Implementation Of Statistical Learning Model For Room Occupancy Detection

Raja Fazliza Raja Suleiman[1], Muhammad Iqbal Nebil[2]

[1]UniKL Robotics and Industrial Automation Center (URIAC),
[1,2]Industrial Automation Section, Universiti Kuala Lumpur Malaysia France Institute, Section 14, Bangi, Selangor, Malaysia

e-mail: [1]fazliza@unikl.edu.my

Abstract
This paper presents several room occupancy detection methods using statistical learning model. Occupancy detection system is mainly used for energy saving in green buildings such as offices and residential apartments. The system will automatically switched-off the lighting, heating or ventilation appliances when the room is empty. The proposed work uses temperature and humidity sensor to detect human presence. All the input values from this sensor are transmitted to an IoT platform called Blynk (for data monitoring), through the medium of an open-source microcontroller board NodeMCU. The collected data is analyzed using two different approaches which are supervised learning model and unsupervised learning model. Results show that for supervised learning, SVM performs slightly better than Decision Tree. While for unsupervised subspace learning, Minimax yields better probability of detection than SVD in worst case criterion.

Keywords: Internet of Thing (IoT), decision tree, minimax, room occupancy detection, subspace learning, SVD, SVM.

1. INTRODUCTION

Room occupancy detection are performed in a building mainly to reduce electricity wastage. Different type of sensors can be used to detect human presence in a room such as temperature, carbon dioxide (C02), humidity and light sensors [1], [2]. Pedersen et al. proposed an occupancy detection method based on indoor climate data trajectory measured from Co2 and passive infrared sensors (PIR) [3]. Other than that, we can find several works in the literature that applied machine learning methods [4] and few of statistical learning model. This is approved by Candanedo quoted possibility to guaranteed high accuracies of detection of occupancy based on statistical approaches by using different type of sensors [2]. Han et al. proposed a statistical-based method to detect occupancy patterns and assessing the quality of the indoor energy-efficient buildings using wireless network of PIR, CO2 and humidity sensors [5]. One of the most common parameter used to measure occupancy is Co2 level inside a room. However, the setback is inaccuracy due to different person gave out different amount of Co2 level.

Learning in general can be categorized into supervised, unsupervised, online and reinforcement learning [6]. Statistical learning allows to study the estimation and trends analysis of a data set. With machine learning, all the input data and expected result are processed to generate the rules, which are then applied to produce original answers.

Supervised learning algorithm are trained as a labelled example, such that the inputs were the desired output is known. The training algorithm learns from a data set fed and mimics the important features through methods like prediction, classification, and gradient descent [7], [8].

Supervised learning uses data with labels, while unsupervised learning uses unlabeled data. Unsupervised learning works well on transactional data. Example of the popular techniques include nearest-neighboring mapping, Support Vector Decomposition (SVD) [9], K-means clustering [10] and minimax [11]. These algorithms allows to segment outliers effectively.

Based on the problem and previous work mentioned in paragraph 1 and 2 and taking the advantages of using IoT technologies in pursuing Industrial Revolution 4.0, this paper propose to study room occupancy detection performance using statistical learning model. The data is collected from humidity and temperature sensor. The data is saved and monitored using Blynk application. Then the data collected is analyzed and trained using two models which is supervised learning (SVM and Decision Tree) and unsupervised subspace learning (minimax and SVD).

2. METHODOLOGY

A. Research Framework

For this research work, we use one input device which is DHT11 sensor to measure the humidity and temperature value. Figure 1 shows the process diagram of our work. The system starts off with the microcontroller NodeMCU ESP8266 being initialized. Then the sensor will measure the humidity and temperature value inside a room. These measured data are stored in cloud and can be monitored through IoT platform that we use (Blynk). Then data will be analyzed in Matlab using machine learning model. Figure 2 shows the working principle of humidity and temperature sensor DHT11. Basically both humidity and temperature use the same principle in DHT11 sensor.

