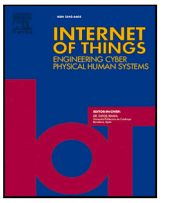
Internet of Things 26 (2024) 101181

Contents lists available at ScienceDirect

Internet of Things

journal homepage: www.elsevier.com/locate/iot



Research article

Leveraging distributed AI for multi-occupancy prediction in Cognitive Buildings

Irfanullah Khan [a,b,*], Franco Cicirelli [a], Emilio Greco [a], Antonio Guerrieri [a,*], Carlo Mastroianni [a], Luigi Scarcello [b], Giandomenico Spezzano [a], Andrea Vinci [a]

[a] ICAR-CNR - Institute for high performance computing and networking - National Research Council of Italy, Rende (CS), Italy
[b] University of Calabria, Rende (CS), Italy

ARTICLE INFO

Dataset link: https://ckan-sobigdata.d4science.org/dataset/multi-sensor_dataset_of_environmental_conditions_in_smart_office

Keywords:
Internet of things
Multi-layer hierarchical federated learning
Edge computing
Long short-term memory neural network
Artificial intelligence
Smart environments
Cognitive buildings

ABSTRACT

Cognitive Buildings are autonomous smart environments capable of setting themselves according to some self-learned rules. Such rules are inferred according to, e.g., the inhabitants' behaviors, users' needs, and specific policies for optimizing security, energy, and comfort management. To do this, it is of foremost importance to gather information about users' habits like room occupancy. Indeed, Cognitive Buildings can effectively exploit information about sensors in the different rooms, thus being able to detect, learn, and forecast the presence of users in the buildings and act in accordance with these predictions. In this direction, 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.

1. Introduction

Cognitive Buildings are smart indoor environments equipped with sensors, actuators, and computational capabilities that combine the benefits of the Internet of Things (IoT) paradigm [1] and the Artificial Intelligence algorithms [2]. Cognitive Buildings are entities able to learn, reason, adapt, and cooperate with each other with the aim of addressing the challenges encountered in the realm of Smart Environments, Smart Cities, and Industry 4.0 [3]. The main benefits that can arise from their adoption include better satisfaction of the users' needs, an increase in productivity, an enhancement of comfort conditions, and an optimization of the energy consumption and operational cost [4].

Cognitive Buildings, and in general cognitive manufacturing, are able to exploit cognitive computing, the industrial Internet of things, and advanced analytics to upgrade manufacturing processes in manners that were not previously conceivable. Cognitive

* Corresponding authors.
E-mail addresses: irfanullah.khan@dimes.unical.it (I. Khan), franco.cicirelli@icar.cnr.it (F. Cicirelli), emilio.greco@icar.cnr.it (E. Greco), antonio.guerrieri@icar.cnr.it (A. Guerrieri), carlo.mastroianni@icar.cnr.it (C. Mastroianni), luigi.scarcello@unical.it (L. Scarcello), giandomenico.spezzano@icar.cnr.it (G. Spezzano), andrea.vinci@icar.cnr.it (A. Vinci).

https://doi.org/10.1016/j.iot.2024.101181
Received 5 January 2024; Received in revised form 1 March 2024; Accepted 3 April 2024
Available online 10 April 2024

computing is a fundamental pillar of the Industry 4.0 evolution [5,6], since it can help to manage the increasing number of variables that can potentially affect manufacturing performance, and to process and analyze the production data collected in real-time. The objective of Industry 4.0 is the design and development of smart products, procedures, and processes [7]. In this context, IoT and Cyber–Physical Systems[1] allow for the interconnection of people, things, and data, thus enabling new ways of organizing and conducting industrial processes [9]. This need emerges in several disciplines, such as: software engineering, control engineering, embedded systems, advanced robotics [10].

A Cognitive Building [11] 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. Designing and implementing such environments is challenging because they are inherently heterogeneous and require transversal technological and methodological skills. Moreover, they have to be pervasive and, at the same time, non-invasive and non-intrusive, i.e., they must not impair people's well-being and must preserve their privacy. Since it has been estimated that, currently, buildings' energy consumption is, in the EU, 40% of the total energy consumption,[2] Cognitive Buildings have to take care of energy optimization issues [12]. By monitoring occupancy presence information, energy-efficient building solutions can reduce energy consumption and promote a healthy and safe environment [13].

Many studies report how IoT devices [14] can be used to monitor environmental parameters such as temperature, humidity, CO_2 levels, and light, in order to determine when people are present in the different areas of the buildings [15]. In addition, the occupancy data is crucial for several reasons, such as [16]: (i) *air quality*: after the COVID-19 pandemic, air quality is more important than ever and is crucial for providing a safe environment; (ii) *air conditioning*: the temperature and operation time of HVAC systems can be set on the basis of the number of users and their preferences; (iii) *indoor lighting systems*: operating times and intensities of lights can be driven by occupancy information; (iv) *security*: for managing security concerns, e.g., in the event of an emergency evacuation, the knowledge about the presence and distribution of individuals in buildings is essential.

Researchers have shown that HVAC management, when assisted by occupancy data, can lead to energy savings ranging from 10%–40% [17], while lighting systems can save up to 75% of the energy [18]. Such savings can be obtained by simply detecting when a space is unoccupied, however, more significant energy efficiency improvements are also possible by knowing or predicting the exact number of people in a room [19–21] since buildings can be optimized with more precise control of lighting system, temperature, and ventilation. As an example, if the actual number of occupants is lower or higher than expected, building systems may over- or under-condition the space, resulting in energy waste and increased costs, or in discomfort conditions.

By collecting data from IoT sensors, machine learning models can learn patterns and make predictions about the usage of the spaces in a building. Even though occupancy could be estimated by exploiting classical machine learning algorithms such as Random Forest [22], more recent Deep Learning-based approaches resulted in better performance and scalability [23]. Among them, Long Short-Term Memory Artificial Neural Networks are more advantageous than others due to their ability to retain long-term memory of past data, thus being the most suitable in the occupancy prediction task [24]. Indeed, LSTM networks [25] have specialized memory cells and gates that control the flow of information, allowing them to store information over a long period of time. This has determined their widespread adoption in modeling sequential data for the prediction of occupancy in environmental monitoring [26].

In addition, people are afraid of solutions that physically wire them to sensors, and thus, they require device-free technologies for their monitoring [27]. So, two conventional approaches have been utilized to collect data for the prediction of room occupancy without exploiting wearable devices: cameras [28] and environmental sensors, e.g., motion [29,30] and CO_2 sensors [22]. However, even if these methods are promising for achieving energy-efficient buildings, they can raise privacy concerns, for example, about how the data is stored, analyzed, and transmitted to the Cloud. If this data is not handled properly, it could be vulnerable to hacking or other forms of data breaches, compromising individuals' privacy. Therefore, handling data security and privacy is essential when implementing occupancy estimation algorithms that use IoT devices [31].

Federated Learning (FL) is an emerging approach for addressing privacy concerns associated with occupancy detection that can give results close to the corresponding centralized model [32]. By using FL, an Edge Computing layer [33] is exploited to perform local computations and local model training [34]. The Edge layer acts as an intermediary layer between the devices and the central server (e.g., the Cloud), and ensures that raw sensor data are not shared with external entities. The exchange of information with the Cloud occurs only for high-level model updates and aggregated gradients, while most communications remain local. This helps to enforce privacy requirements and also reduces the experienced latency, which leads to communication cost savings and more reactive behavior of the applications [35,36].

