Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 171 (2020) 1943–1952

Procedia Computer Science

www.elsevier.com/locate/procedia

Third International Conference on Computing and Network Communications (CoCoNet'19)

IoT-based Indoor Occupancy Estimation Using Edge Computing

Krati Rastogi, Divya Lohani*

Department of Computer Science & Engineering, Shiv Nadar University, Gautam Buddha Nagar, India, 201314

Abstract

Indoor occupancy estimation has become an important area of research in the recent past. This work investigates the feasibility of an Internet of Things (IoT) based university classroom occupancy estimation system. As IoT devices generate voluminous data at high rates, the centralized cloud computing approach is found to generate high latencies. The client server based cloud architecture has been compared with the decentralized edge computing architecture for building the occupancy estimation system. The performance of these architectures has been compared using two performance metrics: latency and throughput. The occupancy estimation models using carbon dioxide and relative humidity as inputs, have been developed using multiple linear regression and quantile regression. Root Mean Square Error (RMSE) and Mean Absolute Percentage Error (MAPE) have been used to compare the performance of our estimation models.

Keywords: Occupancy estimation; Edge Computing; Cloud Computing; Multiple Linear Regression; Quantile Regression

1. Introduction

Indoor occupancy estimation has become an important area of research in the recent past. Information about the number of people entering or leaving a commercial building is useful in estimation of hourly sales, dynamic seat allocation, detection of security threats and building climate control. Entry control and access surveillance in

* Corresponding author. Tel.: +91-120-3819100.
E-mail address: divya.lohani@snu.edu.in

1877-0509
10.1016/j.procs.2020.04.208

shopping malls, airports, bus/train stations and commercial/residential buildings is another important application of occupancy estimation models. Buildings consume about 40% of the global energy, half of which is used for Heating, Ventilation, and Air Conditioning (HVAC) [1]. One way to improve building energy efficiency at low costs is to reduce the energy consumption of HVAC. Occupancy estimation permits maintaining temperature, illuminance and indoor air quality at a comfortable range while minimizing energy use. Accurate occupancy estimates are useful in (1) minimizing energy consumption by controlling HVAC, (2) tracking human movement, (3) maintaining adequate air quality, (4) improving building security, and (5) tracking and rescuing survivors in emergencies such as fires [2][3]. Educational institutions such as schools and universities have classroom occupancy only during daytime, and that too during the lectures hours. The occupant count of students is highly erratic and may fall sharply during weekends, early morning, late afternoon, festive seasons and rainy days. But most educational institutions use HVAC for the entire day without taking student occupancy into consideration [4] [5]. Knowing the occupancy of any indoor space will help university staff make informed choice about its usage, energy savings in HVAC usage, scheduling of activities etc.

IoT has revolutionized the way we monitor and analyse physical phenomenon indoors and outdoors. Small size and low cost sensors and chips relay data and communicate with each other without human involvement, enabling frequent sampling, enhancing the range of monitoring and sampling and decreasing the need of human labour. It is seen that deployment of devices, connections and communication with the Internet becomes easy with the use of IoT. The perpetual connectivity to the Internet and ease of storing high volumes of data in the cloud eliminated the requirement of large storage and communication algorithms with the sensing devices [6]. The edge nodes have high computational capability, storage and internet connectivity [7]. Most of the computational tasks and data processing is performed at these intermediate nodes. As large amount of data can be processed near the source, the requirement of internet bandwidth is reduced. Processing of data near the source also decreases the traffic and network latency. Edge computing increases resilience of the network by enabling functioning of the IoT devices at the edge even when the connectivity to the cloud is lost [8]. Hence, edging compliments the cloud based architecture by allowing the cloud to perform resource intensive, time consuming analytics while short term analytics is handled at the edge. Edge computing is suited in our work as we have deployed a large number of sensor nodes in university classrooms and they continuously stream real time sensor data [9].