B. Hardware Component

1) DHT11 sensor: This sensor is used to monitor and collect both humidity and temperature data. To measure humidity, the sensor uses the humidity sensing component which has two electrodes with moisture holding substrate between them as shown in Figure 2. The conductivity of the substrate or the resistance between these electrodes changes as the humidity varies. The sensor work with 5V DC power supply and have operating range for humidity 0-100% RH and temperature from -40 to 80 Celsius. Sensing period approximately with average 2second. Resolution of sensitivity for humidity is at 0.1% RH and for temperature is at 0.1 Celsius.
2) NodeMCU ESP8266: This controller is a low-cost open-source development board that runs on the ESP8266 Wi-Fi microchip. This board (cf. Figure 3) consists of 16 General Purpose Input/Output (GPIO) pins, Pulse Width Modulator (PWM), serial 2-wire bus interface (Inter-Integrated Circuit, IIC), 1-Wire interface and Analog to Digital Convertor (ADC). The flash memory is 4MB and has internal clock of 80MHz. The RAM size is 50K and it integrates and on chip Wi-Fi Transceiver. NodeMCU operates on an external DC voltage supply of 6 to 24 volts, but it is advisable to use 12 volts maximum to prevent overheating. NodeMCU is ideal for building IoT projects as it consists of all the necessary components as described above, and can be programmed via Arduino IDE. Several research works presented in [12]-[16] also use the same microcontroller which is efficient because it integrates Wi-Fi module on the board

(compared to using Arduino Uno for example, which needs separate Wi-Fi module). Figure 4 depicts the wiring diagram of the sensor DHT11 with the NodeMCU controller.