This paper presents a distributed approach to predict multi-occupancy in Cognitive Buildings. The proposed method encompasses: (i) a multi-layer hierarchy for Federated Learning; (ii) the utilization of IoT devices at the Edge for both data collection and actuation purposes; (iii) the utilization of long short-term memory neural network models for accurate occupancy estimation; (iv) the exploitation of Edge Computing for efficient data processing and occupancy forecasting; (v) the introduction of a versatile design template suitable for developing and implementing real distributed systems for multi-occupancy prediction in Cognitive Buildings. To safeguard the privacy of the data, the distributed approach collects data to train separate local deep learning LSTM models and

[1] In [8], the Internet of Things is defined as a new paradigm in which every device is digitally connected, regardless of their function, and can communicate with other devices and people over communication protocols. In [1], cyber–physical systems, or CPS for short, are defined as sophisticated computer devices that work together to perform functions, control physical elements, and respond to human control. From these definitions, it emerges that CPS focus more on control aspects while IoT focus more on networking and computation.

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

uses an FL process to determine an aggregated LSTM model. Moreover, a multi-layer approach is proposed to reduce the energy consumed for communication, by avoiding continuous data transmission to the Cloud [37]. This paper also shows a preliminary prototype of the presented approach, developed and deployed at ICAR-CNR, Rende, Italy, and reports a comparative analysis with centralized learning and standard Federated Learning approaches.

This paper widely extends the work presented in [38], in which we proposed an approach that highlighted the benefits of integrating IoT and Edge technologies with Federated Learning to detect the presence/absence of occupants. In the past work, we applied an FL model to predict an occupied/non-occupied binary variable for each room in an office building. Moreover, the case study relied on datasets taken from the literature. With respect to the previous work, in this paper, we add the multi-occupants prediction, a multi-layer hierarchical FL architecture, a design template, a real implemented case study, and a comparative analysis.

The structure of the paper is as follows: a review of the relevant literature is provided in Section 2. The proposed approach is presented in Sections 3, Section 4 illustrates a detailed overview of the case study, and Section 5 discusses the outcomes of our approach. Finally, Section 6 concludes the paper and describes some avenues for future research.

2. Related work

The concept of Cognitive Buildings involves the integration of various technologies to improve building operations, optimize energy usage, and enhance occupants' comfort. One important application of these technologies is the forecasting of room occupancy, which involves predicting the number of people that are likely to be present in a given space at a particular time.

Early research in occupancy prediction focused on vision-based methods, which utilized cameras to detect and track people in a building. In [39], the authors propose a method for investigating the correlation between the active WiFi connections and the actual building occupancy, using cameras to obtain the ground-truth data. The proposed method is able to predict the number of occupants in the following day with high accuracy. Similarly, the work presented in [40] uses a vision-based approach to predict the occupancy in an office. More specifically, the paper in [40] evaluates the performance of a deep learning-based occupancy prediction method in two offices by using a camera-based system for counting the people in the office during the training phase. The authors assess the performance of energy-saving strategies for controlling HVAC and lighting based on the implemented system.

Camera-based occupancy prediction is widely used and shows promising results, but it poses privacy issues that have led researchers to investigate alternative methods. Moreover, the use of cameras in offices or houses is perceived as uncomfortable by users. As a solution, researchers are shifting their attention towards sensor-based methods that can collect accurate occupancy data without infringing on privacy.

Sensor-based data collection for monitoring the room environment of a building is a promising technology that has recently gained attention in the literature. When dealing with data from multiple sensors, traditional machine learning may struggle to process and analyze the data effectively. Deep learning, however, is specifically designed to handle complex and large data sets with high levels of accuracy and efficiency. By using neural networks, deep learning can identify intricate patterns and relationships in the data, making it a more suitable approach for occupancy prediction in intelligent building systems. As an example, the work presented in [41] proposes a novel approach to infer occupancy in intelligent buildings. The authors use AI methods such as Linear Regression, Neural Networks, and Random Trees to predict the concentration of CO_2 based on the values of temperature and relative humidity. The aim is to identify the presence of people and improve automation, while saving energy. The authors also created a software tool for real-time monitoring and storage of the collected data using a message-queuing telemetry transport protocol and a CouchDB database.

In [15], the authors propose an LSTM algorithm that uses a combination of temperature, humidity, CO_2 concentration, light, and motion sensors to forecast room occupancy in the short term. This approach allows buildings to become cognitive and self-adapting, thereby achieving energy efficiency and reducing wastage without compromising privacy concerns. Similarly, another recent study [42] employs a machine learning forecasting approach that utilized an LSTM neural network to predict occupancy in buildings and, accordingly, to optimize the energy management of electric heating systems. The study improves the LSTM algorithm's performance using optimization techniques such as Genetic Algorithms (GA) and Particle Swarm Optimization (PSO). Experiments based on real-world datasets showed that the proposal outperformed standard LSTM algorithms. The study in [24] presents a novel algorithm, based on the predicted number of occupants in a room, for the control of ventilation systems. The authors collected data on room occupancy and from the HVAC system. Four different models, including LSTM, were utilized to predict the number of occupants in the room. The results showed that each model was able to predict the number of occupants with 85% accuracy. Two important issues related to our work are the management of privacy and the reduction of energy consumption. These two issues are discussed in the following.

Privacy. The studies mentioned above employ sensors to monitor indoor environments without focusing on privacy management. However, privacy concerns arise, especially when there is the need to share the collected data with a central server or with the Cloud for processing and storage. This can be detrimental to privacy, as it may allow unauthorized parties to access or intercept sensitive information. In particular, a deep view of EU privacy regulations in Cognitive Buildings is given in [43], which can also be used as a useful guideline for designers and developers to create regulations-compliant applications. Some of the approaches used in the literature to preserve privacy regarding the data collected in buildings are Federated Learning [44] and Multi-Layer Hierarchical Federated Learning (MLH-FL) [37]. As highlighted in [45], FL is suited for use cases in which privacy is a key concern, and a clear view and understanding of risk factors enables an FL adopter to successfully build a secure environment. Furthermore,

