A MACHINE LEARNING APPROACH ON OCCUPANT NUMBER PREDICTION FOR INDOOR SPACES

Umit Isikdag*[1], Kemal Sahin[1], Sergen Cansiz[1]

1. Mimar Sinan Fine Arts University, Istanbul, Turkey (umit.isikdag,kemal.sahin)@msgsu.edu.tr,sergencansiz@gmail.com

Commission IV, WG IV/6

KEY WORDS: Indoor, Machine Learning, Occupancy, IoT, Sensors

ABSTRACT:

The knowledge about the occupancy of an indoor space can serve to various domains ranging from emergency response to energy efficiency in buildings. The literature in the field presents various methods for occupancy detection. Data gathered for occupancy detection, can also be used to predict the number of occupants at a certain indoor space and time. The aim of this research was to determine the number of occupants in an indoor space, through the utilisation of information acquired from a set of sensors and machine learning techniques. The sensor types used in this research was a sound level sensor, temperature/humidity level sensor and an air quality level sensor. Based on data acquired from these sensors six automatic classification techniques are employed and tested with the aim of automatically detecting the number of occupants in an indoor space by making use of multi-sensor information. The results of the tests demonstrated that machine learning techniques can be used as a tool for prediction of number of occupants in an indoor space.

1. INTRODUCTION

Occupancy detection and occupant number prediction have a critical importance in order to increase the automation capability of buildings and to strengthen the decision-making capabilities of emergency responders and facility managers. The knowledge about the occupancy of an indoor space can serve to various domains ranging from emergency response to energy efficiency in buildings. The literature in the field presents various methods for occupancy detection. The data gathered from the indoor space for occupancy detection can later be used to predict the number of occupants at a certain indoor space and at a certain time. Automatic prediction of the number of occupants of an indoor space is an active research problem, and today Machine Learning emerges as a key tool to deal with it.

Machine Learning (ML) can be defined as a set of approaches focused on deriving meaningful information from data, based on human guidance or autonomously. There are 4 main categories in ML as Supervised Learning, Unsupervised Learning, Semi-supervised Learning and Reinforcement Learning. The occupant number prediction problem can be dealt with using Supervised Learning approach. In the Supervised Learning approach, a human provides a machine with the training data containing the independent/predictor variables and the correct values of the dependent variable which needs to be predicted later by the machine. Based on this correct value of the dependent variable the machine -learns- the pattern of the data (i.e. the impact of each independent/predictor variable to the value of the dependent variable) and makes predictions for estimation of values of the dependent variable. Because of the training process, the impact (or role) of each variable in the determination of the dependent variable is estimated. The mathematical formalisation of this estimation is known as the "ML Model, or in short "Model". These machine-made predictions (i.e. predicted values of the dependent variable) are then compared with the correct values of the dependent variable by using a test dataset to evaluate the success of the prediction (known as the accuracy of the ML Model).

The aim of this research was to determine the occupancy status and several occupants in an indoor space, through the utilisation of a set of sensors and machine learning techniques. The sensor types used in this research was a sound level sensor, temperature/humidity level sensor and an air quality level detection sensor. Based on data acquired from these sensors automatic classification techniques are employed to detect the number of occupants in an indoor space. Following the background on the subject, data acquisition and machine learning processes in the study are elaborated in the paper.

2. BACKGROUND

The recent studies in the field of occupancy detection and occupant number prediction have mainly concentrated on determining the occupant numbers for energy consumption detection. or energy requirement prediction (e.g. Hailemariam, et al.,2011 and Szczurek et al., 2017). Similarly, Mahdavi (2009), Dobbs and Hencey (2014) and Oldewurtel et al. (2013) have also extensively highlighted the importance of occupancy number as well as their behaviour in building energy consumption, building performance simulation and building control. Studies such as Erickson et al. (2014) and Dong and Andrews (2009) have shown that around one-third of the energy consumed in the buildings can be saved using occupancy-based control. On the other hand, information related to occupancy can also be used for other domains. For instance, Mahdavi (2011) stated the importance of occupant behaviour modelling in the field of occupancy based controls. Yang and Becerik-Gerber (2014) revealed that increased awareness of the indoor environment quality associated health and productivity issue in buildings have been a must-have feature of buildings to consider the indoor environment quality while focusing on energy saving. Occupants, generate heat as well as CO2 into the indoor environment, and they also move indoors, and these factors will influence the indoor environment causing such as

changes at the levels of CO2, door/window status, light levels, and temperature.

