IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 30 July 2024, accepted 18 August 2024, date of publication 26 August 2024, date of current version 3 September 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3449810

RESEARCH ARTICLE

A Real-Life Evaluation of Supervised and Semi-Supervised Machine Learning Approaches for Indirect Estimation of Indoor Occupancy

ALMA MENA-MARTINEZ[1], MANUEL DAVILA DELGADO[2], JOANNA ALVARADO-URIBE[1,3], AND HECTOR G. CEBALLOS[1,3]

[1]School of Engineering and Sciences, Tecnológico de Monterrey, Monterrey, Nuevo Leon 64700, Mexico
[2]Applied Artificial Intelligence (AI) and Business Transformation, Birmingham City University (BCU), B4 7BD Birmingham, U.K.
[3]Institute for the Future of Education, Tecnológico de Monterrey, Monterrey, Nuevo Leon 64700, Mexico

Corresponding author: Alma Mena-Martinez (a00834070@tec.mx)

This work was supported by the Tecnologico de Monterrey.

ABSTRACT Occupancy information is essential for space management, energy efficiency, and in times of the COVID-19 pandemic, for crowd control. Obtaining labeled data is challenging due to hardware limitations, privacy considerations, and the required underlying costs. This study demonstrates the benefits of using Semi-Supervised Learning (SSL) for occupancy estimation in enclosed spaces; which requires less labeled data than other Machine Learning (ML) methods. It presents an empirical comparison between Supervised ML and SSL models in three real-life university classrooms (uncontrolled conditions). The data was collected for three weeks at each classroom using an in-house developed Internet of Things (IoT) device that measures air temperature, relative humidity, and atmospheric pressure. The ground truth records were gathered through manual logging of occupancy levels. Datasets' sizes averaged 2350 entries with only 280 labeled instances per dataset. Support Vector Machine (SVM), Random Forest (RF), and Multi-Layer Perceptron (MLP) were used to define a performance baseline for supervised ML. Self-Training (ST) and Label Propagation (LP) were tested for SSL. ST achieved superior performance compared to baseline models (SVM, RF, MLP) with a highest average accuracy of 87.33% compared to SVM (86.66%). These results demonstrate the effectiveness of SSL for indirect occupancy estimation while reducing the need for extensive data collection and labeling.

INDEX TERMS Occupancy estimation, semi-supervised learning, environmental sensors, machine learning.

I. INTRODUCTION

Occupancy estimation is a key requirement in the optimization of Heating, Ventilation, and Air Conditioning (HVAC) and lighting systems in smart buildings [1], [2]. Accurate occupancy estimation enables optimal control of artificial lighting, natural daylight operation, and utilities' management in connection with safety, fire detection, and protection devices [1]. Furthermore, the COVID-19 pandemic created new needs for building operations' requirements. For instance, to implement social distancing rules, occupancy tracking and monitoring, dynamic HVAC controls, as well as stricter air cleaning requisites [3].

Over the years, several authors have used many occupancy measurement technologies based on sensors. Each sensor can obtain different occupancy information and each one of them has unique advantages and limitations for occupancy measurement. For example, direct methods, such as cameras and pattern recognition, can be used to count the number of occupants; nevertheless, personal privacy is a major problem for their deployment. The utilization of these invasive tools, which could gather extremely delicate information from users, has generated significant concerns. These issues have hindered the acceptance of occupancy detection systems in buildings [4].

The associate editor coordinating the review of this manuscript and approving it for publication was Gang Li.

On the other hand, indirect methods, such as Passive Infrared (PIR), ultrasonic, and acoustic sensors, can only be used to determine whether the room is occupied, rather than determining the actual number of occupants [5]. Ambient sensors, such as temperature, humidity, pressure, and CO_2 concentration, are used frequently in occupancy modeling because of their advantages [6], i.e.,: (i) smart environmental sensors are available for buildings at low cost, (ii) being non-intrusive leading to avoid privacy concerns, and (iii) do not require additional devices or human interventions.

Several methods and algorithms have been developed to estimate or detect the occupancy in buildings [7], [8]. The most used approaches are Hidden Markov Models, as well as the Supervised Machine Learning (ML) models, such as Support Vector Machine (SVM), Random Forest (RF), and Artificial Neural Networks (ANN) [7], [8].

Supervised ML has proven to be very efficient for classification tasks. Nevertheless, a significant disadvantage is that these models need large amounts of labeled training data to produce accurate estimations. Thus, data labeling has been a primary focus of research for ML tasks [9]. In the realm of indoor occupancy, data labeling consists of recording the number of room occupants. In binary classification, it is determined only if the room is occupied or empty. For occupancy levels, the number of occupants determines predefined occupancy levels (i.e., low, medium, or high) depending on the occupancy capacity of the room. In many situations, obtaining high-quality labeled information is expensive and time-consuming as it requires human input and additional devices, such as cameras [10].

In this study, we investigated the use of Semi-Supervised Learning (SSL) for occupancy estimation as it presents significant advantages for applications with limited access to large labeled datasets. This study contributes to the research and practitioner body of knowledge by:

- Evaluating effectiveness of SSL models for indoor occupancy estimation in enclosed spaces.
- Pioneering a systematic comparison between supervised and SSL models for occupancy estimation, using the same datasets collected from the three selected scenarios.
- Focusing on using environmental sensors (temperature, humidity, pressure) to achieve accurate occupancy estimation while minimizing costs and preserving the users' privacy.

The rest of the manuscript is organized as follows. Section II introduces the background and Section III presents the related work. The methodology used for this research is presented in Section IV. Section V introduces the experimental setup and Section VI reports the results. Finally, the conclusion is drawn in Section VII.

II. BACKGROUND

This section provides an overview of the most commonly used ML supervised models alongside a brief introduction to the concept of SSL and two model implementations for indoor occupancy.

A. SUPERVISED MACHINE LEARNING

In supervised ML, a model learns from labeled training data, allowing it to make a prediction about future data [11]. The most common models used for indoor occupancy estimation and detection are briefly explained below.

- **Support Vector Machine (SVM).-** The algorithm is commonly used among researchers for classification and regression tasks. SVM has the ability to handle non-linear problems, giving good results even if there is not enough information about the data. Nevertheless, it is difficult to choose the appropriate kernel solution function and training time is long when using large datasets [12]. The SVM draws margins as boundaries between the classes in the provided datasets. The margins are defined as the distance between two supporting vectors separated by a hyperplane. Furthermore, the algorithm assumes that the data is linearly separable so that the weight associated with support vectors can be drawn easily and the margin computed. Therefore, the assumption makes regularization possible [13]. Mathematically Equation 1 shows the functions that SVM aims to minimize. Where w represents the weight vector, which indicates the direction of the hyperplane. C is a regularization parameter that controls the trade-off between having a smooth decision boundary and classifying the training points correctly. While the term ς_i measures how much a data point i violates the margin or ends up on the wrong side of the decision boundary. Therefore $\sum_{i=1}^{n} \varsigma_i$ is the sum of slack variables ς_i for each data point and $\min \frac{1}{2} w^t w$ is a regularization term that penalizes the magnitude of the weight vector w.

$$\min \frac{1}{2} w^t w + C \sum_{i=1}^{n} \varsigma_i \tag{1}$$

  Equation (1) is subject to $y_i(w^T \phi(x_I) + b) \geq 1 - \varsigma_i, \varsigma_i \geq 0, i = 1, \ldots, n$ where given training vectors $x_i \in R^p$, $i = 1, \ldots, n$ in two classes, and a vector $y \in (1, -1)^n$, the goal is to find $w \in R^p$ and $b \in R$ such that the prediction given by $(w^T \phi(x_I) + b)$ is correct for most samples [13]. b is the bias term that shifts the hyperplane away from the origin and allows the SVM to better fit the data, and ϕ is the kernel function.
- **Random Forest (RF).-** RF is capable of handling non-linear parameters, usually robust to outliers, stable, and less affected by noise. Therefore, it is extremely flexible and has very high accuracy. Nevertheless, it requires a lot of computation and is time-consuming [12]. A RF is a classifier that involves a collection of tree-structured classifiers $h(x, \theta_k), k = 1, \ldots$ where the θ_k are independent identically distributed random vectors and each tree casts a unit vote for the most popular class at input

x. The margin function is $mg(X, Y) = av_k I(h_k(X) = Y) - max_{j \neq Y} av_k I(h_k(X) = j)$ where $I(\bullet)$ is the indicator function. The margin measures the extent to which the average number of votes at X, Y for the right class exceeds the average vote for any other class. The larger the margin, the more confidence in the classification. The generalization error is given by $PE* = P_{X,Y}(mg(X, Y) < 0)$, where the subscripts X, Y indicate that the probability is over the X, Y space [14].

- **Multi-Layer Perceptron (MLP).-** Characterized by its feed-forward structure, the MLP transmits information forward from input to output layers without any feedback loops. Layers that are not directly connected to the environment are called hidden. The power of the MLP comes from non-linear activation functions, which unlock the ability to model complex relationships beyond the limitations of linear models. Common choices for activation functions include the single-pole (logistic) sigmoid and the hyperbolic tangent function [15]. Therefore, MLP learns a function $f(\bullet) : R^m \rightarrow R^o$ by training on a dataset, where m is the number of dimensions for input and o is the number of dimensions for output. Given a set of features $X = x_1, x_2, \ldots, x_n$ and a target y, it can learn a nonlinear function approximator for either classification or regression. The leftmost layer, known as the input layer, consists of a set of neurons $\{x_i \mid x_1, x_2, \ldots, x_m\}$ representing the input features. Each neuron in the hidden layer transforms the values from the previous layer with a weighted linear summation $w_1x_1 + w_2x_2 + \ldots + w_mx_m$, followed by a non-linear activation function $g(\bullet) : R \rightarrow R-$ like the hyperbolic tan function. The output layer receives the values from the last hidden layer and transforms them into output values [16].

B. SEMI-SUPERVISED MACHINE LEARNING

In the standard SSL problem, a model is given two sample sets, the labeled set XL together with its label set YL, and the unlabeled set XU. XL consists of I sample points $XL = X_1, \ldots, X_I$ with the corresponding labels as $YL = Y_1, \ldots, Y_I$, and XU consists of u sample $XU = X_{I+1}, \ldots, X_{I+U}$. The label for XU is $YU = Y_{I+1}, \ldots, Y_{I+U}$, which is unknown. It is also convenient to denote $X = XL \cup XU$, and similarly, $Y = YL \cup YU$, where all $X_i \in X$ and $Y_i \in Y$. Let $n = I + u$. The X can be a regular subset of R_d, a smooth manifold in R_d, or sequences in language problems. Similarly, Y can be binary class labels ± 1, real-valued numbers, or complex structured output. The data in SSL is also called partially observed data [10].

A necessary condition of semi-supervised learning is that the underlying marginal data distribution $p(x)$ over the input space contains information about the posterior distribution $p(y \mid x)$. If this is the case, one might be able to use unlabeled data to gain information about $p(x)$, and thereby about $p(y \mid x)$ [17].

However, how $p(x)$ and $p(y \mid x)$ interact is not always the same. This has given rise to the following semi-supervised learning assumptions [17]:

- The smoothness assumption states that for two input points $x, x' \in X$ that are close to the input space, the corresponding labels y, y' should be the same.
- The low-density assumption implies that the decision boundary of a classifier should preferably pass through low-density regions in the input space.
- Manifold assumption states that the marginal distribution of the data $p(x)$ is supported on a low-dimensional manifold embedded in a high-dimensional ambient space. The conditional distribution $p(y \mid x)$ is smooth, as a function of x, with respect to this low-dimensional manifold.
- Cluster assumption states that the marginal distribution of the data $p(x)$ has multiple clusters, and example points of the same cluster are likely to be of the same class.

Furthermore, the optimization procedure is distinguished between predicting the labels on future test data and predicting the labels on the unlabeled instances in the training sample. An inductive method infers a function $f(x) : X \rightarrow Y$ on the entire space X and afterward returns the evaluations $f(x_i)$ at the test points. In contrast, the transduction method, a function $f(x) : Xu \rightarrow Y$ consists of directly estimating the finite set of unlabeled sample points [18].

Therefore, SSL presents the following advantages: (i) it uses both labeled and unlabeled data, (ii) it may achieve higher or the same level of performance as supervised learning using fewer labeled instances, and (iii) it leads to decrease costs due to the reduction on the labeling data effort.