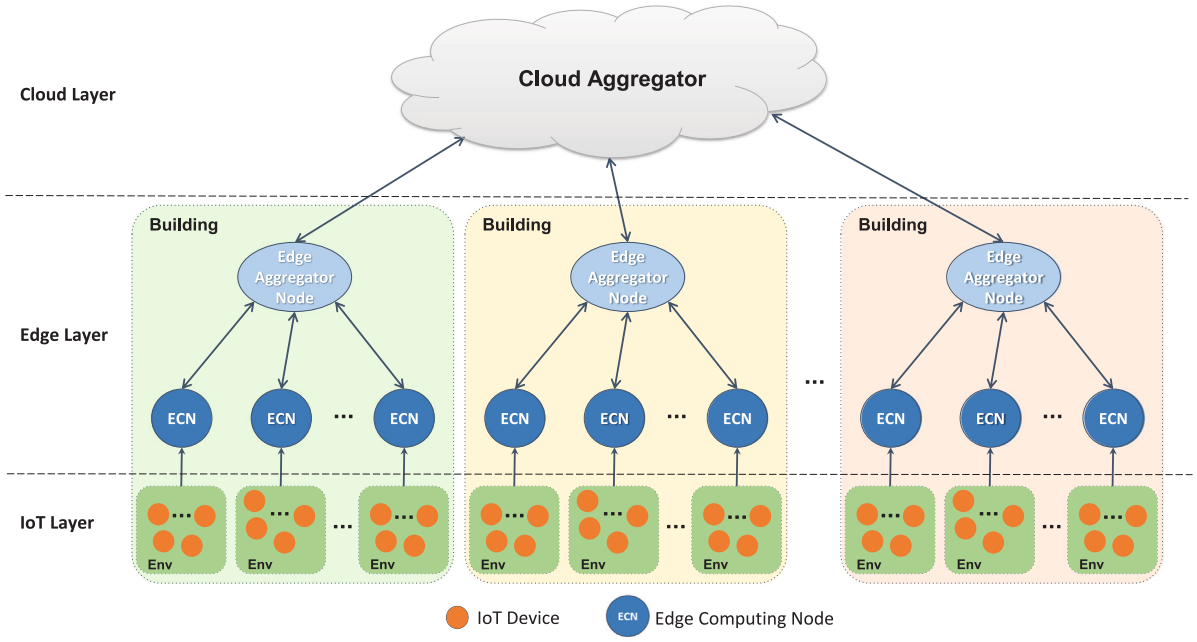


Fig. 1. The reference architecture proposed in the paper.

as reported in [46,47], there are many privacy mechanisms, such as differential privacy [48] and k-anonymity [49], which enforce FL with different privacy guarantees.

Energy. Cognitive and sustainable manufacturing, and Industry 4.0 practices, can help to optimize energy consumption and improve resource efficiency [50,51], which is essential when considering that energy consumption is a major part of costs in buildings and indoor environments. The survey work in [17] summarizes the significant energy savings that can be obtained when applying a wide range of energy efficiency policies in a Cognitive Building context. In particular, a policy based on occupancy forecasting can lead to important energy savings (up to 76%) on the lighting system [18] by automatically turning off the lights according to the change of the occupant's activity level. The MLH-FL approach, already mentioned before [37], besides offering an improvement in privacy preservation, helps in reducing communication burdens with respect to classical Federated Learning algorithms. Specifically, experiments show that this approach achieves a similar accuracy when compared to the classic FL approach, but can ensure an amount of energy saving of about 80%. Such saving is due to a dramatic reduction in the number of communications towards the Cloud since most of the models are processed and aggregated directly in the Edge layer. Also, the work in [52] introduces a promising approach based on MLH-FL. The reduction of energy consumption derives from the tremendous WAN traffic savings, which, in the mentioned work, are estimated to be between 95% and 99%. As a consequence, the monetary cost related to WAN traffic appears to be dramatically reduced between 79% and 97%.

When compared to the mentioned works, our approach is, to the best of our knowledge, the first one that aims to tackle the problem of multi-occupancy forecasting by providing a design template and leveraging Multi-Layer Hierarchical Federated Learning, LSTM, and IoT/Edge technologies.

3. The approach for occupancy prediction

In this paper, we propose an approach for occupancy prediction in Cognitive Buildings, whose main characteristics are: (i) a multi-layer hierarchy for Federated Learning [53], which combines models for presence forecasting at the different levels; (ii) the use of IoT devices at the Edge for data collection and actuation; (iii) long short-term memory neural network models [22,54]; (iv) the exploitation of Edge Computing for processing data and forecasting occupancy; (v) a design template that can be used to design and implement real distributed systems for occupancy prediction in Cognitive Buildings. The following subsections will describe the proposed approach.

3.1. The reference architecture

The reference architecture of the proposal is shown in Fig. 1. 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, 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 [37]. 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 1, 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 [55]. Although very interesting, the evaluation of this specific aspect is postponed to future works.

3.2. The design template

The diagram in Fig. 2 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.

Algorithm 1 The proposed algorithm based on the Multi-Layer Hierarchical Federated Learning.

procedure LSTM_ML_HFL($LowLevelRounds$, $MidLevelRounds$, $HighLevelRounds$)
  $currLLR = 0$, $currMLR = 0$, $currHLR = 0$
  $sendModel(CA, EANs)$ ▷ the initial model is sent from the Cloud to all the EANs
  $sendModel(EANs, ECNs)$ ▷ the initial model is sent from the EANs to all the ECNs
  while $currHLR < HighLevelRounds$ **do**
    while $currMLR < MidLevelRounds$ **do**
      while $currLLR < LowLevelRounds$ **do**
        $collectIoTData(ECNs)$ ▷ data is collected by the ECNs
        $updateInternalModel(ECNs)$ ▷ the models are updated in the ECNs based on the data collected
        $currLLR++$
      $sendModel(ECNs, EANs)$ ▷ the updated models are sent to the respective EANs
      $aggregateModel(EANs)$ ▷ EANs merge the updated models from the related ECNs
      $sendModel(EANs, ECNs)$ ▷ ECNs receive the merged models from the related EANs
      $currMLR++$
    $sendModel(EANs, CA)$ ▷ EANs send the aggregated model to the CA
    $aggregateModel(CA)$ ▷ CA merge the models from all the EANs
    $sendModel(CA, EANs)$ ▷ at each high-level round, all the nodes are updated with the last merged model
    $sendModel(EANs, ECNs)$
    $currLLR = 0$, $currMLR = 0$
    $currHLR++$

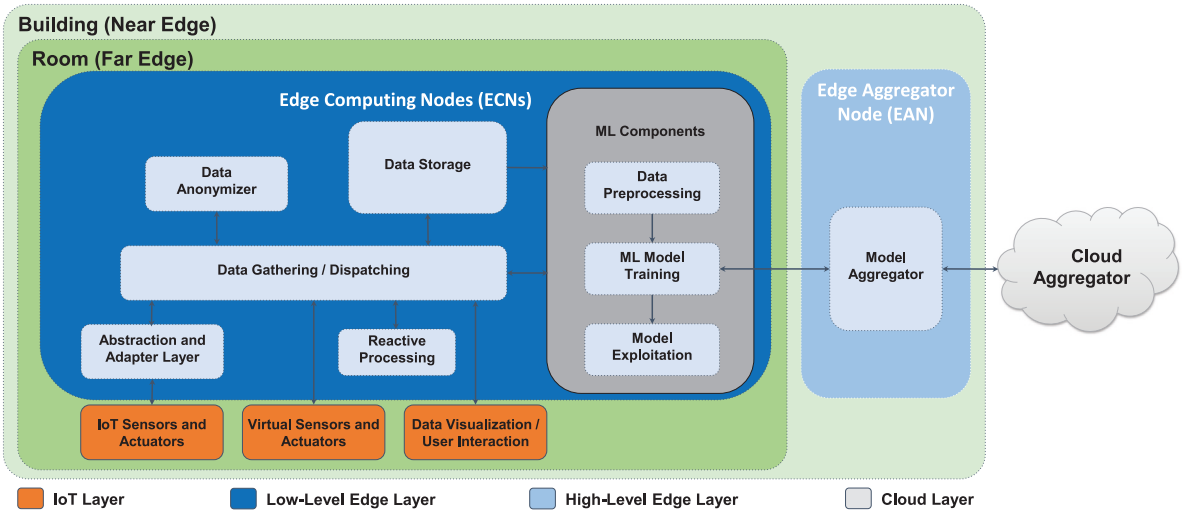


Fig. 2. The proposed Design Template.