Contribution of this work: This work presents indoor occupancy estimation models using of multiple linear regression and quantile regression. The proposed models use CO_2 and relative humidity as inputs, to estimate the occupancy of an indoor space. As the data generated by out IoT setup is very large, so transmitting all the data to the cloud for computation may incur delay in estimation; therefore, the proposed models are made to run on an edge device. Later in the paper, a comparison of edge computing and cloud computing has also been shown.

2. Related Work

In this section, we first discuss about the edge and cloud computing approaches followed by the developments in the realm of occupancy estimation. Three implementations of Edge computing, viz. cloudlet, fog computing and mobile edge computing have been described and compared in [11]. Decision tree algorithm has been used to select one of the three implementations according to a particular application, based on a set of input parameters used for comparison. Zhuo Chen et al. [12] have first outlined the drawbacks of cloud computing. Fog computing, dew computing and mobile edge computing have then been analyzed using application scenarios. Future prospects of the emerging technologies have been in the end. Y. Zhou et al. [13] have demonstrated and evaluated the performance of edge computing in wearable cognitive assistance applications. The authors have demonstrated that the end to end latency decreased by 70% by the introduction of edge computing architecture. A novel approach to improve network performance in the absence of edge computing is proposed in [14]. Simulations to test the proposed approach show substantial improvement in job completion rate although the energy rate is increased. Vehicular fog computing is proposed in [15] to improve the storage and processing power of fog nodes. The proposed fog architecture is implemented and evaluated. No major work has presented a comparative study between cloud and edge computing architectural implementations. This work describes the implementation of cloud and edge based IoT architectures and discusses the comparative performance of the two architectures using suitable metrics.

Determination of occupant count for various purposes has been reported by several authors. Candanedo et al. [16] have predicted occupancy in an office room using data from light, temperature, humidity and CO_2 sensors. The statistical classification models evaluated in this work for prediction are CART, RF, GBM and LDA. Chaoyang Jiang [17] has developed an indoor occupancy estimator for the number of real-time indoor occupants based on CO_2 measurement. The Feature Scaled Extreme Learning Machine (FS-ELM) algorithm, which is a variation of the standard Extreme Learning Machine (ELM), has been explored as occupant estimator. It greatly improves the performance of the standard ELM. [18] have developed an indoor environmental data-driven model for occupancy prediction using machine learning techniques. HMM and CART decision tree algorithms were implemented using MATLAB's Statistics and Machine Learning Toolbox. Bing Dong [19] has presented a methodology reduce energy consumption based on prediction of occupant behavior patterns and local weather conditions. Adaptive Gaussian Process, Hidden Markov Model, Episode Discovery and Semi-Markov Model are modified and implemented into this study. CO_2, acoustics, motion and lighting changes have been used as input features.

An approach to estimate people count in office space using distributed, strategically placed PIR sensors has been presented in [20]. A floor-wide simulation of realistic occupant behaviours was performed to investigate two algorithms to estimate people count per office space. An estimation system to detect the number of persons and their directions of movement at an entrance has been proposed [21]. The output of IR sensor array is binarized using the background mean method and the number of persons and their directions of movement are determined by a pattern recognition algorithm. General state-space models have been developed in [22] that use IR sensor data to estimate occupancy in different rooms of a house has been presented. Actual as well as artificial data obtained by adding noise to data collected in an experimental house is used. 10% - 40% improvement in accuracy was achieved by the proposed method over conventional estimation methods in experimental results.

3. IoT System Architecture and Testbed

In this work, we have used two architectures – cloud based architecture (cloud is used both for processing and storage) and edge based architecture (where cloud is used only to storage) for end to end IoT-based occupancy estimation system:

A. IoT Sensing System

A sensor node is placed in every classroom to collect and transmit the sensed data to IoT cloud. The sensor nodes used in this work are composed of relative humidity (RH), CO_2 and temperature sensors which are connected to Arduino Uno micro-controller. MQ135 gas sensor has been used for detection of CO_2, DHT22 sensor for temperature and humidity and IR sensor for determining the person count (Table 1). The Arduino Uno micro-controller is responsible for collection, calibration and transmission of the sensed data. It is based on ATmega 328P 8-bit micro-controller. The cloud and edge computing architectures are discussed in the following lines.