1) SELF-TRAINING (ST)

The ST is a method based on the Expectation-Maximization (EM) approach that considers a probability density function $p(x \mid \theta)$ which is governed by the set of parameters θ. This method can be explained as follows. Let the complete data be $Z = (X, Y)$, where Y is either the missing data or some hidden variables. Assume a joint density function $p(z \mid \theta) = p(x, y \mid \theta) = p(y \mid x, \theta)p(x \mid \theta)$ [10]. The likelihood function for Z is $L(\theta \mid Z) = p(X, Y \mid \theta)$. Due to the missing values, which are random variables, $L(\theta \mid Z)$ is random. The algorithm first finds the expected value of the log-likelihood $logp(X, Y \mid \theta)$ with respect to the unknown data Y, given the data X and the current parameter estimate $\theta^{(t-1)}$ [10].

$$Q(\theta, \theta^{(t-1)} = E[logp(X, Y \mid \theta) \mid X, \theta^{(t-1)}] \qquad (2)$$

The computation of the Equation (2) is called the E-step. The first argument θ is the parameter to be optimized in the maximization of the likelihood function. The second argument $\theta^{(t-1)}$ is the parameter from the previous estimate. This means the expectation is taken with respect to the distribution determined by the data X and the density function with parameter $\theta^{(t-1)}$. The second step is called the M-step,

which is to maximize the expectation from the E-step: $\theta(t) = argmaxQ\theta, \theta^{(t-1)}$. These two steps are repeated as necessary. Each iteration is guaranteed to increase the log-likelihood and the algorithm is guaranteed to converge to a local maximum of the likelihood function [10].

2) LABEL PROPAGATION (LP)

Label propagation is a family of semi-supervised algorithms based on a graph representation of the dataset. In particular, if we have N labeled points (with bipolar labels $+1$ and -1) and M unlabeled points (denoted by $y = 0$), it is possible to build an undirected graph based on a measure of geometric affinity among samples. If $G = \{V, E\}$ is the formal definition of the graph, the set of vertices is made up of sample labels $V = \{-1, +1, 0\}$, while the edge set is based on an affinity matrix W (often called adjacency matrix when the graph is unweighted), which depends only on the X values, not on the labels. The affinity matrix is normally symmetric and square with dimensions equal to $(N+M)*(N+M)$ and it can be obtained using k-nearest Neighbor (k-NN) or Radial Basis Function (RBF) kernel. In the complete steps of the label propagation algorithm first select an affinity matrix type (k-NN or RBF) and compute W. Second, compute the degree matrix D which is a diagonal matrix where each non-null element represents the degree of the corresponding vertex. This can be the number of incoming edges, or a measure proportional to it. Subsequently, define $Y^{(0)} = Y$ and define $Y_L = \{y_0, y_1, \ldots, y_N\}$. Finally, iterate until convergence (when label assignments stop changing significantly between iterations) [19].

III. RELATED WORK

Considering both supervised ML and SSL approaches, the number of investigations using supervised algorithms is far greater, only two articles were found that use SSL approaches. Of these two articles, neither addresses the estimation of occupancy levels nor uses environmental sensors or applies the ST and LP algorithms. Table 1 summarizes the key aspects of the related work reviewed.

A. INDOOR OCCUPANCY ESTIMATION THROUGH SUPERVISED ML

Supervised models are widely used for indoor occupancy detection and estimation. For instance, to predict the number of occupants Choi et al. [20] collected indoor temperature and CO_2 values, as well as devices' log data in a smart building. The training data was collected for seven days and the testing data for two days. The ground truth data was collected using image data every five minutes by a camera; then, to count detected users the Openpose library was used. The trained models were k-nearest neighbors (k-NN), Gaussian Process (GP), Random Forest (RF), Bayesian Ridge (BR), and Multi-Layer Perceptron (MLP). The best model was MLP with a Mean Absolute Error (MAE) of 0.39. On the other hand, Abade et al. [21] sensed environmental data, such as temperature (in and out), sound, CO_2, and light intensity. The experiments were conducted in a room for two weeks. A simple mechanism with two buttons (blue to enter and red to leave) was used to create the ground truth and register when a person enters or leaves the room. The ML models used to determine whether an occupant was in the room (y = 1) or not (y = 0), as well as to estimate the number of occupants in a room were Linear Regression (LR), SVM, and ANN. Achieving SVM the best F1-score of 29.72% for the multi-classification task.

In addition, other supervised multi-class classifiers were trained to estimate occupancy based on levels such as low, medium, or high. For example Das et al. [22] used environmental sensors to measure CO_2, temperature, humidity, and motion. The ground truth was manually assigned by an operator. The records involved the three classes of occupancy levels: empty when there are no occupants; moderate, when there are four or fewer occupants, and high when there are more than four occupants. The ML models were LR, SVM, and Cosine similarity. The best accuracy achieved considering all the sensor measures was 97.9% by SVM and the Cosine Similarity approach. Similarly, Vela et al. [23] carried out experiments to estimate the occupancy levels (empty, low, medium, and high) in a university fitness gym and a residence living room. The experiments were carried out for 15 days and 11 days, respectively. For the ground truth data, the levels were reported by the gym administrator, while a camera was used to take photos of the living room every 10 min, and afterward, the number of people was annotated. The supervised ML models were k-NN, SVM, and Decision Trees (DT). The higher accuracy was 98.8% using the k-NN algorithm.

Recently, Aliero et al. [24] used a combination of camera and environmental sensing data, along with interactive learning and a rule-based classifier. The study attained a robust occupancy count by considering a multimodal input to a single output regression model. The proposed solution was tested in a living room. The results showed that the RF model achieved high accuracy values of 99.7% for predicting the occupants' presence and 99.35% for predicting their numbers. Likewise, Dutta and Roy [25] introduced the OccupancySense model, which aims to detect human presence and forecast indoor occupancy count. The core of the OccupancySense design utilized CatBoost. The efficacy of the model was verified through a real-life case study conducted in a University research lab. Data on indoor air quality (IAQ) including PM2.5, PM10, CO_2, temperature, and humidity was gathered. The performance of CatBoost indicated that it can accurately detect occupancy and predict headcount, along with the density of room occupancy, with rates of 99.85%, 93.2%, and 95.6%, respectively.

Lastly, Banihashemi et al. [26] explored the use of CO_2 concentration, air temperature, RH, indoor air quality (IAQ), sound pressure level (SPL), and illuminance data for detecting occupancy in office rooms and assessed the impact of the quantity of ground truth data. RF, XGBoost, and Dense

Neural Networks (DNN) models were trained on this dataset. The findings of this research suggest that by using six days of ground truth data, it is possible to achieve accuracy and F1-score for occupancy detection of above 0.95 and 0.93, respectively.

B. INDOOR OCCUPANCY ESTIMATION THROUGH SSL

SSL approach is not common in the realm of indoor occupancy. Some of the researchers that have dabbled are Shih et al. [27] who used a semi-supervised approach to estimate the number of occupants. The approach is based on the Weighted Principal Component Analysis (WPCA), which requires minimal training data, using a wide-band ultrasonic signal into a room to later process the superposition of the reflections recorded by a microphone. The collected data was done over two weeks and the model had an average error of 10% in terms of the maximum capacity. Another researcher was Jin et al. [28] who installed a module sensor that includes an ultrasonic sensor to detect user presence, an acceleration sensor for the movement of chairs, and a Wi-Fi access point to detect the presence of smartphones. The ground truth data was recorded every 5 minutes during the participation of four users in the office at different locations (user at the desk and presence in the office area except for his/her desk). The energy consumption monitoring was collected for 11 months while the data from the module sensor and ground truth data were recorded only for 30 days and 13 days, respectively. The approach Presence-Sense is a zero-training algorithm proposed in the study to infer presence. The accuracy obtained was from 71% to 97%. On the other hand, Wei et al. [29] proposed the new parameter estimation models based on the Maximum Likelihood approach and Bayesian analysis. These models calculate the occupant number independently of the true number of occupants. The proposed models were evaluated by collecting real-time CO_2 concentration, energy consumption of facilities, and make-up air system in a commercial office that contains 36 occupants.

The existing research indicates that Supervised ML is the primary approach used for indoor occupancy estimation. Additionally, environmental sensors, such as CO_2, temperature, light, and motion, are commonly employed in these studies. Therefore, the goal of this study is to fill the gaps in previous research by exploring SSL, like ST and LP, along with non-intrusive sensors (air temperature, RH, and pressure). Furthermore, in contrast to most existing studies, which rely on fully labeled datasets, this research uses both a few labeled and large amounts of unlabeled data to predict indoor occupancy levels in real-life scenarios.

IV. METHODOLOGY

The methodology includes activities related to the data collection, data preprocessing, as well as the selection of ML models and their evaluation. The main activities are illustrated in Fig. 1.

A. DATA COLLECTION AND STORAGE

This first stage develops a robust data collection guideline for continuous indoor occupancy monitoring. The considered tasks are:

- **Data Collection.-** The data was collected using an IoT device designed in-house (patent pending: MX/a/2022/016112) with the idea of facilitating a non-intrusive sensor to preserve users' privacy. The device is constituted of a low-cost and low-energy consumption sensor designed for meteorological monitoring. This sensor allows measuring relative humidity (%), temperature (°C), and atmospheric pressure (hPa) every minute. Sensor ranges and accuracy values are listed below: pressure 300 to 1100 hPa (± 1 hPa accuracy), temperature −40 to 85 °C (± 1 °C accuracy), and relative humidity 0% to 100% (± 3% accuracy) [23]. Furthermore, the device has three buttons that the occupants can press to indicate the real occupancy level segmented into categories (low, medium, and high), see Fig. 2. The other two buttons are used to test the device, one for sending data and the other for enabling the ML model on the device to make real-time predictions (not considered for this study).
- **Cloud storage.-** The data is sent through Wi-Fi to a cloud database, which provides centralized storage, scalability, and accessibility. An IoT communication at Amazon Web Services (AWS) using the Message Queuing Telemetry Transport (MQTT) protocol was previously done in order to send the device measurements as JavaScript Object Notation (JSON) documents to the DynamoDB database [23], [30]. All traffic to and from AWS IoT is sent securely over Transport Layer Security (TLS). AWS cloud security mechanisms protect data as it moves between AWS IoT and other AWS services [31]. Finally, a Comma-Separated Values (CSV) file is downloaded to be cleaned and preprocessed.
- **Ground truth data.-** The occupancy is measured through periodic visual observations, which involve a person monitoring physical occupancy in the enclosed area. The manual recording of occupancy data is intended to provide true values of occupancy; therefore, only it is counted the number of occupants in the room. The records are carried out when the scenarios have a change in their conditions (door/window status open or closed), as well as when the number of occupants changes. The ground truth records include the date, start-end hour (to refer to the period in which the conditions do not change), number of occupants, door status, windows status, air conditioning system status, and temperature (if it applies). These variables and records are added manually to the dataset.
- **Data protection and privacy.-** The data was protected in accordance with the (EU) 2016/679 General Data Protection Regulation (GDPR), the Data Protection Act 2018 (or any successor legislation), and any

TABLE 1. Summary of related work reviewed.

