Machine Learning-based Occupancy Estimation Using Multivariate Sensor Nodes

Adarsh Pal Singh*, Vivek Jain*, Sachin Chaudhari*, Frank Alexander Kraemer†, Stefan Werner† and Vishal Garg*
*International Institute of Information Technology (IIIT), Hyderabad, India
†Norwegian University of Science and Technology (NTNU), Norway
Email: adarshpal.singh@research.iiit.ac.in, jain.vivek@students.iiit.ac.in, sachin.c@iiit.ac.in, vishal@iiit.ac.in, kraemer@ntnu.no, stefan.werner@ntnu.no

***Abstract*—In buildings, a large chunk of energy is spent on heating, ventilation and air conditioning systems. One way to optimize their usage is to make them demand-driven depending on human occupancy. This paper focuses on accurately estimating the number of occupants in a room by leveraging multiple heterogeneous sensor nodes and machine learning models. For this purpose, low-cost and non-intrusive sensors such as CO_2, temperature, illumination, sound and motion were used. The sensor nodes were deployed in a room in a star configuration and measurements were recorded for a period of four days. A regression based method is proposed for calculating the slope of CO_2, a new feature derived from real-time CO_2 values. Supervised learning algorithms such as linear discriminant analysis (LDA), quadratic discriminant analysis (QDA), support vector machine (SVM) and random forest (RF) were used on several different combinations of feature sets. Moreover, multiple performance metrics such as accuracy, F1 score and confusion matrix were used to evaluate the performance of our models. Experimental results demonstrate a maximum accuracy of 98.4% and a high F1 score of 0.953 for estimating the number of occupants in the room. Principal component analysis (PCA) was also applied to evaluate the performance of a dataset with reduced dimensionality.**

***Index Terms*—Internet of Things, Machine Learning, Occupancy Estimation, Wireless Sensor Network.**

I. Introduction

Real-time occupancy information can give rise to intelligent heating, ventilation and air conditioning (HVAC) and lighting systems in buildings which would not only conserve energy, but also provide better comfort to the occupants. With the advent of the Internet of Things (IoT), there are readily available sensors which can measure the environmental parameters. This data can then be analyzed using machine learning (ML) to determine human occupancy without video based systems. Recent studies have demonstrated energy savings up to 30% in buildings where the occupancy pattern was known [1].

Early approaches for occupancy detection and estimation resulted in the use of intrusive systems such as cameras [2], WiFi [3], wearables and RFID [4], [5]. With the rise in concerns regarding privacy, the research in recent years has moved towards the use of non-intrusive environmental sensors for occupancy detection and estimation such as CO_2 [6]–[14], temperature [6], [7], [9]–[14], CO [9], [10], total volatile organic compounds [9], [10], light [6]–[8], [10], [11], motion [7]–[11], sound [6], [8]–[11], humidity [6], [7], [9]–[13], pressure [6], [12], [13] and air-volume [14]. The focus of this paper is also on the use of non-intrusive sensors for occupancy estimation and the following five readily available low-cost sensors have been used in our experiments: CO_2, temperature, light, motion and sound.

A lot of research has been carried out in the literature for occupancy detection, i.e., if the room is occupied or not [6]–[8]. Although detection alone can help in improving energy savings, estimating also the precise number of occupants can make the system even more adaptive and energy-efficient. Therefore, the focus of this paper is on occupancy estimation.

There are quite a few papers on ML based occupancy estimation [10]–[14]. In [10], three ML techniques namely Hidden Markov model (HMM), artificial neural network (ANN) and support vector machine (SVM) were used on a distributed sensor network. It was shown that HMM gives the best performance with 75% accuracy. However, as stated by the authors, the occupancy levels were very dynamic indicating that all labels may not have equal number of data points. As such, F1 score and confusion matrix are more suitable performance metrics for such studies as compared to only accuracy metric used in [10]. In [11], an ambient sensor system was deployed in two labs and a radial basis function (RBF) neural network was used for classification. They, however, did not do cross-validation and their model may fail for large spaces where a single node may not be effective. In [14], another set of sensors comprising of CO_2, air volume, auxiliary and room temperature were used. However, the paper binned occupancy levels instead of giving a point estimate. A similar approach of binning was used in [13], which achieved a high accuracy using a convolutional deep bidirectional long short-term memory approach. The work in [12] used a network of three sensor nodes with multiple sensors and used extreme learning machines (ELM) to implement a wrapper model of feature selection. The method could accurately determine the levels of occupancy but the accuracy degraded when the exact number was needed. However, the estimation could be done for as many as twenty-two occupants.