Current literature indicated six methods for automatic detection of occupancy using input from electronic systems. The first one utilised the behaviour of lighting systems (Nguyen and Aiello, 2013). The second one is the utilisation of behaviour of Ventilation and Air-conditioning (HVAC) systems (Oldewurtel et al., 2013). The third one is making use of radio frequency (RF) signals and is developed on the basis of electromagnetic signal detections (Domdouzis et al., 2007). Fourth method is occupancy detection based on the information from infrared, ultrasound, or video cameras (Gu et al., 2009). The fifth method included a combination of global positioning system (GPS), cellular data, wireless local area network (WLAN) (Liu et al., 2007). Bluetooth technology was also applied in the occupancy detection (Hallberg et al., 2003). The last method contains the use of single or multiple sensors for occupancy detection. The studies in this field showed that the data obtained within this method produces more consistent results (Wang et al., 1999 and Mumma, 2004). For example, in a recent study, Hailemariam et al (2011) reported that inclusion of sensors improved overall occupancy detection accuracy. In addition to occupancy detection, the data gathered from multiple sensors can also be used for prediction of occupant numbers (Candanedo and Feldheim, 2016). Based on the evidence from the literature, the research explained in this paper concentrated on the utilisation of multi-sensor information in order to predict number of occupants at a certain moment in a (closed) indoor space using machine learning.

3. DATA ACQUISITION STRATEGY

The overall research was completed in two stages. The first stage was acquisition of data from an indoor space using multiple sensors. The second stage was the machine learning process with the aim of prediction of number of occupants. The data acquisition stage was an important part of the study as inconsistent or missing data would cause inefficiencies and inaccuracies in the machine learning models. For this reason, much attention has been paid in the data collection/acquisition process as possible. The Internet of Things (IoT) technologies provide reliable means for environmental sensing and data collection from an indoor environment. By using single board computers, sensors and interconnected devices, data can be collected, processed and stored easily and securely. The first phase of the data acquisition strategy was the selection of hardware for data collection. In the study hardware to be utilised was determined as a single-board computer. The reason behind this choice was the popularity of single board computers in today’s R&D projects related to collection of environmental information. Second factor of this choice was their low-price range. The third factor was the existence of lots of documentation regarding the programming of and interaction with these devices. In summary, the hardware choice was to use a single-board computer and sensors attached to them. Single board computers contain microcontrollers or relatively cheap microprocessors that are able to control sensor and actuators, while most of the single-board computer hardware can run specific version of operating systems. As explained in Gajski (1997) a microcontroller is a single integrated circuit, and have some important features. Choosing the correct and the most suitable microcontroller and a single board computer from several different microcontrollers and single board computers was a very critical decision for the research. In this study, commonly used microcontrollers and single board computers were compared and analysed in light of eleven different factors. These factors were system requirements, memory architecture, availability, size, compatibility, power management, manufacturer’s track record, manufacturer’s support, availability for development support. As this evaluation on hardware selection is out of this paper’s scope, this process will not be elaborated more here. As a result of the evaluation process, Arduino UNO which uses Atmel microprocessor was selected as the single board computer for data collection (Figure 1).

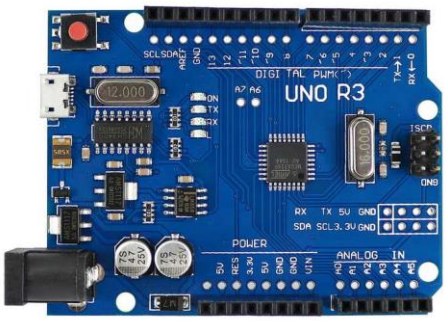

Figure 1 Arduino UNO R3

The environmental variables to be acquired were the temperature/humidity, air quality and the sound level. The data is acquired from computer lab at certain intervals by using DHT11, MQ 135 and FC-04 sensors respectively (Figure 2). DHT 11 was used to acquire a signal value regarding temperature/humidity in the lab, MQ135 were used to acquire a signal value on the air quality and FC-04 was used to acquire a signal value regarding sound levels. MQ 135 sensor used in the study could detect the levels of carbon oxides such as CO2 and CO, and the acquired information is used to take CO2 levels into account in occupant number prediction.

Figure 2 DHT11, MQ-135, FC-04 Sensors

During the collection of the data, the processor of Arduino UNO is used for data processing and for formatting the data in a structured form that can be easily stored in a database. UNO together with WIZNET W5100 Arduino Ethernet shield, and the code prepared for the Atmel microcontroller of UNO, has been utilized for publishing the data in form of an extensible Mark-up Language (XML) document from the local IP address of UNO. The schema of the XML data served by the UNO, was designed by the researchers prior to data collection. Another software component namely Data Acquisition and Update Component (DAUC) module was developed and used to acquire XML data served by the Arduino UNO, and later to populate a MySQL data store with this data. The DAUC module (coded in PHP) is hosted on a virtual machine (VM) running a Linux based OS, and NGINX is used as the web server to run the module. A CRON job is created in the Linux VM to call the DUAC module every 30 seconds, which would catch “a snapshot” of the real-time values acquired/published by the

UNO and update the database with those values along with a timestamp. The overall data acquisition system is illustrated in Figure 3.

