Applied Soft Computing 119 (2022) 108536

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

OccupancySense: Context-based indoor occupancy detection & prediction using CatBoost model

Joy Dutta *, Sarbani Roy

Department of Computer Science and Engineering, Jadavpur University, India

ARTICLE INFO

Article history:
Received 14 February 2021
Received in revised form 12 January 2022
Accepted 22 January 2022
Available online 3 February 2022

Keywords:
Occupancy detection
Occupancy prediction
Indoor air quality
IoT
Context data
Feature engineering
Machine learning
Data fusion
Forecasting
CatBoost

ABSTRACT

Occupancy detection and prediction are two well-established problems which can be improved further to achieve higher accuracy in both cases than the existing solutions. To achieve the desired higher accuracy, proposed OccupancySense model detects human presence and predicts indoor occupancy count by the fusion of Internet of Things (IoT) based indoor air quality (*IAQ*) data along with static and dynamic context data which is a unique approach in this domain. This data fusion helps us to achieve higher forecasting accuracy along with the integration of state of the art gradient boosting based categorical features supported CatBoost algorithm. For comparison, other commonly used machine learning classification and regression algorithms, e.g., Multiple Linear Regression (MLR), Decision Tree (DT), Random Forests (RF) and Support Vector Machine (SVM) for regression and Logistic Regression (LR), Naïve Bayes (NB), Decision Tree (DT) and Random Forest (RF), Support Vector Machine (SVM) for classification, were also assessed during this experiment. Out of these, CatBoost outperformed other models when considered in terms of accuracy. Hence, CatBoost is used as the core of the OccupancySense design and we have validated the proposed model by a real-world case study with continuous 91 days of indoor data, having 33 unique external features. These features are collected directly as well as derived from the collected data. To handle these features, feature engineering plays a key role in the OccupancySense model. The speciality of this model is, it is non-intrusive one but have high predictive power. It can detect occupancy and predicts headcount along with occupancy density of the room pretty accurately with 99.85%, 93.2% and 95.6% respectively (with 10 fold cross-validation) which outperforms other state of the art models.

1. Introduction

Indoor Air quality is very crucial when it comes to human health. This is one of the most ignored areas that needs to be focused more at present. Poor indoor air affects us both knowingly or unknowingly. Nowadays, people spend more than two-thirds of his life in closed indoor. This time includes the sleeping hour as well as the working hour. Hence, healthy indoor is very much important when considering Indoor Air Quality (*IAQ*). In real life, *IAQ* can be up to 10 times inferior to the outdoor air [1]. The reason is that in closed areas potential pollutants grow faster than the open space. Statistics also suggest the same. Practically, when different case studies are done in different countries, it is found that in developing countries, the health impacts of indoor air pollution far more significant those of outdoor air pollution. The list of major contaminants present in the *IAQ* are Carbon dioxide (CO_2), Particulate Matter (PM)2.5, PM10, total volatile organic compounds (TVOC), etc. [2] The level of these pollutants is directly correlated to indoor occupancy as humans are, in general, the major source of indoor pollutants [3]. Thus, for proper indoor pollution monitoring and for controlling the pollution level inside for maintaining *IAQ*, proper prediction of occupancy plays an important role. The concept is true vice versa as well, i.e., in order to improve occupancy detection and prediction, IAQ data can also be used [2].

Occupancy detection and prediction are two very well known problems in the research area. Along with *IAQ* maintenance, some major benefits come along with proper occupancy detection and prediction. Firstly, it can help to control energy consumption in developed countries. Heating, Ventilation and Air Conditioning (HVAC) systems are currently the sources of half of the country's total energy consumption [4]. Hence, to make buildings energy-efficient, these HVAC systems should be designed in an efficient manner such that it can meet the low energy goal of the country. This HVAC systems are strongly correlated with the occupancy parameter. The better this parameter is estimated, the more efficient the whole system will be. Due to the presence of human *IAQ* along with heat load generated by human metabolism and electrical and electronic appliances usage, indoor gets virulent.

* Corresponding author.
E-mail address: joydutta.rs@jadavpuruniversity.in (J. Dutta).

https://doi.org/10.1016/j.asoc.2022.108536
1568-4946/

Hence, proper prediction of occupancy helps to reduce energy consumption in the buildings and keeps the indoor healthy with the proper and controlled use of HVAC systems. Secondly, it helps to manage the space present in the smart building more efficiently. This helps building admin to identify if space is under-utilized or over-crowded. Effective space management keeps a place cosy in terms of comfort for the occupants. Thirdly, based on occupancy, indoor electrical appliances can be controlled smartly which helps to reduce the overall energy consumption of the building that includes automatic climate control of the room along with basic appliances like light, fan control etc.

Hence, we understand that accurate occupancy detection and prediction has huge benefits from many aspects. Now, there are two ways to predict indoor occupancy. The first one is the intrusive one, whereas the second one is non-intrusive one [4–6]. The intrusive one is more accurate but compromises with privacy whereas non-intrusive solutions struggle with accuracy but privacy is maintained. We found that intrusive solution can achieve 100% accuracy by using multiple camera modules [5] but the privacy of the occupant is compromised. Hence, the requirement of non-intrusive solution comes into the picture as privacy is one of the most important concern nowadays. Thus, our challenge here is to predict occupancy as accurately as an intrusive solution can by utilizing the non-intrusive approach that respects the privacy of the occupant [6]. For this type of solution, we need to use non-intrusive sensors which do not concern the privacy aspect [5]. Now, which sensors are appropriate for this solution, i.e., which sensors can provide sufficient information that can replace the requirement of the camera module is the biggest concern [4]. There are different approaches to predict occupancy non-intrusively and those are discussed in details in the related work section.

Here, in this work, we have found from our related study [2] and experimental research [3] that, *IAQ* along with the context information is the best combination to provide a non-intrusive but accurate alternative solution in this aspect. Here to predict indoor occupancy correctly, which *IAQ* and context factors to be considered, is another dimension of the problem. Because there are multiple *IAQ* factors which are affected by human presence, we have considered the following factors that we found most important according to our research, i.e., AQI, PM2.5, PM10, CO_2, Temperature and Humidity which is similar as mentioned in the article [2]. These factors are in turn dependent on different static (constant) and dynamic contexts (non-constant). Different static context involves the various constant indoor context of the room, e.g., room's heat isolation, ventilation, type of location, type of area, the context of the area, whereas, major dynamic contexts are status of the ac, fan, doors and windows, cleaning of the room in progress or not, physical activity of the people, outside weather context, weather season, etc. Hence, to achieve our goal to predict indoor occupancy as accurately as an intrusive solution, we have considered all these above mentioned non-intrusive factors. We found that our proposed non-intrusive approach (OccupancySense) has the potential to predict the occupancy more accurately than any other non-intrusive model present and can be compared with intrusive models. This is the main difference in our work with the rest of the previous work in this field.

According to our findings, the reason for achieving this accuracy is that it is not only considering direct features (*IAQ*) but also the root cause of change in these direct features. These causes are nothing but various context information which are affected by occupancy directly. In this industrial 4.0 revolution, with the help of IoT, considering these features for occupancy modelling is easier than before. Thus, here we aim to predict indoor occupancy detection as well as prediction (headcount) accurately using *IAQ* and all these context information.

Hence, the problem can be stated as follows :

Given a time series indoor air pollution data along with corresponding non-intrusive static and dynamic context information based on previous events that have been observed and collected at regular time intervals for a specific indoor location, we aim to predict next moment's indoor occupancy at that location.

Our main objective is here to accurately predict three different subjects, i.e., detection of indoor occupancy, prediction of occupancy density and exact headcount based on our collected data. The main contribution of this research work can be summarized as follows:

- Identification of the feature set for indoor context to get the best result by true integration of static and dynamic context along with Indoor Air Quality (*IAQ*).
- Getting the best accuracy level in terms of both Occupancy detection and prediction to date as compared to its predecessors.

The rest of the paper is structured as follows. In Section 2, some of the most relevant works related to this indoor occupancy detection and prediction are identified and our contribution has been pointed out in this context. Then in Section 3, the proposed OccupancySense model has been discussed. Next, Section 4 describes the case study with details. After this, Section 5 comprises of the implementation details along with results for the proposed OccupancySense model. Finally, Section 6 concludes the paper.

2. Related work

Occupancy detection and prediction is a domain where various options are tested thoroughly. Out of which CO_2 based occupancy detection is the most common one [6,7] as CO_2 is the most important factor that helps us to understand occupancy. But, only CO_2 based systems are not as accurate as a multi-sensor network [4,5,8] is formed for occupancy detection and prediction. A fusion of IoT based sensor data and non-intrusive environment sensor also showed promising signature in this research [9,10]. Different machine learning and artificial intelligence based techniques are also tested in different scenarios to predict occupancy and has been proven to be a successful alternative to existing methods [11–13].

In [6], a novel Feature Scaled Extreme Learning Machine (FS-ELM) algorithm is proposed where they have tested the system with 4 tolerance level. This tolerance level is high if you want fine, granular and exact headcount in terms of accuracy. Whereas in [7], the relationship between occupancy and internal structure of the time series was disclosed to detect how human factor influences *IAQ*. They proposed to use pattern matching represented by different occupancy profile. They achieved 82% true positive and 22% false-positive detections.

