrieri, Giandomenico Spezzano, “Dynamic Occupant Behaviors: Paving the way for Energy-efficient and Privacy-Preserving Smart Buildings (invited talk at 2nd International Conference on Recent Trends in Statistics and Data Analytics in NUST, Islamabad, Pakistan).

6.3 Future Work

Although this thesis has contributed to the growing research literature on using cutting-edge technologies for occupancy prediction and activity recognition in CBs, there is still a gap for improvement in both occupancy prediction and occupants’ activity recognition. Regarding occupancy prediction, integrating additional sensors, such as smart thermostats, might provide more accurate temperature and light intensity monitoring. Furthermore, our research concentrated only on indoor environmental parameters; however, further investigations might include outside elements, such as meteorological conditions and seasonal aspects, which directly affect interior environments. Digital twins could enhance our methodology by monitoring energy optimization and occupant comfort in real-time. Further experiments could explore the transferability of learned models across similar buildings to optimize energy and comfort management. In the context of activity recognition, we may consider incorporating additional UWB radars and expanding the scope of activities to include those performed outside the line of sight, such as those behind walls. We aim to further develop our research by focusing on recognizing human activities and tracking people’s movements or locations inside buildings, which will provide essential input for future CBs. Furthermore, we want to explore transfer learning methods in activity recognition to adjust our model to new environments with little retraining. Finally, we need to generalize our model to provide a flexible solution that applies to diverse situations without requiring much more adjustment in the surrounding environment.

Bibliography

- [1] F. Cicirelli, A. Guerrieri, A. Vinci, and G. Spezzano, *IoT Edge Solutions for Cognitive Buildings*. Springer.
- [2] M. Amadeo, F. Cicirelli, A. Guerrieri, G. Ruggeri, G. Spezzano, and A. Vinci, “When edge intelligence meets cognitive buildings: The cogito platform,” *Internet of Things*, vol. 24, p. 100908, 2023.
- [3] S. D’Oca, T. Hong, and J. Langevin, “The human dimensions of energy use in buildings: A review,” *Renewable and Sustainable Energy Reviews*, vol. 81, pp. 731–742, 2018.
- [4] L. Scarcello, F. Cicirelli, A. Guerrieri, C. Mastroianni, G. Spezzano, and A. Vinci, “Pursuing energy saving and thermal comfort with a human-driven drl approach,” *IEEE Transactions on Human-Machine Systems*, pp. 1–13, 2022.
- [5] D. Chwieduk, “Towards sustainable-energy buildings,” *Applied energy*, vol. 76, no. 1-3, pp. 211–217, 2003.
- [6] I. Khan, E. Greco, A. Guerrieri, and G. Spezzano, “Occupancy prediction in buildings: State of the art and future directions,” *Device-Edge-Cloud Continuum: Paradigms, Architectures and Applications*, pp. 203–229, 2023.
- [7] W. Guo and M. Zhou, “Technologies toward thermal comfort-based and energy-efficient hvac systems: A review,” in *2009 IEEE international conference on systems, man and cybernetics*, pp. 3883–3888, IEEE, 2009.