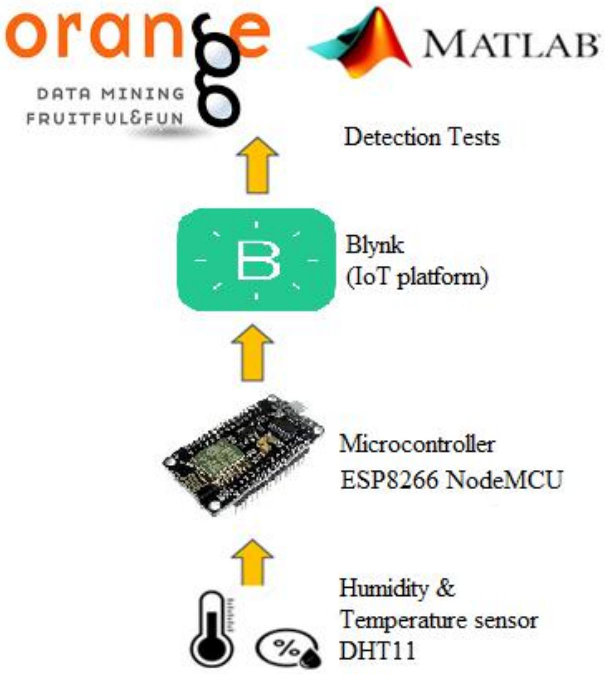

Figure 1: The proposed framework diagram

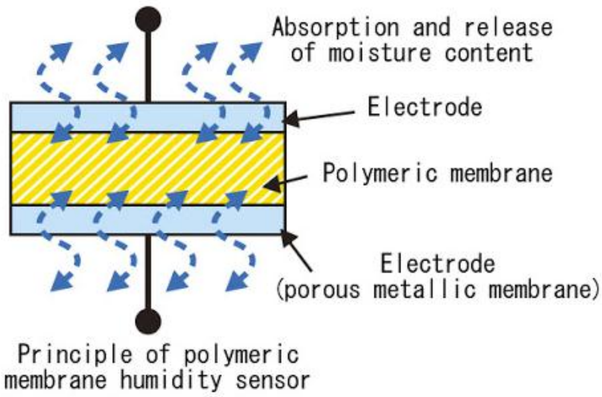

Figure 2: DHT11 working principle

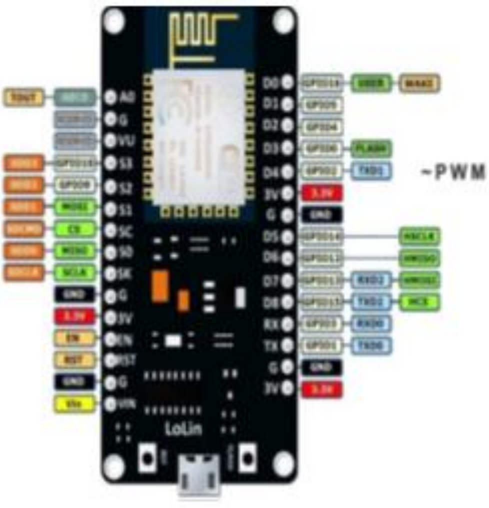

Figure 3: NodeMCU ESP8266 pin assignment

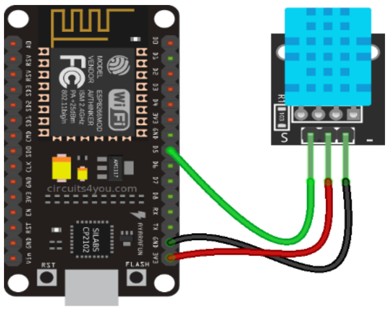

Figure 4: Wiring Connection of DHT11 and ESP8266

C. Statistical Detection Test via Unsupervised Subspace Learning

1) *Hypothesis detection test*: The signal detection problem involves here decide whether a signal is absent or is present in a noisy data set. For this work, the scenario is to detect whether the room is empty, or the room is occupied. A binary hypothesis model can be written as:

$$\begin{cases} H_0: Room\ empty \\ H_1: Room\ occupied + noise \end{cases} \quad (1)$$

Considering a data set $\boldsymbol{S} \in \mathbb{R}^{N\times L}$, the hypothesis model:

$$\begin{cases} H_0: \boldsymbol{x} = \boldsymbol{n}, \quad \boldsymbol{n} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}) \\ H_1: \boldsymbol{x} = \boldsymbol{S\alpha} + \boldsymbol{n}, \qquad \|\boldsymbol{\alpha}\|_0 = 1 \end{cases} \quad (2)$$

The detection test based on Generalized Likelihood Ratio:

$$T(\boldsymbol{x}, \boldsymbol{S}) = \max_{\boldsymbol{\alpha}:\|\boldsymbol{\alpha}\|_0=1} \frac{p(\boldsymbol{x}; \boldsymbol{S\alpha})}{p(\boldsymbol{x}; \mathbf{0})} \gtrless_{\mathcal{H}_0}^{\mathcal{H}_1} \gamma' \quad (3)$$

For subspace learning, the binary hypothesis model of (2) reduces to:

$$\begin{cases} H_0: \ \boldsymbol{x} = \boldsymbol{n}, \quad \boldsymbol{n} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}) \\ H_1: \boldsymbol{x} = \boldsymbol{D\beta} + \boldsymbol{n}, \quad \| \boldsymbol{\beta} \|_0 = 1 \end{cases} \quad (4)$$

where $\boldsymbol{D} \in \mathbb{R}^{N\times K}$ is the trained or learned subspace of lower dimension than $S \in \mathbb{R}^{N\times L} (K < L)$. The associated detection test becomes [11]:

$$T(\boldsymbol{x}, \boldsymbol{D}) = \max_{\boldsymbol{\beta}:\|\mathbb{B}\|_0=1} \frac{p(\boldsymbol{x}; \boldsymbol{D\beta})}{p(\boldsymbol{x}; \boldsymbol{0})} \gtrless_{\mathcal{H}_0}^{\mathcal{H}_1} \xi' \quad (5)$$

For the experiment, in each run, the ratio of train data to test data is 70%:30%. The experiments are conducted in 100 runs, where for each run, the train data and test data are permuted.

2) *Minimax subspace learning*: Minimax method seeks to maximize the worst Probability of Detection, or equivalently minimize the maximum Probability of False Alarm. From previous work presented in [18], it was shown that minimax subspace learning for one-dimension can be obtained by solving the following:

$$\boldsymbol{d}^* = \arg\max_{d:\|d\|_2=1} \min_{i=1,\dots,L} (\boldsymbol{d}^\top s_i)^2 = \arg\max_{d:\|d\|_2=1} \min_{i=1,\dots,L} |\boldsymbol{d}^\top s_i| \quad (6)$$

where d is the trained or learned subspace obtained using Minimax and SVD method.