TABLE 1. DETAILS OF SENSORS USED

Parameter	Unit	Sensor
Temperature	^oC	DHT22
Relative Humidity	%	DHT22
CO_2	ppm	MQ135
Person count	Persons	Sharp GP2Y0A02YK0F Infrared Proximity Sensor

(a) Cloud based architecture: The sensor nodes use Wifi module *ESP 8266* to transmit the sensed RH, CO_2 and temperature values to the IoT cloud. As shown in Fig. 1, the data sensed by the IR, DHT22 and MQ135 sensors is collected by the micro-controller board. The micro-controller board transmits the sensed data to the IoT cloud using the *ESP 8266* Wifi module. Estimation models are applied to the aggregated data at the cloud and the results are provided to the end users in form of plots and alerts. The occupancy estimates are made available to the end users with the help of an Android application.

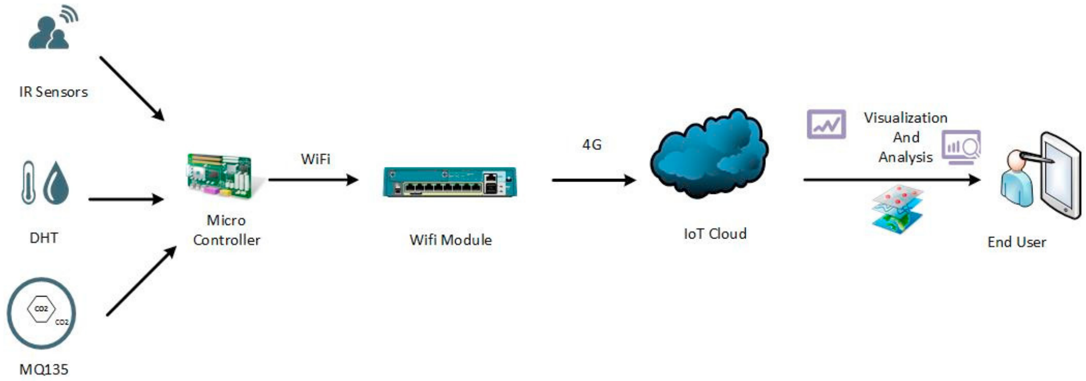

Fig. 1: Hardware architecture of the proposed cloud computing approach

(b) Edge computing architecture: In the edge computing architecture, the sensor node transmits the sensed data to the smart edge first instead of uploading it directly to the cloud. The sensor node communicates with the edge device, in our case is a Raspberry Pi 3 model B has been used as the edge node (Fig. 2). Raspberry Pi is a single board computer with 1.2 GHz, 64-bit quad-core processor, 1 GB RAM and executes a Linux based OS. The estimation models run on the RPi edge node and only the result analysis is uploaded on the cloud. The occupancy estimates are made available to the end users with the help of an Android application.