- **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.
- **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.

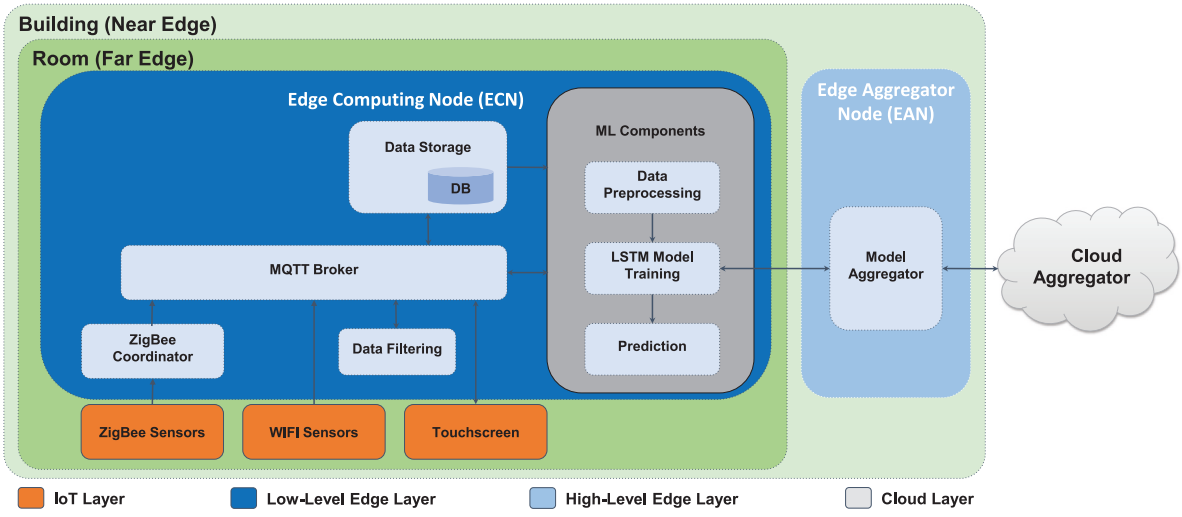


Fig. 3. The main components implemented for the case study.

- **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. 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 (see Section 3), 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 [43].

Fig. 3 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.

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 Waspmote IoT Sensors[3] 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 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 in Section 3, 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 8 GB 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.

4.1. 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 1.

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.

[3] Waspmote: https://development.libelium.com/waspmote-technical-guide/waspmote-kit.

Table 1
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 <5 m, angle 100°.
Xiaomi Mijia Smart Light Sensor	Zigbee 3 HA	battery	Light [lux]	range: from 0 to 83000 lux.
Waspmote 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.

Fig. 4. The touchscreen connected to the $EDGE_I$ Raspberry Pi.

4.2. Wifi sensors

In our case study, we used the Waspmote WiFi Pro node equipped with the Waspmote Gases Sensor Board PRO[4]. The board is customized with a Carbon Dioxide (CO_2) and a Carbon Monoxide (CO) gas sensor (see Table 1). As soon as the Waspmote 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.

4.3. 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"/"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 Fig. 4. The touchscreen also displays real-time data coming from different sensors, including CO_2, 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 Fig. 5. 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 CO_2 concentration, temperature, humidity, light intensity, and CO level, which are then displayed on the screen.

4.4. 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

[4] Waspmote Gases Sensor Board PRO. https://development.libelium.com/gases_pro_sensor_guide/gases-pro-sensor-board-calibrated.

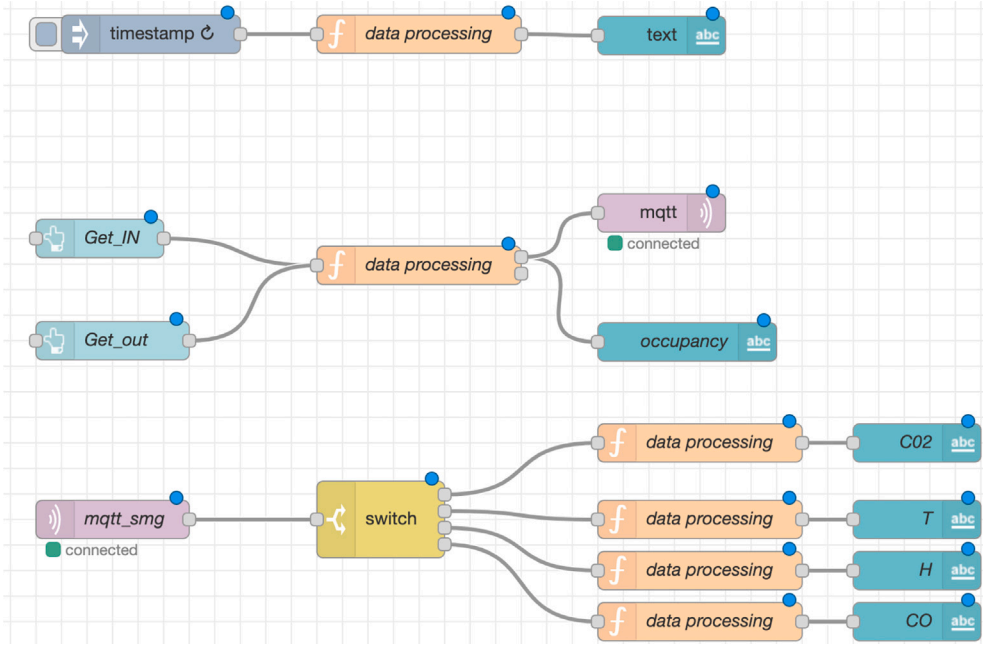


Fig. 5. The realized Node-Red application.

cleaned and normalized. Hence, the Data Preprocessing component 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 Preprocessing component has been purposely designed to deal with missing values, execute data normalization, and make data transformation. These operations are detailed in the following.

4.4.1. 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 [56] was implemented, which is a statistical method in which a missing value is replaced by the previously observed one.

4.4.2. 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 [57] 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.

4.4.3. 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.

4.5. 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.

4.6. Model aggregator

This component, implemented at the Building level, is devoted to the aggregation of different trained LSTM models according to the *FedAvg* [58] algorithm. We chose FedAvg, rather than other algorithms for the merging of LSTM models, such as *FedProx* [59] and *FedPso* [60], 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.

5. Experimental evaluation

After the proposal of our approach in Section 3 and the introduction of our case study in Section 4, 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 Fig. 6, which involves all the sensors deployed in the IoT Laboratory, i.e., the room equipped with the largest variety of sensors. A correlation matrix shows how strongly the pairs of variables are related in a dataset. If we have a set of variables $X_1, X_2, \ldots, X_n$, 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:

$$r_{ij} = \frac{\text{cov}(X_i, X_j)}{\sqrt{\text{var}(X_i) \cdot \text{var}(X_j)}} \tag{1}$$

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:

$$R = \begin{bmatrix} 1 & r_{12} & \ldots & r_{1n} \\ r_{21} & 1 & \ldots & r_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ r_{n1} & r_{n2} & \ldots & 1 \end{bmatrix} \tag{2}$$

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 (Fig. 6), it is clear that the most useful sensors to infer and predict the value of occupancy, set manually through the virtual sensor of Fig. 3, and represented by the *Occupancy Counter* field in Fig. 6, are the light (illuminance), the CO_2, and the PIR (passive infrared) sensors. 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.