3) *SVD subspace learning:* Let's consider a data set S. The subspace of $\boldsymbol{S}$ (lower dimension, say $\widehat{\boldsymbol{S}} \in \mathbb{R}^{N\times L}$ can be approximated, or can be learned based on The Eckart-Young Theorem [9]:

$$\min_{\widehat{\boldsymbol{S}}} \left\|S - \widehat{\boldsymbol{S}}\right\|_F^2 \quad Subject\ to\ rank(\widehat{\boldsymbol{S}}) < rank(\boldsymbol{S}) \quad (7)$$

where $\|.\|_F$ denotes the Frobenius norm. This problem can be solved through SVD (of rank $\widehat{\boldsymbol{S}}$, say K). The approximate matrix is:

$$\widehat{\boldsymbol{S}} = \mathbf{U}\boldsymbol{\Sigma}_{\mathrm{K}}\mathbf{V}^\top \quad (8)$$

where $\boldsymbol{\Sigma}_K \in \mathbb{R}^{N\times L}$ is a diagonal matrix containing the *K* largest singular value of $\boldsymbol{S}$, $\boldsymbol{U} \in \mathbb{R}^{N\times N}$ and $\boldsymbol{V} \in \mathbb{R}^{L\times L}$ are respectively left-singular and right-singular matrices. $\boldsymbol{\Sigma}_{\mathrm{K}}\mathbf{V}^\top$ is a sparse matrix by rows, and $\mathbf{U}_{\mathrm{K}}$ represents $\boldsymbol{S}$ in lower dimension.

D. *Detection Test via Supervised Learning*

1) *SVM:* Support Vector Machine (SVM) is an algorithm to find a hyperplane in an N-dimensional space where the data points are classified. SVM finds the optimized hyper plane that separates training samples of different classes as maximum as possible [7]. Based on Figure 5,

$$g(x) = w^T x + b \quad (9)$$

Maximize k such that:

$$\begin{aligned} -w^T x + b &\geq k \text{ for } d_i = 1 \\ -w^T x + b &\leq k \text{ for } d_i = -1 \end{aligned} \quad (10)$$

Value of *g(x)* depends upon $\|w\|$:

(a) keep $\|w\| = 1$ and maximize *g(x)* or,
(b) *g(x)* ≥ 1 and minimize $\|w\|$.

2) Decision Tree: As the name indicated, Decision Tree method has several branches. It consists of decision nodes as the parent and terminal nodes as the child. Each branch represents the outcome of the test, and each leaf node represents a class label after computation of all parameters set, as shown in Figure 6.

Let's denote $\widehat{\pi}_{mc}$ the fraction of training data in Region R_m that belongs to class c. In growing a decision tree, one of these functions need to be minimized:

(a) Classification Error Rate

$$E = 1 - arg \max_{c} \widehat{\pi}_{mc} \tag{11}$$

(b) Gini Index (Purity) / Cross-Entropy (Deviance)

$$G = \sum_{c=1}^{C} \widehat{\pi}_{mc} \ (1 - \widehat{\pi}_{mc}) \tag{12}$$

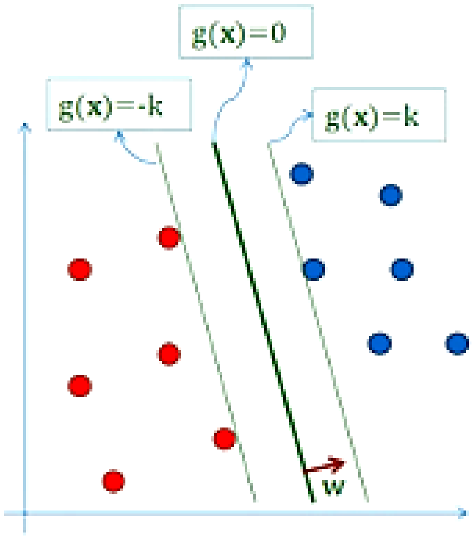

Figure 5: SVM Hyperplane

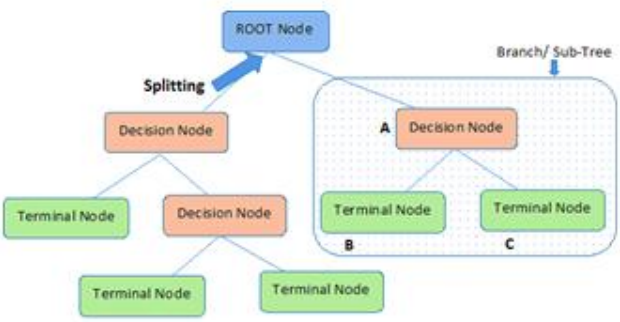

Figure 6: The concept of Decision Tree method

3. RESULT AND DISCUSSION

A. Monitoring System and Database

Blynk application allowed user to store data through Superchart Widgets. The function of Superchart is to display output value in graph display. User need to connect the Blynk application with email address and manually saved the data. Data saved will automatically transfer to email address which user connect to. The data is stored in CSV format by default. The CSV data which received is based on virtual pin user selected. In this project, Humidity data is assigned to V5 pin while temperature is in V6 pin. From the different CSV file, the dataset is combine into one Excel file to use as dataset for Matlab and Orange analysis. Figure 7 shows the example of monitoring interface designed using Blynk for this work.