In this paper, we aim to estimate the number of occupants

978-1-5386-4920-6/18/$31.00 ©2018 IEEE

(between 0 and 3) in a room by using multiple heterogeneous sensor nodes with various ML techniques such as linear discriminant analysis (LDA), quadratic discriminant analysis (QDA), SVM (Linear), SVM (RBF) and random forest (RF). In particular, our contributions are as follows:

- Unlike previous approaches of using a single heterogeneous sensor node, we deploy multiple such nodes in a room all transmitting data to the sink periodically.
- The performance comparison in terms of estimation accuracy and F1 score is carried out for various combinations of features, both homogeneous and heterogeneous.
- Data preprocessing and feature engineering techniques to handle such a vast dataset and infer new features such as *slope of* CO_2 to boost performance are discussed.
- Apart from supervised learning techniques, principal component analysis (PCA) is also employed to see how well a transformed but reduced feature set performs as compared to the original but large feature set.

The structure of the rest of the paper is as follows: Section II describes the data logging sensor network and discusses feature engineering as well as ML techniques used in this work. Section III presents the experiments performed on the dataset and discusses the results obtained with different models. The paper is concluded in Section IV.

II. Methodology

A. Experimental Setup

Fig. 1 shows the test lab in which the wireless sensor network (WSN) was deployed to record the data. The lab is a (6 m x 4.6 m) room with four office desks. The room has a big window with blinds at the rear end under node S7 and a self closing glass door for entry/exit under node S6. Note that no HVAC systems were in use while performing the experiments. The network is essentially a Zigbee based star network with seven slave nodes feeding data to the master node. The intuition behind having multiple multivariate nodes was that such a system could be deployed more reliably in large spaces than a single node.

Five different types of non-intrusive sensors were used in this experiment: temperature, illumination, sound, CO_2 and passive infrared (PIR). The CO_2 and sound sensors needed manual calibration. Table I lists the accuracy and resolution of each sensor used. As it is evident from Fig. 1, sensor nodes S1-S4 were deployed at the desks (referred to as desk nodes). Since there were multiple desks in the room, the desk nodes

TABLE I
SPECIFICATIONS OF THE SENSORS USED IN THIS EXPERIMENT

Sensor	Parameter	Resolution	Accuracy
BH1750	Light	1 Lux	1.2 times
MAX4466	Sound	0.01V*	-
MH-Z14A	CO_2	5ppm	$\pm$50ppm
Digital PIR	Motion	-	-

*Sound level considered in terms of voltage and not dB.

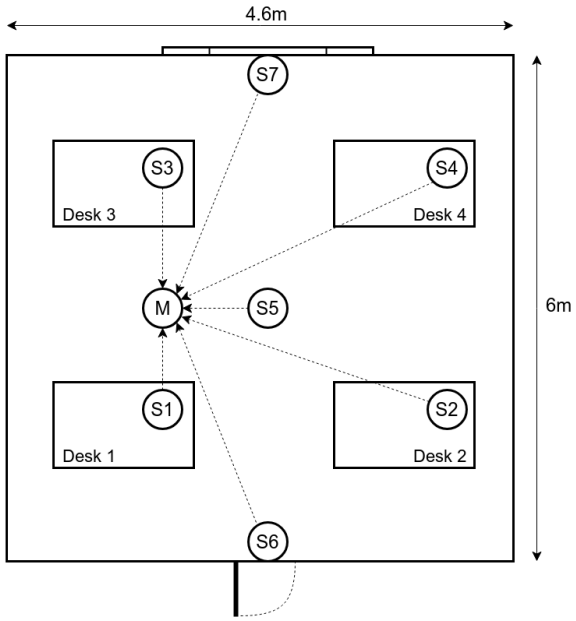

Fig. 1. A star network based data acquisition system deployed in a test room.

were planned to have low-cost sensors and therefore have temperature, light and sound sensors only. Node S5 had a CO_2 sensor which was kept in the middle to get the best possible reading of the room. Nodes S6 and S7 only contain PIR sensors and were deployed on the ceiling at an angle that maximized the sensor's field of view for motion detection.