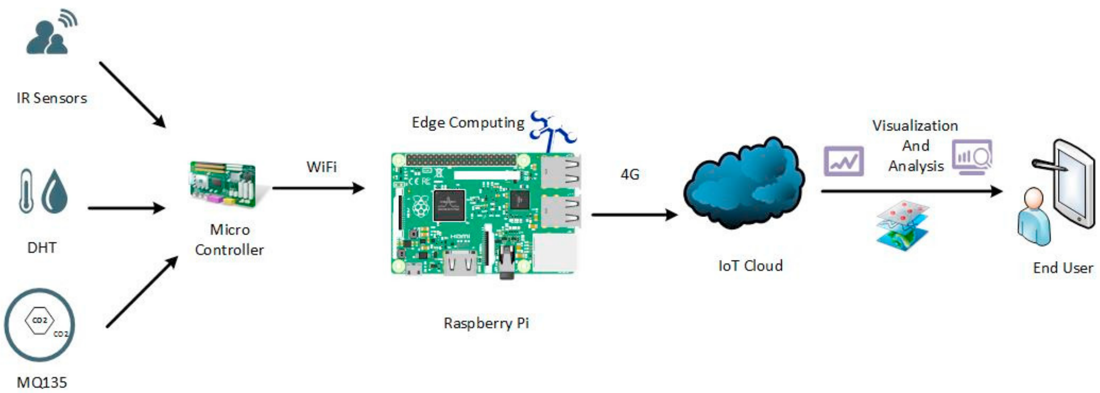

Fig. 2: Hardware architecture for the edge computing approach

B. Experimental Setup and Data Collection

The study has been conducted in the classrooms of the Engineering block of the university. The rooms have centralized heating, ventilating and air conditioning (HVAC) system. The classroom windows are fully covered and the doors are closed during the experiment. The sensor node was placed at the centre of the classrooms. The average dimensions of the classrooms under study are 144×40×16 feet.

The sensor node was kept in classroom and data was collected at a rate of 1 sample per minute. Relative humidity (RH), CO_2 and temperature values collected by the sensor node have been used to estimate occupancy inside the classroom during lectures. Each experimental observation lasted for two hours and occupancy, RH and CO_2 concentration was recorded using our setups every minute. Average values of RH and CO_2 have been used to estimate occupancy. Temperature has been kept constant during the experiment.

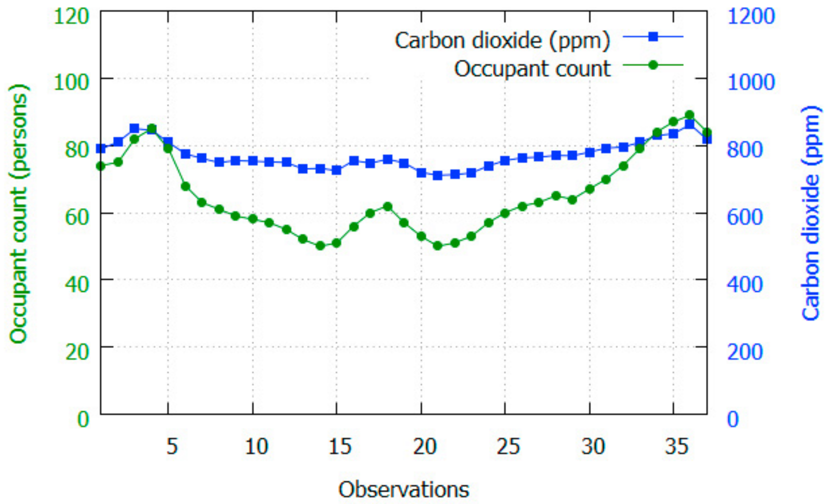

Fig. 3: Plot of CO_2 and occupancy count against 2-hour observations

Fig. 3 and 4 show the variation in indoor occupant count and RH at different periods of observation against the number of occupant students present inside. Each observation lasted for 2 hours and average values of recorded RH and CO_2 were used to plot the graph. It can be seen that with the change in occupancy both RH and CO_2 change.