Now, when instead of using only a single CO_2 sensor, environmental sensing is utilized in [4], authors utilize sensor fusion of multiple environmental parameters (e.g. temperature) in their proposed framework. Their proposed Sensing by Proxy leads to fractional accuracy level (hourly) with root mean square error (*RMSE*) 0.6044 (fractional person), while the best alternative by Bayes net is 1.2061 (fractional person) when considered with 7 participants in the experiment room and limited test data. Also, heterogeneous sensor network [5] is using motion detection, power consumption, CO_2 concentration sensors, microphone or door/window positions for occupancy prediction. This approach uses machine learning along with heterogeneous sensors and utilizes a camera module for a limited time. Five occupancy levels were defined to generate decision trees because of the maximum number of occupants in the office and have achieved an accuracy of 81%. In [8] also, Double-beam, Pressure mat, Acoustic, PIR

motion and Carbon dioxide sensor is utilized together with sensor fusion concept. In this context, the measurement of Normalized Root Mean Square Error (NRMSE) facilitates the comparison between models with different scales in terms of RMSE. This results in an hourly average of 89%–98% accuracy in presence detection, 0.08–0.11 (NRMSE) in occupancy density and 0.13–0.16 (NRMSE) in predicting the exact number of headcounts when at max 13 occupants were present.

In this regard, IoT can be very effective, and the same is explored in [9]. IoT based sensor data fusion is the key concept of the paper which is utilized here for sensing the occupancy in smart buildings for a better prediction. Researchers predict indoor occupancy as high as 99.09% by utilizing the data collected by this IoT based system where different indoor parameters (e.g., temperature, humidity, CO_2, light etc.) are considered and by using Dempster–Shafer evidence theory for sensor data fusion purpose. On the other hand, in [10], the authors use the concept of environment sensor fusion from indoor ambience parameters e.g., CO_2, volatile organic compounds (VOCs), temperature and humidity to predict occupancy by collecting data from a 4-student apartment from 49 days training data. They also compare different supervised machine learning algorithm in this purpose and achieved occupancy detection and headcount prediction with 81.1% and 64.7% respectively. In [11], Decision tree model along with Hidden Markov model is used for occupancy prediction by considering time, environmental and energy data. Here, hourly level data is collected for the experiment and corresponding case study handles 5 occupants in the room for occupancy prediction which provide us up to 89.5%. By applying machine learning based approach in [12], i.e., kNN, DT, MLP and GRU machine learning techniques satisfactory performance has been achieved which in turn helps to improve building's HVAC control system performance. In another research [13], occupancy information is obtained from images by CNN-based density estimation methods and achieved Average all-day prediction accuracy of 83.12% for occupancy of the target building.

Hence, it is found that there is scope to improvements in the accuracy in the occupancy detection as well as prediction. Though the occupancy detection domain is almost saturated, the struggle for reaching higher accuracy still exists in occupancy prediction category (head count).

3. OccupancySense: Context-based non-intrusive model for predicting occupancy

The proposed OccupancySense model for predicting indoor occupancy is divided into five major steps—out of these, the very first step is data collection. We found that there is neither standardized context-based dataset available with corresponding time series *IAQ* data that can be used as a benchmark, nor it is available in the form of an open-source project (to the best of our knowledge). Hence proper data collection is the first significant step to achieve.

After data collection, the second step is to prepare that data for machine learning model fitting, i.e., apply feature engineering on the data to make the data applicable to machine learning-based models. In real life, collected raw data cannot be directly applied in machine learning models for target prediction. Also, collected context data cannot be used directly in any model. Thus, to solve this it is required to process *IAQ* data along with context data, in such a way, that can be applied to any machine learning model. For this purpose, the second step is further sub-categorized as a compendium of initial data analysis, encoding context data, data preprocessing and feature extraction to produce the model ready dataset.

Now when this model ready data set is prepared, it is found that this dataset has many features to consider if any machine learning model is applied to this model ready data. Not all models can handle a large number of features due to the increase in the model's complexity. Also, considering extra features may lead to model overfitting. Hence to reduce the feature space, feature selection is the third step. This step also helps us to identify the more important features that need to be considered for the problem in order to achieve a standard accuracy.

Next, in fourth step, we map the problem in the machine learning domain. In this work, three issues are getting addressed, i.e., Occupancy Detection, Occupancy Density Prediction and Occupancy Headcount prediction. These three problems need three different approaches to handle since three problems are inherently different according to the problem's nature.

Then, the CatBoost Model is being studied in details and established as the best suited model in this context. After this, the model ready data is applied to the selected CatBoost model for performance evaluation in the result section after understanding the case study in details. In the result section, different machine learning models are also considered and compared with the selected CatBoost model with performances of these individual models to predict indoor Occupancy for future instances. We train these models with the model ready dataset and predict the indoor occupancy and verify our claim. These are, as a whole, the fifth and final step for the OccupancySense. Now, each of these steps will be discussed in details below.

3.1. Create occupancy prediction dataset

As mentioned earlier, one of the major problems of the context-based research is the unavailability of standard data which can be used for testing the proposed model's performance. Hence, to find the effectiveness of the inclusion of context data we have to build our own set up to collect both indoor pollution data along with context data and build a database based on which further investigation can be done into this research area. So, preparation of the database is the first step of the experiment upon which the rest of the experiment is based. The features that are being considered here are mentioned in Fig. 1. Here, different data sources are further discussed in the case study section separately in details.

Next, for preparing the dataset for the experiment, feature engineering plays a key role to understand the collected data in a more sensible way. Here, feature engineering steps include initial data analysis, context data encoding, data preprocessing and feature extraction. Here, collected raw data is statistically analysed first and the same is visualized for clear understanding. Then the collected context data is encoded numerically such that the dataset can be applied to the machine learning models. After that, pre-processing is done on the data to make it model ready. For data pre-processing, the following issues need to be taken care of: missing data, outliers present in the data, data skewness and device anomaly. Details of each of the mentioned step are discussed in the case study section. Next, to improve the model's performance further, information is extracted from the timestamp associated with each instance. By this feature extraction, much more relevant information is gathered from the time stamp which is useful for seasonal forecasting. This will, in turn, enrich the collected dataset as this feature extraction will now be utilized by the model to predict the next moment's indoor occupancy forecasting.

In this context, it is found that sensor fusion and incorporation of IoT is playing a key role. From the investigation, we found that IoT along with sensor fusion can be much more useful than individual sensor use. Here, we have used three types of data fusion based on relations between the data sources. These are complementary, redundant and cooperative fusion [14]. Out of

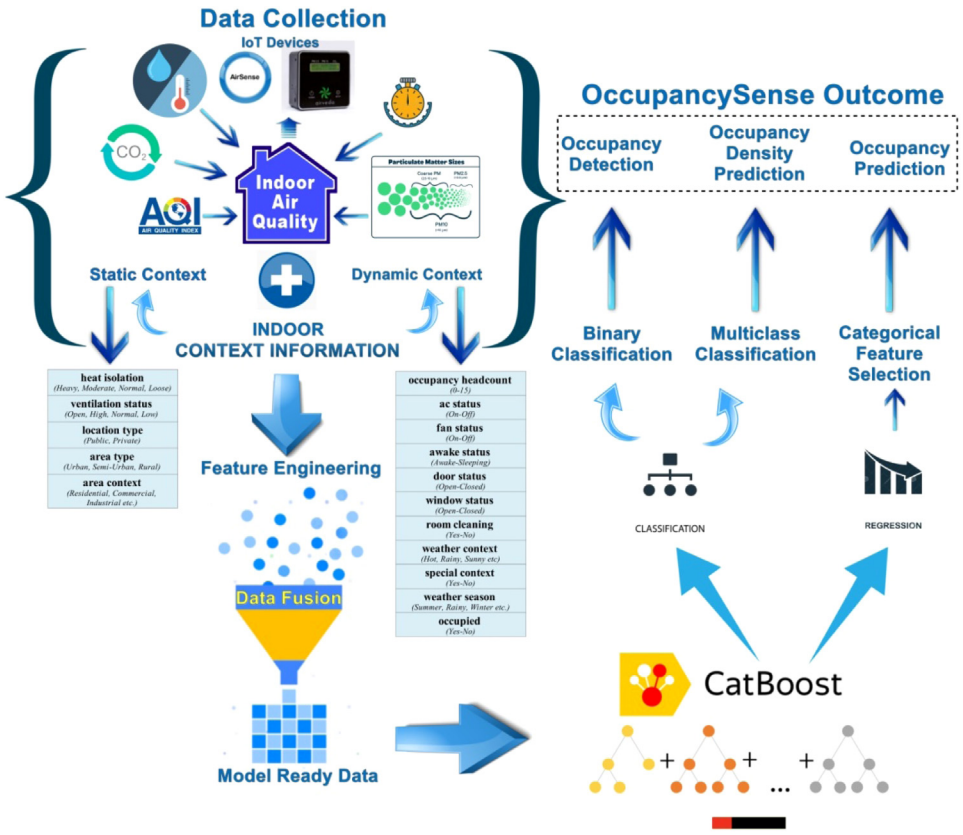

Fig. 1. OccupancySense model workflow.

these techniques, complementary fusion predicts different parts of the target, hence, increases overall predictivity of the system. In redundant fusion, more than one resource is considered for target prediction to gain the overall prediction confidence. Lastly, cooperative fusion is used when more than two resources are merged to get new information from the same source data which helps us to predict target more accurately as that is built from the collected data. Here, using these three types of fusion, we aim to predict indoor occupancy more precisely than other existing models in this domain from our collected data. This is further discussed in the case study section. As a resource, relevant features from the collected data that are directly affected by occupancy (i.e., *IAQ* and dynamic context) and some location-wise constant features (i.e., static context) in our model have been considered as the main factors to predict occupancy accurately.