Author	Time	O.R.	Scenario	Sensor	G.T.	Dataset	L.T	Testing	Approach	Result
Banihashemi *et al.* **[26]**	9 m	1	Off.	temp, light, CO_2, IAQ, hum, SPL	B	115199	–	–	RF, XGBoos, DFNN, CDBLSTM	F1 score 93%
Aliero *et al.* **[24]**	28 d	1, 2	Off.	temp, hum, CO_2, illumi-nance, thermal camara sensor	–	2668	–	–	RF, NBC, SVM, ANN, LR	F1 score 87.66% - 99.35%
Dutta *et al.* **[25]**	3 m	1, 2	Laboratory	PM2.5, PM10, CO_2, temp, hum	App	129600	–	–	CatBoost, LR, NB, DT, RF, SVM, MLR	F1 score 88.85% - 95.2%, MAE 0.256 – 1.89
Wei *et al.* **[29]**	5d	2	Off.	CO_2, PM2.5, temp, hum, e.c.	C	240	–	–	Bayesian analysis, Maximum Likelihood	NRMSE 0.1115 and 0.1235
Choi *et al.* **[20]**	9d	2	Off.	CO_2	C	4337	3035	1302	k-NN, GP, RF, BR, MLP	MAE 0.21 – 1.84
Das *et al.* **[22]**	-	3	Off.	CO_2, temp, hum, mot.	M	-	-	-	SVM, LR. Cosine Similarity	Acc 79.7% – 98.9%
Vela *et al.* **[23]**	6d	3	Gym	temp., hum., press.	M	857	821	36	k-NN, SVM, DT	Acc 97.2% – 100%
	11d		Living Room		C	29751	23800	5951		Acc 83.5% – 98.8%
Abade *et al.* **[21]**	2w	1 , 2	Off.	temp, CO_2, Noise, light	B	14400	10080	4320	LR, SVM, ANN	F1 Binary 0-95.60% Multi-class 24.43-29.72%
Shih *et al.* **[27]**	2w	2	10 rooms (class, lab)	Ult. Horn, mic.	C	18200	500	-	WPCA	Avg. Error 0.26 – 2.36
Jin *et al.* **[28]**	11m, 1m	1	Off.	elec met., ult, acc. ,Wi-Fi point	M	–	–	–	Pres.Sense	Acc 71% – 97%

The symbol - means information not available.
Columns: *Time:* **data collection period in days(d), weeks(w) or months (m);** *Occupancy resolution (O.R.):* **detection presence (1), number of occupants (2), occupancy levels(3);** *Scenario:* **office (off.), University gym (Gym), residential living room, classroom (class), and laboratory (lab);** *Sensor:* **Carbon dioxide sensor (CO_2), Temperature (temp), Humidity (hum), motion sensor (mot), Barometric Pressure (press), Ultrasonic horn (Ult.Horn), Microphone (mic), Electricity consumer meter (elec met), ultrasonic sensor (ult), accelerometer (acc), and Wi-Fi points access (Wi-Fi point);** *Ground truth (G.T):* **camera (C), manually (M), mechanism with buttons (B);** *Dataset:* **total number of entries per dataset;** *Labeled training data (L.T):* **total of labeled entries used for training;** *Testing:* **total of entries used for testing;** *Approach:* **k-Nearest Neighbor (k-NN), Gaussian Process (GP), Random Forest (RF), Bayesian Ridge (BR), Multi-Layer Perceptron (MLP), Linear Regression (LR), Support Vector Machine (SVM), Artificial Neural Network (ANN), Decision Trees (DT), Weighted Principal Component Analysis (WPCA), Presence Sense (Pres.Sense);** *Result:***Accuracy (Acc), F1-Score (F1), Mean Absolute Error (MAE).**

other legislation directly relating to privacy laws that apply (together "the Data Protection Legislation") [32]. In order to address user privacy concerns, any additional personal information or user behaviors were not collected. Furthermore, individuals were informed about the presence of these devices and the purpose of data collection. Likewise, strong security measures were implemented to protect data from unauthorized access, breaches, and misuse. Since the testbed scenarios involved university classrooms, it was determined that the environmental data and occupant counts did not have a significant impact on user privacy.

B. DATA PREPROCESSING

To begin with the training of the models, it is necessary to prepare the data. Therefore, a pipeline with meticulous attention to innovation and effectiveness is proposed to ensure the integrity and reliability of the information used for

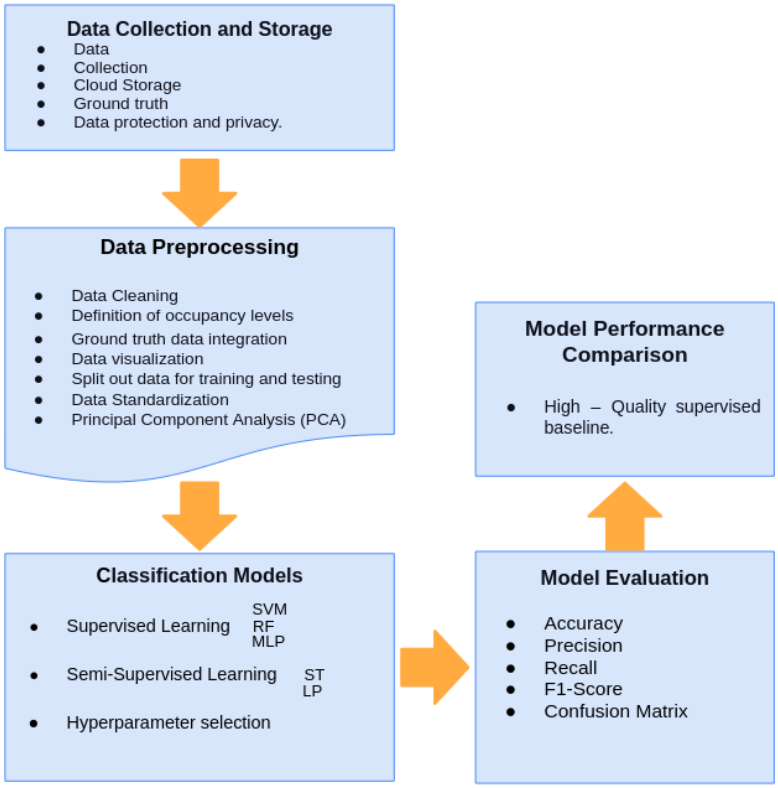

FIGURE 1. Experimental methodology.

FIGURE 2. The in-house developed environmental IoT device.

subsequent analysis The following tasks were carried out for each CSV file downloaded per scenario:

- **Data cleaning.-** To ensure reliable data, first, the raw data is separated by MAC address. Furthermore, to avoid having too many instances with no occupants, the data is selected based on weekdays (Monday Friday), and work hours (7 am 7 pm) [33]. Afterward, the missing values are counted. If the missing values are less than 1%, they are removed [34]. In contrast, they are filled. On the other hand, outliers are not removed because, during the data collection, the indoor conditions were not controlled. As a consequence, the models should learn using this information from real scenarios.
- **Definition of occupancy levels.-** To provide a consistent ground truth, the indoor occupancy levels were defined using the maximum occupancy of the selected scenario and divided into three levels (low, medium, and high) [23], [35]. Mathematically, the interval of occupants (Occ) for the levels can be expressed as $Occ = \frac{roommax.capacity}{3}$. Subsequently, the occupancy levels are established following these rules: $Empty = 0, 1 \leq Low \leq Occ, Occ + 1 \leq Medium \leq Occ * 2$, and $(Occ * 2) + 1 \leq High$.
- **Ground truth data integration.-** Followed, to create labeled data the ground truth related to the number of occupants (Num. Occ) is added manually. Then, a conditional formula (if) is used to transform the Num. Occ into the labels (E, L, M, H). That is, when there were no occupants (value 0), these values were represented as "E". On the other hand, if the number of occupants is equal to or less than the established "low" level, the values in this range are represented as "L". Furthermore, all the values equal to or more than the established "high" level is represented as "H". Finally, the rest of the values should fall in the "medium" level category which is represented as "M". For all the unlabeled data the label assigned was -1.
- **Data Visualization.** To identify potential anomalies or patterns a graph of the timeline of the environmental variables is plotted.
- **Split out the data for training and testing.-** To avoid over-fitting and ensure unbiased model evaluation, each dataset was divided into 20% for testing and 80%

for training using a random method [36], [37]. Due to the size of labeled data, a validation dataset or cross-validation method is not considered in this study. For supervised models, the dataset involves only labeled data. In contrast, for the semi-supervised models, the training data involves both labeled and unlabeled data. Unlabeled data is represented by "-1".

- **Data Standardization.-** Many ML algorithms are sensitive to the scale and magnitude of the features. Thus, having features within a similar scale allows comparison of feature importance and helps algorithms converge faster, hence improving performance and training times [38]. Therefore, all the data is standardized using the StandardScaler function from the scikit-learn library [16]. Standardization is the process of centering the variable at zero and standardizing the variance to 1. The standard score of a sample x is calculated using Equation (3).

$$z = \frac{x - u}{s} \tag{3}$$

where u is the mean of the training samples and s is the standard deviation of the training samples.

- **Principal Component Analysis (PCA).-** PCA is conducted to combine highly dimensional features into independent principal components through an orthogonal linear transformation that explains a maximum amount of the variance [16], [39], as well as to reduce the noise from the data by identifying the principal components that capture the most significant variations in the dataset. In this study, the three components are retained; therefore, the essential information remains while some of the noise is potentially reduced [40], [41]. Mathematically, the transformation is defined by a set of size l of p-dimensional vectors of weights or coefficients $W_k = (W_1, \ldots, W_p)_k$ that map each row vector x_i of X to a new vector of principal component scores $t_i = (t_1, \ldots, t_l)_i$ given by $tk_i = x_i * w_k$ for $i = 1, \ldots, n$ $k = 1, \ldots, l$. In such a way that the individual variables $t_1, \ldots, t_l$ of t considered over the data set successively inherit the maximum possible variance from X, with each coefficient vector w constrained to be a unit vector (where l is usually selected to be strictly less than p to reduce dimensionality).

C. CLASSIFICATION MODELS

To estimate the indoor occupancy level, two different approaches of ML are selected: 1) Supervised learning, which is often used in the reviewed literature [22], [42], [43], and 2) Semi-supervised learning (SSL), which is rarely applied to this topic [44] and is halfway between supervised and unsupervised learning. In addition to unlabeled data, the SSL algorithms are provided with some supervision information but not necessarily for all examples [45].

- **Supervised Machine Learning.-** To make a baseline, the SVM, RF, and MLP algorithms are selected to classify the four occupancy levels (empty, low, medium, and high). These are the most used supervised ML algorithms in the literature related to occupancy estimation.
- **Semi-Supervised Machine Learning.-** The ST and LP were selected based on the literature review [17] and the availability of libraries in Python version 3.8.10. The main objective of applying semi-supervised models is to evaluate their potential in classifying indoor occupancy levels (E, L, M, H).
- **Hyperparameters selection.-** To find the best model configuration for optimal results the hyperparameters for each model were selected using a GridSearchCV function [16]. The initial values are presented in Table 2, which were set up according to the literature review [21], [23], [46]. If Λ is a set indexed by K configuration variables (for example, for neural networks it would be the learning rate, the number of hidden units, the strength of weight regularization, etc.), then grid search requires that a set of values for each variable is chosen $(L^1, \ldots, L^k)$ [47]. In a grid search, the set of trials is formed by assembling every possible combination of values, thereby the number of trials in a grid search is $S = \prod_{k=1}^{K} |L^{(k)}|$ elements. The function carries out a cross-validation for evaluating the models. To choose the winning parameters, the function reports a weighted average of all the test set scores, in which each one is weighted by the probability that its $\lambda(s)$ is, in fact, the best.

TABLE 2. Parameters established for the GridSearchCV function.

Estimator	Parameter	Values
SVM	C	1, 10, 100, 1000
	gamma	1, 0.1, 0.01, 0.001, 0.0001
	kernel	rbf
RF	n_estimators	Range(2, 30)
	max_depth	Range(2,15)
	criterion	gini, entropy
MLP	hidden layer	[(range(5,15), range(5,15))], [(range(5,15), range(5,15), range(5,15))], [(range(5,15))]
	activation	tanh, relu
	solver	adam
	alpha	0.0001, 0.05, 0.1, 1, 10
	learning rate	constant
LP	kernel	k-NN, rbf
	n_neighbors	(range(2, 20)
	gamma	(range(-2, 20)

D. MODEL EVALUATION

To measure the performance of the models, the metrics used are Accuracy, Precision, Recall, F1-Score, and Confusion matrix. Employing multiple metrics offers a multifaceted view of model performance, avoiding overreliance on a single metric.

- **Confusion matrix.-** It shows a more detailed breakdown of correct and incorrect classifications for each class [48]. The confusion matrix is constructed by comparing the predicted class with the known ground truth class. Each row of the matrix represents the true label (TL) and each column of the matrix represents the prediction of the classifier. For each data instance X with TL class Y and predicted class Z, the matrix cell corresponding to the $Y - th$ row and $Z - th$ column is incremented, counting the number of times that the object of class Y is assigned to a class Z. In this way, the raw confusion matrix is constructed, where diagonal elements of the matrix represent the number of accurate classifications for each class, while off-diagonal elements represent misclassifications, as shown in Fig. 3.

	True Labels	
Predicted Labels	TP	FP
	FN	TN

FIGURE 3. Construction of the Confusion Matrix.