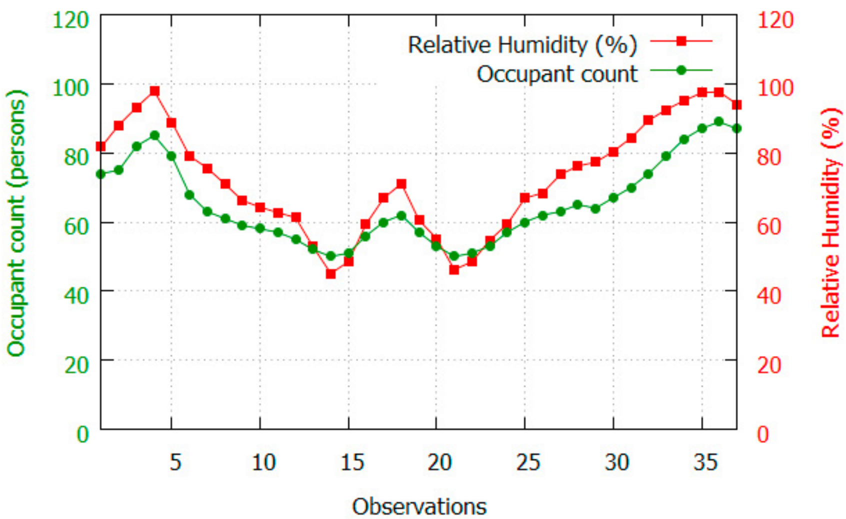

Fig. 4: Plot of RH and occupancy count against 2-hour observations

4. Occupancy Estimation Models

In this work, occupancy estimation systems have been developed using RH and CO_2. Regression based models tend to determine a relationship between dependent (RH and CO_2) and independent variables, the independent variable being the occupancy to be estimated.

(a) *Multiple Linear Regression Model*: The occupancy estimate (**Y**) depends on multiple inputs (**X_1, X_2, .. ,X_n**), represented by the equation:

$$Y = a_1X_1 + a_2X_2 + .. + a_nX_n + c \quad (1)$$

Where,

***Y** = dependent variable (occupancy count),*

***X_1, X_2, .. , X_n**= independent variables (RH and CO_2),*

***a_1, a_2, .., a_n**= coefficients of independent variables, and*

***c** = intercept.*

Using MLR, the estimation equation that uses CO_2 as input is given as:

$$Y = -155.73 + 0.286 \times X \quad (2)$$

The estimation equation using only RH as input is obtained as:

$$Y = 13.56 + 0.711 \times X \quad (3)$$

MLR equation that uses both RH and CO_2 is found to be:

$$Y = -80.966 + 0.336 \times RH + 0.158 \times CO_2 \quad (4)$$

(b) Quantile Regression Model: The linear regression technique is highly sensitive to outliers. An outlier is an observation that significantly differs from other observations of the time-series. Spikes and dips in the observed time-series of occupant count is best handled using quantile regression. Quantile regression is an extension to the linear regression technique, and serves to nullify or offset the effect of outliers on the trend line or regression line. To define quantile regression model, we consider 'n' observations of relative humidity (X_1) and carbon dioxide concentration (X_2). The occupancy count Y_i is bound within a known interval y_{min} and y_{max}:

$$Y_i = X_{1i}.\beta_{1i} + X_{2i}.\beta_{2i} + \epsilon_i \quad (5)$$

Where, Yi is the occupancy to be estimated, X_1 is relative humidity value, X_2 is the CO_2 concentration and $\beta_i = \{\beta_1, \beta_2, .. \beta_n\}$ and it represents the unknown regression parameters. The 'p' quantile of the conditional distribution of the estimated occupancy y_i, given relative humidity x_{1i} and CO_2 concentration $x_{2i}$is described as:

$$Q_y(p) = x_{1i}.\beta_{1p} + x_{2i}.\beta_{2p} \quad (6)$$

Where, ($0 < p < 1$) indicates the proportion of the population having scores below the quantile at p. In this way, we are able to eliminate the outliers, and get better trend-line. For any quantile p, there exists a fixed set of parameters p and the non-decreasing function 'h' from the interval (y_{min}, y_{max}) to the real line such that:

$$h\{Q_y(p)\} = x_{1i}.\beta_{1i} + x_{2i}.\beta_{2i} \quad (7)$$

The logistic transformation is defined as:

$$h(y_i) = \log(\frac{y_i - y_{min}}{y_{max} - y_i}) = \log(it\ (y_i)) \qquad (8)$$

We get the inverse function by integrating equations (7) and (8):

$$Q_y(p) = \frac{exp(x_i\ .\beta_p).y_{max} + y_{min}}{1 + exp(x_i\ .\beta_p)} \qquad (9)$$

Inference on $Q_y(p)$ can be made through the inverse transform in equation (6):

$$Q_{h(y_i)}(p) = Q\log(it(y_i))(p) = x_{1i}.\beta_{1i} + x_{2i}.\beta_{2i} \qquad (10)$$