Thus, the model ready feature engineered dataset is prepared. This ready dataset now can be directly applied to other machine learning models as well for occupancy forecasting. The procedure, that is applied here, is a generic one. The model ready dataset is valid not only for any specific model (e.g., CatBoost) but also for other supervised as well as unsupervised machine learning models.

3.2. Identification of important features: Feature selection from the model ready dataset

In indoor occupancy prediction, feature selection can play an important role in enhancing the model's performance. It is the method for reducing data dimension while doing predictive analysis without affecting the model's performance, thereby reducing its complexity. Simple models are easier to interpret, have shorter training times and they also enhance generalization by reducing overfitting. Feature selection also helps a model to handle bad learning behaviour in high dimensional spaces. This method uses the variable ranking technique to select the variables for ordering. By ranking, it means how much useful and important each feature is expected to be for the target prediction. It can select the subsets of variables independently of the chosen predictor.

Here, in the model ready dataset, 33 features are being considered in total by including all *IAQ*, context and time stamp extracted features. All the features are not of similar importance. Also, considering all the features may lead to model overfitting for different models. This makes the model more composite in terms of complexity. Dropping features randomly is also not a smart job. It may discard an important feature for the problem which was required otherwise for a better prediction. Besides, there is plenty of other aspects to a problem. It is very much possible, that for a specific perspective a feature is not important but in another aspect may require that feature for accurate modelling. For example, in this specific indoor pollutant (CO_2) prediction problem, important features will not be similar to indoor occupancy problem. Even, it is possible that if a different indoor pollutant is selected., e.g., PM2.5, a different feature set will be selected for its prediction. Thus, based on the chosen problem/pollutant, this selected feature set will vary. Hence, proper selection of features is very important in terms of the OccupancySense model's performance. Presently, this model focuses on the indoor Occupancy prediction problem only.

Feature selection generally reduces a model's complexity but it has its own cost associated with it. If one is using a reduced set of features for training the model to reduce the model's complexity, the accuracy of the model also reduces. The same is true for our case as well. This concept is valid in general for all machine learning based models.

In this work, we have used the filter method for fast feature space reduction to achieve optimal model performance. This is

done without increasing the time complexity of the whole system or losing information about the dataset. It is the fastest statistical tool to reduce the feature set [15]. Filter methods are generally the first step in any machine learning modelling exercise. They provide quick and easy sanity checks for the features that immediately allow us to reduce the feature space and get rid of un-useful features. The filter method is fast and has the capability of selecting variables independent of the selected model. It relies only on the characteristics of the data. Here, the feature selection method contains the removal of duplicate, constant, quasi-constant, and correlated features. Constant features are those features that do not vary with time. Hence, they do not add any extra information that enhances the model's performance. On the other hand, here a quasi-constant feature means those features whose 98% values are identical. These sorts of variables generally produce very little information on the overall system. On the other hand, correlated predictor variables provide redundant information. If two features are highly correlated, the second one will add little information over the first one, hence removing the second one helps to reduce the dimension. The same concept is applied for duplicate features also as they carry no extra information and thus need to be discarded from the feature set. Applying these four filtering techniques reduce the feature space and help us to identify important features for the mentioned specific problem. Mapping of this feature selection procedure in our specific problem is further discussed in the case study section.

3.3. Mapping of the problem in machine learning domain

Here, in this research, we aim to predict three different things, i.e., indoor occupancy detection, occupancy density prediction and exact headcount prediction based on the model ready data. The first problem is occupancy detection, i.e., at a specific moment whether an occupant is present inside the room or not. Hence, this problem can be formulated as a binary classification problem. It has only two specific outcomes, either it will provide information that occupant is present, i.e., 1 is the classification result or will provide information that the room is empty, i.e., 0 in the classification result.

The next problem here is to predict the occupancy density. For this purpose, the occupancy level is divided into 5 categories in general for any indoor, i.e., Very Low, Low, Medium, High and Very High. Based on the available historical data, it is considered that the maximum numbers of occupants present in the room in near past as the highest and considering zero as the lowest. The categories can be divided as follows: 0%–20% as very low, 20%–40% is low, 40%–60% is medium, 60%–80% is high and 80%–100% is very high which is verified by a frequent occupant of that indoor. Hence, the occupancy density prediction is mapped as a multiclass classification problem where based on different rooms, the number of occupants belonging to each category will change.

Finally, the last problem is an accurate indoor occupancy prediction (headcount) which is, by nature, a regression problem. The reason is that the target variable (indoor occupancy value) is having a real and continuous value. Since the occupancy prediction (headcount) problem is a problem of regression, the way of measuring impurity is variance. So, while training, the feature importance is measured by calculating how much a feature reduces the variance. The feature, that reduces the variance the most, is the most important feature of all. The same concept is applied for the rest of the features as well. Here, headcount prediction produced is in floating format which is converted to integer for measuring the model's performance as floating human presence is not possible.

3.4. Basic concepts

3.4.1. Study of CatBoost model

Since we are handling various context information which is mainly mapped as categorical variables, hence the success of OccupancySense model largely depends on the proper handling of these categorical variables. For selecting the best model for this research problem, it is initially found, that gradient boosting may be the primary method for this problem with the mentioned heterogeneous features, noisy data, and complex dependencies. This has come to our mind by looking at the working procedure of gradient boosting. It is a process of creating ensemble predictor by performing gradient descent in a functional space which actually shows how multiple weak predictors (which are base predictors) iteratively can merge greedily to create a strong predictor. But, it is found that though this process is proven to be a very effective solution in most of the cases, two major problems identified in it which stops us to choose this model. These flaws are prediction shift and target leakage. Both the problems are very crucial for the present scenario as we want to achieve the best in class accuracy for our problem in the real-life case study.

To understand this, one needs to understand target statistics (TS) first [16]. To handle a large number of categories, a popular method that is used to handle categories is target statistics (TS). This estimates expected target value in each category. Generally, TS can be considered as a simple statistical model predicting the target and it suffers from conditional shift caused by the target leakage. To understand target leakage, in a simple manner, one can think about a scenario where some information from target is used to predict the target itself. This leakage consequently increases the risk of overfitting on the training data, especially when the data is small. Similar target leakage also exists in standard gradient boosting algorithms. In reality, prediction shift is caused by a special kind of target leakage. When a prediction model is obtained after several steps of boosting by relying on the targets of all training examples, generally leads to a shift of the distribution of function for a training. This finally leads to a prediction shift of the learned model. We can identify this problem as a special kind of target leakage. Catboost has implemented a technique called ordering principle (Ordered TS) which solves the problem of target leakage in both cases.

This motivates us to work with CatBoost [16–18] model which handles these two problems efficiently using the concept ordered boosting. We can map our research problem to this CatBoost model as follows:

We assume that the observed dataset is $\mathscr{D} = \{(X_k, y_k)\}_{k=1,\ldots,n}$, where $X_k = (x_k^1, \ldots, x_k^m)$ is a random vector of m features including *IAQ* and context features and $y_k \in \mathbb{R}$ is the target which is the occupancy, which can be either binary (in case of occupancy detection) or numerical (occupancy density prediction) response. Collected $\{(X_k, y_k)\}$ are independent and distributed according to some distribution P. This distribution can be either skewed or can be normal. Based on location and different context parameters, this distribution varies in reality. Next work is to train the model with the collected model ready data. Here, the goal of the learning phase is to train the function $F : \mathbb{R}^m \rightarrow \mathbb{R}$ which minimizes the expected loss $\mathscr{L}(F) := \mathbb{E}L(y, F(x))$ in terms of the supervised target in the training set. Here, L is the smooth loss function which we need to choose from the model performance metrics accordingly.

To understand the working procedure of this model, it is important to understand its working principle. This model follows the structure of the oblivious tree which is symmetric in nature. The features ('m' features of collected data) that we have considered for the problem, are used to divide the tree in left and right partitions. Hence, we can represent the same as, with

our collected dataset $\mathscr{D} = \{X_i\}_{i=1,\ldots,n}$ are partitioned based on individual features with left and right subsets as $\{X_i^l\}_{i=1,\ldots,n}$ and $\{X_i^r\}_{i=1,\ldots,n}$. Here, the objective function for this model can be stated as:

$$argmin_r\{P(r, y, M)\} = argmin\frac{1}{\sum_{i=1}^{n} \mid x_i \mid} \times (\sum_{i=1}^{n} \mid X_i^L \mid var(y(X_i^L)) + \mid X_i^R \mid var(y(X_i^R))) \quad (1)$$

where r signifies the decision rule, y is the target function and M is the function to measure the optimality of the decision rule r. For reference, all the critical points considered for the CatBoost algorithm are mentioned in [16].