- **Accuracy.-** The accuracy metric measures how often the classifier makes the correct prediction. To interpret the result, the accuracy is 1.0 when all the predicted labels match the true labels; otherwise, it is 0 [16]. To estimate the accuracy of a test, it is necessary to calculate the True Positive (TP) predictions that represent the labels predicted correctly as the positive class and the True Negative (TN) predictions that represent the labels predicted correctly as the negative class in all evaluated cases [49]. Mathematically, this can be stated as shown in the Equation (4).

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{4}$$

where false positive (FP) is the number of cases incorrectly identified as the positive class and false negative (FN) represents the number of cases incorrectly identified as the negative class.

- **Precision and Recall.-** Precision is the fraction of all the items that the classifier predicted to be relevant, but are truly relevant. Whereas, Recall is the fraction of all the truly relevant items that are successfully classified [50]. Both metrics are formally represented by Equation (5) and Equation (6), respectively.

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

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

- **F1-Score.-** This metric can be interpreted as a weighted harmonic mean of the precision and recall. Both precision and recall contribute equally to the F1 score's calculation. The score can achieve its highest value at 1 and lowest at 0 [50]. The formula is expressed in Equation (7).

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

E. MODEL PERFORMANCE COMPARISON

To demonstrate the potential of the SSL techniques, a systematic comparison between both learnings (supervised and SSL) is conducted.

- **HighQuality Supervised Baseline.-** This metric establishes a reference point for comparison, ensuring evaluation against a well-established and validated approach. The comparison of the SSL algorithm uses a supervised algorithm as its baseline [45]. If M is the accuracy of a supervised model and M' is the accuracy of an SSL model the difference between both helps to decide which model is better. If $M - M' > 0 \rightarrow$ M is better, else $M - M' < 0 \rightarrow$ M' is better.

V. EXPERIMENTAL SETUP

In this section, the setup of the experiments and the deployment of devices are described. All the scenarios presented were carried out at the University of the West of England (UWE), Bristol, United Kingdom. The classrooms, offices, and meeting rooms are named with the following nomenclature #X###. The first number of the room indicates the floor, then X means the name of the building, and the three last digits correspond to the room number. Firstly, each experiment is described in detail, and, at the end of this Section, key information related to the experiments is summarized in Table 3.

A. EXPERIMENT 1

The experiment was carried out during October 10 28, 2022, at PC Lab 6 × 233 (Fig. 4). The room has 26 computers distributed on three tables. Furthermore, the room is equipped with a desktop for the professor, four whiteboards on the front and back walls, one convector heater, and a projector. The area is 76 m^2. During the experiment, the HVAC system maintained an average temperature of 23.5°C and the door mostly stayed closed. Whereas the average of the external weather conditions was around 18.85°C and 56% of RH.

Three devices were deployed (shown as orange boxes in Fig. 4) and each device was placed on the center of the tables. The ground truth data was recorded manually during the first and second weeks of data collection. Finally, according to the maximum occupancy, the occupancy levels were established as "E"= 0, "L"= 1 −8, "M"= 9- 17, and "H" = +18.

B. EXPERIMENT 2

This experiment was carried out in the classroom 4Q04 (Fig. 5) from October 19th to November 11th, 2022. Four devices were installed on the walls of the classroom (shown as orange boxes in Fig. 5) at an approximate height of 1.20 m from the ground. The classroom has a surface of 109.18 m^2 and a maximum capacity of 56 students. Hence,

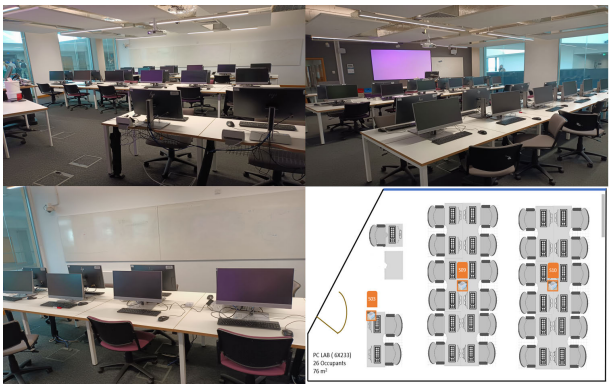

FIGURE 4. Setup of the first experiment in the PC Lab 6 × 233.

FIGURE 6. Setup of the third experiment in the Classroom 5 × 102.

the occupancy levels were stated as "E" = 0, "L" = 1 18, "M"= 19 36, and "H"=+37. The ground truth data was recorded during the first week of data collection. The classroom is equipped with three convector heaters on the back wall, an HVAC system, two projectors, four whiteboards, and a computer desk for the professor. Furthermore, the classroom has eight awning windows (100 × 50 cm), generally, three of them were opened during the experiments. Nevertheless, it was not possible to verify the windows or HVAC system status all the time. Likewise, the average of the external weather conditions was around 18°C and 67.50% of RH.

FIGURE 5. Setup of the second experiment in the Classroom 4Q04.

C. EXPERIMENT 3

Four devices were installed on the tables of the 5 × 102 room (shown as orange boxes in Fig. 6) from November 1st to 24th, 2022. The classroom has an area of 65.12 m^2 and a maximum capacity of 25 occupants. Therefore, the occupancy levels were stated as "E"=0, "L"= 1- 8, "M"= 9 16, "H"= +17. The ground truth data was recorded during the second and third weeks of data collection. The classroom is equipped with four flat screens on the tables, six whiteboards along the walls, one projector, one convector heater, and four awning windows that were opened most of the time. The average of the external weather conditions was around 12.89°C and 70% of RH.

VI. RESULT

Firstly, for each experimental scenario, measurements were collected and labeled correctly, despite facing challenges during device deployment and ground truth collection. The most common technical issues were the users sometimes plugged off the IoT device, as well as Wi-Fi network failures. Likewise, the devices had to be placed on tables or walls where the electrical contacts were available. Furthermore, it was not always possible to check the classroom conditions before collecting the ground truth. On the other hand, each CSV file was cleaned based on the methodology outlined in Section IV-B. As the data from devices were processed individually, no data-filling methods were necessary for handling missing values, ensuring that only the available measurements in the CSV files were considered.

All the datasets used to train the models have the same number of columns, which correspond to Date (Day and Time), Temperature (°C), Humidity (%), Pressure (hPa), Occupancy (Number of occupants), Label (occupancy levels: E, L, M, H, -1). Table 4 presents the number of entries and the amount of labeled and unlabeled data. Furthermore, it summarizes the training and testing size, as well as the classes used for training. The "Labeled Training Data" column presents the amount of labeled data used for supervised models. In contrast, the training data for semi-supervised models involves the values in the columns "Unlabeled" and "Labeled Training Data". Therefore, the dataset for semi-supervised learning is larger than the dataset for supervised learning.

All of the graphs reflect the readings obtained from the IoT devices. According to the statistical analysis, the data presented some outliers due to the natural variations in real-life scenarios. Fig. 7 shows the statistical analysis of the datasets corresponding to classroom 4Q04. The box plot indicates that the average temperature is similar across sensors, but there are outliers present. This implies that while the typical temperature range may be similar,

TABLE 3. Summary of the experimental setups.

Exp.	Place	Date	Max. Occ	Area (m^2)	Occupancy Levels			HVAC	Windows	Avg. Temperature (°C) Outdoor
					L	M	H			
1	PC LAB 6X233	10-28 Oct	26	76.00	1-8	9-16	+17	Yes	No	18.85
2	Classroom 4Q04	19 Oct-11 Nov	56	109.18	1-18	19-36	+37	Yes	Yes	18.00
3	Classroom 5X102	1-24 Nov	25	65.12	1-8	9-16	+17	No	Yes	12.89

TABLE 4. Description of the datasets per experiment.

Exp	Device ID	Entries	Labeled	Unlabeled	% Labeled	Labeled Training Data	Testing Data	Training Classes			
								E	L	M	H
1	S09	2557	276	2281	10.79%	220	56	35	77	106	2
	S10	3019	360	2659	11.92%	288	72	50	92	131	15
	S03	1693	371	1322	21.91%	296	75	36	98	150	12
2	S03	1849	134	1715	7.25%	107	27	11	64	32	0
	S10	1641	124	1517	7.56%	99	25	13	56	30	0
	S11	1692	129	1563	7.62%	103	26	8	61	34	0
	S05	942	133	809	14.12%	106	27	12	70	24	0
3	S03	3769	426	3343	11.30%	340	86	113	124	103	0
	S09	3462	407	3055	11.76%	325	82	121	93	111	0
	S10	2603	408	2195	15.67%	326	82	125	95	106	0
	S11	2648	361	2287	13.63%	288	73	95	93	100	0

some sensors recorded significantly higher temperatures than usual. Similarly, the humidity data shows that humidity levels typically vary between 12% and 17%. However, it is important to note that outliers were present for all sensors, indicating instances where humidity readings deviated significantly from the usual range. Therefore, during the cleaning process, outliers were intentionally retained because they represent the realism of the uncontrollable real-life scenarios. That is, professors and students conducted their daily activities without restrictions. Some of the variations could be influenced by the device's placement. That means the devices near windows or doors could be affected by ventilation rates, as well as device resets.

On the other hand, according to Viani et al. [51], the environmental variables, such as air temperature and humidity, of an enclosed space are modified due to the number of occupants and their activities. Hence, graphs are performed to plot the temperature, humidity, and occupancy timeline of one week for each experiment. The plots highlight the changes between temperature and humidity regarding the occupancy level.

For instance, in experiment 1 shown in Fig. 8, the temperature sensed by the IoT device presented a large increment when the PC Lab was occupied at the medium level. Generally, the RH tended to decrease and the temperature to increase. Following this pattern, it is possible to figure out when the classroom was occupied or not even though the data is not all labeled. On the other hand, the behavior of the RH and temperature were unusual in experiment 3 (Fig. 9). The RH increased when the temperature also increased (gray circle). Another unexpected behavior was that for a short period, the temperature dropped drastically (orange circle). Thus, these behaviors do not contribute to visualizing any patterns in this scenario, as presented in experiment 1.

In addition, Fig. 10 presents a correlation matrix of the environmental variables and occupancy level in Experiment 2 (4Q04 Classroom). The main goal is to provide a statistical measure of the degree to which the variables are correlated with each other and with indoor occupancy. In general, temperature has the highest correlation values (coefficient around +0.68) with indoor occupancy. This means that when occupancy levels increase, temperature also tends to increase, and vice versa. Conversely, pressure has a negative correlation (coefficient around -0.60).

A. HARDWARE AND SOFTWARE SPECIFICATIONS

All the models were performed in a laptop computer with the following specifications: processor AMD® Ryzen 7 pro

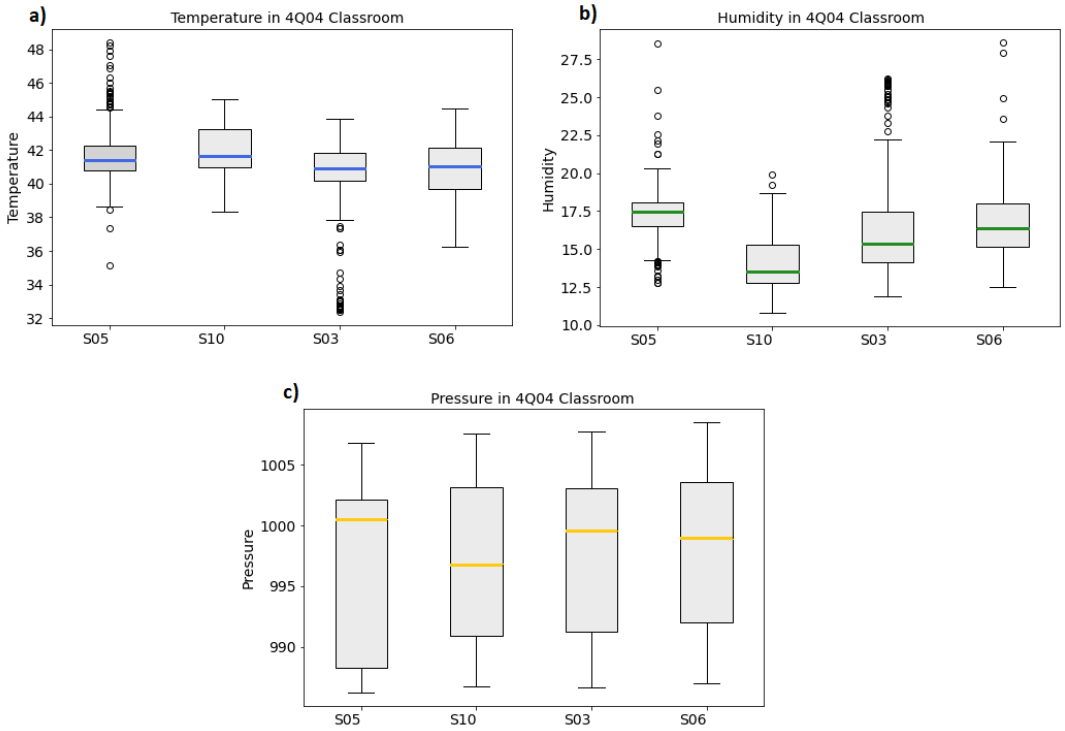

FIGURE 7. Data distribution for Temperature (a), RH (b), and Pressure (c) grouped by the Device ID. Experiment 2 in the 4Q04 Classroom.