- [8] S. Colace, S. Laurita, G. Spezzano, and A. Vinci, “Room occupancy prediction leveraging LSTM: an approach for cognitive and self-adapting buildings,” in *IoT Edge Solutions for Cognitive Buildings - Technology, Communications and Computing* (F. Cicirelli, A. Guerrieri, A. Vinci, and G. Spezzano, eds.), pp. 197–219, Springer, 2023.
- [9] H. N. Rafsanjani, C. R. Ahn, and M. Alahmad, “A review of approaches for sensing, understanding, and improving occupancy-related energy-use behaviors in commercial buildings,” *Energies*, vol. 8, no. 10, pp. 10996–11029, 2015.
- [10] V. Oikonomou, F. Becchis, L. Steg, and D. Russolillo, “Energy saving and energy efficiency concepts for policy making,” *Energy policy*, vol. 37, no. 11, pp. 4787–4796, 2009.
- [11] T. Leephakpreeda, “Adaptive occupancy-based lighting control via grey prediction,” *Building and environment*, vol. 40, no. 7, pp. 881–886, 2005.
- [12] K. Bouchard, J. Maitre, C. Bertuglia, and S. Gaboury, “Activity recognition in smart homes using uwb radars,” *Procedia Computer Science*, vol. 170, pp. 10–17, 2020.
- [13] J. Virote and R. Neves-Silva, “Stochastic models for building energy prediction based on occupant behavior assessment,” *Energy and buildings*, vol. 53, pp. 183–193, 2012.
- [14] I. Khan, O. Zedadra, A. Guerrieri, and G. Spezzano, “Occupancy prediction in iot-enabled smart buildings: Technologies, methods, and future directions,” *Sensors*, vol. 24, no. 11, p. 3276, 2024.
- [15] N. E. Klepeis, W. C. Nelson, W. R. Ott, J. P. Robinson, A. M. Tsang, P. Switzer, J. V. Behar, S. C. Hern, and W. H. Engelmann, “The national human activity pattern survey (nhaps): a resource for assessing exposure to environmental pollutants,” *Journal of exposure science & environmental epidemiology*, vol. 11, no. 3, pp. 231–252, 2001.
- [16] Y. Fan and C. Fang, “Ghg emissions and energy consumption of residential buildings—a systematic review and meta-analysis,” *Environmental Monitoring and Assessment*, vol. 195, no. 7, p. 885, 2023.

- [17] C. Ionescu, T. Baracu, G.-E. Vlad, H. Necula, and A. Badea, “The historical evolution of the energy efficient buildings,” *Renewable and Sustainable Energy Reviews*, vol. 49, pp. 243–253, 2015.
- [18] R. Pacheco, J. Ordóñez, and G. Martínez, “Energy efficient design of building: A review,” *Renewable and sustainable energy reviews*, vol. 16, no. 6, pp. 3559–3573, 2012.
- [19] Y. Ding, W. Chen, S. Wei, and F. Yang, “An occupancy prediction model for campus buildings based on the diversity of occupancy patterns,” *Sustainable Cities and Society*, vol. 64, p. 102533, 2021.
- [20] I. Khan, A. Guerrieri, G. Spezzano, and A. Vinci, “Occupancy prediction in buildings: An approach leveraging lstm and federated learning,” in *2022 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress (DASC/PiCom/CBDCCom/CyberSciTech)*, pp. 1–7, IEEE, 2022.
- [21] I. Khan, A. Guerrieri, F. Cicirelli, A. S. McGough, and G. Spezzano, “Human-centric based energy and comfort optimization in cognitive buildings: A review,” in *2024 IEEE Conference on Pervasive and Intelligent Computing (PICom)*, pp. 98–104, IEEE, 2024.
- [22] I. Khan, F. C. Delicato, E. Greco, M. Guarascio, A. Guerrieri, and G. Spezzano, “Occupancy prediction in multi-occupant iot environments leveraging federated learning,” in *2023 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress (DASC/PiCom/CBDCCom/CyberSciTech)*, pp. 0036–0043, 2023.
- [23] I. Khan, F. Cicirelli, E. Greco, A. Guerrieri, C. Mastroianni, L. Scarcello, G. Spezzano, and A. Vinci, “Leveraging distributed ai for multi-occupancy prediction in cognitive buildings,” *Internet of Things*, vol. 26, p. 101181, 2024.