Fig. 2 shows the block architecture of the sensor nodes as well as the master. In each sensor node, the Arduino Uno microcontroller board sampled data from the sensors and transmitted it periodically via a Zigbee module every 30 s. The temperature, light and CO_2 sensors were sampled only once in the 30 s time frame since these quantities seldom change in such a short duration. The PIR and sound sensors, however, need constant polling or else the events of interest are lost. Since the output pin of the PIR sensor remains high for about 3 s in repeat trigger mode, the Arduino polled the PIR every 2.5 s to check for the motion events. If even a single motion event was captured in the frame of 30 s, a '1' was sent to the master. For sound sensors, the algorithm churned out the maximum peak to peak voltage that was achieved in the time frame of 30 s. The master node only had a Zigbee radio for receiving data from the sensors and appending it to a file after merging the current time-stamp.

B. Data Preprocessing and Feature Engineering

The following tasks were done to convert the raw data received from the sensor nodes into a usable dataset:

1) All sensor nodes did not send their data at the same time and a few seconds of variation was found between the arrival time of different data points. Since a single dataset was required, we merged the timestamps within

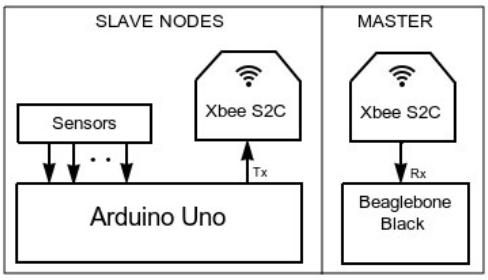

Fig. 2. Block architecture of the sensor nodes (left) and the master (right).

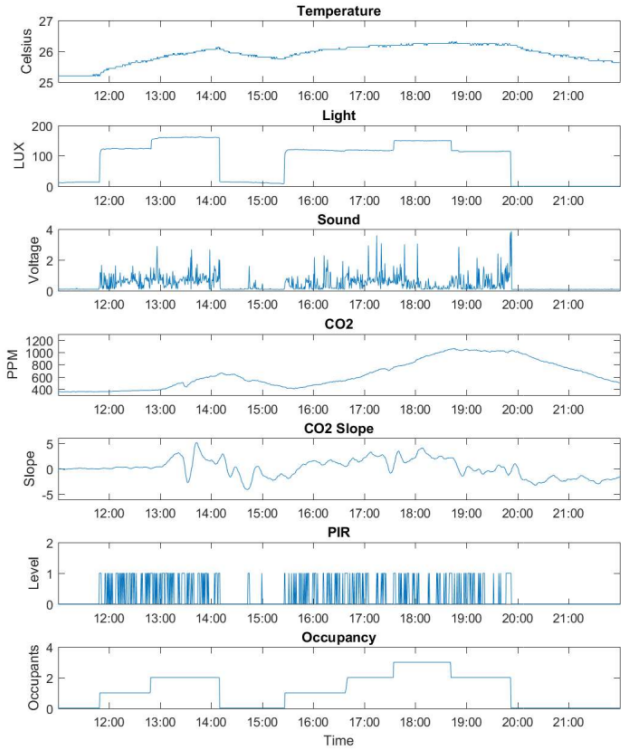

Fig. 3. Data from a few representative sensors and the CO_2 slope for a period of about 10 hours on 23/12/2017. The variation in these features is juxtaposed with the ground truth shown in the occupancy plot. All sensors except CO_2 and PIR belong to node S1.

a given time frame into a common vector.

2) Feature vectors with missing data were deleted since we had no concrete model to approximate PIR and sound values from historical data. This resulted in fewer but more credible vectors in our dataset. Missing data can be an issue if the system is real time.