As compared with other regression methods for occupancy estimation, quantile regression is most suited for our problem of estimating occupancy in classrooms and laboratories. The occupancy in the University classrooms and laboratories is sporadic. The occupancy is high during regular lectures and classes, and variable at other times. This gives rise to outliers in occupancy time-series. Quantile regression equation for our estimation model using only CO_2 as input is:

$$Y = -147.37 + 0.275\times X \qquad (11)$$

The estimation equation using only RH is given as:

$$Y = 15.36 + 0.681\times X \qquad (12)$$

Quantile regression equation that uses both RH and CO_2 is found to be:

$$Y = -87.09 + 0.302\times RH + 0.169\times CO_2 \qquad (13)$$

5. Performance Evaluation of Occupancy Estimation Models

To evaluate the performance of multiple linear regression based occupancy model and quantile regression based model, following metrics have been used:

(i) The *coefficient of determination* (R^2): It is defined as

$$R^2 = 1 - \frac{u}{v} \qquad (14)$$

Where 'u' is the residual sum of squares and 'v' is the total sum of squares. The best possible value is 1.

(ii) *Root Mean Square Error (RMSE)*: It is the square root of the second moment of the difference between actual and predicted indoor occupant counts. As errors are squared before averaging, RMSE is sensitive to outliers. RMSE is represented as

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - f(x_i))^2} \qquad (15)$$

Where,
N is the number of observations,

y_i is the occupancy count at time instant i,
x_i is the input vector (time), and
f is the estimation algorithm.

(iii) Mean Absolute Percentage Error (MAPE): MAPE is the measure of accuracy of an estimation algorithm. It is described by the equation below

$$MAPE = \frac{1}{N}\sum_{i=0}^{N} \left| \frac{y_i - f(x_i)}{y_i} \right| * 100 \tag{16}$$

Where,
N is the number of observations,
y_i is the occupancy count at time instant i,
x_i is the input vector (time), and
f is the estimation algorithm.

TABLE 2. COMPARISON OF ESTIMATION MODELS' R^2, RMSE AND MAPE

Estimation Model	**R^2**	**RMSE (persons)**	**MAPE (%)**
M.L.R.	0.88	2.47	2.92
Q.R.	0.91	2.65	2.51

MAPE is lesser for the QR model, indicating that the estimates with QR are more accurate than that of the MLR model. RMSE is lesser for MLR than QR, which shows that MLR performs fewer or lesser large errors.

6. Edge Computing vs. Cloud Computing Architectures

Following network metrics have been used to compare the performances of the two architectures:

(i) Latency: Latency is the time required to transmit the data packets across a network. The sensed data is uploaded directly to IoT cloud in the cloud computing approach and via smart edge device in the edge computing approach, and occupancy estimates and other information is accessed by the end user. The time duration from sensing of the parameters to end user availability of occupancy estimates using those sensed values is calculated for both the approaches. The end-to-end delay is a sum of network delay, processing delay and buffering delay.

(a) Network Delay: The delay caused while the packets are travelling through the network is known as network delay. It is the sum of transmission delays from sensor node to smart edge, edge node to cloud server and cloud server to the Android application in case of edge computing approach and the sum of delays from sensor node to IoT cloud and cloud server to the Android application in case of cloud computing approach.

(b)Processing Delay: The delay experienced due to data processing tasks such as calibration at the sensor node and the running time of the occupancy estimation algorithms is termed as processing delay. The estimation algorithms are stored in form of a python code in the edge node in case of the edge computing approach and in the AWS cloud in the cloud computing approach.

The end-to-end delay or the average latency for both approaches is computed in this work using a packet sniffing tool called *wireshark*. Packet analysers or sniffers are computer programs that can intercept incoming and outgoing packets in a network. Using the captured packets, delay for the two approaches is calculated. An ICMP packet is used to calculate the round trip time (RTT) between sensor node and end user device with the Android application. RTT measures the time spent in sending a packet from the sensor node to receiving the acknowledgment packet from the end user smartphone. RTT thus obtained is used to compute network delay. End-to-end delay is the sum of network delay (=RTT/2) and processing delay (Table 3). Buffering delay is not considered while calculating the end-to-end delay as it is small as compared with other delays as approximately the same for both the architectural approaches.