- [24] I. Khan, A. Guerrieri, E. Serra, and G. Spezzano, “A hybrid deep learning model for uwb radar-based human activity recognition,” *Internet of Things*, p. 101458, 2024.
- [25] H. T. Ebuy, H. Bril El Haouzi, R. Benelmir, and R. Pannequin, “Occupant behavior impact on building sustainability performance: A literature review,” *Sustainability*, vol. 15, no. 3, p. 2440, 2023.
- [26] F. S. Hafez, B. Sa’di, M. Safa-Gamal, Y. Taufiq-Yap, M. Alrifayy, M. Seyedmahmoudian, A. Stojcevski, B. Horan, and S. Mekhilef, “Energy efficiency in sustainable buildings: a systematic review with taxonomy, challenges, motivations, methodological aspects, recommendations, and pathways for future research,” *Energy Strategy Reviews*, vol. 45, p. 101013, 2023.
- [27] I. Khan, T. Zhang, X. Xu, S. Shan, A. Khan, and S. Ahmad, “Priority-based content dissemination in content centric vehicular networks,” in *2018 2nd IEEE Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC)*, pp. 2005–2009, IEEE, 2018.
- [28] K. Sayed and H. A. Gabbar, “Building energy management systems (bems),” *Energy conservation in residential, commercial, and industrial facilities*, pp. 15–81, 2017.
- [29] A. A. Aguilera, R. F. Brena, O. Mayora, E. Molino-Minero-Re, and L. A. Trejo, “Multi-sensor fusion for activity recognition—a survey,” *Sensors*, vol. 19, no. 17, p. 3808, 2019.
- [30] G. Chimamiwa, M. Alirezaie, F. Pecora, and A. Loutfi, “Multi-sensor dataset of human activities in a smart home environment,” *Data in Brief*, vol. 34, p. 106632, 2021.
- [31] B. Vidya and P. Sasikumar, “Wearable multi-sensor data fusion approach for human activity recognition using machine learning algorithms,” *Sensors and Actuators A: Physical*, vol. 341, p. 113557, 2022.
- [32] V. Singh, S. Singh, and P. Gupta, “Real-time anomaly recognition through cctv using neural networks,” *Procedia Computer Science*, vol. 173, pp. 254–263, 2020.

- [33] M. Straczekiewicz, P. James, and J.-P. Onnela, “A systematic review of smartphone-based human activity recognition methods for health research,” *NPJ Digital Medicine*, vol. 4, no. 1, p. 148, 2021.
- [34] H. Saha, A. R. Florita, G. P. Henze, and S. Sarkar, “Occupancy sensing in buildings: A review of data analytics approaches,” *Energy and Buildings*, vol. 188, pp. 278–285, 2019.
- [35] R. Lavanya, C. Murukesh, and N. Shanker, “Development of machine learning based microclimatic hvac system controller for nano painted rooms using human skin temperature,” *Journal of Electrical Engineering & Technology*, pp. 1–12, 2022.
- [36] M. Cheraghinia, A. Shahid, S. Luchie, G.-J. Gordebeke, O. Caytan, J. Fontaine, B. Van Herbruggen, S. Lemey, and E. De Poorter, “A comprehensive overview on uwb radar: Applications, standards, signal processing techniques, datasets, radio chips, trends and future research directions,” *arXiv preprint arXiv:2402.05649*, 2024.
- [37] V. Lafontaine, K. Bouchard, J. Maitre, and S. Gaboury, “Denoising uwb radar data for human activity recognition using convolutional autoencoders,” *IEEE Access*, 2023.
- [38] G. Moro, F. Di Luca, D. Dardari, and G. Frisoni, “Human being detection from uwb nlos signals: accuracy and generality of advanced machine learning models,” *Sensors*, vol. 22, no. 4, p. 1656, 2022.
- [39] N. Tekin, A. Acar, A. Aris, A. S. Uluagac, and V. C. Gungor, “Energy consumption of on-device machine learning models for iot intrusion detection,” *Internet of Things*, vol. 21, p. 100670, 2023.
- [40] M. Urban, R. Stewart, S. Basford, Z. Palmer, and J. Kaufman, “Estimating building occupancy: a machine learning system for day, night, and episodic events,” *Natural Hazards*, pp. 1–20, 2023.
- [41] J. Kim, J. Bang, A. Choi, H. J. Moon, and M. Sung, “Estimation of occupancy using iot sensors and a carbon dioxide-based machine learning model with ventilation system and differential pressure data,” *Sensors*, vol. 23, no. 2, p. 585, 2023.