3) The final dataset had over 10,000 points and 16 features where each feature was the data of a particular sensor. After looking at the time series plots of the sensors given in Fig. 3, CO_2 data seemed to give an excellent indication for the number of occupants in the room. However, it took several minutes for the readings to rise or fall to a steady state. Therefore, a new feature was derived in the form of slope of CO_2. This was calculated by fitting a linear regression in a window of 25 points at each instance and calculating the slope of the line. This parameter of 25 was obtained by trial and error with respect to the classification accuracy metric.

4) The ground truth was established manually wherein each person who entered or left the room signed the exact time along with the desk number in a register.

C. Machine Learning Algorithms

Supervised and unsupervised learning algorithms are extensively used in ML. We first built models using the former approach and then moved on to the latter for dimensionality reduction using PCA. Four supervised learning techniques were used: LDA, QDA, RF and SVM. The first three are inherently multiclass classifiers whereas SVM uses one-vs-one scheme to achieve multiclass classification. LDA and QDA assume a multivariate Gaussian distribution with μ_k and Σ_k for class conditional probability $P(X|y=k)$ where, k denotes the class and μ_k and Σ_k denote the mean and covariance matrix of class k, respectively. In LDA, $\Sigma_k = \Sigma$. The predictions are made using Bayes' rule:

$$P(y=k|X) = \frac{P(X|y=k)P(y=k)}{\sum_i P(X|y=i)P(y=i)} \quad (1)$$

Here, the class prior probability $P(y=k)$ is learned from the training data along with μ_k and Σ_k. There is no tunable hyperparameter.

In SVM, each n-dimensional feature vector is basically a point in an n-dimensional feature space, where n is the number of features present in the dataset. Each vector belongs to one of the k classes. Unlike LDA and QDA, SVM does not make any assumptions about the data. The algorithm attempts to fit an optimal hyperplane between the two classes with the help of support vectors. Therefore, for k classes, the number of classifiers learned by the algorithm are $\frac{k(k-1)}{2}$ which are put to a majority vote. A non-linear separation boundary can be obtained by using a kernel. SVM has a tunable penalty hyperparameter for indicating the tolerable error or misclassifications in fitting the hyperplane.

The RF classifier is an ensemble of decision trees where each tree is grown on some part of the dataset with replacement and has a vote. The final outcome is the average of the votes of all the trees. A decision tree, as the name suggests, is a tree-shaped structure in which each internal node represents a logical test on some feature and each leaf node represents one of the outcome classes. The tree forming algorithm used in this experiment is the classification and regression tree (CART) algorithm which utilizes the gini index for determining the quality of a split. The number of trees is a tunable hyperparameter along with a few others for pruning and preventing overfitting. We tuned the minimum samples split for the latter case.

PCA is a dimension-reducing unsupervised procedure in which a dataset having multiple and possibly correlated features is decomposed into a set of orthogonal components that capture the maximum amount of variance. This procedure is scale-variant. The reduced dataset is then fitted with the aforementioned supervised models for performance evaluation.

III. Experiments and Results

The ML algorithms discussed in the previous section were implemented using Scikit-learn [15]. Metrics such as accuracy, F1 score and confusion matrix were evaluated using 10-fold cross-validation. Since the data is of time-series nature, data was not shuffled prior to cross validation to avoid data points similar to the test data getting into the training data. In case of SVM, the training set features were normalized to have zero mean and unit variance since the algorithm is scale-variant. The normalization constants from the training set were used to scale the testing set at each iteration of the cross-validation loop. Apart from linear SVM, we also evaluated the results with RBF kernel to allow for non-linear classification boundaries. The penalty hyperparameter was varied from 10^{-4} to 10^{4} for each feature set and the best metric value is reported. For RF, the number of trees in the forest was kept at 30 and the metrics were calculated by averaging over 100 iterations. Since our dataset is skewed with more points corresponding to an empty room (zero occupancy), macro F1 score, which calculates metric for labels separately, is more reliable than micro F1 score [16]. Table II shows the accuracy and macro F1 scores of different combinations of features present in our dataset by the various employed ML algorithms.