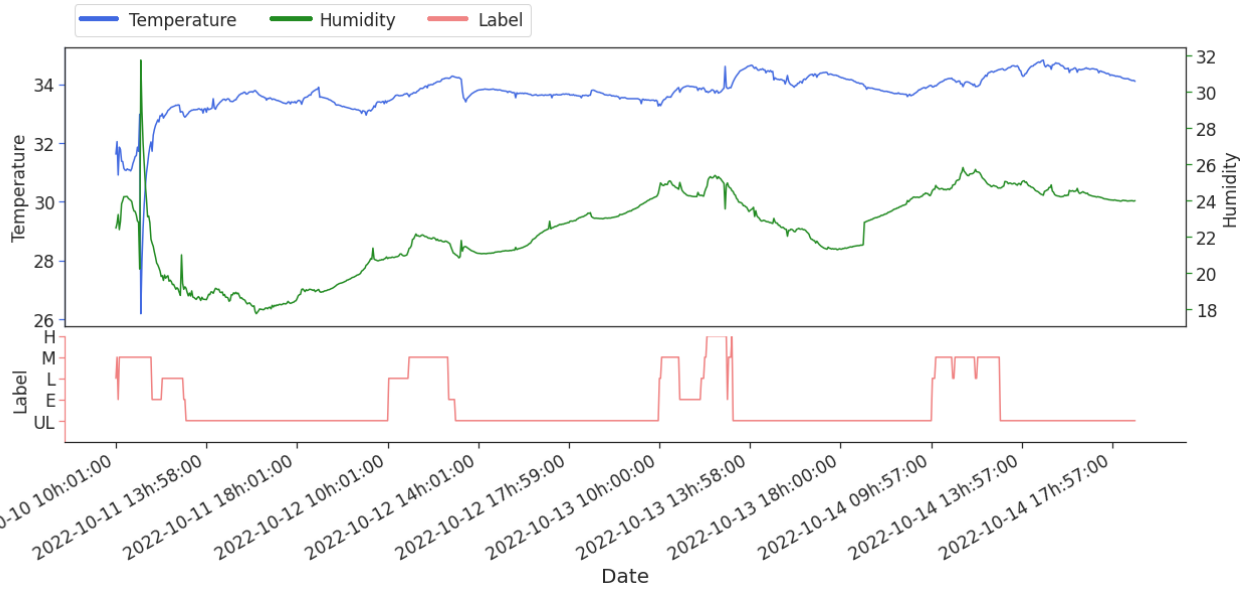

FIGURE 8. Temperature (blue), humidity (green), and occupancy level (red) of Experiment 1. Data from October 10th to 14th, 2022, of the device S03.

4750u with Radeon graphics × 16, RAM of 46.3 GB, SSD capacity of 1.0 TB, and operating system Ubuntu 20.04.4 LTS - 64 bit. Moreover, the Python 3.8.10 language and scikit-learn library [16] were used for programming. On the other hand, the datasets were divided into 80% for training and 20% for testing. Due to the size of labeled data, a validation dataset or cross-validation method is not considered in this study.

From hyperparameter selection, the best combination of parameters used to train the models is presented in Table 5. These parameters achieved the best results according to the GridSearchCV selection. The self-training model is trained

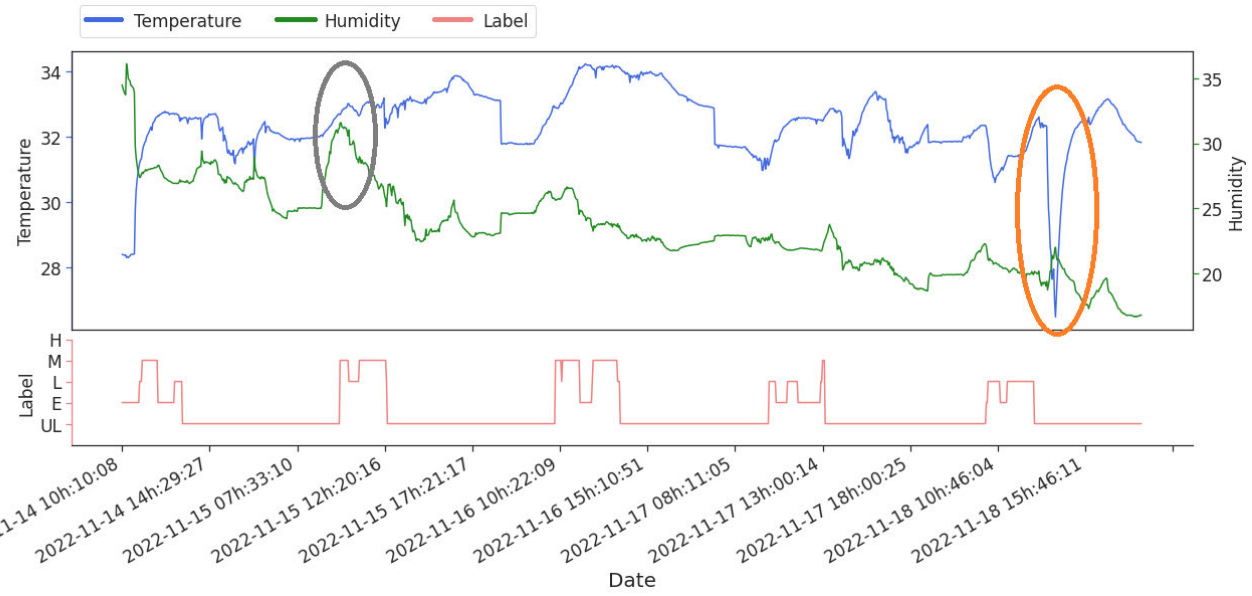

FIGURE 9. Temperature (blue), humidity (green), and occupancy level (red) of Experiment 3. Data from November 14th to 18th, 2022, of the device S10.

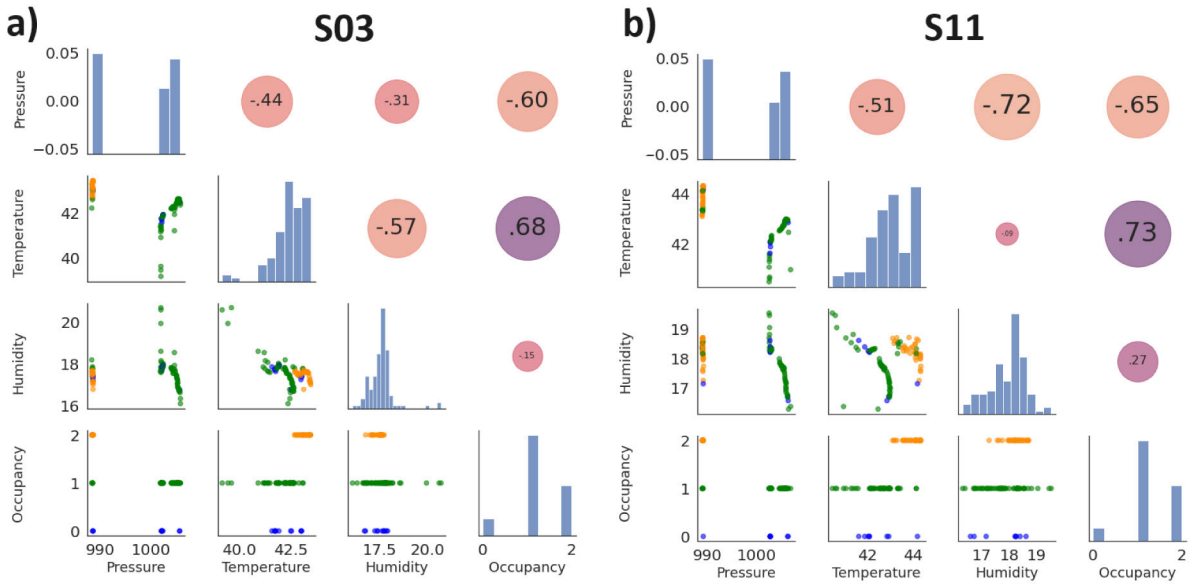

FIGURE 10. Correlation coefficients between variables and indoor occupancy by Device ID. Experiment 2 in 4Q04 Classroom. a) Device installed near the door, and b) Device placed on walls at the back of the classroom near windows and tables.

using 'threshold' as a criterion with a value of 0.75, whereas for its base estimator (SVM) the parameters selected are the same results as those suggested for the supervised model.

B. MODEL EVALUATION

The classification task for indoor occupancy levels was evaluated through different metrics, such as accuracy, precision, recall, and F1-score. Furthermore, a confusion matrix was developed to have a visual identification of the errors.

The confusion matrix provided a visual analysis of the misclassification error per model. In general, classification errors occurred between the "E" and "L" levels, as well as in the "L" and "M" levels. Fig. 11 presents the confusion matrix of the device S10 in Experiment 1. Specifically, the confusion matrices a) and b) illustrate that SVM and RF have similar errors by classifying the "M" level. ST (d) can classify the Medium (M) and High (H) levels better than supervised models (a, b, and c). The worst classification for all the classes was made by LP. These findings are similar for all the confusion matrices constructed, which is why, and due to the length of the manuscript, they were not included in this manuscript.

TABLE 5. The best combination of parameters used to train the models.

	Supervised models								
	SVM		RF			ANN			
Exp.	C	Gamma	N estimators	Depth	Criterion	Hidden layers	Activation	Solver	Alpha
1	1000	1	13	13	gini	(14, 11, 14)	relu	adam	0.0001
2	1000	1	25	4	gini	(6, 12, 13)	relu	adam	0.01
3	1000	1	11	9	gini	(13, 14, 13)	relu	adam	0.0001
	Semi-supervised models								
	ST					LP			
	Threshold	Criterion				Gamma	Neighbors	Kernel	
1	0.75	threshold				10	3	rbf	
2	0.75	threshold				2	3	rbf	
3	0.75	threshold				2	3	rbf	

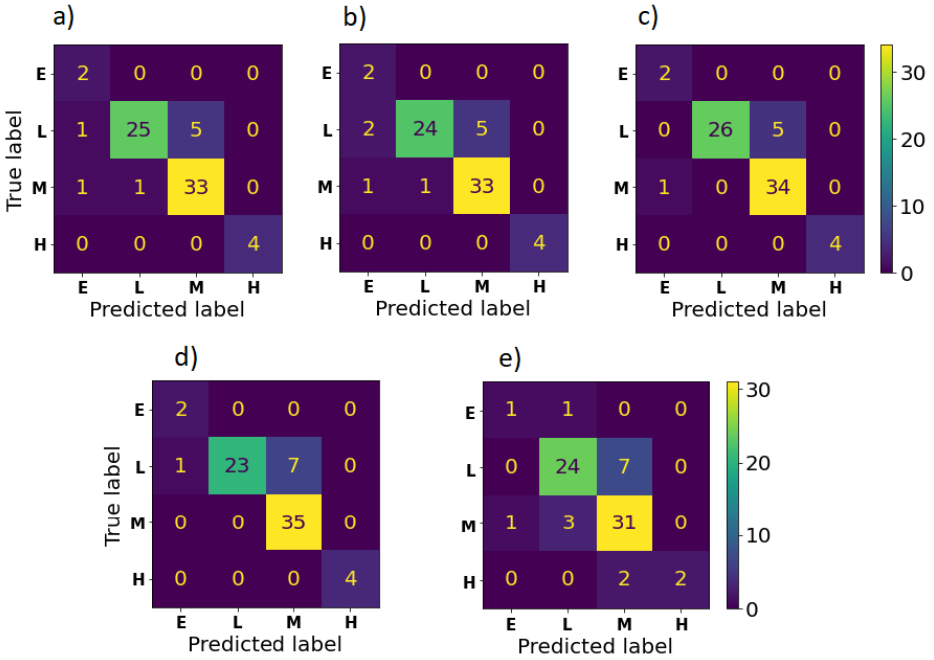

FIGURE 11. Confusion Matrix of the device S10 in Experiment 1. a) SVM, b) RF, c) MLP, d) ST, and e) LP.