We can broadly classify the advantage of CatBoost model in two folds. The first one is its exceptional capabilities to handle categorical data which is used fully to predict the target (e.g., occupancy here). Ordering principle-based target statistics is utilized here on the training data to predict the target. This has the capability to effectively reduce information loss as well as overfitting which resolves the target leakage problem effectively. Now, here as we mentioned earlier, we have a dataset $\mathcal{D}$ having a random vector of m features, the kth training is replaced by a numerical feature (shown in Eq. (2) below) [16], according to the ordered TS to achieve the above-mentioned advantage.

$$\hat{x}_{\Phi_{p,k}} = \mathbb{E}(y|x_{\Phi_p} = x_{\Phi_{p,k}}) = \frac{\sum_{j=1}^{p-1}[x_{\Phi_{j,k}} = x_{\Phi_{p,k}}]Y_{\Phi_j} + aP}{\sum_{j=1}^{p-1}[x_{\Phi_{j,k}} = x_{\Phi_{p,k}}] + a} \quad (2)$$

where, $[x_{\Phi_{j,k}} = x_{\Phi_{p,k}}] = 1$ if $x_{\Phi_{j,k}} = x_{\Phi_{p,k}}$ or 0 otherwise. Here, $a > 0$ stands for the weight of the prior distribution P.

Which effectively denotes that the categorical variable $x_{\Phi_{p,k}}$ can be substituted with the average level value of the same category. CatBoost has also the capability of combining different categorical features which we provide at the time of model training and can create a new more effective category internally in a greedy fashion when it tends to split a new tree.

The second reason, it builds tree avoiding gradient bias which is one of our model's major requirement. This, in turn, stops the prediction shift. To achieve the same it made some modification in the tree structure. This modification of the Gradient Boosted Decision Tree procedure is called the ordered boosting which is the heart of this CatBoost procedure. Here, decision trees are considered as base predictors for this model. Hence, balanced trees are utilized for speeding up the execution as well as testing time. These trees are less prone to overfitting and can handle categorical features more efficiently by using the target statistics (TS) than any other method available currently. TS along with the ordering principle helps CatBoost to minimize information loss while handling categorical features.

This model can handle both classifications as well as a regression problem with the same confidence and with high accuracy. The reason for achieving the same is evaluating feature importance efficiently. This actually measures the change is prediction due to a feature. Hence, feature with high importance is responsible for greater influence in the final prediction in CatBoost model. In our case, based on the requirement, the feature list can be extended further. Hence to handle all those features, OccupancySense design methodology should not change frequently. So, keeping in mind this concept, we found Catboost model is the most suitable model to serve the core of the proposed system for this purpose. This model is capable to handle more features in future without an issue without changing system design. Here, if a new feature is planned to be incorporated in the system, then that particular feature should be incorporated in the model ready dataset as any feature cannot be treated directly in any machine learning model. Once the model ready data set is prepared, OccupancySense will rank this feature according to their actual importance and will produce required results accordingly.

The reason of the same is at the heart of the OccupancySense model, CatBoost is there and this particular feature of the original CatBoost model is inherited by OccupancySense model. In reality, Catboost can understand the most important features because of its own design mechanism. Hence, once the feature set is finalized for the proposed model, catboost will automatically order the features according to their importance for OccupancySesne. Because of this, OccupancySense model can handle categorical features very nicely which helps the model to outperform other relevant machine learning models in performing classification and regression.

3.4.2. Model performance metrics

As we have seen Section 3.3, mapping of our problem requires both classification and regression. Hence, to measure the performance of our model, these both aspects are required to be evaluated. Since the nature of classification and regression are different, hence different standard matrices are needed to be considered to measure the different types of model's performance.

For regression, we have considered three metrics here, i.e., coefficient of determination (R^2), the root mean square error ($RMSE$), and the mean absolute error (MAE). These three statistical matrices are used to understand how good our prediction model performs with respect to the observed value. R^2 metric measure the goodness of fit of a model which ranges the value from 0 to 1 which signifies variance explained by the model concerning the total variance. The value close to 1 refers to a more perfect fit for the model which in turn refers to the model's greater predictive power. R^2 is calculated according to Eq. (3).

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_{pred,i} - y_{obs,i})^2}{\sum_{i}^{n}(y_{obs,i}, y_{obs\_avg})^2} \quad (3)$$

where $y_{pred,i}$ is the predicted value, $y_{obs,i}$ is the observed value, y_{obs_avg} is the mean of observed value, n is the number of observations.

Next metric that we have considered here is $RMSE$. This measures the standard deviation of the prediction errors which means it gives us an estimate of how our predicted values deviates from the values observed. This is calculated as shown in Eq. (4). Another similar metric that we have considered here to measure regression prediction is MAE which is measured according to Eq. (5). It measures the difference between the predicted value and the actual value. Generally, a high R^2 value (Close to 1) and low $RMSE$ and MAE values (based on the target's average value) is desired for any model prediction.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_{pred,i} - y_{obs,i})^2} \quad (4)$$

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_{pred,i} - y_{obs,i}| \quad (5)$$

Here, to measure the classification accuracy, log-loss, ROC_AUC and F1 Score are used. One of the most basic metrics that is used mostly, is the accuracy. This is a ratio of total correct predictions to the total number of predictions. We calculate this metric using Eq. (6) below.

$$Accuracy = \frac{TP + FN}{TP + FP + TN + FN} \quad (6)$$

where True Positive (TP), True Negative (TN), False Positive (FP) and False Negative (FN) construct the confusion matrix to find out the accuracy.

However, there are cases when this metric is not ideal (e.g., class imbalance). Hence, log-loss, ROC_AUC and F1 Score are used to check the classification model's performance accurately. Out of these, logarithmic loss (which is related to entropy) measures the accuracy by penalizing false classifications. Any classification model's goal is to reduce this value and make it as small as possible. Ideally, this value should be zero. This log-loss function is expressed as:

$$H_p(q) = -\frac{1}{N}\sum_{i=1}^{N} y_i \times \log(p(y_i)) + (1 - y_i) \times \log(1 - p(y_i)) \quad (7)$$

where y is the label (1 and 0) and $p(y)$ is the predicted probability for all N instances.

Another classification metric which is used here to measure the classification model's performance is ROC_AUC score. ROC (Receiver Operating Characteristics) is a probability curve and AUC (Area Under The Curve) together measure the degree of separability between classes. Hence, a higher the value (Best case value = 1) of this score represents better separability.

Another metric that is considered here to measure classification accuracy is the F1 score. This classification measure is particularly useful when imbalanced class distribution exists in the data. The fact is, in a real-life problem, the presence of imbalanced class is very normal. Hence, this metric is particularly useful along with the accuracy which is not capable to handle this properly. Hence, using the concept of TP, TN, FP and FN, we measure the value of this F1-Score with the help of Recall and Precision as shown in Eq. (8), (9), (10) respectively.

$$F1\_score = \left(\frac{Recall^{-1} + Precision^{-1}}{2}\right)^{-1} = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (8)$$

where.

$$Precision = \frac{TP}{TP + FP} \quad (9)$$

$$Recall = \frac{TP}{TP + FN} \quad (10)$$

Lastly, we have used a general statistical model, i.e., K-fold cross-validation for both classification, as well as for regression. This is a tested statistical method used widely to evaluate a machine learning model. This is used to select a model out of many predictive machine learning models based unseen data. This is heavily used in applied machine learning domain and our one is from the same domain. Here, we have 2 key-words in the name of this performance measure metric, i.e., K- Fold and Cross-Validation. The term cross-validation is a resampling procedure used to evaluate machine learning models on a limited data sample whereas K-fold refers to the number of data sample groups created from the whole data for measuring the model's performance. Here, we have chosen as K = 10, this means 10 fold cross-validation is used throughout the experiment for checking our model's validity.

4. Case study

To check the validity of the proposed system, a real-world case study has been made. In this study, to prepare the database, a University research lab is chosen, e.g., DST-FIST lab of Computer Science & Engineering Department, Jadavpur University for the study of indoor environment located at Kolkata, West Bengal, India. This lab has a sitting arrangement for 15 persons. This computer lab is composed of approx. 300 square feet area, having 3 work stations, 12 Pc and 1 Server. This lab has 6 tube lights, 2 ceiling fans, 2 AC's in the lab for inside climate control. It has 1 wooden door for entry and exit, a large clear window glass area (90sqft approx.). This lab is situated in an urban area, normally isolated from outside heat and the ventilation system is also normal in this lab. It is situated on the 3rd floor of the building 'Prayukti Bhaban' which is an institutional building at approx. 40ft height.

We have used AirSense [19,20] for IoT based *IAQ* data collection and airveda [21] for the ground truth of the data in this lab. By the term *IAQ*, time-series data of AQI, PM2.5, PM10, CO_2, Temperature and Humidity is referred. For the context data, we have depended on the participating volunteers of the lab, who are research scholars of the same department. The data is collected in the above-mentioned lab for the duration starting from 20th June 2019 to 18th September 2019. The *IAQ* tuple is collected from the standard device on minute level granularity, i.e., 129 600 instances which consists of 907 200 data points. The unique context features of 16 types were collected with 30 min granularity, i.e., 4320 instances having 73 440 data points. Features extracted from the timestamp and other raw feature data takes the granularity of the source. Hence, the overall data points further increase as the number of features increases.