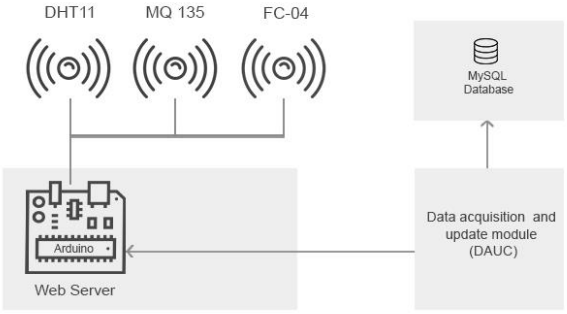

Figure 3 Data Acquisition System

The location of the sensors and seating order within the computer lab is illustrated in Figure 4. Sensors hardware has been located in the middle of the computer laboratory and in front of the student seating rows. Number of occupants is recorded manually prior to the start of the lecture and entrance to and exit from the lab is prohibited during the experiment hours. Occupant’s counts is a required parameter for the supervised machine learning algorithms, as they require at least one response variable to train and test the system. In the implementation of the supervised machine learning algorithms, the real number of occupants in the room during the environmental data collection process has been used for validating the prediction results (i.e. predicted number of occupants).

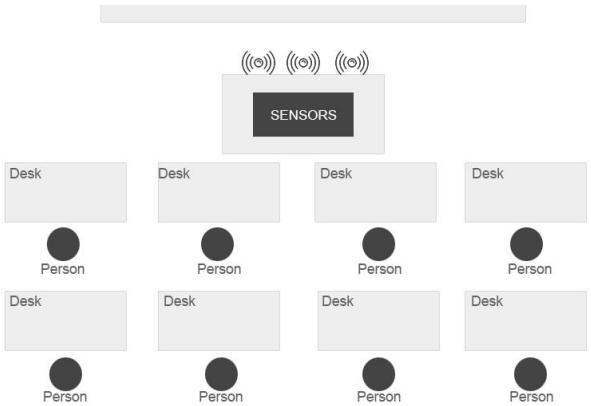

Figure 4 Sensor’s location and the Computer Lab’s seating order

The data collected at the end of the experiment, and used to train/test the system, was consisted of a total of 1974 records and stored in form of a database table in a MySQL database. The fields stored in the table were Id, Date and signal levels coming from three sensors. An extract from the data table is provided in Figure 5.

Row No.	ID	Person	Date	Mic	mQ135	humidity
1	2247	5	2017-11-09 13:11:58	825	356	41
2	2248	5	2017-11-09 13:12:32	825	350	41
3	2249	5	2017-11-09 13:13:05	820	342	40
4	2250	5	2017-11-09 13:13:37	819	335	40
5	2251	5	2017-11-09 13:14:10	826	333	40
6	2252	5	2017-11-09 13:14:42	826	329	40
7	2253	5	2017-11-09 13:15:15	828	325	40
8	2254	5	2017-11-09 13:15:45	825	319	40
9	2255	5	2017-11-09 13:16:18	827	314	40
10	2256	5	2017-11-09 13:16:50	828	311	39
11	2257	5	2017-11-09 13:17:22	827	308	39
12	2258	9	2017-11-09 13:17:49	825	306	39
13	2259	9	2017-11-09 13:18:21	827	303	39
14	2260	9	2017-11-09 13:18:54	831	301	38
15	2261	9	2017-11-09 13:19:26	826	298	38
16	2262	9	2017-11-09 13:19:59	827	296	37
17	2263	9	2017-11-09 13:20:31	830	294	36

Figure 5 Overview of the data

The following section elaborates on the learning process.

4. THE MACHINE LEARNING PROCESS

In the ML process of the study, randomly selected training sets from the data and predictor variables in them such as sound level, humidity, CO2 level, are used to train a ML Model to predict the dependent variable i.e. number of occupants in a computer lab. First, a training subset is randomly selected from the dataset for training the machine, and the remaining part of the dataset is considered as the test set. Once the training of the machine is completed using the training set and a machine learning model is defined, then the test set is fed into the trained ML Model, and the trained Model is asked to make predictions of the dependent variable using the test data. Following this, the predictions are done by the machine using the test data, and these predictions are then used for evaluating the success of the ML model by comparing the predicted dependent variables(i.e. predicted number of occupants) with the correct values of the dependent variables(i.e. real number of occupants) in the test set . This process has been carried out various times by selecting different training and test subsets from the dataset to determine the success rate of the predictions accurately. In machine learning, this training-testing loop is known as cross-validation. The test results are then provided in form of a table which is known as the Contingency Matrix. The Contingency Matrix is a tool that depicts the success rate of predictions in a clear form.