TABLE 3. COMPARISON OF PERFORMANCE FOR EDGE AND CLOUD COMPUTING APPROACHES

Parameter	Architecture	
	Cloud	Edge
RTT (seconds)	00.931	00.685
Duration (bits tranfer in sec)	63.846	12.454
Throughput (MBps)	00.158	00.073

(ii) Throughput: It is defined as the quantity of data being sent/received by unit of time. As seen in Table 2, the throughput of the edge node is about double than that of a single sensor node that directly uploads sensed data to the AWS cloud. Bit-rate or bits transferred per second (duration) are higher for the cloud architecture as sensed data is continuously uploaded to the IoT cloud. The data transmission from the edge device to the cloud is not continuous. Instead, it is in the form of bursts and takes place only after aggregation, processing and application of estimation algorithms takes place.

7. Conclusion and Future Work

IoT based occupancy estimation models are developed taking into account indoor environmental parameters. IoT sensor setup uses an array of sensors to collect relative humidity (RH), CO_2 concentration and occupant count values. Multiple linear regression and quantile regression based techniques have been used to estimate indoor occupancy using RH and CO_2 concentrations. Quantile regression based estimation models have been found to be the most accurate with a MAPE of 2.51%. On comparing the edge based IoT system with cloud based IoT system, it was found that edge processing performed better in terms of RTT and network traffic. As a future work, we propose to build and evaluate occupancy estimation system using non-linear estimation models such as support vector machines and artificial neural networks on the edge.

References

[1] Frauke Oldewurtel, David Sturzenegger, Manfred Morari, "Importance of occupancy information for building climate control, *Applied Energy*," Volume 101, 2013, Pages 521-532, ISSN 0306-2619.

[2] K. Akkaya, I. Guvenc, R. Aygun, N. Pala and A. Kadri, "IoT-based occupancy monitoring techniques for energy-efficient smart buildings," *2015 IEEE Wireless Communications and Networking Conference Workshops* (WCNCW), New Orleans, LA, 2015, pp. 58-63.

[3] Timilehin Labeodan et al., "On the application of wireless sensors and actuators network in existing buildings for occupancy detection and occupancy-driven lighting control," *Energy and Buildings,* 127 (2016) 75–83.

[4] Theis Heidmann Pedersen, Kasper Ubbe Nielsen, Steffen Petersen, "Method for room occupancy detection based on trajectory of indoor climate sensor data," *Building and Environment*, 115 (2017) 147-156.

[5] M. K. Masood, Yeng Chai Soh and V. W. Chang, "Real-time occupancy estimation using environmental parameters," *2015 International Joint Conference on Neural Networks (IJCNN)*, Killarney, 2015, pp. 1-8.

[6] P. Rathore, A. S. Rao, S. Rajasegarar, E. Vanz, J. Gubbi and M. Palaniswami, "Real-Time Urban Microclimate Analysis Using Internet of Things," in IEEE Internet of Things Journal, vol. 5, no. 2, pp. 500-511, April 2018.

[7] C. Esposito, A. Castiglione, F. Pop and K. R. Choo, "Challenges of Connecting Edge and Cloud Computing: A Security and Forensic Perspective," in IEEE Cloud Computing, vol. 4, no. 2, pp. 13-17, March-April 2017. doi: 10.1109/MCC.2017.30

[8] J. Pan and J. McElhannon, "Future Edge Cloud and Edge Computing for Internet of Things Applications," in IEEE Internet of Things Journal, vol. 5, no. 1, pp. 439-449, Feb. 2018. doi: 10.1109/JIOT.2017.2767608