These collected context data is a combination of static and dynamic context data. For any particular place, this static information (e.g., heat_isolation, ventilation_status, location_type, area_type, area_context) remains fixed all the data collected in that particular location and changes only if the location changes. Hence, considering different data collected from different locations will make the dataset robust and more accurate for general occupancy detection and prediction purpose. Here, for this work, a specific research lab has been considered, hence, these static information remains fixed throughout the experimental dataset. The reason for considering same is that neither we can change the location, nor the building construction. But, these features have effects on the indoor air quality (IAQ) which in turn play an important role in our research for estimating indoor occupancy. The parameters, which are considered as static, will not be static when this model will be deployed in different real world scenarios, i.e., rooms in different locations, at different heights and with different configuration features. Thus, if this model is used in different rooms, in various conditions, the effect of different parameters can be more prominent. Rather, in these cases, contexts which we have considered here as static, will not be static any more. But for this particular case study, these data are treated as constant data. The general model will require those static contexts to make the model global.

Here, the variation of the dataset is only due to the change in dynamic context data. By the term dynamic context data, we have considered here the following ones, e.g., the status of the ac, fan, doors and windows ((Open |≋| / Close |×|)), cleaning of the room in progress or not, physical activity of the people (Awake ⊤ - Sleeping ¬) etc., outside weather context(Hot, Rainy, Sunny etc.), weather season(Summer, Rainy, Winter etc.), presence of any special context and numbers of humans along with occupancy for the ground truth purpose.

After collecting the data, the first job to make the data model ready. Hence, the first step of creating model ready data preparation is the initial data analysis. It is found that there are some missing data points in the collected raw data, i.e., a small portion, i.e., 1.7% of data is missing in the collected dataset from the air quality measuring device. The next step in the initial data analysis is to inspect the nature of the data. It is found that different types of variables in the collected data which includes the numerical, categorical and date–time variables. Here, device collected features are generally of integer types and continuous in nature. Here, it is also noticed that IoT based AirSense collected data

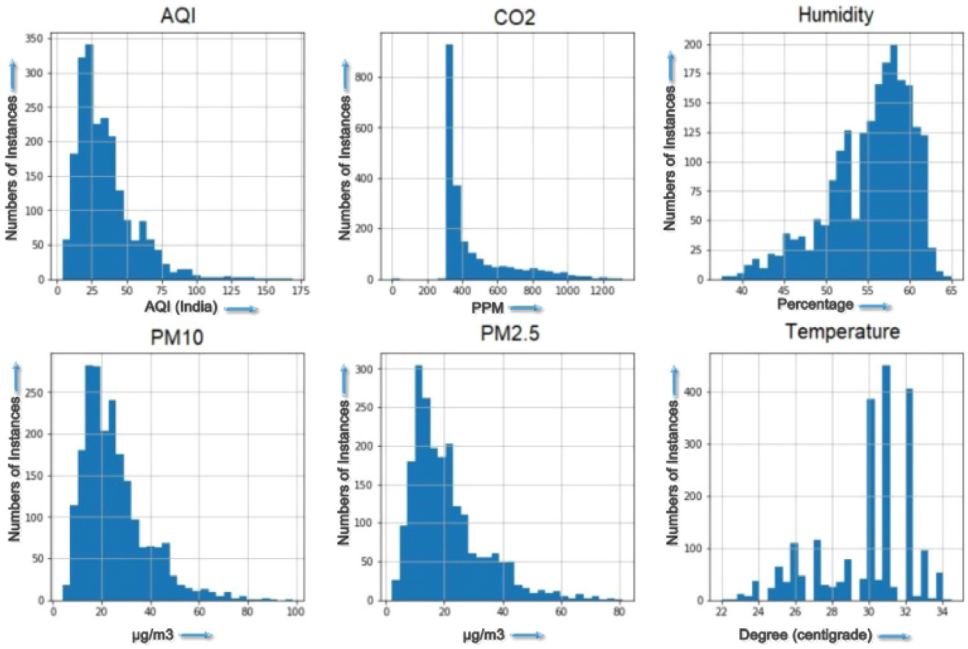

Fig. 2. Histogram for AirSense Collected *IAQ* data.

are numerical but not homogeneous in nature and distribution of these data features is not normal always. Hence, skewness is found in the data. The same is reflected in the histogram of the collected data shown in Fig. 2 below.

While inspecting the nature of the data further, it is found that context data features are a mixture of numerical and categorical data types. Hence, context data encoding is used here to handle these collected features. Handling numerical data is relatively straightforward compared to various types of categorical data. Here, context data are mapped to categorical data and required special attention such that it can be a part of model ready data. Different encoding schemes are utilized to handle diverse context information. Here, according to the problem's nature, the collected categorical context data is encoded as of ordered ordinal categorical variables, discrete numerical variables and nominal categorical variables. Here, nominal categorical variables are further sub-categorized as binary as well as numeric variables for a better model fitting.

Now, it is time to pre-process the data. It is found the data has some specific types of anomalies within the collected data in terms of consistency, presence of skewness as well as outliers. Here, for the dataset, the device's anomaly, missing data, variable distributions and magnitudes are handled specifically in the data preprocessing section. All these mentioned factors otherwise affect the model's performance as these factors distort the target variable's distribution as well as characteristics of the data. This, in turn, affects the model's prediction capability also. Finally, required features are extracted (e.g. year, dt_week, dt_month, dt_day, day_of_week, weekend, hour, minute, second, holiday) from the timestamp which finally makes the data model ready. Details of preparing model ready data procedure is followed here according to [22] authors as mentioned in their research paper.

This model ready data contains all the mentioned features that are collected from the real world indoor. To get the most out of these features, data fusion is applied to these collected data. To achieve this, we have to consider complementary, redundant as well as the cooperative fusion here. Here, in all the cases, our target is to predict the occupancy. Next, we have applied Error-Trend-Seasonality (ETS) decomposition to statistically analyse the data further. It is a model used for the time series decomposition. It decomposes the series into the error, trend and seasonality component. Decomposition provides a useful information related to the nature of the data. It is a univariate forecasting model used when dealing with time-series data. Here, in our case, we are working with time series data and here by using ETS decomposition, we find out that, there is seasonality (weekly) as well as trend (additive) present in our data (reflected in Fig. 3). Using this statistical tool, we also have found that there is a prominent error part present in the data. The speciality of this decomposition technique is that the observed data can be recreated when you add up all the three decomposed parts. Thus, if you can decompose properly and get a clear idea about the decomposed components (changing pattern of the components) properly then by utilizing that knowledge, you can predict future occupancy as well. Thus, here our aim is to predict each part of the statistical decomposition procedure, such that we can predict occupancy more accurately. This decomposition gives us an internal understanding of the data. From the statistical analysis, we have found that CO_2 is the most important component to predict the indoor occupancy. But considering components—temperature, humidity, AQI along with CO_2 becomes an example of redundant data fusion which helps us to understand indoor occupancy from the *IAQ* factors. The reason is that all the factors have some predictive power but when used as a whole, the overall predictive power of the system increases. On the other hand, timestamp extracted features are utilized as cooperative fusion. For example, date, hour, holiday and weekend can provide us some specific time-related information that one of these features alone cannot provide. Mapping the same in our case study, if a running date in a year is a holiday declared by the government, then the behaviour in the indoor will be changed due to this announcement. Hence, when all these features are fused, then it will be useful to understand the overall internal behaviour for that specific date. Seasonality alone cannot decode such an event. The trend takes a similar path as well. Now, the part of the data, that cannot be described using trend and seasonality is an error. To explain this error part, we need to consider all possible context information such that the maximum error can be explained. Here, different context features are working together to explain the maximum of the error portion and is the case of

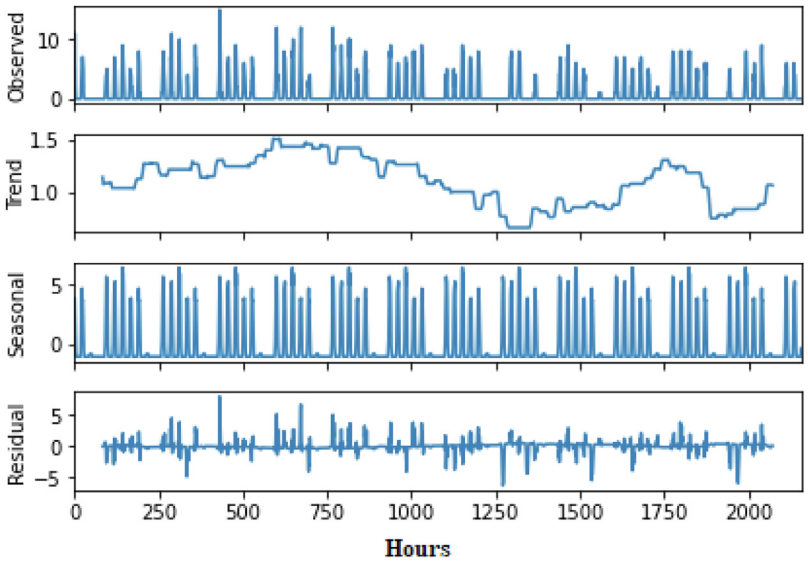

Fig. 3. ETS Decomposition of the Occupancy Headcount Data.

complementary fusion. Different contexts are useful to explain the different unexplained part of the error which helps the overall system to predict the target more accurately, thus increasing overall predictivity of the system. This approach is particularly useful for the realization of our CatBoost model.