In Machine Learning several ML techniques have to be taken into account and tested in order discover the fittest (most accurate) ML Model and ML technique, as various ML techniques will output different models (Moraru, 2010) and the success rate of predictions would change from one technique to another. Thus, in this study we implemented six different learning techniques as Naïve Bayes, Generalized Linear Model (GLM), Decision Tree, Random Forrest, Gradient Boosted Trees (GBT) and Deep learning, to evaluate the success of prediction in a more comprehensive manner.

The academic version of an off-the-shelf machine learning tool “rapidminer Studio” was used, to implement the six classification techniques mentioned here. The rapidminer Studio has an analysis option called “Auto Model” which was utilised by the researchers to implement and test the techniques. As the recorded number of occupants of the lab were in the narrow range of 0-10 (i.e. in form of integer values including 0 and 10), the problem is handled by the authors as a classification problem. 11 classes were identified for training the model as provided in Table 1.

Class Name	Lower bound	Upper bound	The class represents existence of … occupants in the room
R0	0	0.909	0
R1	0.909	1.818	1
R2	1.818	2.727	2
R3	2.727	3.636	3
R4	3.636	4.545	4
R5	4.545	5.455	5
R6	5.455	6.364	6
R7	6.364	7.273	7
R8	7.273	8.182	8
R9	8.182	9.091	9
R10	9.091	11	10

Table 1 Class Definitions

Every algorithm mentioned above is implemented in radipminer Studio, and the training results of the six different ML algorithms were evaluated using 5-fold cross-validation. The following will elaborate on the implemented ML techniques/algorithms and on the performance of each of them.

4.1 Naïve Bayes

The first algorithm tested in the study was Naïve Bayes (NB). Naïve Bayes is one of the efficient learning algorithms in machine learning. The algorithm stems from the Bayes theory which focuses on conditional probability. Basically, Bayes theorem depends on conditional and marginal probabilities of two random states. $x = (x_1, x_2, x_3 \dots x_k)$ is unclassified d-dimensional sample and $C = (C_1, C_2, C_3 \dots C_k)$ is the classes. $P(x|C_k)$ represents the probability of obtaining x when hypothesis C_k is true (Ren et al., 2009).The equation for obtaining $P(x|C_k)$ is

$$P(x|C_k) = \frac{P(x|C_k)P(C_k)}{\sum_{k'} P(x|C_{k'})P(C_{k'})}$$

The fundamental assumption of Naive Bayes is that, given the value of dependent variable, i.)the value of any independent variable is independent of the value of any other independent variable(rapidminer Studio Documentation, 2018a) and ii.) all the independent variables independently contribute to the probability that dependent variable would have that given value. For example, a fruit may be considered to be an apple (i.e. dependent var.) if it is red, round, and about 3 inches in diameter (independent vars.). Even if these features depend on each other or upon the existence of the other features, all of these properties (independent vars.) independently contribute to the probability that this fruit is an apple and that is why the algorithm is known as 'Naive'. (Analyticsvidhya, 2017). The NB algorithm can work properly with the data sets which have a few variables in order to classify unknown data. Recently, Rahman and Han (2017) focused on estimating number of occupants in a room according to CO2 concertation in the room by using neural network and Bayesian MCMC methods. That study compared the two machine learning models in order designate an optimal significant result. According to their findings the accuracy of neural network is influenced by the complexity of input (CO2 concertation), but Bayesian MCMC is not influenced. This result shows that Bayesian techniques such as Naïve Bayes classification can be optimal for determining the occupant numbers. Table 2 presents the Contingency Matrix for NB model which would be useful for evaluating the ML model that is trained and tested. The rows of the matrix (P0..P10) shows the class predictions for each occupancy state (e.g. such as 0 person, 1 person,….9 person,10 person) , while the columns of the matrix(R0…R10) shows the number of real occupancy classes of the test data. For example the value of Cell(P0,R0) = 171 indicates that 171 predictions made by the machine as -the room being empty- for the given humidity,CO2, and sound level values, are actually correct. On the other hand, for instance the value of Cell (P3,R0)=44 indicates that the machine predicted 44 times that there should be 3 persons in the room for the given humidity, CO2, and sound level values, but in reality there was nobody (i.e. 0 person) in the room. In summary the diagonal of the matrix shows the number of correct classifications, while all other values present false classifications.