- [42] S. Sra, S. Nowozin, and S. J. Wright, *Optimization for machine learning*. Mit Press, 2012.
- [43] K. Mehrotra, C. K. Mohan, and S. Ranka, *Elements of artificial neural networks*. MIT press, 1997.
- [44] M. V. Valueva, N. Nagornov, P. A. Lyakhov, G. V. Valuev, and N. I. Chervyakov, “Application of the residue number system to reduce hardware costs of the convolutional neural network implementation,” *Mathematics and Computers in Simulation*, vol. 177, pp. 232–243, 2020.
- [45] S. Dupond, “A thorough review on the current advance of neural network structures,” *Annual Reviews in Control*, vol. 14, pp. 200–230, 2019.
- [46] Y. Yu, X. Si, C. Hu, and J. Zhang, “A review of recurrent neural networks: Lstm cells and network architectures,” *Neural computation*, vol. 31, no. 7, pp. 1235–1270, 2019.
- [47] F. Cicirelli, A. Guerrieri, A. Mercuri, G. Spezzano, and A. Vinci, “IteMa: A methodological approach for cognitive edge computing iot ecosystems,” *Future Generation Computer Systems*, vol. 92, pp. 189–197, 2019.
- [48] G. Aloï, G. Fortino, R. Gravina, P. Pace, and C. Savaglio, “Simulation-driven platform for edge-based aal systems,” *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 2, pp. 446–462, 2021.
- [49] C. Savaglio and G. Fortino, “A simulation-driven methodology for iot data mining based on edge computing,” *ACM Trans. Internet Technol.*, vol. 21, mar 2021.
- [50] A. Metwaly, J. P. Queralta, V. K. Sarker, T. N. Gia, O. Nasir, and T. Westerlund, “Edge computing with embedded ai: Thermal image analysis for occupancy estimation in intelligent buildings,” in *Proceedings of the INTelligent Embedded Systems Architectures and Applications Workshop 2019*, pp. 1–6, 2019.
- [51] A. Bourechak, O. Zedadra, M. N. Kouahla, A. Guerrieri, H. Seridi, and G. Fortino, “At the confluence of artificial intelligence and edge

- computing in iot-based applications: A review and new perspectives,” *Sensors*, vol. 23, no. 3, 2023.
- [52] A. Alsalemi, Y. Himeur, F. Bensaali, and A. Amira, “Smart sensing and end-users’ behavioral change in residential buildings: An edge-based internet of energy perspective,” *IEEE Sensors Journal*, vol. 21, no. 24, pp. 27623–27631, 2021.
- [53] I. Khan, A. Guerrieri, G. Spezzano, and A. Vinci, “Occupancy prediction in buildings: An approach leveraging lstm and federated learning,” in *2022 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress (DASC/PiCom/CBDCCom/CyberSciTech)*, pp. 1–7, 2022.
- [54] M. Xia, D. Jin, and J. Chen, “Short-term traffic flow prediction based on graph convolutional networks and federated learning,” *IEEE Transactions on Intelligent Transportation Systems*, 2022.
- [55] A. Guzzo, G. Fortino, G. Greco, and M. Maggiolini, “Data and model aggregation for radiomics applications: Emerging trend and open challenges,” *Information Fusion*, vol. 100, p. 101923, 2023.
- [56] T. Vafeiadis, S. Zikos, G. Stavropoulos, D. Ioannidis, S. Krinidis, D. Tzovaras, and K. Moustakas, “Machine learning based occupancy detection via the use of smart meters,” in *2017 International Symposium on Computer Science and Intelligent Controls (ISCSIC)*, pp. 6–12, IEEE, 2017.
- [57] B. Yang, F. Haghighat, B. C. Fung, and K. Panchabikesan, “Season-based occupancy prediction in residential buildings using machine learning models,” *e-Prime - Advances in Electrical Engineering, Electronics and Energy*, vol. 1, p. 100003, 2021.
- [58] W. Wang, J. Chen, and T. Hong, “Occupancy prediction through machine learning and data fusion of environmental sensing and wi-fi sensing in buildings,” *Automation in Construction*, vol. 94, pp. 233–243, 2018.