In the first phase of supervised learning, only homogeneous fusion of data was done. This is documented in the first half of Table II. It can be seen from Table II that the proposed CO_2 slope feature shows promising results. It performs better than CO_2 for most algorithms and the performance improves significantly when both the features are combined. Other features, except for light, exhibit a good accuracy but a poor F1 score. The need for combining more features stems from this observation. Heterogeneous fusion of data was done in the next phase of supervised learning which is documented in the second half of Table II. To generate these feature combinations, we kept adding one sensor type at a time in a greedy fashion until we got the complete dataset. Light, however, was considered only in the end. From the first half of Table II, it can be seen that the best performance for the homogeneous case is achieved with light sensors with an accuracy of 97.3% and an F1 score of 0.929. This heavy bias towards light can be explained from the observation that in most cases, people tend to switch on the lights above their desks when they arrive and turn them off when they leave. However, when this assumption fails or is not valid, relying only on light sensors for occupancy estimation would leave our system vulnerable to false positives. For example, a person may leave the room with lights on. Also, in systems where the lights in a room need to be controlled based on the occupancy status, light cannot be taken as a feature. Therefore, light was not considered in the beginning of heterogeneous fusion even though it showed the best performance. The second last row of Table II is the complete dataset devoid of all the light sensors.

TABLE II
Cross Validation Accuracy and F1 score for Various Feature Sets and Algorithms

Feature	Metric	LDA	QDA	SVM (Linear)	SVM (RBF)	RF
Temp{1,2,3,4}	A	0.840	0.862	0.866	0.895	0.869
	F1	0.479	0.590	0.554	0.730	0.657
Light{1,2,3,4}	A	0.973	0.919	0.973	0.973	0.972
	F1	0.928	0.854	0.929	0.927	0.925
Sound{1,2,3,4}	A	0.851	0.879	0.875	0.885	0.887
	F1	0.449	0.544	0.542	0.591	0.601
PIR{6,7}	A	0.869	0.869	0.870	0.870	0.870
	F1	0.474	0.474	0.466	0.460	0.460
CO_2	A	0.809	0.808	0.812	0.812	0.763
	F1	0.383	0.409	0.286	0.314	0.329
Slope	A	0.852	0.831	0.870	0.870	0.876
	F1	0.387	0.394	0.462	0.510	0.564
CO_2, Slope	A	0.891	0.867	0.890	0.888	0.873
	F1	0.556	0.590	0.592	0.635	0.559
Temp{1,2,3,4}, CO_2, Slope	A	0.903	0.881	0.904	0.912	0.894
	F1	0.653	0.680	0.667	0.750	0.684
Temp{1,2,3,4}, CO_2, Slope, Sound{1,2,3,4}	A	0.920	0.908	0.933	0.924	0.918
	F1	0.735	0.749	0.793	0.782	0.731
Temp{1,2,3,4}, CO_2, Slope, Sound{1,2,3,4}, PIR{6,7}	A	0.922	0.910	**0.934**	0.924	0.919
	F1	0.737	0.748	**0.793**	0.780	0.734
Temp{1,2,3,4}, CO_2, Slope, Sound{1,2,3,4}, PIR{6,7}, Light{1,2,3,4}	A	0.980	0.957	0.982	**0.984**	0.978
	F1	0.946	0.911	0.948	**0.953**	0.933

*The numbers in curly bracket denotes the Sensor ID.
**A denotes Accuracy and F1 denotes macro F1 score.

As expected, the complete dataset (last row of Table II) which includes all the sensors, performs the best at estimating the number of occupants accurately. SVM with RBF kernel gives the best accuracy of 98.4% with an F1 score of 0.953. Other algorithms also exhibit similar performance except for QDA, which settles at an F1 score of 0.911. In the complete dataset devoid of all the light features, a good accuracy of 93.4% and a moderately high F1 score of 0.793 is achieved with linear SVM. The confusion matrices for the best case in both the feature sets are shown in Table III and Table IV.