	R0	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	%
P0	171	0	0	0	0	0	0	0	0	12	0	93.44
P1	0	0	0	0	0	0	0	0	0	0	0	0.00
P2	0	0	0	0	0	0	0	0	0	0	0	0.00
P3	44	0	0	246	0	0	0	10	0	0	0	82
P4	0	0	0	0	0	1	0	0	0	0	0	0.0
P5	16	0	2	0	0	220	0	0	0	56	0	74.83
P6	0	0	0	0	0	0	119	0	0	0	0	100
P7	13	0	0	1	0	0	0	56	0	0	0	80
P8	0	0	0	0	0	0	0	0	0	0	0	0.00
P9	8	0	0	0	0	122	1	0	0	320	0	70.95
P10	0	0	0	0	2	2	0	0	1	0	158	96.93
%	67.86	0	0	99.60		63.77	99.17	84.85	0.00	82.47	100	

Table 2 Contingency Matrix for NB

The ML model generated using Naïve Bayes algorithm has been successful in predicting the number of occupants with 81.59% +/-1.99% success rate. The second algorithm tested was the linear model.

4.2 Generalized Linear Model (GLM)

Generalized linear model (GLM) is a mature technique for modeling data which has large amount of variables and observations. This technique depends on associations/correlations between predictor and the response variables. The model parameters represent the strength of the associations (Cantoni and Ronchetti, 2011). Generalized Linear Model consists of linear predictor and other two functions;

- Linear predictor with $\hat{y}_i, i = (1, \dots, p)$ as response observations and $x_j, j = (1, \dots, p)$ as explanatory variables(i.e. the well-known regression equation);

$$\hat{y}_i = \beta_0 + \beta_1 x_{1i} + \cdots + \beta_p x_{pi} + \epsilon_i$$

- Link function which represents the dependency of mean to linear predictor;

$$g(\mu_i) = y_i$$

- The variance function which represents the dependency of variance to mean;

$$Var(Y_i) = \phi V(\mu)$$

The GLM is also a supervised learning method similar to NB. In order to estimate number of occupants in a room with this technique sound level, humidity, CO2 levels are used as predictor variables and number of occupants is used as the response variable. Table 3 presents the Contingency Matrix which would be useful for evaluating the ML model that is trained and tested using GLM method.

	R0	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	%
P0	189	0	0	0	0	0	39	26	0	12	0	74.41
P1	0	0	0	0	0	0	0	0	0	0	0	0.00
P2	0	0	0	0	0	0	0	0	0	0	0	0.00
P3	55	0	0	247	0	0	0	40	0	0	0	72.22
P4	0	0	0	0	0	1	0	0	0	0	0	0.0
P5	8	0	1	0	0	326	0	0	0	56	0	83.38
P6	0	0	0	0	0	0	81	0	0	17	0	82.65
P7	0	0	0	0	0	0	0	0	0	0	0	0
P8	0	0	0	0	0	0	0	0	0	0	0	0
P9	0	0	1	0	0	13	1	0	0	315	0	95.74
P10	0	0	0	0	2	0	0	0	1	0	158	94.61
%	75.00	0	0	99.60	0	94.49	94.49	0	0.00	82.47	100	

Table 3 Contingency Matrix for GLM

The ML model generated using GLM algorithm has been successful in predicting the number of occupants with 83.24% +/- 2.27% success rate, which showed a better performance compared to NB. The third algorithm tested was Deep Learning.

4.3 Deep Learning

Deep learning is frequently used for large-scale and complex data sets such as image and speech recognition. The technique can also be a good option for occupancy detection as of sensor data might include many observations depending on the information acquisition procedures. Deep Learning (DL) algorithms have representation-learning layer structure and run on multiple representation layers. Representation-learning is the method which allows a machine fed by raw data to automatically discover the new classes in it. Thanks to this structure it is more convenient than other machine learning methods while working on complex raw data (LeCun et al., 2015). The rapidminer Studio executes the DL algorithm using H2O ML package. The implemented algorithm in this research is based on a multi-layer feed-forward artificial neural network that is trained with stochastic gradient descent using back-propagation. The network can contain a large number of hidden layers consisting of neurons with tanh, rectifier and maxout activation functions. (rapidminer Studio Documentation, 2018b) . Table 4 presents the Contingency Matrix which would be useful for evaluating the ML model that is trained and tested using DL method.

	T0	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	%
P0	197	0	0	2	0	0	0	16	0	12	0	91.63
P1	0	0	0	0	0	0	0	0	0	0	0	0.00
P2	0	0	0	0	0	0	0	0	0	0	0	0.00
P3	48	0	0	245	0	0	0	37	0	0	0	74.24
P4	0	0	0	0	0	0	0	0	0	0	0	0.0
P5	4	0	2	0	0	318	0	0	0	14	0	94.08
P6	3	0	0	0	0	0	120	2	0	16	0	85.11
P7	0	0	0	0	0	0	0	11	0	0	0	100
P8	0	0	0	0	0	0	0	0	0	0	0	0
P9	0	0	0	0	0	21	1	0	0	358	0	94.46
P10	0	0	0	0	2	0	0	0	1	0	158	94.61
%	78.17	0	0	99.19	0	92.17	100	16.67	0.00	92.27	100	

Table 4 Contingency Matrix for DL

The ML model generated using DL algorithm has been successful in predicting the number of occupants with 88.99% +/- 2.39% success rate, which showed a better performance compared to NB and GLM. The fourth algorithm tested was Decision Tree.