- [59] S. Colace, S. Laurita, G. Spezzano, and A. Vinci, “Room occupancy prediction leveraging LSTM - An approach for cognitive and self-adapting buildings,” in *IoT Edge Solutions for Cognitive Buildings*. (F. Cicirelli, A. Guerrieri, G. Spezzano, and A. Vinci, eds.), ch. 9, Springer, 2022.
- [60] O. Bouhamed, M. Amayri, and N. Bouguila, “Weakly supervised occupancy prediction using training data collected via interactive learning,” *Sensors*, vol. 22, no. 9, 2022.
- [61] M. Khalil, S. McGough, Z. Pourmirza, M. Pazhoohesh, and S. Walker, “Transfer learning approach for occupancy prediction in smart buildings,” in *2021 12th International Renewable Engineering Conference (IREC)*, pp. 1–6, 2021.
- [62] Y. Jin, D. Yan, X. Kang, A. Chong, S. Zhan, *et al.*, “Forecasting building occupancy: A temporal-sequential analysis and machine learning integrated approach,” *Energy and Buildings*, vol. 252, p. 111362, 2021.
- [63] S. H. Ryu and H. J. Moon, “Development of an occupancy prediction model using indoor environmental data based on machine learning techniques,” *Building and Environment*, vol. 107, pp. 1–9, 2016.
- [64] Y. Peng, A. Rysanek, Z. Nagy, and A. Schlüter, “Using machine learning techniques for occupancy-prediction-based cooling control in office buildings,” *Applied energy*, vol. 211, pp. 1343–1358, 2018.
- [65] W. Wang, J. Chen, and T. Hong, “Occupancy prediction through machine learning and data fusion of environmental sensing and wi-fi sensing in buildings,” *Automation in Construction*, vol. 94, pp. 233–243, 2018.
- [66] I. P. Mohottige, H. H. Gharakheili, T. Moors, and V. Sivaraman, “Modeling classroom occupancy using data of wifi infrastructure in a university campus,” *IEEE Sensors Journal*, vol. 22, no. 10, pp. 9981–9996, 2022.
- [67] J. Dutta and S. Roy, “Occupancysense: Context-based indoor occupancy detection & prediction using catboost model,” *Applied Soft Computing*, vol. 119, p. 108536, 2022.

- [68] A. Akbar, M. Nati, F. Carrez, and K. Moessner, "Contextual occupancy detection for smart office by pattern recognition of electricity consumption data," in *2015 IEEE international conference on communications (ICC)*, pp. 561–566, IEEE, 2015.
- [69] W. Kleiminger, C. Beckel, T. Staake, and S. Santini, "Occupancy detection from electricity consumption data," in *Proceedings of the 5th ACM Workshop on Embedded Systems For Energy-Efficient Buildings*, pp. 1–8, 2013.
- [70] W. Kleiminger, C. Beckel, and S. Santini, "Household occupancy monitoring using electricity meters," in *Proceedings of the 2015 ACM international joint conference on pervasive and ubiquitous computing*, pp. 975–986, 2015.
- [71] C. Beckel, W. Kleiminger, R. Cicchetti, T. Staake, and S. Santini, "The eco data set and the performance of non-intrusive load monitoring algorithms," in *Proceedings of the 1st ACM conference on embedded systems for energy-efficient buildings*, pp. 80–89, 2014.
- [72] M. S. Aliero, M. F. Pasha, D. T. Smith, I. Ghani, M. Asif, S. R. Jeong, and M. Samuel, "Non-intrusive room occupancy prediction performance analysis using different machine learning techniques," *Energies*, vol. 15, no. 23, p. 9231, 2022.
- [73] J. Liu, H. Mu, A. Vakil, R. Ewing, X. Shen, E. Blasch, and J. Li, "Human occupancy detection via passive cognitive radio," *Sensors*, vol. 20, no. 15, p. 4248, 2020.
- [74] A. Mohammadabadi, S. Rahnama, and A. Afshari, "Indoor occupancy detection based on environmental data using cnn-xgboost model: Experimental validation in a residential building," *Sustainability*, vol. 14, no. 21, p. 14644, 2022.
- [75] S. Mahjoub, S. Labdai, L. Chrifi-Alaoui, B. Marhic, and L. Delahoche, "Short-term occupancy forecasting for a smart home using optimized weight updates based on ga and pso algorithms for an lstm network," *Energies*, vol. 16, no. 4, p. 1641, 2023.