5.1. Setup

The Multi-Layer Hierarchical FL training and prediction tasks have been realized by using *Python*[5] (Release 3.7.0) and a set of libraries, among which *TensorFlow*[6] (version 2.8.2), *TensorFlow Federated*[7] (version 0.16.1), and *Pandas* (version 1.3.5)[8]. As explained above, the considered part of the dataset consists of the data coming from the light, the CO_2, 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 Fig. 7.

[5] Python 3.7.0. https://www.python.org/downloads/release/python-370/.

[6] TensorFlow Homepage. https://www.tensorflow.org/.

[7] TensorFlow Federated. https://www.tensorflow.org/federated.

[8] Pandas - Python Data Analysis Library. https://pandas.pydata.org/.

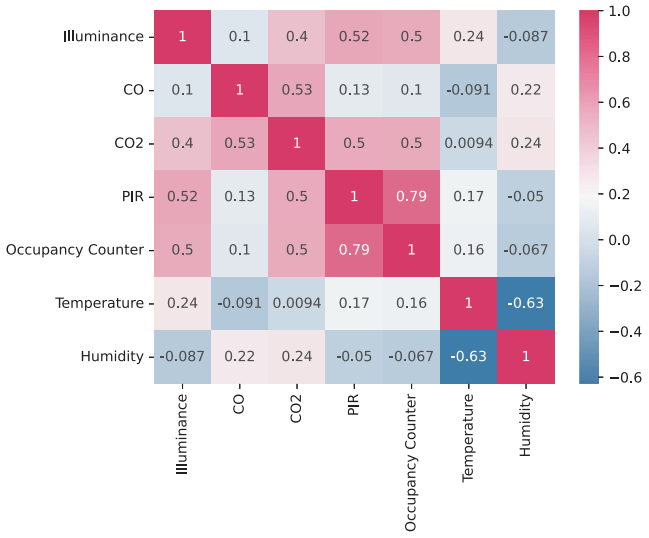


Fig. 6. Correlation matrix of the sensors deployed in the IoT Laboratory.

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 CO_2, 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 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 [61] 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 [62], 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 min ahead) by using the previous 60 observations of data as input (i.e., 60 min 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 Section 3, 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.

5.2. 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:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(x_i - y_i)^2 \tag{3}$$

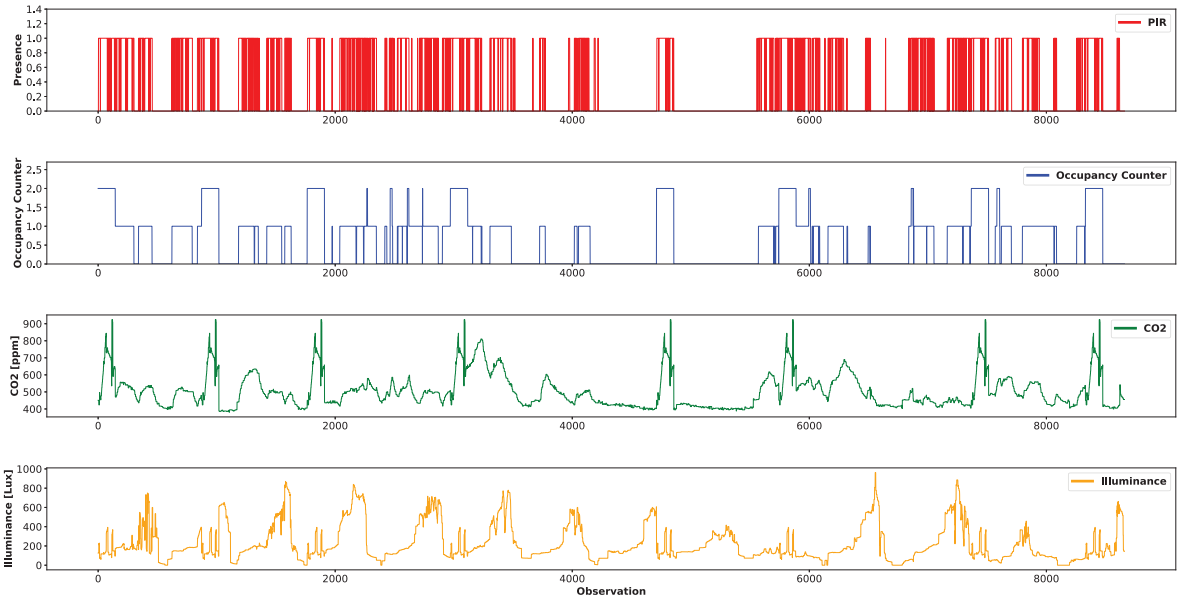


Fig. 7. The Testset used in the implemented scenario.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|x_i - y_i| \tag{4}$$

$$Precision = \frac{TP}{TP + FP} \tag{5}$$

$$Recall = \frac{TP}{TP + FN} \tag{6}$$

$$F1 = \frac{2 * Precision * Recall}{Precision + Recall} \tag{7}$$

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{8}$$

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 prediction classes.

5.3. 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 Fig. 8, 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 Fig. 8 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 (see Section 5.1), and, for each HighLevelRounds, we executed 10 LowLevelRounds. We observed improvements on the results at each HighLevelRound iteration, as displayed in Table 2. Further iterations beyond 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 3. The *F1 score* for the *occupancy class 0, the occupancy class 1, and the multi-occupancy classes* reached the values 0.88 and