The data from humidity and temperature sensor are collected with 100 measurements for two room conditions: occupied and unoccupied. Figure 8 shows the scatter graph of humidity vs temperature value with two type of room conditions. The graph was generated according to the values that are collected from DHT11 sensor. The data were recorded at different time at different external weather. The room temperature is set to 25 degree Celsius using air conditioner controller, set as the default temperature of the room. Blue dots represent occupied room condition and orange dots represent unoccupied room condition.

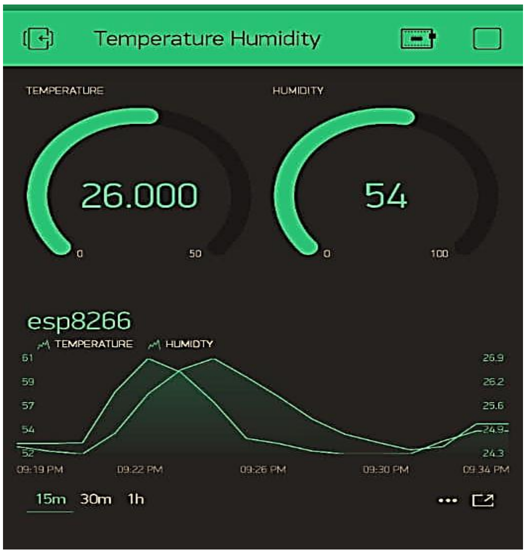

Figure 7: Blynk monitoring page

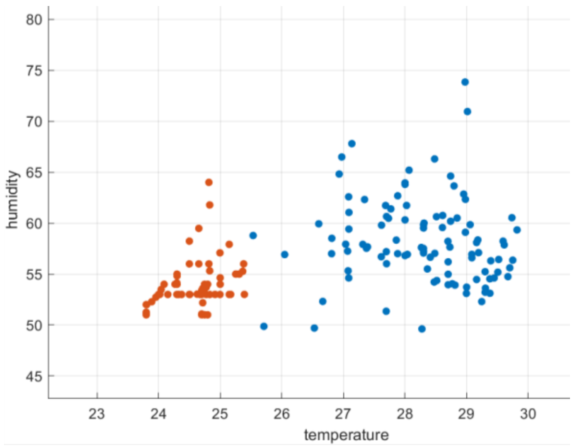

Figure 8: Scatter plot of the collected dataset

B. Detection Test Performance via Unsupervised Subspace Learning in Worst Case Criterion

The result of the detection testing for subspace learning is shown in Figure 9. Blue line represents Minimax while red dotted-line represents (K)-SVD (where K=1, thus equivalent to SVD). Y-Axis represent the minimum probability detection Pdet of the data result. X-Axis represent runs with maximum 100 runs which we set on in this experiment.

Based on this graph in Figure 9, Minimax show a better performance in worst case scenario. The minimum Probability of detection (Pdet) is higher in overall runs compared to

SVD. The obvious comparison can be seen at run 21 which SVD minimum Pdet drops significantly than the Minimax. The other obvious Pdet minimum for SVD are at runs 31, 40 and 65. For 100 number of runs, minimax shows better performance than SVD: in some several worst cases, the minimum Pdet for SVD is lower than the minimum Pdet for minimax. This indicate that SVD might loses some of the detection signal to detect an occupied room.