1) EVALUATION USING ACCURACY

Table 6 summarizes the accuracy achieved by the supervised models. The best results are highlighted in blue; however, in general, all the devices reached a high accuracy (up to 80%). In Experiment 1, RF achieved the highest average accuracy of 89%, outperforming SVM and ST, which reached an average of 87%. Experiment 2 showed a similar trend, with ST and MLP achieving the top average accuracy of 91%, followed by SVM and RF with 87%. However, Experiment 3 had a change: SVM with 86% outperformed ST and RF with 84%. Across all experiments, the LP model consistently exhibited the lowest accuracy, averaging 70%.

The accuracy by device presents some differences not only for the supervised models but also for the SSL models. The accuracy of some devices using SSL is so close to the supervised model's accuracy; in fact, in some cases, it is higher. For instance, in the device S05 of Experiment 2, the average accuracy of SSL is better (91%) than supervised models (average of 83.67%). On the other hand, the device S03 in Experiment 1 achieved an average accuracy very close (87%) to that of the supervised models (89%). A deeper analysis is needed to determine why some devices can obtain a better estimate than others. However, this analysis falls beyond the scope of the current study.

TABLE 6. Model evaluation using the accuracy metric.

		Supervised ML Models				Semi-supervised ML Models		
Exp.	Device ID	SVM	RF	MLP	Avg Device Accuracy	ST	LP	Avg Device Accuracy
1	S09	80	89	80	83	82	79	80.5
	S10	89	88	92	89.67	89	81	85
	S03	91	89	87	89	89	85	87
	Avg.	87	89	86		87	82	
2	S03	89	93	93	91.67	89	74	81.5
	S10	92	92	92	92	88	84	86
	S11	85	77	92	84.67	92	69	80.5
	S05	81	85	85	83.67	93	89	91
	Avg.	87	87	91		91	79	
3	S03	83	87	79	83	86	60	73
	S09	87	80	77	81.33	78	70	74
	S10	89	84	87	86.67	87	77	82
	S11	84	84	82	83.33	85	74	79.5
	Avg.	86	84	81		84	70	

2) EVALUATION USING PRECISION AND RECALL

Interpreting precision and recall metrics on a class-by-class basis allows to understand how each model performs for specific classes, which can be important if certain classes are more critical or imbalanced. Fig. 12 shows the models' performance in all the scenarios.

In Experiment 1 (PC Lab 6 × 233), ST achieves the highest precision (macro average 79%) across all classes, highlighting its strong overall performance. MLP and RF also show competitive precision (macro average 77.55%) and recall (macro average 81%) values. LP and SVM showed similar performance, with a macro average precision of around 73% and recall of 68%, slightly lower than the top performers. Similarly, in Experiment 2 (Classroom 4Q04), SVM and ST consistently perform well, with a macro average precision of 65% and 68%, and a macro average recall of 61% and 67%, respectively. RF also shows strong performance with a macro average precision and recall of 62%. Similarly, MLP presents competitive results, with a macro average precision of 62% and recall of 57%. Conversely, LP struggles on most metrics, with a macro average precision of 39% and recall of 47%. Finally, Experiment 3 (Classroom 5 × 102) shows that SVM achieves the highest macro average precision (65%) and recall (63%) across all classes. MLP and ST also demonstrate competitive performance, with a macro average precision and recall of 63%. RF performs reasonably well, with a macro average precision and recall of 61%, while LP lags behind in most metrics, with a macro average precision of 54% and recall of 51%.

3) EVALUATION USING F1-SCORE

Fig. 13 displays the F1-score (macro average) and accuracy for each model per scenario. In Experiment 1 (PC Lab 6 × 233), RF and ST are the high-performance models in terms of accuracy and F1-score, where RF achieved the highest accuracy (89%) and F1-score (78%). While ST seems to offer a good trade-off between accuracy (87%) and F1-score (79%). On the other hand, MLP, SVM, and LP show competitive performance, with accuracies ranging from 82% to 87% and F1-scores from 69% to 79%. Similarly, the results of Experiment 2 (Classroom 4Q04) indicate that SVM and ST stand out as the high-performance models with the highest F1-scores (62% and 66%, respectively). RF performs well in terms of accuracy (87%) but has a slightly lower F1-score (61%). MLP, although having high accuracy (91%), has a low F1-score (57%), indicating potential issues with precision or recall. LP lags behind in both accuracy (79%) and F1-score (42%). Finally, the results of Experiment 3 Classroom 5 × 102) highlight that SVM performs the best in terms of F1-score (64%) and accuracy (86%), indicating a good balance between precision and recall. RF and ST are competitive with an accuracy of around 84% and F1-scores of 60% and 63%, respectively. While MLP has an acceptable performance with an accuracy of 81% and F1-Score of 62%. In contrast, LP has the lowest accuracy (70%) and F1-score of 52% in this scenario.

C. MODEL PERFORMANCE COMPARISON

The potential of some SSL techniques to achieve comparable or even superior performance compared to fully supervised approaches was demonstrated through the High-Quality Supervised Baseline evaluation.

By comparing the supervised and SSL models, some differences in the accuracy achieved by the SSL algorithms can be shown. The main purpose is to consider the fully supervised models with limited labeled data as a baseline. Thereby, the SSL models are compared to the baseline to determine whether the combination of labeled and unlabeled

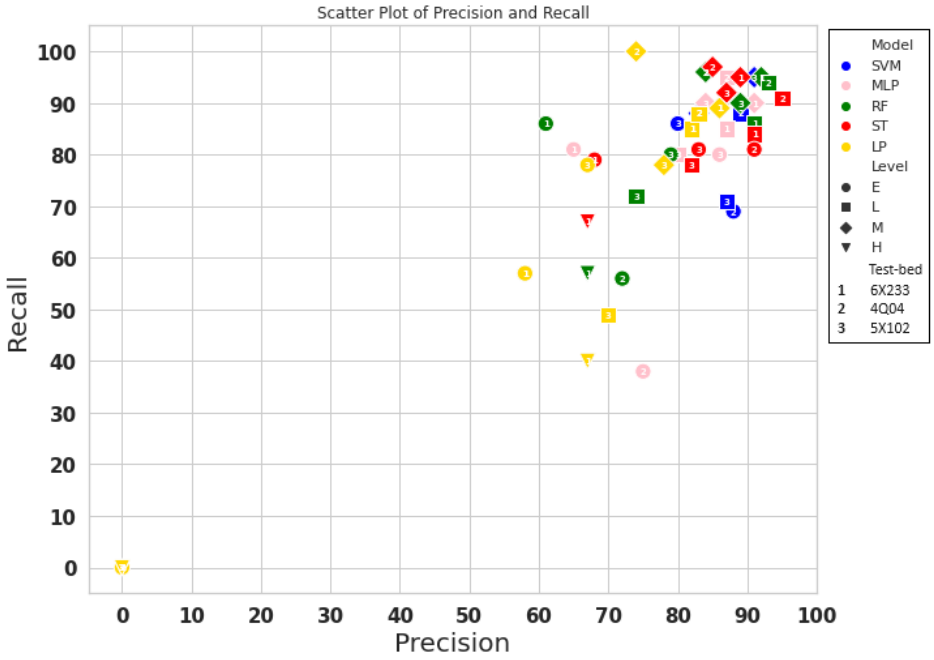

FIGURE 12. Precision and recall evaluation for each scenario and class.

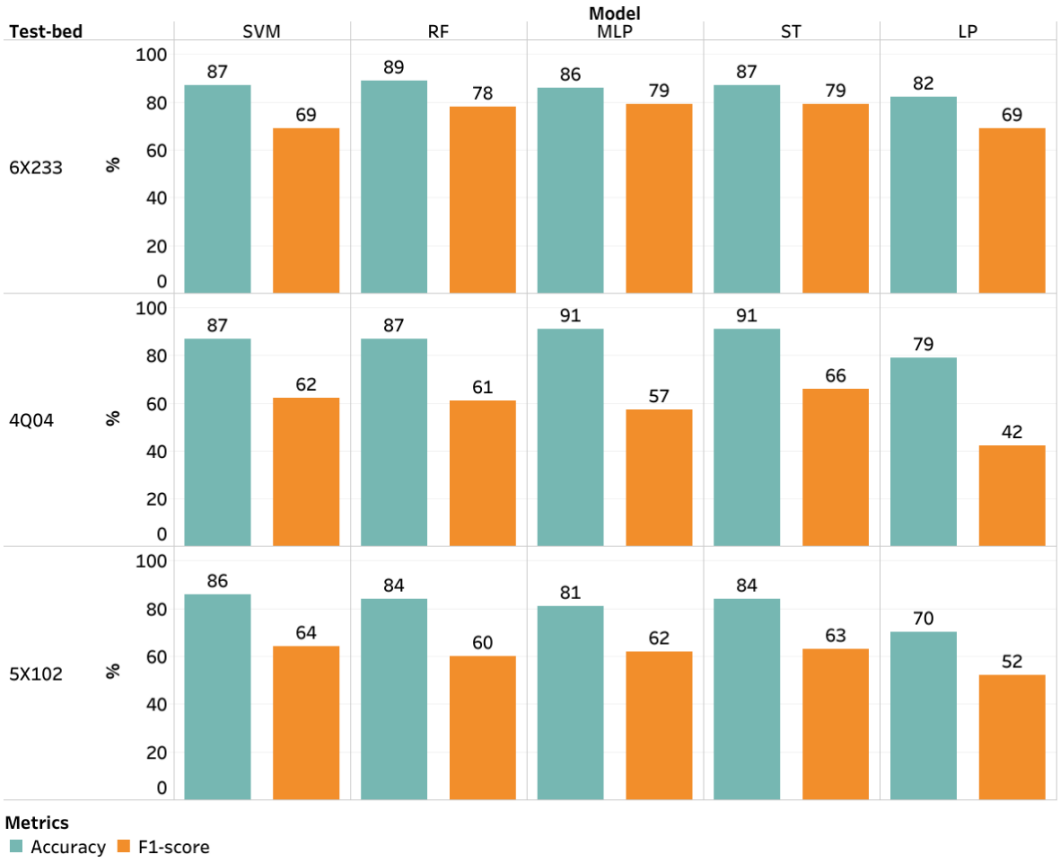

FIGURE 13. Model evaluation using F1-score and accuracy.

data achieves a better performance. Table 7 presents the results of the accuracy comparison among the models. The difference in accuracy between supervised models and SSL models is shown using a yellow-to-blue gradient scale.

TABLE 7. HighQuality supervised baseline results. Accuracy comparison between supervised and semi-supervised algorithms.

Exp.	Device ID	SVM-ST	RF-ST	MLP-ST	SVM-LP	RF-LP	MLP-RF
1	S09	-2	7	-2	1	10	1
1	S10	0	-1	3	8	7	11
1	S03	2	0	-2	6	4	2
1	Average	0	2	-0.34	5	7	4.66
2	S03	0	4	4	15	19	19
2	S10	4	4	4	8	8	8
2	S11	-7	-15	0	16	8	23
2	S05	-12	-8	-8	-8	-4	-4
2	Average	-3.75	-3.75	0	7.75	7.75	11.5
3	S03	-3	1	-7	23	27	19
3	S09	9	2	-1	17	10	7
3	S10	2	-3	0	12	7	10
3	S11	-1	-1	-3	10	10	8
3	Average	1.75	-0.25	-2.75	15.5	13.5	11

Yellow cells indicate cases where SSL achieved higher accuracy, while blue cells show where supervised learning performed better. Considering the average accuracy, the ST model is usually better than any supervised model. Nevertheless, regarding the accuracy per device, ST is usually better than RF and MLP. In contrast, SVM and ST are tied in this comparison. On the other hand, LP was the worst model in this research. Nevertheless, it is worth exploring different tuning parameters, as well as other scenarios to evaluate its performance. To have a visual analysis of the performance of the supervised and SSL models, Fig. 14 shows the predicted labels for each model. The ground truth data is represented by a black line, while the predictions made by the models are represented using different colors. The ST model (purple line) makes predictions similar to the SVM (yellow line), and RF models (red line) which are almost close to the ground truth.