- [76] A. Husnain and T.-Y. Choe, "Smart occupancy detection system based on long short-term memory units," *Journal of Computers*, vol. 31, no. 5, pp. 159–175, 2020.
- [77] S. Pešić, M. Tošić, O. Iković, M. Radovanović, M. Ivanović, and D. Bošković, "Bleamat: Data analytics and machine learning for smart building occupancy detection and prediction," *International Journal on Artificial Intelligence Tools*, vol. 28, no. 06, p. 1960005, 2019.
- [78] Y.-M. Chang, C.-H. Chen, J.-P. Lai, Y.-L. Lin, and P.-F. Pai, "Forecasting hotel room occupancy using long short-term memory networks with sentiment analysis and scores of customer online reviews," *Applied Sciences*, vol. 11, no. 21, p. 10291, 2021.
- [79] E. Hitimana, G. Bajpai, R. Musabe, L. Sibomana, and J. Kayalvizhi, "Implementation of iot framework with data analysis using deep learning methods for occupancy prediction in a building," *Future Internet*, vol. 13, no. 3, p. 67, 2021.
- [80] S. Kim, S. Kang, K. R. Ryu, and G. Song, "Real-time occupancy prediction in a large exhibition hall using deep learning approach," *Energy and Buildings*, vol. 199, pp. 216–222, 2019.
- [81] Y. Yang, Y. Yuan, T. Pan, X. Zang, and G. Liu, "A framework for occupancy prediction based on image information fusion and machine learning," *Building and Environment*, vol. 207, p. 108524, 2022.
- [82] I. B. Arief-Ang, M. Hamilton, and F. D. Salim, "A scalable room occupancy prediction with transferable time series decomposition of co2 sensor data," *ACM Transactions on Sensor Networks (TOSN)*, vol. 14, no. 3-4, pp. 1–28, 2018.
- [83] H. Zou, Y. Zhou, J. Yang, and C. J. Spanos, "Device-free occupancy detection and crowd counting in smart buildings with wifi-enabled iot," *Energy and Buildings*, vol. 174, pp. 309–322, 2018.
- [84] J. Choi, H. Ge, and N. Koshizuka, "Iot-based occupants counting with smart building state variables," in *2020 IEEE 29th International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE)*, pp. 171–176, IEEE, 2020.

- [85] B. Qolomany, A. Al-Fuqaha, D. Benhaddou, and A. Gupta, "Role of deep lstm neural networks and wi-fi networks in support of occupancy prediction in smart buildings," in *2017 IEEE 19th International Conference on High Performance Computing and Communications; IEEE 15th International Conference on Smart City; IEEE 3rd International Conference on Data Science and Systems (HPCC/SmartCity/DSS)*, pp. 50–57, IEEE, 2017.
- [86] Z. Chen, R. Zhao, Q. Zhu, M. K. Masood, Y. C. Soh, and K. Mao, "Building occupancy estimation with environmental sensors via cd-blstm," *IEEE Transactions on Industrial Electronics*, vol. 64, no. 12, pp. 9549–9559, 2017.
- [87] V. Chidurala and X. Li, "Occupancy estimation using thermal imaging sensors and machine learning algorithms," *IEEE Sensors Journal*, vol. 21, no. 6, pp. 8627–8638, 2021.
- [88] E. Longo, A. E. Redondi, and M. Cesana, "Accurate occupancy estimation with wifi and bluetooth/ble packet capture," *Computer Networks*, vol. 163, p. 106876, 2019.
- [89] I. G. Dino, E. Kalfaoglu, O. K. Iseri, B. Erdogan, S. Kalkan, and A. A. Alatan, "Vision-based estimation of the number of occupants using video cameras," *Advanced Engineering Informatics*, vol. 53, p. 101662, 2022.
- [90] H. Choi, J. Lee, Y. Yi, H. Na, K. Kang, and T. Kim, "Deep vision-based occupancy counting: Experimental performance evaluation and implementation of ventilation control," *Building and Environment*, vol. 223, p. 109496, 2022.
- [91] M. Mosteiro-Romero, I. Hischier, J. A. Fonseca, and A. Schlueter, "A novel population-based occupancy modeling approach for district-scale simulations compared to standard-based methods," *Building and Environment*, vol. 181, p. 107084, 2020.
- [92] C. Fauzi, S. Sulisty, *et al.*, "Group activity recognition method based on camera in the building," *Procedia Computer Science*, vol. 161, pp. 1019–1027, 2019.