After creating the model ready data, we try to shrink the feature space for the betterment of a machine learning model in general where other popular models are considered for performance comparison due to presence of more number of features. The goal of doing so is to identify important features for the researchers. These selected features are recorded from the indoor. This reduced set of model ready data is applicable for these general machine learning models. The reason is that not all machine learning models are designed to handle a large number of features efficiently. However, our prime focus for finding a selection of features here is purely for identifying the most significant features for this general machine learning approach. Here, applying these four filtering techniques reduces the number of features present in the data from 33 to 10 which reduces the feature space 30%. It is found that removing extra features from the dataset, reduces the data dimension but not hampering the model fittings which are measured here in terms of R square metric with 10 fold cross-validation. Since the problem is a regression problem, R square is used here as the primary pre-set evaluation. Here, by considering all the matrices, i.e., all 33 features for model fittings, we were getting R square value of 94.9% whereas after applying filter method the percentage of variance explained becomes 94.2%. These selected 10 features are *AQI*, CO_2, *Temperature*, *Humidity*, *holiday*, *hour*, *fan_status*, *weather_context*, *dt_day* and *day_of_week*.

To work with the selected 10 features, AQMD device of AirSense has been used along with digital signal sensing for relay module. Here required physical sensors are embedded in the AQMD Device which senses *AQI*, CO_2, *Temperature*, *Humidity* using MQ135, MG 811, DHT11 sensors. Here, other information is collected using free APIs. For example, current weather of any city is obtained using OpenWeathermap API in Python. Time is collected from the system and using holidays library (Python library for generating country, province and state specific sets of holidays on the fly), holiday related data has been received. Here, information about status of the fan is received by sensing the digital pin of Arduino which has been designed to control the status of relay module that controls the fan. Effect of considering these selected features are discussed in the result section further.

Finally, the CatBoost model is applied for the realization of the OccupancySense model on the model ready data (after applying data fusion) for Occupancy Detection, Occupancy Density Prediction, and Occupancy Headcount Prediction and compared those with other popular models in their specific problem domain which is reflected in the result section.

5. Implementation and result

To implement the OccupancySense model, python is used. This implementation includes data cleaning, feature engineering, feature extraction, feature selection as well as time series forecasting model preparation. To implement the same, we have utilized statmodels library for statistical analysis, machine learning library scikit-learn, mlxtend for data science tasks and CatBoost library along with common modules of python. This CatBoost is a fast, scalable, high-performance gradient boosting on decision trees library, used for ranking, classification, regression and other ML tasks [18]. In this work, CatBoost is the core of OccupancySense and it is utilized in its full extent for classification as well as regression also.

After creating the model ready data, important features are shortlisted using filter method which is cross-verified by utilizing mlxtend library's SequentialFeatureSelector by applying step forward feature selection technique along with the scikit-learn library. This is done to identify the most important features that are contributing most towards the model's performance. These features are cross verified here with the result of features selected by the filter method of feature selection. Rest of the features are also important but the significance is less than the first ten features as mentioned previously.

Next, the CatBoost model is compared along with some other popular machine learning models for both binary and multiclass classification as well as for regression purpose. For this comparison purpose, we have considered Logistic Regression (LR), Naïve Bayes (NB), Decision Tree (DT) and Random Forest (RF), Support Vector Machine (SVM) for classification and Multiple Linear Regression (MLR), Decision Tree (DT), Random Forest (RF), Support Vector Regressor (SVR) for regression.

For validity testing of the model, different matrices are used to measure the accuracy of the classification and regression prediction. For classification error metrics utilized here are Accuracy (equivalent to Jaccard score), Log-loss, ROC-AUC, F1 Score

Table 1
Performance comparison of different classifiers when 3 weeks model ready data is considered for occupancy detection (classification problem) without context information (considering multi-sensor fusion based on *IAQ* only).

Classification models ⇒						
Error Metric ⇓	Logistic regression	Naïve Bayes	Kernel support vector machine	Decision tree	Random forest	CatBoost
Accuracy	0.967	0.937	0.972	0.954	0.972	0.975
Log-loss	1.12	2.189	0.961	1.602	0.961	0.854
ROC-AUC	0.956	0.94	0.962	0.953	0.964	0.964
F1 score	0.937	0.885	0.946	0.914	0.946	0.952
10F CV score (%)	96	93	96	95	96	97
10F CV Std. deviation (%)	0.85	1.79	1.08	1.84	1.37	0.96

whereas, for regression, corresponding error matrices are Mean Square Error (*MSE*), Mean Absolute Error (*MAE*), Root Mean Square Error (*RMSE*) and Mean Absolute Percentage Error (*MAPE*), to find the effectiveness of individual model settings. Furthermore, 10 Fold Cross-Validation (10F CV) is used here to check the model's validity for both classification as well as or regression to estimate the skill of the machine learning model on unseen data.

First, we estimate how accurately OccupancySense can estimate indoor occupancy. For this, our model will provide a binary output, i.e., whether the room is empty (0) or not (1). The same is checked by us for both with only *IAQ* and also *IAQ* with Context and compared with performance of other machine learning models. The result of the first case (prediction based on *IAQ* only) is reflected in the Table 1. Here, we find that CatBoost outperforms other relevant models in the machine learning domain. This result is generated by utilizing the first 10 week's data for training purpose and next 3 weeks data for prediction purpose and the result is based on an hourly level granularity. The accuracy reflected in 1 that is achieved here by taking an average of 3 week's prediction accuracy's average. This signifies the robustness of the model.

Next, in the second case, i.e., by considering the entire context feature set along with *IAQ*, CatBoost achieves 99.85% occupancy detection accuracy along with 100% 10 fold cross-validation accuracy along with 0.15% standard deviation which is highest in the category as mentioned in Table 3 . Here, we have noted that corresponding Log-loss, ROC-AUC and F1 Score are 0.5, 0.999 and 0.997 respectively. Rest of the model are also performing similar impressive accuracy. Here, by considering all the 33 features, we get approximately 99–99.5% (at maximum) accuracy with all the above-mentioned classification models. The reason for achieving this accuracy with other models is because of considering the context information factor along with the *IAQ*. However, for result visualization purpose (to closely observe how the proposed model is performing), we have shown three days of occupancy detection and prediction of OccupancySense is reflected in Fig. 4 below. Based on our study, CatBoost performs better than other machine learning models even when limited numbers of context features are considered. Hence, our default choice is CatBoost model for the OccupancySense as it performs overall well in all the situations. This signifies that using this CatBoost model we can achieve almost zero error in the prediction of occupancy detection which is the best possible achievement in this domain. What we understand is that context plays the most crucial role here. Hence, the inclusion of information on the context data from the non-intrusive sensors will play a major role in occupancy detection in the near future.

After this, the model is tested for predicting occupancy density category. Before predicting the headcount in a particular room, this occupancy density prediction is beneficiary for the local admin as it informs how the room is congested in a broadway. This is often useful as it signals us about the future *IAQ* change in the room as well as other intrinsic information about the room. This prediction of occupancy density is a generalized technique that can be used for different room sizes as well. Here, the occupancy density is categorized to 5 different levels for all the indoors i.e., Very Low (0), Low (1), Medium (2), High (3) and Very High (4)- which are constant. For this particular case study, occupancy density ranges with headcount can be referred as density_category (headcount) format as '0' (0–2), '1' (3–5), '2' (6–9), '3' (10–12) and '4' (13–15) respectively. It is found, using OccupancySense (CatBoost Classifier) indoor occupancy density can be predicted with an average of 95.6% when tested on three weeks data which is again the category highest concerning previous works of occupancy density prediction as discussed in the related work section before. Because of the use of ordered boosting intrinsically, CatBoost outperforms other multiclass classification models here as well. The result of Catboost model for occupancy density prediction is reflected in Fig. 5 below. Here, during the test data collection duration, numbers of occupants present in the lab are relatively low, i.e., 6 occupants as compared to lab's capacity, i.e., 14 (highest headcount during this entire data collection duration). Hence, only three categories can be visualized (i.e., 0, 1 and 2) in Fig. 5. Also, note, for the result as shown in the figure, all the features have been considered for achieving the above-mentioned accuracy for occupancy density prediction. This again signifies the effectiveness of context information in occupancy density prediction. However, without context information, using the environmental sensor fusion alone reaches the Occupancy density prediction accuracy up to 92.8%, i.e. 2.8% accuracy is reduced because of not-considering the context information.

Finally, its time to predict the headcount. This is the crucial part of the experiment. This experiment is based on non-intrusive sensor data, hence preserves privacy. Here, all the collected features, i.e., *IAQ* as well as context, are utilized for prediction. From the experiment, it is found that OccupancySense predicts headcount with 93.2% accuracy and having *RMSE* value of 0.54 for three-day hourly prediction average (fraction of a human being) which is also the category highest. It is also noted by us that if context information is not considered, then the 10 fold cross-validation's accuracy reduces and gives us 89.3% accuracy with *RMSE* 0.75. This signifies that context information plays a very crucial role in terms of accuracy for both occupancy detection and prediction. Further study in this regard reveals that even with the increase of the test case study duration, i.e., starting from a single day to three weeks, the average performance of CatBoost model constantly predict better than the rest of the models. Comparison with different models along with the CatBoost is reflected in Table 2 below.