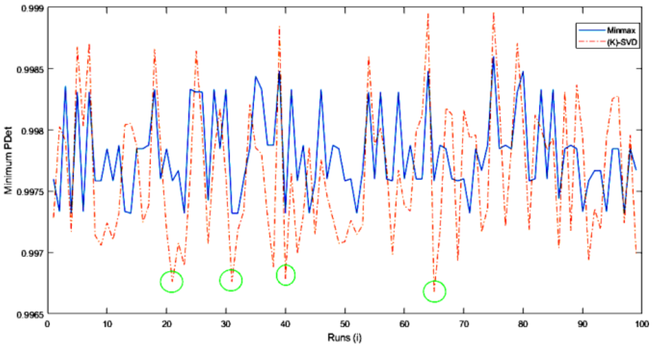

Figure 9: Probability of detection (Pdet) of SVD versus Minimax. Blue line represents Minimax while red dotted-line represents (K)-SVD. The minimum Pdet for Minimax method is higher in overall runs compared to SVD. The obvious comparison can be seen at runs 21, 31, 40 and 65 where the minimum Pdet of SVD minimum Pdet drops significantly than the Minimax.

C. Detection Test Performance via Supervised Learning

Classification learner with 5 fold cross validation is studied using two methods which are Support Vector Machine (SVM) and Decision Tree. Using room condition as response while humidity and temperature value as features. H1 is classified as room occupied and H0 room unoccupied. 100 data value of each room condition is selected randomly. 70% of data is used as training data and 30% used as test data on Matlab classification learner application and compared using Orange.

Figure 10 and Figure 11 depict the result of this experiment. Using SVM gives the highest area under curve of the ROC (Receiver Operating Characteristic) of the true positive rate (TPR) and false positive rate (FPR) which can be considered as 1.0 followed by Decision Tree with 0.98. SVM and Tree almost yield the same percentage but the curve line between these 2 classes give a different result.

For Tree method (Figure 10), it starts at TPR 0.98 and FPR 0.03. Then, achieve TPR 1.0 at estimation FPR 0.06. For SVM method (Figure 11), it starts at TPR 0.99 at FPR 0. Then achieve TPR 1.0 at estimation FPR 0.3. This can be concluded both classes have a pattern of performance result even though the coverage is the same and SVM method give more better accuracy than Tree using this dataset with humidity and temperature level features.

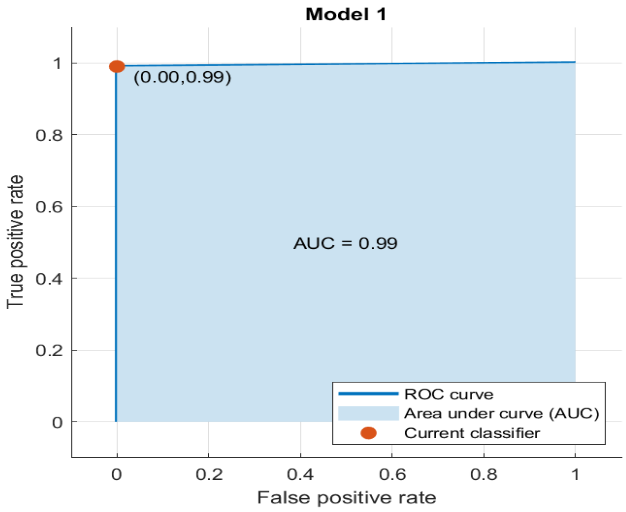

Figure 10: Decision Tree ROC Graph

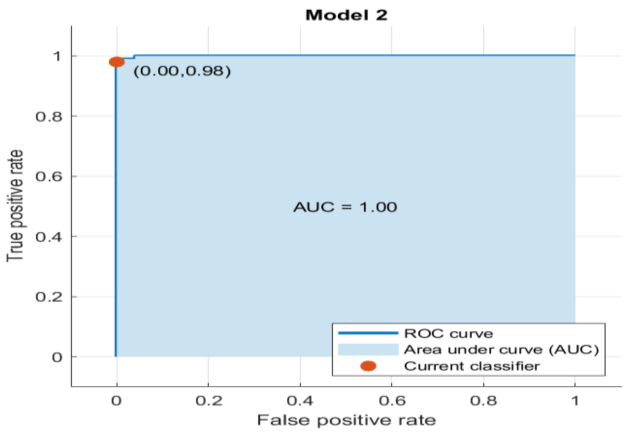

Figure 11: SVM ROC graph

4. CONCLUSION AND FUTURE WORK

The detection of occupancy in a room using unsupervised subspace leaning model in worst criterion, with IoT technology was proposed in this research work. Comparison with supervised learning model was also executed to learn the trends. With further implementation of this research work, energy can be saved thus increases green-index of buildings.