SVM with RBF kernel performed better than linear SVM for most cases. The difference is significant when considering fewer features (rows 1, 3, 5, 6, 7, 8 of Table II) while the F1 score for both are comparable when the number of features are high (rows 9, 10, 11 of Table II). In the latter case, linear SVM can be considered as a better choice than SVM with a non-linear kernel because of faster computations. We can say that the Gaussian assumption for the data holds for most

TABLE III
CONFUSION MATRIX FOR LINEAR SVM CASE FOR THE COMPLETE DATASET DEVOID OF LIGHT FEATURES

	Predicted 0	Predicted 1	Predicted 2	Predicted 3
Actual 0	8117	43	41	27
Actual 1	104	336	19	0
Actual 2	65	48	502	133
Actual 3	21	7	154	512

TABLE IV
CONFUSION MATRIX FOR SVM WITH RBF KERNEL CASE FOR THE COMPLETE DATASET

	Predicted 0	Predicted 1	Predicted 2	Predicted 3
Actual 0	8196	1	3	28
Actual 1	0	453	6	0
Actual 2	0	0	712	36
Actual 3	10	1	67	616

cases as the performance of LDA and QDA is similar to that of SVM.

The best accuracy, as described above, is achieved when all the 16 features are considered. Since the four light features were already giving an excellent accuracy and F1 score, we performed PCA on the feature set without light (second last row of Table II) in an attempt to reduce the dimension from 12 to a significantly smaller value. Fig. 4 and Fig. 5 show the variation in accuracy and F1 score of all the five ML models respectively with the number of PCA components. Even with as low as four components, SVM with RBF kernel gives an accuracy of around 92% and a moderate F1 score of 0.72. Linear SVM also shows a similar performance.

IV. CONCLUSION AND FUTURE WORK

In this paper, we proposed a deployment scheme involving multiple multivariate sensor nodes for occupancy count estimation. We described various methods to process large amounts of data obtained from the WSN. The proposed slope of CO_2 feature, estimated using linear regression, improved the accuracy and F1 metric. The results show a promising 98.4% accuracy of occupancy estimation with a high F1 score of 0.953 using SVM with RBF kernel. Results for various other combinations of features have also been documented. An attempt was made to reduce the dimensions of the dataset using PCA. It has been shown that even with just four principal components, an accuracy of 92% and a moderate F1 score of 0.72 is achievable when light features are not considered.

The experiments in this work were conducted in a small room. We plan to extend this model to large workspaces in the future. Certain derived features like time and type of day can also be taken into account provided our dataset is large and spans multiple weeks. We also plan to conduct real-time experiments in the near future.

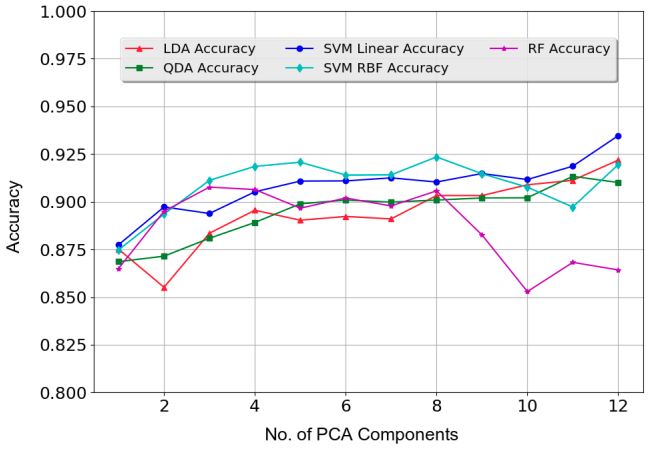

Fig. 4. Accuracy of different ML models with respect to the number of PCA components.

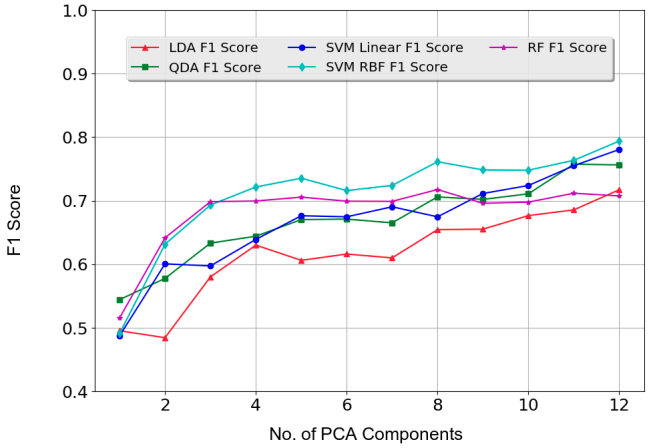

Fig. 5. F1 score of different ML models with respect to the number of PCA components.