[9] J. Ren, G. Yu, Y. He and G. Y. Li, "Collaborative Cloud and Edge Computing for Latency Minimization," in IEEE Transactions on Vehicular Technology, vol. 68, no. 5, pp. 5031-5044, May 2019. doi: 10.1109/TVT.2019.2904244

[10] H. El-Sayed et al., "Edge of Things: The Big Picture on the Integration of Edge, IoT and the Cloud in a Distributed Computing Environment," in IEEE Access, vol. 6, pp. 1706-1717, 2018. doi: 10.1109/ACCESS.2017.2780087

[11] K. Dolui and S. K. Datta, "Comparison of edge computing implementations: Fog computing, cloudlet and mobile edge computing," *2017 Global Internet of Things Summit (GIoTS)*, Geneva, 2017, pp. 1-6.

[12] Y. Zhou, D. Zhang and N. Xiong, "Post-cloud computing paradigms: a survey and comparison," in Tsinghua Science and Technology, vol. 22, no. 6, pp. 714-732, 2017.

[13] Zhuo Chen et al., "An empirical study of latency in an emerging class of edge computing applications for wearable cognitive assistance," in Proceedings of the Second ACM/IEEE Symposium on Edge Computing (SEC '17), ACM, New York, USA, Article 14, 2017.

[14] S. Sthapit, J. Thompson, N. M. Robertson and J. R. Hopgood, "Computational Load Balancing on the Edge in Absence of Cloud and Fog," in IEEE Transactions on Mobile Computing, vol. 18, no. 7, pp. 1499-1512, 1 July 2019.

[15] M. Sookhak et al., "Fog Vehicular Computing: Augmentation of Fog Computing Using Vehicular Cloud Computing," in IEEE Vehicular Technology Magazine, vol. 12, no. 3, pp. 55-64, Sept. 2017.Luis M. Candanedo et al., "Accurate occupancy detection of an office room from light, temperature, humidity and CO_2 measurements using statistical learning models," Energy and Buildings 112 (2016) 28–39.

[16] Chaoyang Jiang, "Indoor occupancy estimation from carbon dioxide concentration," Energy and Buildings 131 (2016) 132–141.

[17] Seung Ho Ryu et al., "Development of an occupancy prediction model using indoor environmental data based on machine learning techniques," Building and Environment 107 (2016) 1-9.

[18] Bing Dong, "A real-time model predictive control for building heating and cooling systems based on the occupancy behavior pattern detection and local weather forecasting," BUILD SIMUL (2014) 7: 89–106.

[19] I. P. Mohottige and T. Moors, "Estimating Room Occupancy in a Smart Campus using WiFi Soft Sensors," 2018 IEEE 43rd Conference on Local Computer Networks (LCN), Chicago, IL, USA, 2018, pp. 191-199.doi: 10.1109/LCN.2018.8638098.

[20] F. Wahl, M. Milenkovic and O. Amft, "A Distributed PIR-based Approach for Estimating People Count in Office Environments," 2012 IEEE 15th International Conference on Computational Science and Engineering, Nicosia, 2012, pp. 640-647. doi: 10.1109/ICCSE.2012.92.

[21] K. Hashimoto, K. Morinaka, N. Yoshiike, C. Kawaguchi and S. Matsueda, "People count system using multi-sensing application," Proceedings of International Solid State Sensors and Actuators Conference (Transducers '97), Chicago, IL, USA, 1997, pp. 1291-1294 vol.2.doi: 10.1109/SENSOR.1997.635472.

[22] D. Linzmeier, "Pedestrian detection with thermopiles using an occupancy grid," *Proceedings. The 7th International IEEE Conference on Intelligent Transportation Systems (IEEE Cat. No.04TH8749)*, Washington, WA, USA, 2004, pp. 1063-1068. doi: 10.1109/ITSC.2004.1399054.

[23] Brennan, Colin & W. Taylor, Graham & Spachos, P. (2018). Designing Learned CO_2-based Occupancy Estimation in Smart Buildings. IET Wireless Sensor Systems. 8. 10.1049/iet-wss.2018.5027.