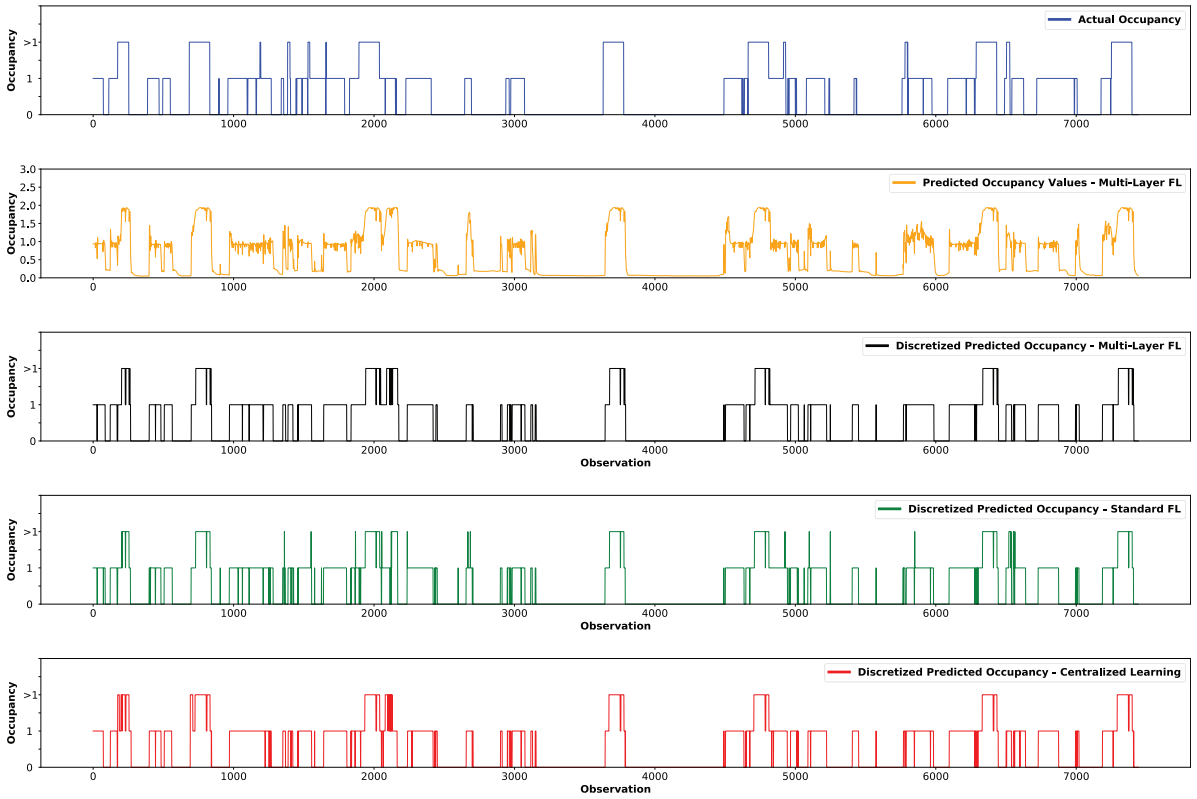


Fig. 8. Actual vs Predicted occupancy for the considered case study.

Table 2
Performance of the proposed model.

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

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 3 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 Fig. 8 with green and red lines. The related evaluation metrics, including *Accuracy, Macro Precision, Macro Recall, and Macro F1-score*, are reported in Table 4. `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`.

In conclusion, we would like to summarize the key contributions and outcomes of our experimental evaluation: (i) we built a model, based on Federated Learning and LSTM, which is able to predict the occupancy level in buildings rooms by distinguishing among three different classes, namely *Occupancy class 0*, *Occupancy class 1*, and *multi-occupancy*; (ii) we computed the main evaluation metrics, which show that our proposal can globally reach an accuracy and a F1-score of 84.5%; (iii) we compared our Multi-Layer Hierarchical FL approach to the standard FL and the centralized learning approach. The comparison shows that our approach is comparable to the other approaches in terms of accuracy and F1-Score, keeps the benefits introduced by the standard FL, i.e., privacy preservation and latency reduction, and in addition, leads to energy savings in terms of communication.

Table 3
Evaluation matrix table for the considered case study.

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

Table 4
Evaluation metrics computed for the comparison analysis of different methods. Bold values highlight the best method for a given measure.

Methods	*Accuracy*	*Macro Precision*	*Macro Recall*	*Macro F1 Score*
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**

6. Conclusions

This paper presented a novel approach that combines advanced technologies to predict multi-occupancy in Cognitive Buildings for optimized energy usage and enhanced building performance. More in detail, the approach comprehends multi-layer hierarchy for Federated Learning, utilization of IoT devices at the Edge, implementation of long short-term memory neural network models, and Edge Computing. The approach also introduces a versatile design template for developing real distributed systems for occupancy prediction. A real-world implementation of the approach was developed at the ICAR-CNR office, where the approach achieved a global *accuracy* of 84.5% in forecasting 10 minutes-ahead room multi-occupancy while preserving data privacy. The approach is validated by carrying out a comparative analysis with centralized learning and standard federated learning.

Although this research contributed to the growing body of literature on using advanced technologies to tackle energy waste in buildings, there is still room for improvement. For instance, more sensors can be utilized to determine and forecast not only the presence but also the activities carried out in the building's rooms. Further studies can focus on the optimization of the parameters that determine the numbers of iterations performed at the different levels of the Multi-Layer Hierarchical Federated Learning. Furthermore, the approach can be enhanced by applying control strategies and specific actuations on the environment, in accordance with the occupancy predictions, in order to pursue energy-saving policies, e.g., by properly controlling lighting and HVAC systems. To further enforce the privacy of the implemented system, some mechanisms can also be introduced, such as the differential privacy and the k-anonymity. Our approach can also be extended to consider preliminary training for new buildings through the adoption of digital twins implemented on purpose. Finally, more experiments can be conducted on the possible transfer of a learned model among similar buildings.

CRediT authorship contribution statement

Irfanullah Khan: Writing – original draft, Validation, Software, Data curation, Conceptualization, Methodology. **Franco Cicirelli:** Writing – review & editing, Supervision, Methodology, Funding acquisition, Conceptualization. **Emilio Greco:** Validation, Software, Data curation. **Antonio Guerrieri:** Writing – original draft, Supervision, Methodology, Funding acquisition, Conceptualization, Writing – review & editing. **Carlo Mastroianni:** Funding acquisition, Validation, Writing – review & editing. **Luigi Scarcello:** Writing – review & editing, Data curation. **Giandomenico Spezzano:** Writing – review & editing, Supervision, Methodology, Conceptualization. **Andrea Vinci:** Writing – review & editing, Supervision, Methodology, Funding acquisition, Conceptualization.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Franco Cicirelli, Emilio Greco, Antonio Guerrieri, Andrea Vinci reports financial support was provided by European Union. Antonio Guerrieri, Carlo Mastroianni, Andrea Vinci reports financial support was provided by Italian Ministry of University and Research. If there are other authors, they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The experimental data has been shared in the SoBigData.it platform at: https://ckan-sobigdata.d4science.org/dataset/multi-sensor_dataset_of_environmental_conditions_in_smart_office.

Acknowledgments

This work has been partially supported by: Project SoBigData.it, SoBigData.it receives funding from European Union – NextGenerationEU – National Recovery and Resilience Plan (Piano Nazionale di Ripresa e Resilienza, PNRR) – Project: "SoBigData.it – Strengthening the Italian RI for Social Mining and Big Data Analytics" – Prot. IR0000013 – Avviso n. 3264 del 28/12/2021; ICSC — Italian Research Center on High Performance Computing, Big Data and Quantum Computing, funded by European Union — NextGenerationEU, PUN: B93C22000620006; the National Research Council of Italy (CNR), "Le Scienze per le TRansizioni Industriale, Verde ed Energetica": Towards Sustainable Cognitive Buildings (ToSCoB) project, CUP B53C22010110001; European Union - NextGenerationEU - the Italian Ministry of University and Research, PRIN 2022 Project "INSIDER: INtelligent ServIce Deployment for advanced Cloud-Edge integRation", grant n. 2022WWSCRR, CUP H53D23003670006; and European Union - NextGenerationEU - the Italian Ministry of University and Research, PRIN 2022 Project "COCOWEARS" (A framework for COntinuum COmputing WEARable Systems), grant n. 2022T2XNJE, CUP B53D23013190006.