ACKNOWLEDGMENT

This work was supported in part by the Research Council of Norway for which the authors are thankful. The authors are also thankful to the Center for International Mobility (CIMO grant no. Intia-1-2016-03) and Aalto University, Finland, for their financial support.

REFERENCES

[1] V. Garg and N. Bansal, "Smart occupancy sensors to reduce energy consumption," *Energy and Buildings*, vol. 32, no. 1, pp. 81 – 87, 2000.

[2] V. Erickson, S. Achleitner, and A. Cerpa, "POEM: Power-efficient occupancy-based energy management system," in *Proc. 12th Int. Conf. Inform. Process. Sensor Netw. (IPSN)*. New York, NY, USA: ACM, 2013, pp. 203–216.

[3] B. Balaji, J. Xu, A. Nwokafor, R. Gupta, and Y. Agarwal, "Sentinel: Occupancy based HVAC actuation using existing WiFi infrastructure within commercial buildings," in *Proc. 11th ACM Conf. Embedded Networked Sensor Systems*. New York, NY, USA: ACM, 2013, pp. 17:1–17:14.

[4] J. Scott et al., "Preheat: Controlling home heating using occupancy prediction," in *Proc. 13th Int. Conf. Ubiquitous Comput.*, ser. UbiComp. New York, NY, USA: ACM, 2011, pp. 281–290.

[5] N. Li, G. Calis, and B. Becerik-Gerber, "Measuring and monitoring occupancy with an RFID based system for demand-driven HVAC operations," *Autom. construction*, vol. 24, pp. 89–99, 2012.
[6] I. Ang, F. Salim, and M. Hamilton, "Human occupancy recognition with multivariate ambient sensors," in *IEEE Int. Conf. Pervasive Comput. and Commun. Workshops (PerCom Workshops)*, Mar. 2016, pp. 1–6.
[7] L. Candanedo and V. Feldheim, "Accurate occupancy detection of an office room from light, temperature, humidity and CO_2 measurements using statistical learning models," *Energy and Buildings*, vol. 112, pp. 28 – 39, 2016.
[8] E. Hailemariam, R. Goldstein, R. Attar, and A. Khan, "Real-time occupancy detection using decision trees with multiple sensor types," in *Proc. Symp. Simulation Architecture and Urban Des. (SimAUD)*, San Diego, CA, USA, 2011, pp. 141–148.
[9] K. Lam et al., "Occupancy detection through an extensive environmental sensor network in an open-plan office building," *IBPSA Building Simulation*, vol. 145, pp. 1452–1459, 2009.
[10] B. Dong et al., "An information technology enabled sustainability test-bed (itest) for occupancy detection through an environmental sensing network," *Energy and Buildings*, vol. 42, no. 7, pp. 1038 – 1046, 2010.
[11] Z. Yang, N. Li, B. Becerik-Gerber, and M. Orosz, "A multi-sensor based occupancy estimation model for supporting demand driven HVAC operations," in *Proc. Symp Simulation Architecture and Urban Des. (SimAUD)*, San Diego, CA, USA, 2012, pp. 2:1–2:8.
[12] M. Masood, Y. Soh, and V. Chang, "Real-time occupancy estimation using environmental parameters," in *Int. Joint Conf. on Neural Netw. (IJCNN)*, Jul. 2015, pp. 1–8.
[13] Z. Chen et al., "Building occupancy estimation with environmental sensors via CDBLSTM," *IEEE Trans. Ind. Electron.*, vol. 64, no. 12, pp. 9549–9559, Dec 2017.
[14] A. Dey et al., "Namatad: Inferring occupancy from building sensors using machine learning," in *IEEE 3rd World Forum Internet of Things (WF-IoT)*, Dec. 2016, pp. 478–483.
[15] F. Pedregosa et al., "Scikit-learn: Machine learning in python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, Nov. 2011.
[16] P. Perner, *Machine Learning and Data Mining in Pattern Recognition: 10th International Conference, MLDM 2014, St. Petersburg, Russia, July 21-24, 2014, Proceedings*. Springer International Publishing, 2014.