4.4 Decision Tree

Tree-based ML algorithms stem from two roots; Decision Tree(DT) and Regression Tree algorithms. Decision Tree is used for classifying the data while Regression Tree is used for predicting continuous variables. The Decision Tree algorithm has the potential of obtaining better result than other machine learning methods on multi-class prediction due to its tree-like model (Silva-Palacios et al., 2017). A decision tree is a tree where each node represents an independent variable, each link (branch) represents a condition/decision (rule) and each leaf represents an outcome(i.e. a class of the dependent variable) (Sanjeevi,2017). The whole idea of the algorithm is to create a tree like this for classification of the dependent variable by learning from the pattern of the data.

Candanedo and Feldheim (2016) used Decision Tree algorithm in order to estimate number of occupants by using sound, CO2, lighting, power use and motion sensors data. According to their results the success rate of the algorithm has been found between 81% and 98.441%. When only lighting was included into model as the independent variable success rate has been found 81%, when only sound was included the success rate was 90.78% and when only CO2 was included the success rate was 94.78%. This result shows that CO2 concentration in the room was found as the most indicative variable. Table 5 presents the Contingency Matrix which would be useful for evaluating the ML model that is trained and tested using DT method in this research.

	T0	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	%
P0	239	0	0	3	0	6	0	1	0	2	0	95.22
P1	0	0	0	0	0	0	0	0	0	0	0	0.00
P2	0	0	0	0	0	0	0	0	0	0	0	0.00
P3	9	0	0	243	0	0	0	3	0	0	0	95.29
P4	0	0	0	0	2	1	0	0	0	0	0	66.67
P5	3	0	2	0	0	336	0	0	0	3	0	97.67
P6	1	0	0	0	0	0	120	0	0	0	0	99.17
P7	0	0	0	1	0	0	0	62	0	0	0	98.41
P8	0	0	0	0	0	0	0	0	0	0	1	0
P9	0	0	0	0	0	2	1	0	0	383	0	98.48
P10	0	0	0	0	0	0	0	0	1	0	157	99.37
%	94.84	0	0	98.38	0	97.39	100	93.94	0.00	98.71	99.37	

Table 5 Contingency Matrix for DT

The ML model generated using DT algorithm has been successful in predicting the number of occupants with 97.53% +/- 0.70% success rate, which showed a better performance compared to NB, GLM, DL. The success rate of the algorithm in this case is similar to the upper bound of success rate mentioned by Candanedo and Feldheim (2016). The fifth algorithm tested was Random Forest.

4.5 Random Forest

Random Forest (RF) one of the most used supervised learning algorithms in machine learning thanks to its flexible structure which can be applied for regression and classification tasks. Random Forest as it’s name suggests is based on decision tree algorithm. Random Forest basically depends to combining many binary decision trees. It includes the bagging method which combine all independent predictors in order to use for model averaging. (Geuner et al., 2010).

Previously Candanedo and Feldheim (2016) stated that Random Forest algorithms shows poor performance in occupant number prediction using multi-sensor information, mentioning that this is very likely due to the presence of highly correlated variables in the model. Table 6 presents the Contingency Matrix which would be useful for evaluating the ML model that is trained and tested using RF method in our study.

	T0	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	%
P0	225	0	0	1	0	1	1	0	0	0	0	98.68
P1	0	0	0	0	0	0	0	0	0	0	0	0.00
P2	0	0	0	0	0	0	0	0	0	0	0	0.00
P3	10	0	0	245	0	0	0	11	0	0	0	92.11
P4	0	0	0	0	0	0	0	0	0	0	0	0.00
P5	18	0	2	0	0	343	0	0	0	71	0	79.40
P6	1	0	0	0	0	0	119	0	0	0	0	99.17
P7	0	0	0	1	0	0	0	55	0	0	0	98.21
P8	0	0	0	0	0	0	0	0	0	0	0	0
P9	0	0	0	0	0	0	0	0	0	317	0	100
P10	0	0	0	0	2	1	0	0	1	0	158	97.53
%	89.29	0	0	99.19	0	99.42	99.17	83.33	0.00	81.70	100	

Table 6 Contingency Matrix for RF

The ML model generated using DT algorithm in our study has been successful in predicting the number of occupants with 92.47% +/- 1.17% success rate, which showed a better performance compared to NB, GLM, DL, but the performance was worse than DT. Similar to the findings of Candanedo and Feldheim (2016) in our case DT also performs much better than RF for occupant number prediction. The sixth algorithm tested was Gradient Boosted Trees.