Here, OccupancySense is used for predicting occupancy count for various time durations, i.e., starting from a single day to three weeks. In this model, regression error metric *MSE*, *MAE*, *RMSE* and *MAPE* are considered for different forecasting durations. Performance of other regression models are also considered here for reference. Here, we have considered various time durations to compare performance of different machine learning model's performance with our proposed approach. This also helps us to understand how the performance of a machine learning model changes with different forecasting durations. In this study, four different durations i.e., single day, three days, seven days and

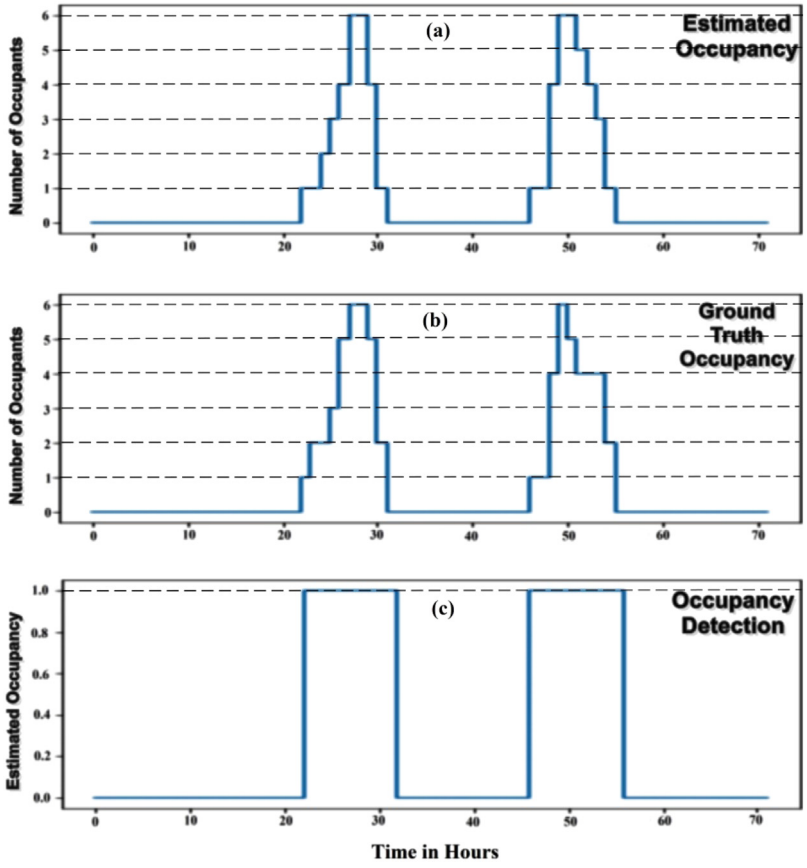

Fig. 4. Occupancy detection and prediction using OccupancySense (Using CatBoost Classifier and Regressor) by utilizing sensor fusion when tested on 3 days data (a) Occupancy in terms of headcount prediction (b) Ground Truth Occupancy count and (c) Occupancy detection for the mentioned duration.

Table 2
Performance comparison of occupancy headcount prediction for 3 week's average (regression problem) with context information (considering multi-sensor fusion based on *IAQ* along with static and dynamic context).

Regression models ⇒					
Error metric ⇓	Multiple linear regression	Kernel support vector regression	Decision tree regression	Random forest regression	CatBoost regression
10F CV score	0.868	0.901	0.863	0.924	0.929
10F CV Std. deviation	0.018	0.019	0.044	0.026	0.019
MAE	0.549	0.342	1.89	1.892	0.256
MSE	0.837	0.593	1.062	0.486	0.474
R^2	0.848	0.893	0.807	0.912	0.949
RMSE	0.915	0.77	1.031	0.697	0.672

fourteen days are considered for forecasting occupancy count in the indoor as shown in Fig. 6 below.

After analysing all the models considered here, we have found that the performance of RF and CatBoost are comparable in terms of *RMSE* but when it is tested with 10 fold cross-validation, CatBoost Outperforms even the Random Forest also. It is also noticed that even other popular machine learning models which are so-called, not so advanced, those are also performing pretty well. The reason is that getting all the context information along with *IAQ* from the training data and the same is reflected in the performance comparison graph above as well. This is showing us that "Data is the new fuel". This signifies that if one can collect proper data, prepare well the same for model fittings, then even standard models (well established and well known) will perform as compared to an advanced state of the art models. This signifies that if the correct data is merged with a proper model by considering all the facts attached to the data, then that is a blessing. Here, the same thing is done by us. It is also identified here that the new fuel is *IAQ* and context data features for our this specific problem. Not only this, based on the requirement, it is also identified the machine learning model which satisfies the need of the problem as well as data, is CatBoost model. All the results discussed here, in this result section, are proof of this fact. Here, in all the aspects, CatBoost is superior to the other relevant models, in this particular context. This CatBoost model outperforms all the other models in terms of accuracy. So, we can say that the Catboost model successfully classifies all the test cases and predicts indoor headcount most accurately which is backed by Fig. 6(b).

Thus, from the above discussion, we found that CatBoost is the most effective model for this specific problem. Now, to clearly understand the effect of context data, and the usefulness of feature selection, we have compared performance of this proposed OccupncySense model with various input data, e.g., only *IAQ*, *IAQ* along with all the context data and only selected features (i.e., AQI, CO_2, Temperature, Humidity, holiday, hour, fan status, weather context, date of the month and day of the week) from the model ready data. From the experiment, it is found that our proposed model performed best (both for occupancy detection and prediction) when all the features are considered, i.e., *IAQ*

Table 3
Performance comparison of OccupancySense with different feature set for three week's average.

Problem⇒	Occupancy detection (Classification)			Occupancy prediction (Regression)		
Selected features for OccupancySense ⇓	Accuracy	F1 score	10F CV score	R^2	*RMSE*	10F CV score
IAQ	97.5	0.952	97	0.914	0.75	89.3
Selected features	99.3	0.987	98.4	0.942	0.702	91.8
IAQ + Context	100	0.997	99.85	0.949	0.672	92.9

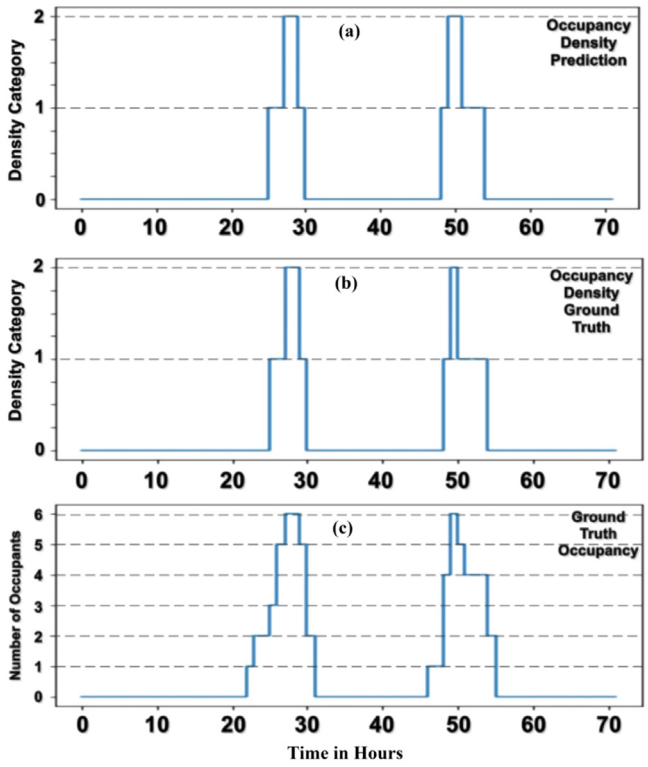

Fig. 5. Occupancy density prediction using OccupancySense (Using CatBoost Classifier) by utilizing sensor fusion when tested on 3 days data (a) Occupancy density prediction (b) Ground Truth Occupancy density and (c) ground truth Occupancy.

along with all the collected context. We also note that using environmental sensing features (*IAQ*), this model is showing the minimum accuracy (For both occupancy detection and prediction). However, selected features are actually creating a balanced tradeoff. Details of the same is reflected in Table 3 below where we have shown the comparative study for different performance metrics for the proposed OccupancySense model for continuously 3 weeks long forecasting with different input features.

Hence, we understand the importance of proper context data in a machine learning model to perform better. So, by looking at the performance, it can be said that this OccupancySense model has the potential to be utilized in the industry level solution further.