References

[1] V. Lesch, M. Züfle, A. Bauer, L. Iffländer, C. Krupitzer, S. Kounev, A literature review of IoT and CPS—What they are, and what they are not, J. Syst. Softw. 200 (2023) 111631, http://dx.doi.org/10.1016/j.jss.2023.111631.
[2] F. Cicirelli, A. Guerrieri, A. Vinci, G. Spezzano, IoT Edge Solutions for Cognitive Buildings, Springer Nature, 2022.
[3] J. Ploennigs, A. Ba, M. Barry, Materializing the promises of cognitive IoT: How cognitive buildings are shaping the way, IEEE Internet Things J. 5 (4) (2018) 2367–2374, http://dx.doi.org/10.1109/JIOT.2017.2755376.
[4] S. Rinaldi, A. Flammini, M. Pasetti, L.C. Tagliabue, A.C. Ciribini, S. Zanoni, Metrological issues in the integration of heterogeneous IoT devices for energy efficiency in cognitive buildings, in: 2018 IEEE International Instrumentation and Measurement Technology Conference, I2MTC, 2018, pp. 1–6, http://dx.doi.org/10.1109/I2MTC.2018.8409740.
[5] A.V. Carvalho, A. Chouchene, T.M. Lima, F. Charrua-Santos, Cognitive manufacturing in industry 4.0 toward cognitive load reduction: A conceptual framework, Appl. Syst. Innov. 3 (4) (2020) http://dx.doi.org/10.3390/asi3040055.
[6] C.D. Serge Bonnaud, A. Kohler, Industry 4.0 and Cognitive Manufacturing, Tech. Rep., IBM Global Markets, 2024, URL https://www.ibm.com/downloads/cas/M8J5BA6R. (Accesed on February 2024).
[7] Y. Lu, Industry 4.0: A survey on technologies, applications and open research issues, J. Ind. Inform. Integr. 6 (2017) 1–10, http://dx.doi.org/10.1016/j.jii.2017.04.005.
[8] S. Fábio, C. Analide, Sensorization to promote the well-being of people and the betterment of health organizations, in: Applying Business Intelligence to Clinical and Healthcare Organizations, 2016, pp. 116–135, http://dx.doi.org/10.4018/978-1-4666-9882-6.ch006.
[9] J. Cecil, S. Albuhamood, A. Cecil-Xavier, P. Ramanathan, An advanced cyber physical framework for micro devices assembly, IEEE Trans. Syst. Man Cybern.: Syst. 49 (1) (2019) 92–106, http://dx.doi.org/10.1109/TSMC.2017.2733542.
[10] J. Tavčar, I. Horváth, A review of the principles of designing smart cyber-physical systems for run-time adaptation: Learned lessons and open issues, IEEE Trans. Syst. Man Cybern.: Syst. 49 (1) (2019) 145–158, http://dx.doi.org/10.1109/TSMC.2018.2814539.
[11] M. Amadeo, F. Cicirelli, A. Guerrieri, G. Ruggeri, G. Spezzano, A. Vinci, When edge intelligence meets cognitive buildings: The COGITO platform, Internet Things 24 (2023) 100908, http://dx.doi.org/10.1016/j.iot.2023.100908.
[12] S. D'Oca, T. Hong, J. Langevin, The human dimensions of energy use in buildings: A review, Renew. Sustain. Energy Rev. 81 (2018) 731–742.
[13] L. Scarcello, F. Cicirelli, A. Guerrieri, C. Mastroianni, G. Spezzano, A. Vinci, Pursuing energy saving and thermal comfort with a human-driven DRL approach, IEEE Trans. Hum.-Mach. Syst. (2022) 1–13, http://dx.doi.org/10.1109/THMS.2022.3216365.
[14] L. Atzori, A. Iera, G. Morabito, The Internet of Things: A survey, Comput. Netw. 54 (15) (2010) 2787–2805, http://dx.doi.org/10.1016/j.comnet.2010.05.010.
[15] S. Colace, S. Laurita, G. Spezzano, A. Vinci, Room occupancy prediction leveraging LSTM: An approach for cognitive and self-adapting buildings, in: F. Cicirelli, A. Guerrieri, A. Vinci, G. Spezzano (Eds.), IoT Edge Solutions for Cognitive Buildings - Technology, Communications and Computing, Springer, 2023, pp. 197–219, http://dx.doi.org/10.1007/978-3-031-15160-6_9.
[16] Z. Chen, C. Jiang, L. Xie, Building occupancy estimation and detection: A review, Energy Build. 169 (2018) 260–270.
[17] V. Oikonomou, F. Becchis, L. Steg, D. Russolillo, Energy saving and energy efficiency concepts for policy making, Energy Policy 37 (11) (2009) 4787–4796.
[18] T. Leephakpreeda, Adaptive occupancy-based lighting control via grey prediction, Build. Environ. 40 (7) (2005) 881–886.
[19] T. Hong, D. Yan, S. D'Oca, C.-f. Chen, Ten questions concerning occupant behavior in buildings: The big picture, Build. Environ. 114 (2017) 518–530.
[20] J. Ahmad, H. Larijani, R. Emmanuel, M. Mannion, A. Javed, Occupancy detection in non-residential buildings–A survey and novel privacy preserved occupancy monitoring solution, Appl. Comput. Inform. (2020).
[21] I. Khan, E. Greco, A. Guerrieri, G. Spezzano, Occupancy prediction in buildings: State of the art and future directions, in: Device-Edge-Cloud Continuum: Paradigms, Architectures and Applications, Springer, 2023, pp. 203–229.
[22] J. Kim, J. Bang, A. Choi, H.J. Moon, M. Sung, Estimation of occupancy using IoT sensors and a carbon dioxide-based machine learning model with ventilation system and differential pressure data, Sensors 23 (2) (2023) 585.
[23] Z.D. Tekler, A. Chong, Occupancy prediction using deep learning approaches across multiple space types: A minimum sensing strategy, Build. Environ. 226 (2022) 109689.
[24] Z. Jiang, Z. Deng, X. Wang, B. Dong, PANDEMIC: Occupancy driven predictive ventilation control to minimize energy consumption and infection risk, Appl. Energy 334 (2023) 120676.
[25] R.C. Staudemeyer, E.R. Morris, Understanding LSTM–A tutorial into long short-term memory recurrent neural networks, 2019, arXiv preprint arXiv:1909.09586.
[26] E. Hitimana, G. Bajpai, R. Musabe, L. Sibomana, J. Kayalvizhi, Implementation of IoT framework with data analysis using deep learning methods for occupancy prediction in a building, Future Internet 13 (3) (2021) 67.
[27] J. Yang, H. Zou, H. Jiang, L. Xie, Device-free occupant activity sensing using WiFi-enabled IoT devices for smart homes, IEEE Internet Things J. 5 (5) (2018) 3991–4002, http://dx.doi.org/10.1109/JIOT.2018.2849655.
[28] I.G. Dino, E. Kalfaoglu, O.K. Iseri, B. Erdogan, S. Kalkan, A.A. Alatan, Vision-based estimation of the number of occupants using video cameras, Adv. Eng. Inform. 53 (2022) 101662.
[29] M.S. Aliero, M.F. Pasha, D.T. Smith, I. Ghani, M. Asif, S.R. Jeong, M. Samuel, Non-intrusive room occupancy prediction performance analysis using different machine learning techniques, Energies 15 (23) (2022) 9231.
[30] S. Hu, P. Wang, C. Hoare, J. O'Donnell, Building occupancy detection and localization using CCTV camera and deep learning, IEEE Internet Things J. 10 (1) (2022) 597–608.