- [93] C. M. Ranieri, S. MacLeod, M. Dragone, P. A. Vargas, and R. A. F. Romero, “Activity recognition for ambient assisted living with videos, inertial units and ambient sensors,” *Sensors*, vol. 21, no. 3, p. 768, 2021.
- [94] P. Ganesh, R. E. Idgahi, C. B. Venkatesh, A. R. Babu, and M. Kyrarini, “Personalized system for human gym activity recognition using an rgb camera,” in *Proceedings of the 13th ACM International Conference on PErvasive Technologies Related to Assistive Environments*, pp. 1–7, 2020.
- [95] A. Prasad, A. K. Tyagi, M. M. Althobaiti, A. Almulihi, R. F. Mansour, and A. M. Mahmoud, “Human activity recognition using cell phone-based accelerometer and convolutional neural network,” *Applied Sciences*, vol. 11, no. 24, p. 12099, 2021.
- [96] K. Peppas, A. C. Tsolakis, S. Krinidis, and D. Tzovaras, “Real-time physical activity recognition on smart mobile devices using convolutional neural networks,” *Applied Sciences*, vol. 10, no. 23, p. 8482, 2020.
- [97] P. W. Tien, S. Wei, J. K. Calautit, J. Darkwa, and C. Wood, “Vision-based human activity recognition for reducing building energy demand,” *Building Services Engineering Research and Technology*, vol. 42, no. 6, pp. 691–713, 2021.
- [98] D. Liciotti, E. Frontoni, P. Zingaretti, N. Bellotto, and T. Duckett, “Hmm-based activity recognition with a ceiling rgb-d camera,” in *International Conference on Pattern Recognition Applications and Methods*, vol. 2, pp. 567–574, SciTePress, 2017.
- [99] J. Shi, D. Zuo, and Z. Zhang, “An energy-efficient human activity recognition system based on smartphones,” in *2020 7th International Conference on Soft Computing & Machine Intelligence (ISCFMI)*, pp. 177–181, IEEE, 2020.
- [100] X. Cheng, L. Zhang, Y. Tang, Y. Liu, H. Wu, and J. He, “Real-time human activity recognition using conditionally parametrized convolutions on mobile and wearable devices,” *IEEE Sensors Journal*, vol. 22, no. 6, pp. 5889–5901, 2022.

- [101] D. J. Cook, N. C. Krishnan, and P. Rashidi, “Activity discovery and activity recognition: A new partnership,” *IEEE transactions on cybernetics*, vol. 43, no. 3, pp. 820–828, 2013.
- [102] R. Gómez-Ramos, J. Duque-Domingo, E. Zalama, and J. Gómez-García-Bermejo, “An unsupervised method to recognise human activity at home using non-intrusive sensors,” *Electronics*, vol. 12, no. 23, p. 4772, 2023.
- [103] J. Maitre, K. Bouchard, C. Bertuglia, and S. Gaboury, “Recognizing activities of daily living from uwb radars and deep learning,” *Expert Systems with Applications*, vol. 164, p. 113994, 2021.
- [104] M. Hämäläinen, L. Mucchi, S. Caputo, L. Biotti, L. Ciani, D. Marabissi, and G. Patrizi, “Ultra-wideband radar-based indoor activity monitoring for elderly care,” *Sensors*, vol. 21, no. 9, p. 3158, 2021.
- [105] A. Beaulieu, F. Thullier, K. Bouchard, J. Maître, and S. Gaboury, “Ultra-wideband data as input of a combined efficientnet and lstm architecture for human activity recognition,” *Journal of Ambient Intelligence and Smart Environments*, vol. 14, no. 3, pp. 157–172, 2022.
- [106] F. M. Noori, M. Z. Uddin, and J. Torresen, “Ultra-wideband radar-based activity recognition using deep learning,” *IEEE Access*, vol. 9, pp. 138132–138143, 2021.
- [107] T. Imbeault-Nepton, J. Maitre, K. Bouchard, and S. Gaboury, “Filtering data bins of uwb radars for activity recognition with random forest,” *Procedia Computer Science*, vol. 201, pp. 48–55, 2022.
- [108] Q. Li, J. Liu, R. Gravina, W. Zang, Y. Li, and G. Fortino, “A uwb radar-based adaptive method for in-home monitoring of elderly,” *IEEE Internet of Things Journal*, 2023.
- [109] K. Pan, W.-P. Zhu, and M. Hasannezhad, “Self-attention cnn based indoor human events detection with uwb radar,” *Journal of the Franklin Institute*, vol. 361, no. 14, p. 107090, 2024.
- [110] L. Huang, A. Zhu, M. Qian, and H. An, “Human fall detection with ultra-wideband radar and adaptive weighted fusion,” *Sensors*, vol. 24, no. 16, p. 5294, 2024.