4.6 Gradient Boosted Trees

In contrast to the Random Forest learning algorithm, Gradient Boosted Trees (GBT) uses boosting method which predictors are not taken independently, but in a sequence. By sequentially applying weak classification algorithms to the incrementally changed data, a series of decision trees are created that produce an ensemble of weak prediction models to reach a stronger prediction model. (rapidminer Studio Documentation, 2018c). In the case of independent variables such as humidity, temperature, CO2 and sound, being dependent on each other (e.g. if humidity level has an impact on temperature or if CO2 level in the room would seem to have an impact on temperature value not based on physics rules, but if the pattern derived from data indicates that), the GBT model would then have high accuracy rate while estimating number of occupants. Table 7 presents the Contingency Matrix which would be useful for evaluating the ML model that is trained and tested using GBT method in our study.

	T0	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	%
P0	245	0	0	3	0	0	0	0	0	0	0	98.79
P1	0	0	0	0	0	0	0	0	0	0	0	0.00
P2	0	0	1	0	0	2	0	0	0	0	0	33.33
P3	7	0	0	244	0	0	0	1	0	0	0	96.83
P4	0	0	0	0	1	1	0	0	0	0	0	50.00
P5	0	0	1	0	1	342	0	0	0	1	0	99.13
P6	0	0	0	0	0	0	119	0	0	0	0	100.00
P7	0	0	0	0	0	0	0	65	0	0	0	100.00
P8	0	0	0	0	0	0	0	0	0	0	0	0
P9	0	0	0	0	0	0	1	0	0	387	0	99.74
P10	0	0	0	0	0	0	0	0	1	0	158	99.37
%	97.22	0	50	98.79	50	99.42	99.17	98.48	0.00	99.74	100	

Table 7 Contingency Matrix for GBT

The ML model generated using GBT algorithm has been successful in predicting the number of occupants with 98.80% +/- 0.37% success rate, which showed a better performance compared to all other methods.

5. DISCUSSION AND CONCLUSION

The research has focused on determining the occupancy status of and number of occupants in an indoor space, through the utilisation of a set of sensors and machine learning techniques. Six different ML algorithms were evaluated using 5-fold cross-validation. The research provided evidence that machine learning can be a useful approach to predict indoor occupancy. The accuracy of the results was high for Gradient Boosted Trees and Decision Trees, which indicated that the Tree-based algorithms provide more efficient results to predict the number of occupants based on multi-sensor information. Random Forest and Deep Learning appeared as applicable algorithms for some configurations of this problem. In the last stage of the research the success of the algorithms is tested once again by removing the information acquired from one sensor from the three sensor setup. In this case the tests covered three combinations, DHT11-MQ 135, MQ135-FC04, DHT11-FC04. The results of these tests and success rate of algorithms for each combination is provided in Table 8. The first finding of this test was combination of information coming from all sensors (DHT11, MQ135, FC-04) produces better results than all other options.

Model	DHT11,MQ135,FC-04	DHT11,MQ135	MQ135,FC-04	DHT11,FC-4
Naïve Bayes	81.63%	83.0%	77.5%	72.6%
G. Linear Model	83.2%	76.3%	66.7%	71.9%
Deep Learning	89.0%	84.8%	76.7%	76.6%
Decision Tree	97.5%	94.8%	87.7%	92.0%
Random Forest	92.5%	90.5%	89.2%	89.8%
G. B. Trees	98.8%	95.5%	91.7%	92.7%

Table 8 Comparison of Success Rates

Our results contradict with the findings of Candanedo and Feldheim (2016) who argue that removing information about a sensor can increase the prediction success of the ML algorithms. In fact similar to findings of the previous research in the domain, our findings also confirm that carbon oxides (i.e. measured by MQ135) are the most significant indicators for occupant number prediction. This is followed by temperature/humidity (i.e. measured by DHT11) and least significant indicator was found as the sound in the room (i.e. measured by FC-04) Future research will focus on, collection of more data for different space usage types and conducting tests with different algorithms.

REFERENCES

Analyticsvidhya, 2017, Navie bayes explained, https://www.analyticsvidhya.com/blog/2017/09/naive-bayes-explained/ (15 June 2018)

Candanedo L.M., Feldheim V., 2016. Accurate occupancy detection of an office room from light, temperature, humidity and CO2 measurements using statistical learning models, *Energy Build*. 112, 28–39.

Cantoni E., Ronchetti E., 2011. Robust Inference for Generalized Linear Models, *Journal of the American Statistical Association*, 96:455, 1022-1030, DOI: 10.1198/016214501753209004

Liu, H., Darabi H., Banerjee P., Liu J., 2007. Survey of wireless indoor positioning techniques and systems, *IEEE Trans. Syst. Man Cybern. C: Appl. Rev. 37* (6) 1067–1080.