D. DISCUSSION

Even though indoor occupancy has been studied over the last few years, there is a lack of systematic studies comparing the state-of-the-art and semi-supervised learning (SSL) has not been widely explored for this task. Furthermore, it is not possible to conduct a systematic benchmark of the existing works in literature as different metrics, parameters, and datasets are used. In other words, it is extremely challenging to fully compare the performance of all the models discussed in the literature section due to variations in measurement data and the lack of accessibility to source datasets. Additionally, the experimental conditions and building configurations differ among them, making impossible a full comparison. Nevertheless, a discussion of the existing work and their relative performances is presented in this article, highlighting how this study contributes to filling the gaps in the literature.

In a first inspection, the existing research indicates that Supervised ML is the primary method used for indoor occupancy estimation. Studies conducted by [21], [22], and [23] have all employed supervised learning models, such as k-NN, SVM, and RF. The performance of their models exceeds 90% of accuracy and an error of less than 1.84 in some of the scenarios selected. Additionally, environmental sensors like CO_2, temperature, and light are commonly employed in these studies [20], [21], [22], [23]. Moreover, combining camera and environmental data seems promising for its high performance (F1-score of 99.35%) while potentially mitigating privacy issues [24]. Nevertheless, the data used in all these researches are completely labeled, which implies methods for labeling, such as cameras or user interaction to press buttons. However, studies such as those conducted by [26] and [29] have emphasized the challenge of obtaining high-quality labeled data.

On the other hand, SSL appears to be less common. Studies by [27] and [28] have explored SSL approaches, but with limitations in terms of sensor types and accuracy when compared to supervised methods. For example, ultrasonic sensors were used in both works, and no study considered environmental variables. Given the gaps in the literature outlined above, a systematic comparison between supervised and SSL models in the field of occupancy estimation was conducted in this work. A full and valid comparison among models was ensured by using the same datasets collected from the three selected scenarios and the same metrics. As described in the Results Section, the applied SSL models used only a few labeled data, which helped achieve an accuracy of around 87% to 93%. These results are comparable to the best-supervised ML models.

Nevertheless, there are some limitations of this study that should be considered. For instance, the experiments were carried out in real-life scenarios (uncontrolled conditions) and the sensor placement analysis was not considered during the deployment. Therefore, it was impossible to control certain factors, such as window status, indoor temperature, and room capacity. This last factor caused some data classes to not be labeled during the ground truth record. Furthermore, the ground truth data was collected only over a one-week period. Additionally, the study focused on a limited set of variables: air temperature, RH, and pressure.

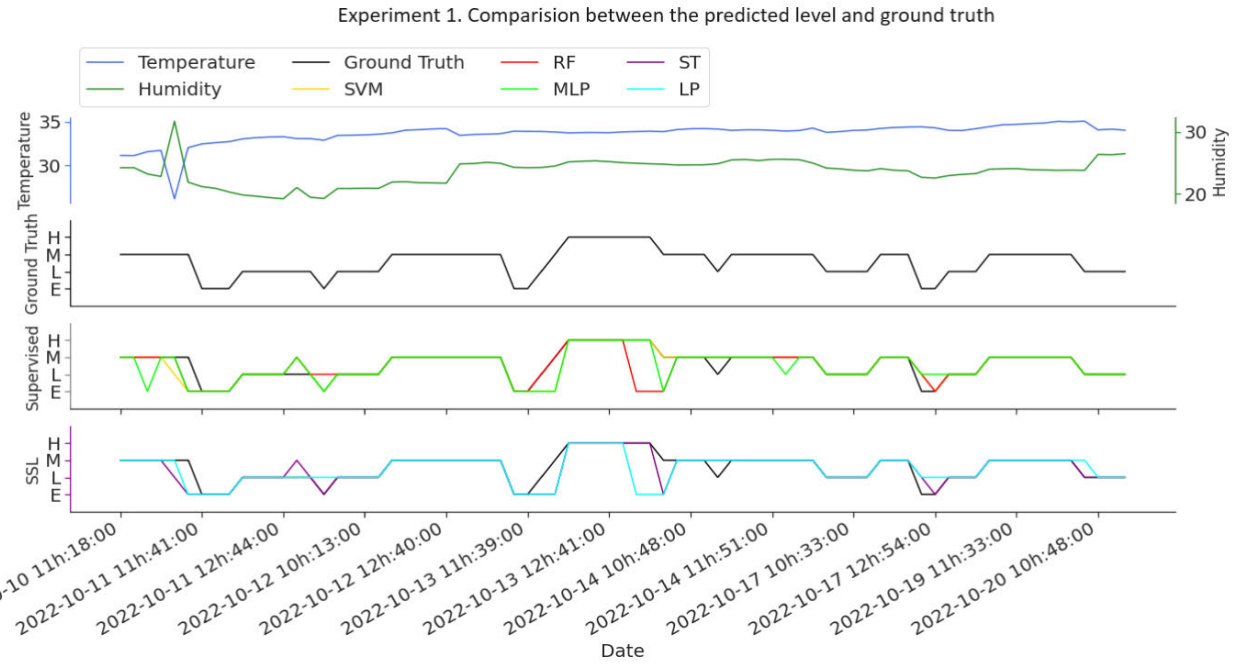

FIGURE 14. Predicted labels from the S03 device test data in experiment 1.

VII. CONCLUSION

This study demonstrated the potential of exploiting a few labeled data and a large amount of unlabeled data to estimate indoor occupancy levels using indirect methods that measure air temperature, relative humidity, and atmospheric pressure. Furthermore, to the best of the authors' knowledge, this study is a pioneer in conducting a systematic comparison between supervised ML and SSL approaches in the field of occupancy estimation.

The results showed that, in general, the accuracy obtained by the ST model is similar and, in some cases, is better than the supervised models. This highlights the effectiveness of SSL approaches, which leverage abundant unlabeled data alongside a limited number of labeled samples. Specifically, the ST model outperformed both the RF and MLP models. It demonstrated comparable performance to SVM, emphasizing ST viability as a competitive alternative by estimating occupancy levels in enclosed spaces with an average accuracy of 87.33%. It is noteworthy that ST was also able to classify occupancy levels that eluded the supervised models.

The finding that ST has had the best results in estimating occupancy levels using environmental variables (indirect method) leads to two other major insights: 1) the future data collection process for this investigation will require less effort and cost. This is because the ground truth data was recorded for a relatively short period, whereas the device continued to collect measurements autonomously for an extended period without the need for human intervention. 2) The study showed a significant relationship between environmental factors and occupation. Specifically, it is observed that temperature, humidity, and pressure within the enclosed areas presented variations corresponding to changes in occupancy levels.

The results presented in this study suggest that it is possible to estimate occupancy levels using only a portion of labeled data, which leads to reducing the cost of labeling data and facilitating deployment in real-world scenarios. This semi-supervised non-intrusive monitoring approach could be adopted to control HVAC and lighting systems in classrooms, meeting rooms, and other facilities in response to real-time occupancy estimations, and thus reduce energy consumption. Furthermore, SSL models can assist facility managers in monitoring space utilization and enhancing space allocation strategies. As well as support security systems, such as intrusion detection, particularly by integrating occupancy detection using SSL models along with conventional and affordable sensors.

Future work will focus on carrying out additional analyses to find the optimal parameters for the SSL models, as well as explore other SSL approaches. Furthermore, it would be valuable to investigate the generalization of the ST model in various indoor environments. In addition, it is necessary to analyze the effects of sensor placement to help in understanding why some devices achieve better occupancy estimations. Similarly, additional work is needed to determine why environmental measurements present unusual behaviors and to determine optimal device location considering the proximity to windows, doors, or a crowded table.

ACKNOWLEDGMENT

The authors would like to thank the University of the West of England for its support in carrying out this research project. Furthermore, the author Alma Mena-Martinez would like to thank Tecnológico de Monterrey and CONAHCYT for the Ph.D. studentship.

ABBREVIATIONS LIST

Abbreviation	Definition.
ANN	Artificial Neural Networks.
AWS	Amazon Web Services.
BR	Bayesian Ridge.
CO2	Carbon Dioxide.
CSV	Comma-Separated Values.
DNN	Dense Neural Network.
DT	Decision Tree.
E	Empty Level.
FN	False Negative.
FP	False Positive.
GP	Gaussian Process.
H	High Level.
HVAC	Heating, Ventilation, and Air Conditioning.
IAQ	Indoor Air Quality.
IoT	Internet of Things.
JSON	JavaScript Object Notation.
k-NN	K-Nearest Neighbor.
L	Low Level.
LP	Label Propagation.
LR	Linear Regression.
M	Medium Level.
MAE	Mean Absolute Error.
ML	Machine Learning.
MLP	Multi-Layer Perceptron.
MQTT	Message Queuing Telemetry Transport.
PCA	Principal Component Analysis.
PIR	Passive Infrared sensor.
RF	Random Forest.
RH	Relative Humidity.
SPL	Sound Pressure Level.
SSL	Semi-Supervised Learning.
ST	Self-Training.
SVM	Support Vector Machine.
TLS	Transport Layer Security.
TN	True Negative.
TP	True Positive.
WPCA	Weighted Principal Component Analysis.

REFERENCES

[1] A. Verma, S. Prakash, V. Srivastava, A. Kumar, and S. C. Mukhopadhyay, "Sensing, controlling, and IoT infrastructure in smart building: A review," *IEEE Sensors J.*, vol. 19, no. 20, pp. 9036–9046, Oct. 2019.

[2] I. Rafiq, A. Mahmood, U. Ahmed, A. R. Khan, K. Arshad, K. Assaleh, N. I. Ratyal, and A. Zoha, "A hybrid approach for forecasting occupancy of building's multiple space types," *IEEE Access*, vol. 12, pp. 50202–50216, 2024.

[3] M. Umair, M. A. Cheema, O. Cheema, H. Li, and H. Lu, "Impact of COVID-19 on IoT adoption in healthcare, smart homes, smart buildings, smart cities, transportation and industrial IoT," *Sensors*, vol. 21, no. 11, p. 3838, Jun. 2021.

[4] J. Jiang, C. Wang, T. Roth, C. Nguyen, P. Kamongi, H. Lee, and Y. Liu, "Residential house occupancy detection: Trust-based scheme using economic and privacy-aware sensors," *IEEE Internet Things J.*, vol. 9, no. 3, pp. 1938–1950, Feb. 2022.

[5] Y. Wei, L. Xia, S. Pan, J. Wu, X. Zhang, M. Han, W. Zhang, J. Xie, and Q. Li, "Prediction of occupancy level and energy consumption in office building using blind system identification and neural networks," *Appl. Energy*, vol. 240, pp. 276–294, Apr. 2019.

[6] C. Jiang, Z. Chen, L. C. Png, K. Bekiroglu, S. Srinivasan, and R. Su, "Building occupancy detection from carbon-dioxide and motion sensors," in *Proc. 15th Int. Conf. Control, Autom., Robot. Vis. (ICARCV)*, Nov. 2018, pp. 931–936.

[7] A. R. Mena, H. G. Ceballos, and J. Alvarado-Uribe, "Measuring indoor occupancy through environmental sensors: A systematic review on sensor deployment," *Sensors*, vol. 22, no. 10, p. 3770, May 2022.

[8] H. Saha, A. R. Florita, G. P. Henze, and S. Sarkar, "Occupancy sensing in buildings: A review of data analytics approaches," *Energy Buildings*, vols. 188–189, pp. 278–285, Apr. 2019.

[9] Y. Roh, G. Heo, and S. E. Whang, "A survey on data collection for machine learning: A big data–AI integration perspective," *IEEE Trans. Knowl. Data Eng.*, vol. 33, no. 4, pp. 1328–1347, Apr. 2021.

[10] X. Zhou and M. Belkin, "Semi-supervised learning," in *Academic Press Library in Signal Processing*, vol. 1. Amsterdam, The Netherlands: Elsevier, 2014, ch. 2, pp. 1239–1269.

[11] S. Raschka, *Python Machine Learning*. Birmingham, U.K.: Packt Publishing Ltd, Sep. 2015.