[31] G. Chaopeng, L. Zhengqing, S. Jie, A privacy protection approach in edge-computing based on maximized dnn partition strategy with energy saving, J. Cloud Comput. 12 (1) (2023) 1–16.
[32] H.G. Abreha, M. Hayajneh, M.A. Serhani, Federated learning in edge computing: A systematic survey, Sensors 22 (2) (2022) 450.
[33] W. Shi, J. Cao, Q. Zhang, Y. Li, L. Xu, Edge computing: vision and challenges, IEEE Internet Things J. 3 (5) (2016) 637–646.
[34] C. Savaglio, G. Fortino, A simulation-driven methodology for IoT data mining based on edge computing, ACM Trans. Internet Technol. (TOIT) 21 (2) (2021) 1–22.
[35] J. Ren, G. Yu, Y. He, G.Y. Li, Collaborative cloud and edge computing for latency minimization, IEEE Trans. Veh. Technol. 68 (5) (2019) 5031–5044, http://dx.doi.org/10.1109/TVT.2019.2904244.
[36] Q. Xia, W. Ye, Z. Tao, J. Wu, Q. Li, A survey of federated learning for edge computing: Research problems and solutions, High-Confidence Comput. 1 (1) (2021) 100008, http://dx.doi.org/10.1016/j.hcc.2021.100008.
[37] F. De Rango, A. Guerrieri, P. Raimondo, G. Spezzano, A novel edge-based multi-layer hierarchical architecture for federated learning, in: 2021 IEEE DASC/PiCom/CBDCom/CyberSciTech, IEEE, 2021, pp. 221–225.
[38] I. Khan, A. Guerrieri, G. Spezzano, A. Vinci, Occupancy Prediction in Buildings: An approach leveraging LSTM and Federated Learning, in: The IEEE 2022 DASC/PiCom/CBDCom/CyberSciTech, IEEE, 2022.
[39] N. Alishahi, M.M. Ouf, M. Nik-Bakht, Using WiFi connection counts and camera-based occupancy counts to estimate and predict building occupancy, Energy Build. 257 (2022) 111759.
[40] H. Choi, C.Y. Um, K. Kang, H. Kim, T. Kim, Application of vision-based occupancy counting method using deep learning and performance analysis, Energy Build. 252 (2021) 111389.
[41] J. Vanus, O. M. Gorjani, P. Bilik, Novel proposal for prediction of CO2 course and occupancy recognition in intelligent buildings within IoT, Energies 12 (23) (2019) 4541.
[42] S. Mahjoub, S. Labdai, L. Chrifi-Alaoui, B. Marhic, 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 16 (4) (2023) 1641.
[43] G. Iacovone, G. Cerchiara, L. Cappiello, G. Strazza, E. Sangiorgio, D. D'Eliso, Needs analysis, protection, and regulation of the rights of individuals and communities for urban and residential comfort in cognitive buildings, in: F. Cicirelli, A. Guerrieri, A. Vinci, G. Spezzano (Eds.), IoT Edge Solutions for Cognitive Buildings, Springer International Publishing, Cham, 2023, pp. 75–102, http://dx.doi.org/10.1007/978-3-031-15160-6_4.
[44] M. Xia, D. Jin, J. Chen, Short-term traffic flow prediction based on graph convolutional networks and federated learning, IEEE Trans. Intell. Transp. Syst. (2022).
[45] V. Mothukuri, R.M. Parizi, S. Pouriyeh, Y. Huang, A. Dehghantanha, G. Srivastava, A survey on security and privacy of federated learning, Future Gener. Comput. Syst. 115 (2021) 619–640, http://dx.doi.org/10.1016/j.future.2020.10.007.
[46] Q. Li, Z. Wen, Z. Wu, S. Hu, N. Wang, Y. Li, X. Liu, B. He, A survey on federated learning systems: Vision, hype and reality for data privacy and protection, IEEE Trans. Knowl. Data Eng. 35 (4) (2023) 3347–3366, http://dx.doi.org/10.1109/TKDE.2021.3124599.
[47] N. Truong, K. Sun, S. Wang, F. Guitton, Y. Guo, Privacy preservation in federated learning: An insightful survey from the GDPR perspective, Comput. Secur. 110 (2021) 102402, http://dx.doi.org/10.1016/j.cose.2021.102402.
[48] C. Dwork, F. McSherry, K. Nissim, A. Smith, Calibrating noise to sensitivity in private data analysis, in: S. Halevi, T. Rabin (Eds.), Theory of Cryptography, Springer Berlin Heidelberg, Berlin, Heidelberg, 2006, pp. 265–284.
[49] K. El Emam, F.K. Dankar, Protecting privacy using k-anonymity, J. Am. Med. Inform. Assoc. 15 (5) (2008) 627–637, http://dx.doi.org/10.1197/jamia.M2716.
[50] A. Jamwal, R. Agrawal, M. Sharma, A. Giallanza, Industry 4.0 technologies for manufacturing sustainability: A systematic review and future research directions, Appl. Sci. 11 (12) (2021) http://dx.doi.org/10.3390/app11125725.
[51] S. Bag, J. Pretorius, Relationships between industry 4.0, sustainable manufacturing and circular economy: Proposal of a research framework, Int. J. Organ. Anal. 30 (4) (2022) http://dx.doi.org/10.1108/IJOA-04-2020-2120.
[52] J. Yuan, M. Xu, X. Ma, A. Zhou, X. Liu, S. Wang, Hierarchical federated learning through LAN-wan orchestration, 2020, arXiv preprint arXiv:2010.11612.
[53] F. De Rango, A. Guerrieri, P. Raimondo, G. Spezzano, HED-FL: A hierarchical, energy efficient, and dynamic approach for edge federated learning, Pervasive Mob. Comput. (2023) 101804, http://dx.doi.org/10.1016/j.pmcj.2023.101804.
[54] M.P. Canino, E. Cesario, A. Vinci, S. Zarin, Epidemic forecasting based on mobility patterns: An approach and experimental evaluation on COVID-19 data, Soc. Netw. Anal. Min. 12 (1) (2022) 116, http://dx.doi.org/10.1007/s13278-022-00932-6.
[55] X. Qiu, T. Parcollet, J. Fernandez-Marques, P.P.B. de Gusmao, Y. Gao, D.J. Beutel, T. Topal, A. Mathur, N.D. Lane, A first look into the carbon footprint of federated learning, 2021, arXiv preprint arXiv:2102.07627.
[56] Y. Zhang, P.J. Thorburn, Handling missing data in near real-time environmental monitoring: A system and a review of selected methods, Future Gener. Comput. Syst. 128 (2022) 63–72.
[57] L. Huang, J. Qin, Y. Zhou, F. Zhu, L. Liu, L. Shao, Normalization techniques in training dnns: Methodology, analysis and application, IEEE Trans. Pattern Anal. Mach. Intell. (2023).
[58] B. McMahan, E. Moore, D. Ramage, S. Hampson, B.A. y Arcas, Communication-efficient learning of deep networks from decentralized data, in: Artificial Intelligence and Statistics, PMLR, 2017, pp. 1273–1282.
[59] T. Li, A.K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, V. Smith, Federated optimization in heterogeneous networks, in: Proceedings of the 1st Adaptive & Multitask Learning Workshop, Long Beach, California, 2019.
[60] S. Park, Y. Suh, J. Lee, Fedpso: federated learning using particle swarm optimization to reduce communication costs, Sensors 21 (2) (2021).
[61] L. Mba, P. Meukam, A. Kemajou, Application of artificial neural network for predicting hourly indoor air temperature and relative humidity in modern building in humid region, Energy Build. 121 (2016) 32–42.
[62] G. Hinton, N. Srivastava, K. Swersky, Lecture 6a: Overview of Mini-Batch Gradient Descent, 2020, https://www.cs.toronto.edu/~tijmen/csc321/slides/lecture_slides_lec6.pdf.