The main objective of this work is to analyze several learning methods performance applied on data collected from temperature and humidity sensors for room occupancy detection using IoT technology. Results shown that for supervised learning, SVM performs slightly better than Decision Tree. While for unsupervised sunspace learning, minimax yields better probability of detection than SVD in worst case criterion. The implementation of statistical learning model has shown a good detection performance for room occupancy detection.

For future works, other suitable sensor such as light intensities sensor and ultrasonic sensor can be added to the system. Using more parameters as the features might result to a better occupancy detection performance.

REFERENCES

[1] Z Sun, S Wang and Z Ma. In-situ implementation and validation of a CO_2-based adaptive demand-controlled ventilation strategy in a multi-zone office building. *Building and Environment.* 2011; **46**, 124-133.

[2] LM 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*. 2016; **112**, 28-39.
[3] TH Pedersen, KU Nielsen and S Petersen. Method for room occupancy detection based on trajectory of indoor climate sensor data. *Building and Environment.* 2017; **115**, 147-156.
[4] T Labeodan, W Zeiler, G Boxem and Y Zhao. Occupancy measurement in commercial office buildings for demand-driven control applications—A survey and detection system evaluation. *Energy and Buildings.* 2015; **93**, 303-314.
[5] Z Han, RX Gao and Z Fan. Occupancy and indoor environment quality sensing for smart buildings. *In*: 2012 IEEE International Instrumentation and Measurement Technology Conference Proceedings, Graz, Austria. 2012, p. 882-887.
[6] T Poggio, L Rosasco, et al. Statistical Learning Theory and Applications, 2012.
[7] E Hailemariam, R Goldstein, R Attar and A Khan. Real-time occupancy detection using decision trees with multiple sensor types. *In*: Proceedings of the 2011 Symposium on Simulation for Architecture and Urban Design, Society for Computer Simulation International, San Diego, CA. 2011, p. 141-148.
[8] BE Boser, IM Guyon and VN Vapnik. A training algorithm for optimal margin classifiers. *In*: Proceedings of the fifth annual workshop on Computational learning theory (COLT '92). Association for Computing Machinery, New York, NY. 1992, p. 144-152.
[9] GH Golub, A Hoffman and GW Stewart. A generalization of the Eckart-Young-Mirsky matrix approximation theorem. *Linear Algebra and its applications*. 1987; **88**, 317-327.
[10] S Lloyd. Least squares quantization in PCM. *IEEE transactions on information theory*. 1982; **28**, 129-137.
[11] RFR Suleiman, D Mary and A Ferrari. Dimension Reduction for Hypothesis Testing in Worst-Case Scenarios. *IEEE Transactions on Signal Processing*. 2014; **62**, 5973-5986.
[12] A Škraba, A Koložvari, D Kofjač, R Stojanović, V Stanovov and E Semenkin. Prototype of group heart rate monitoring with NODEMCU ESP8266. *In*: 2017 6th Mediterranean Conference on Embedded Computing (MECO), Bar, Montenegro. 2017, p. 1-7.
[13] DA Aziz. Webserver Based Smart Monitoring System Using ESP8266 Node MCU Module. *International Journal Of Scientific & Engineering Research*. 2018; **9**, 801-807.
[14] S Shakthidhar, P Srikrishnan, S Santhosh and MK Sandhya. Arduino and NodeMcu based Ingenious Household Objects Monitoring and Control Environment. *In*: 2019 Fifth International Conference on Science Technology Engineering and Mathematics (ICONSTEM). 2019, p. 119-124.
[15] H Ouldzira et al. Remote monitoring of an object using a wireless sensor network based on NODEMCU ESP8266. *Indonesian Journal of Electrical Engineering and Computer Science*. 2019; **16**, 1154-1162.
[16] L Shkurti et al. Development of ambient environmental monitoring system through wireless sensor network (WSN) using NodeMCU and "WSN monitoring". *In*: 2017 6th Mediterranean Conference on Embedded Computing (MECO). 2017, p. 1-5.
[17] R Gupta, and KK Singh. Blynk-based Water Level Monitoring System. *Current Trends in Signal Processing*. 2018; **8**, 1-3.
[18] R. F. R. Suleiman, D. Mary and A. Ferrari. Minimax sparse detection based on one-class classifiers. *In*: 2013 IEEE International Conference on Acoustics, Speech and Signal Processing. 2013, p. 5553-5557.