- [111] W. Zhang and L. Hong, “Radar-based human activity recognition using multi-domain maps and feature fusion convolutional neural network,” in *Proceedings of the 2024 3rd Asia Conference on Algorithms, Computing and Machine Learning*, pp. 277–281, 2024.
- [112] L. M. Candanedo and V. Feldheim, “Accurate occupancy detection of an office room from light, temperature, humidity and co2 measurements using statistical learning models,” *Energy and Buildings*, vol. 112, pp. 28–39, 2016.
- [113] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, “Communication-efficient learning of deep networks from decentralized data,” in *Artificial intelligence and statistics*, pp. 1273–1282, PMLR, 2017.
- [114] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, “Federated optimization in heterogeneous networks,” *Proceedings of Machine learning and systems*, vol. 2, pp. 429–450, 2020.
- [115] J. Wang, Q. Liu, H. Liang, G. Joshi, and H. V. Poor, “Tackling the objective inconsistency problem in heterogeneous federated optimization,” *34th Conference on Neural Information Processing Systems (NeurIPS 2020), Vancouver, Canada, 2020*.
- [116] F. De Rango, A. Guerrieri, P. Raimondo, and G. Spezzano, “A novel edge-based multi-layer hierarchical architecture for federated learning,” in *2021 IEEE DASC/PiCom/CBDCCom/CyberSciTech*, pp. 221–225, IEEE, 2021.
- [117] X. Qiu, T. Parcollet, J. Fernandez-Marques, P. P. B. de Gusmao, Y. Gao, D. J. Beutel, T. Topal, A. Mathur, and N. D. Lane, “A first look into the carbon footprint of federated learning,” *arXiv preprint arXiv:2102.07627*, 2021.
- [118] G. Iacovone, G. Cerchiara, L. Cappiello, G. Strazza, E. Sangiorgio, and D. D’Eliso, *Needs Analysis, Protection, and Regulation of the Rights of Individuals and Communities for Urban and Residential Comfort in Cognitive Buildings*, pp. 75–102. Cham: Springer International Publishing, 2023.

- [119] Y. Zhang and P. J. Thorburn, “Handling missing data in near real-time environmental monitoring: A system and a review of selected methods,” *Future Generation Computer Systems*, vol. 128, pp. 63–72, 2022.
- [120] L. Huang, J. Qin, Y. Zhou, F. Zhu, L. Liu, and L. Shao, “Normalization techniques in training dnns: Methodology, analysis and application,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.
- [121] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, “Federated optimization in heterogeneous networks,” *Proceedings of the 1 st Adaptive & Multitask Learning Workshop, Long Beach, California, 2019*, 2019.
- [122] S. Park, Y. Suh, and J. Lee, “Fedpso: federated learning using particle swarm optimization to reduce communication costs,” *Sensors*, vol. 21, no. 2, 2021.
- [123] L. Mba, P. Meukam, and A. Kemajou, “Application of artificial neural network for predicting hourly indoor air temperature and relative humidity in modern building in humid region,” *Energy and Buildings*, vol. 121, pp. 32–42, 2016.
- [124] G. Hinton, N. Srivastava, and K. Swersky, “Neural networks for machine learning lecture 6a overview of mini-batch gradient descent,” *Cited on*, vol. 14, no. 8, p. 2, 2012.
- [125] K. M. Rashid and J. Louis, “Times-series data augmentation and deep learning for construction equipment activity recognition,” *Advanced Engineering Informatics*, vol. 42, p. 100944, 2019.
- [126] M. Wang, S. Lu, D. Zhu, J. Lin, and Z. Wang, “A high-speed and low-complexity architecture for softmax function in deep learning,” in *2018 IEEE asia pacific conference on circuits and systems (APCCAS)*, pp. 223–226, IEEE, 2018.
- [127] I. Stolarek, A. Samelak-Czajka, M. Figlerowicz, and P. Jackowiak, “Dimensionality reduction by umap for visualizing and aiding in classification of imaging flow cytometry data,” *Iscience*, vol. 25, no. 10, 2022.

- [128] K. Weiss, T. M. Khoshgoftaar, and D. Wang, “A survey of transfer learning,” *Journal of Big data*, vol. 3, pp. 1–40, 2016.