[12] B. Akkaya and N. Colakoglu, "Comparison of multi-class classification algorithms on early diagnosis of heart diseases," in *Proc. Int. Conf. Recent Develop. Data Sci. Bus. Anal.*, 2019, pp. 1–24.

[13] M. W. Berry, A. Mohamed, and B. W. Yap, *Supervised and Unsupervised Learning for Data Science*. Cham, Switzerland: Springer, 2020.

[14] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001, doi: 10.1023/A:1010933404324.

[15] M.-C. Popescu, V. Balas, L. Perescu-Popescu, and N. Mastorakis, "Multilayer perceptron and neural networks," *WSEAS Trans. Circuits Syst.*, vol. 8, pp. 1–7, Jul. 2009.

[16] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, 2011.

[17] X. Yang, Z. Song, I. King, and Z. Xu, "A survey on deep semi-supervised learning," *IEEE Trans. Knowl. Data Eng.*, vol. 109, no. 2, pp. 1–20, May 2022, doi: 10.1109/TKDE.2022.3220219.

[18] Y. Zhou, Z. Yu, J. Li, Y. Huang, and G. Zhang, "A data mining model for building occupancy estimation based on deep learning methods," *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 609, no. 7, Sep. 2019, Art. no. 072029, doi: 10.1088/1757-899X/609/7/072029.

[19] G. Bonaccorso, A. Fandango, and R. Shanmugamani, *Python: Advanced Guide To Artificial Intelligence: Expert Machine Learning Systems and Intelligent Agents Using Python*. Birmingham, U.K.: Packt Publishing, 2018.

[20] J. Choi, H. Ge, and N. Koshizuka, "IoT-based occupants counting with smart building state variables," in *Proc. IEEE 29th Int. Conf. Enabling Techn.*, Sep. 2020, pp. 171–176.

[21] B. Abade, D. Perez Abreu, and M. Curado, "A non-intrusive approach for indoor occupancy detection in smart environments," *Sensors*, vol. 18, no. 11, p. 3953, Nov. 2018.

[22] A. Das, R. Gupta, and S. Chakraborty, "A study on real-time edge computed occupancy estimation in an indoor environment," in *Proc. Int. Conf. Commun. Syst. Netw. (COMSNETS)*, Jan. 2020, pp. 527–530.

[23] A. Vela, J. Alvarado-Uribe, M. Davila, N. Hernandez-Gress, and H. G. Ceballos, "Estimating occupancy levels in enclosed spaces using environmental variables: A fitness gym and living room as evaluation scenarios," *Sensors*, vol. 20, no. 22, p. 6579, Nov. 2020.

[24] 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, Dec. 2022.

[25] J. Dutta and S. Roy, "OccupancySense: Context-based indoor occupancy detection & prediction using CatBoost model," *Appl. Soft Comput.*, vol. 119, Apr. 2022, Art. no. 108536.

[26] F. Banihashemi, M. Weber, F. Deghim, C. Zong, and W. Lang, "Occupancy modeling on non-intrusive indoor environmental data through machine learning," *Building Environ.*, vol. 254, Apr. 2024, Art. no. 111382.

[27] O. Shih, P. Lazik, and A. Rowe, "AURES: A wide-band ultrasonic occupancy sensing platform," in *Proc. 3rd ACM Int. Conf. Syst. Energy-Efficient Built Environments*, Nov. 2016, pp. 157–166.

[28] M. Jin, R. Jia, Z. Kang, I. C. Konstantakopoulos, and C. J. Spanos, "PresenceSense: Zero-training algorithm for individual presence detection based on power monitoring," in *Proc. 1st ACM Conf. Embedded Syst. Energy-Efficient Buildings*, Nov. 2014, pp. 1–10.

[29] Y. Wei, S. Wang, L. Jin, Y. Xu, and T. Ding, "Indoor occupancy estimation from carbon dioxide concentration using parameter estimation algorithms," *Building Services Eng. Res. Technol.*, vol. 43, no. 4, pp. 419–438, Jul. 2022, doi: 10.1177/01436244211060903.

[30] (2022). *Tutorial: Storing Device Data a DynamoDB Table*. [Online]. Available: https://docs.aws.amazon.com/iot/latest/developerguide/iot-ddb-rule.html

[31] *Aws IoT Core: Developer Guide*, Amazon Web Services, Inc., Seattle, WA, USA, 2023.

[32] J. Whitbread. (2021). *Data ProtectionPolicy*. [Online]. Available: https://www.uwe.ac.uk/study/it-services/information-security-toolkit/information-security-policies

[33] R. Adeogun, I. Rodriguez, M. Razzaghpour, G. Berardinelli, P. H. Christensen, and P. E. Mogensen, "Indoor occupancy detection and estimation using machine learning and measurements from an IoT LoRa-based monitoring system," in *Proc. Global IoT Summit (GIoTS)*, Jun. 2019, pp. 1–5.

[34] E. Acuña and C. Rodriguez, "The treatment of missing values and its effect on classifier accuracy," in *Classification, Clustering, and Data Mining Applications*. Berlin, Germany: Springer, 2004, pp. 639–647.

[35] C. Jiang, Z. Chen, R. Su, M. K. Masood, and Y. C. Soh, "Bayesian filtering for building occupancy estimation from carbon dioxide concentration," *Energy Buildings*, vol. 206, Jan. 2020, Art. no. 109566.

[36] A. G. V. Kreinovich and O. Kosheleva, "Why 70/30 or 80/20 relation between trainingand testing sets: A pedagogical explanation," Dept.al Technical Reports, Univ. Texas at El Paso, El Paso, TX, USA, Tech. Tech. Rep. UTEP-CS-18-09, 2018.

[37] V. R. Joseph, "Optimal ratio for data splitting," *Stat. Anal. Data Mining: ASA Data Sci. J.*, vol. 15, no. 4, pp. 531–538, Apr. 2022.

[38] S. Galli, *Python Feature Engineering Cookbook : Over 70 Recipes for Creating, Engineering, and Transforming Features to Build Machine Learning Models*. Birmingham, U.K.: Packt Publishing, 2020.

[39] Z. Yang and B. Becerik-Gerber, "Cross-space building occupancy modeling by contextual information based learning," in *Proc. 2nd ACM Int. Conf. Embedded Syst. Energy-Efficient Built Environments*, Nov. 2015, pp. 177–186, doi: 10.1145/2821650.2821668.

[40] S. Ahmed, U. Kamal, T. R. Toha, N. Islam, and A. B. M. A. Al Islam, "Predicting human count through environmental sensing in closed indoor settings," in *Proc. 15th EAI Int. Conf. Mobile Ubiquitous Syst. Comput., Netw. Services*, Nov. 2018, pp. 49–58.

[41] U. Kamal, S. Ahmed, T. R. Toha, N. Islam, and A. B. M. Alim Al Islam, "Intelligent human counting through environmental sensing in closed indoor settings," *Mobile Netw. Appl.*, vol. 25, no. 2, pp. 474–490, Apr. 2020, doi: 10.1007/S11036-019-01311-W.

[42] F. Viani, "Opportunistic occupancy estimation in museums through wireless sensor networks," *Microw. Opt. Technol. Lett.*, vol. 57, no. 8, pp. 1975–1977, Aug. 2015, doi: 10.1002/MOP.29239.

[43] W. Wang, J. Chen, and T. Hong, "Occupancy prediction through machine learning and data fusion of environmental sensing and Wi-Fi sensing in buildings," *Autom. Construct.*, vol. 94, pp. 233–243, Oct. 2018.

[44] Y. Jin, D. Yan, A. Chong, B. Dong, and J. An, "Building occupancy forecasting: A systematical and critical review," *Energy Buildings*, vol. 251, Nov. 2021, Art. no. 111345.

[45] O. Chapelle, B. Scholkopf, and A. Zien, *Semi-Supervised Learning (Adaptive Computation and Machine Learning)*. Cambridge, MA, USA: MIT Press, 2006.

[46] Z. Yang, N. Li, B. Becerik-Gerber, and M. Orosz, "A systematic approach to occupancy modeling in ambient sensor-rich buildings," *Simulation*, vol. 90, no. 8, pp. 960–977, Jul. 2013.

[47] J. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization," *J. Mach. Learn. Res.*, vol. 13, no. 2, pp. 1–20, 2012.

[48] A. Zheng, N. Shelby, and E. Volckhausen, "Evaluating machine learning models," *Mach. Learn. AWS Cloud*, vol. 1, p. 3, Jul. 2019.

[49] A. Baratloo, M. Hosseini, A. Negida, and G. El Ashal, "Part 1: Simple definition and calculation of accuracy, sensitivity and specificity.," *Emergency*, vol. 3, pp. 9–48, Jun. 2015.

[50] A. Zheng, *Evaluating Machine Learning Models*. Sebastopol, CA, USA: O'Reilly Media, Sep. 2015.

[51] F. Viani, A. Polo, F. Robol, G. Oliveri, P. Rocca, and A. Massa, "Crowd detection and occupancy estimation through indirect environmental measurements," in *Proc. 8th Eur. Conf. Antennas Propag.*, Apr. 2014, pp. 2127–2130.

ALMA MENA-MARTINEZ received the B.S. degree in mechatronics engineering and the M.Sc. degree (Hons.) in environmental systems. She is currently pursuing the Ph.D. degree in computer science with Tecnológico de Monterrey (ITESM), Mexico. Her research interests include semi-supervised as well as unsupervised machine learning models, data fusion, and the IoT embedded systems. She is interested in addressing real-world problems, such as reducing energy consumption in buildings and homes, improving collaborative workspace management, and developing indoor occupancy monitoring systems.

MANUEL DAVILA DELGADO received the Ph.D. degree in computational mechanics, topology optimization, and evolutionary structural optimization. He is responsible for delivering high-impact research at the intersection of key digital technologies that enable digital transformation. He is an expert in developing novel digital solutions for smart infrastructure and digital engineering. He has over 16 years of experience applying research in varied areas of digital engineering. He has carried out research with Eindhoven University of Technology, The Netherlands; the National University of Singapore; the University of Cambridge, U.K.; and the University of the West of England, U.K. He has authored over 50 peer-reviewed international publications and secured research grants from Innovate UK and the Centre for Digital Built Britain in excess of £3.6 million (PI: £840K, Co-I: 2.83 million). His research has helped top companies in the U.K., such as Balfour Beatty, Costain,

and Airbus, to adopt novel emerging technologies and develop digital solutions that enable data-driven decision-making. His research interests and expertise lie at the intersection of digital twins, machine learning, and immersive technologies. In 2019, he was the winner of the J. James R. Croes Medal, awarded by American Society of Civil Engineers (ASCE), for the article ''Structural Performance Monitoring Using a Dynamic Data-Driven BIM Environment.'' The award recognizes the most notable contribution to engineering research in terms of the practical value of the research and its impact on engineering practice.

JOANNA ALVARADO-URIBE is currently the Data Hub Leader with the Department of Living Laboratory and Data Hub, Institute for the Future of Education (IFE), Tecnológico de Monterrey, Mexico. She is also an adjunct Researcher with the Advanced Artificial Intelligence Research Group, School of Engineering and Sciences, Tecnológico de Monterrey. She has participated in different projects between the EU and Mexico focused on strengthening cooperation in ICT research and innovation, accelerating the dissemination and adoption of the FIWARE technologies in Mexico, and developing FIWARE-based applications oriented to smart cities and data and context management. She is collaborating on research, development, and entrepreneurship projects related to indoor monitoring using environmental variables, machine learning algorithms, and the Internet of Things. Moreover, she is a member of the National Research System (SNI) Rank 1, an adherent member of Mexican Academy of Computing (AMEXCOMP), and a member of the CONACYT Collaboration Network on Applied Computational Intelligence (RedICA).

HECTOR G. CEBALLOS received the master's and Ph.D. degrees in intelligent systems from Tecnológico de Monterrey, in 2010. He is currently the Director of the Living Laboratory and Data Hub, Institute for the Future of Education (IFE), Tecnológico de Monterrey. He has worked as an expert consultant for banks and IT companies. He is ascribed to the IFE's Research Unit on Engaging and Motivating Learning Models. He is the author of more than 60 papers in journals and conferences. His main research interests include machine learning, data science, process mining, and causality, applied to research and learning analytics. He is a member of Mexican National System of Researcher (SNI) and an adherent member of Mexican Academy of Computing (AMEXCOMP).