Dobbs J.R., Hencey B.M., 2014. Model predictive HVAC control with online occupancy model, *Energy Build*. 82, 675–684.

Domdouzis K., Kumar B., Anumba C., 2007. Radio-frequency identification (RFID) applications: a brief introduction, *Adv. Eng. Inf.* 21 (4) 350–355.

Dong B., Andrews B., 2009. Sensor-based occupancy behavioral pattern recognition for energy and comfort management in intelligent buildings, *in: Proceedings of Building Simulation*, pp. 1444–1451.

Erickson V.L., Carreira-Perpinán M.Á., Cerpa A.E., 2014, Occupancy modeling and prediction for building energy management, *ACM Trans. Sens. Netw.* (TOSN)10 (3) 42.

Gajski D.D. 1997. Principles of Digital Design. *Englewood Cliffs, NJ: Prentice-Hall*

Geuner R., Poggi J.M., Tuleau-Malot C., 2010. Variable Selection Using Random Forest Pattern Recognition Letters 31,F pp. 2225–2236

Gu Y., Lo A., Niemegeers I., 2009. A survey of indoor positioning systems for wireless personal networks, *IEEE Commun. Surv. Tutor.* 11, 13–32

Hallberg J., Nilsson M., Synnes K., 2003. Positioning with Bluetooth, *in: 10th International Conference on Telecommunications, ICT*, 952, pp.954–958.

Hailemariam E., Goldstein R., Attar R., Khan A., 2011. Real-time occupancy detection using decision trees with multiple sensor types, *in: Presented at the Proceedings of the 2011 Symposium on Simulation for Architecture and Urban Design*, Boston, Massachusetts.

LeCun Y., Bengio Y., Hinton, G., 2015. Deep learning, Nature 521 (7553) (2015) 436-444.

Mahdavi A., 2009, Patterns and implications of user control actions in buildings, *Indoor Built Environ*. 18 (5)

Mahdavi A. 2011. People in building performance simulation, *in: J.L.M. Hensen*, R.Lamberts (Eds.), *Building Performance Simulation for Design and Operation*,*Spon Press*, ISBN 978-0-415-47414-6.

Mumma S.A., 2004 Transient occupancy ventilation by monitoring CO2, in: *ASHRAE IAQ Applications*, pp. 21–23.

Nguyen T.A., Aiello M., 2013. Energy intelligent buildings based on user activity: a survey, *Energy build.* 56 244–257.

Oldewurtel F., Sturzenegger D., Morari M., 2013. Importance of occupancy information for building climate control, *Appl. Energy* 101, 521–532

Rahman H., Han H., 2017. Occupancy Estimation Based on Indoor CO2 Concentration: Comparison of Neural Network and Bayesian Methods, *International Journal of Air-Conditioning and Refrigeration*, 25,3

Ren J., Lee S.D., Chen X., Kao B., Cheng R., 2009, Naive bayes classification of uncertain data, Ninth IEEE International Conference on Data Mining, 944-949

rapidminer Studio Documentation, 2018a, Naïve bayes, https://docs.rapidminer.com/8.1/studio/operators/modeling/predictive/bayesian/naive_bayes.html (15 June 2018)

rapidminer Studio Documentation, 2018b, Deep learning https://docs.rapidminer.com/8.1/studio/operators/modeling/predictive/neural_nets/deep_learning.html (15 June 2018)

rapidminer Studio Documentation, 2018c, Gradient boosted tree,https://docs.rapidminer.com/latest/studio/operators/modeling/predictive/trees/gradient_boosted_trees.html (12 June 2018)

Silva-Palacios D., Ferri C., Ram´ırez-Quintana M. J., (2017). Improving Performance of Multiclass Classification by Inducing Class Hierarchies, Performance of Multiclass, *Procedia Computer Science* 108C, 1692–1701

Sanjeevi, 2017, Deep math machine learning AI, https://medium.com/deep-math-machine-learning-ai/chapter-4-decision-trees-algorithms-b93975f7a1f1 (1 June 2018).

Szczurek, A., Maciejewska, M., Pietrucha, T. 2017. Occupancy Detection using Gas Sensors. *Proceedings of the 6th International Conference on Sensor Networks*. doi:10.5220/0006207100990107

Wang S., Burnett J., Chong H., 1999. Experimental validation of CO2-based occupancy detection for demand controlled ventilation, *Indoor Built Environ.* (8) 377–391.

Yang Z., Becerik-Gerber, B., 2014. The coupled effects of personalized occupancy profile based HVAC schedules and room reassignment on building energyuse, *Energy Build*. 78 113–122.