6. Conclusion

Because of the privacy issue, a non-intrusive yet highly accurate solution is required for the Occupancy detection and prediction problem. This is a proven fact that even highly correlated single sensor cannot do this task properly. Hence to achieve satisfactory results which are actually a need of time nowadays, fusing multiple sensor readings could be even more productive. Here, environment sensor fusion can play a crucial role. Hence, indoor air quality can be a good predictor for this problem. However, by digging deeper, it is found that these *IAQ* is depending on multiple factors, i.e., static as well as the dynamic context of that specific room. This thought process leads to a very exciting open research challenge of occupancy detection and prediction. This research helps to accurately predict occupancy in a non-intrusive but in an accurate manner. Here, we have used IoT for *IAQ* information collection and similarly participatory sensing for context information collection. Using this *IAQ* and context information, we have achieved an accuracy of 99.85% in terms of occupancy detection, 95.6% for occupancy density prediction and 92.9% for predicting the headcount in the room after taking the average value for the forecasting for 3 weeks long. It is also found that CatBoost is the most appropriate model to use as the core of the OccupancySense design. This model is robust, can handle more features (in future) easily without an issue. Since the CatBoost model is used as the core of OccupancySense, our proposed model inherits the features of the CatBoost. Hence, the functionality that the CatBoost model supported, is also supported by our OccupancySense model as well. It outperforms other models in this domain because of the inclusion of context information with *IAQ* which proves that data is the new fuel now. We can also understand that the importance of context information is huge in present-day applications as there is the scope of improvement in almost all smart city-related applications by merging context data along with present smart city application models. However, this work can be further tested more with more different places dataset. Here, the data is collected by utilizing IoT as well as participatory sensing where human volunteers provide us with accurate context information. In future, our plan is to make the whole system automatic by utilizing IoT [23,24], such that the context information acquirement system becomes automatic and to make this solution industry-ready. Along with this, in future we plan to design an industry ready energy efficient and healthy HVAC system by utilizing OccupancySense.

CRediT authorship contribution statement

Joy Dutta: Conceptualization, Investigation, Methodology, Software, Validation, Formal analysis, Resources, Data curation, Writing – original draft, Writing – review & editing, Visualization. **Sarbani Roy:** Supervision, Conceptualization, Investigation, Writing – review & editing, Visualization, Project administration, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

The research work of Joy Dutta is funded by "Visvesvaraya PhD Scheme, Ministry of Electronics & IT, Government of India". This research work is also supported by the project entitled "Participatory and Realtime Pollution Monitoring System For Smart City", funded by Higher Education, Science & Technology and Biotechnology, Department of Science & Technology, Government of West Bengal, India.

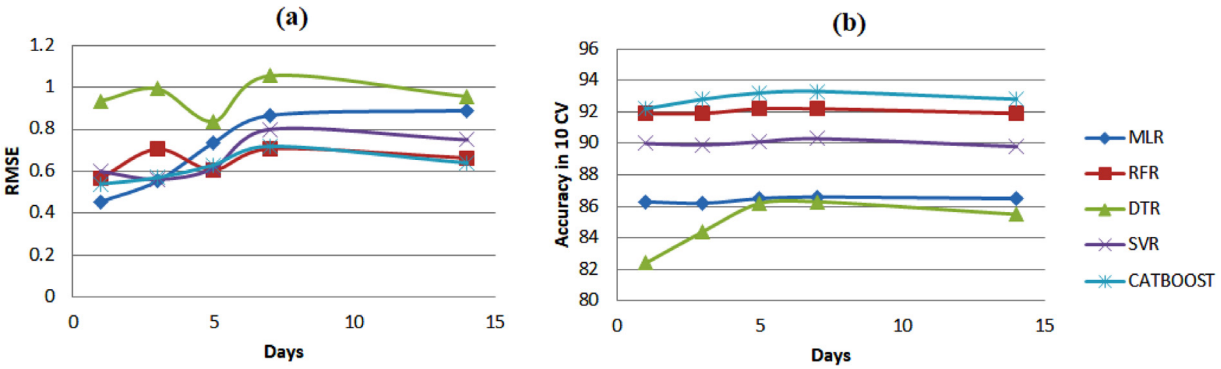

Fig. 6. Performance comparison different machine learning models for Occupancy Prediction (headcount) using (a) *RMSE* (b) 10 fold Cross-Validation for different forecasting time durations.

References

[1] B.N.A. Kankaria, S.K. Gupta, Indoor air pollution in India: Implications on health and its control, Indian J. Commun. Med. Off. Publ. Indian Assoc. Prevent. Soc. Med. 39 (4) (2014) 203–207, http://dx.doi.org/10.4103/0970-0218.143019.

[2] J. Dutta, S. Roy, IndoorSense: context based indoor pollutant prediction using SARIMAX model, Multimedia Tools Appl. 80 (13) (2021) 19989–20018, http://dx.doi.org/10.1007/s11042-021-10666-w.

[3] J. Dutta, S. Roy, Indoor air pollutant prediction using time series forecasting models, in: Advances in Intelligent Systems and Computing, Vol. 1286, in: Emerging Technologies in Data Mining and Information Security, Springer, Singapore, 2021, pp. 6639–6649, http://dx.doi.org/10.1007/978-981-15-9927-9_48.

[4] M. Jin, N. Bekiaris-Liberis, K. Weekly, C.J. Spanos, A.M. Bayen, Occupancy detection via environmental sensing, IEEE Trans. Autom. Sci. Eng. 15 (2) (2018) 443–455, http://dx.doi.org/10.1109/TASE.2016.2619720.

[5] M. Amayri, A. Arora, S. Ploix, S. Bandhyopadyay, Q.-D. Ngo, V.R. Badarla, Estimating occupancy in heterogeneous sensor environment, Energy Build. 129 (2016) 46–58, http://dx.doi.org/10.1016/j.enbuild.2016.07.026, URL http://www.sciencedirect.com/science/article/pii/S0378778816306223.

[6] C. Jiang, M.K. Masood, Y.C. Soh, H. Li, Indoor occupancy estimation from carbon dioxide concentration, Energy Build. 131 (2016) 132–141, http://dx.doi.org/10.1016/j.enbuild.2016.09.002, URL https://www.sciencedirect.com/science/article/pii/S0378778816308027.

[7] A. Szczurek, M. Maciejewska, A. Wyłomańska, R. Zimroz, G. Żak, A. Dolega, Detection of occupancy profile based on carbon dioxide concentration pattern matching, Measurement 93 (2016) 265–271, http://dx.doi.org/10.1016/j.measurement.2016.07.036, URL https://www.sciencedirect.com/science/article/pii/S0263224116303955.

[8] S. Zikos, A. Tsolakis, D. Meskos, A. Tryferidis, D. Tzovaras, Conditional random fields - based approach for real-time building occupancy estimation with multi-sensory networks, Autom. Construction 68 (2016) 128–145, http://dx.doi.org/10.1016/j.autcon.2016.05.005, URL https://www.sciencedirect.com/science/article/pii/S0926580516300851.

[9] N. Nesa, I. Banerjee, IoT-based sensor data fusion for occupancy sensing using Dempster–Shafer evidence theory for smart buildings, IEEE Internet Things J. 4 (5) (2017) 1563–1570, http://dx.doi.org/10.1109/JIOT.2017.2723424.

[10] L. Zimmermann, R. Weigel, G. Fischer, Fusion of nonintrusive environmental sensors for occupancy detection in smart homes, IEEE Internet Things J. 5 (4) (2018) 2343–2352, http://dx.doi.org/10.1109/JIOT.2017.2752134.

[11] S.H. Ryu, H.J. Moon, Development of an occupancy prediction model using indoor environmental data based on machine learning techniques, Build. Environ. 107 (2016) 1–9, http://dx.doi.org/10.1016/j.buildenv.2016.06.039, URL https://www.sciencedirect.com/science/article/pii/S0360132316302463.

[12] M. Esrafilian-Najafabadi, F. Haghighat, Impact of occupancy prediction models on building hvac control system performance: application of machine learning techniques, Energy Build. 257 (2022) 111808, http://dx.doi.org/10.1016/j.enbuild.2021.111808.

[13] Y. Yang, Y. Yuan, T. Pan, X. Zang, G. Liu, A framework for occupancy prediction based on image information fusion and machine learning, Build. Environ. 207 (2022) 108524, http://dx.doi.org/10.1016/j.buildenv.2021.108524.

[14] F. Castanedo, A review of data fusion techniques, Sci. World J. (2013) http://dx.doi.org/10.1155/2013/704504, URL https://www.hindawi.com/journals/tswj/2013/704504/.

[15] A. Bommert, X. Sun, B. Bischl, J. Rahnenführer, M. Lang, Benchmark for filter methods for feature selection in high-dimensional classification data, Comput. Statist. Data Anal. 143 (2020) 106839, http://dx.doi.org/10.1016/j.csda.2019.106839, URL http://www.sciencedirect.com/science/article/pii/S016794731930194X.

[16] P. Liudmila, G. Gleb, V. Aleksandr, D.A. Veronika, G. Andrey, CatBoost: Unbiased boosting with categorical features, in: Proceedings of the 32nd International Conference on Neural Information Processing Systems, in: NIPS'18, Curran Associates Inc., Red Hook, NY, USA, 2018, pp. 6639–6649.

[17] Y. Pan, L. Zhang, Data-driven estimation of building energy consumption with multi-source heterogeneous data, Appl. Energy 268 (2020) 114965, http://dx.doi.org/10.1016/j.apenergy.2020.114965, URL http://www.sciencedirect.com/science/article/pii/S0306261920304773.

[18] Yandex, CatBoost: A high-performance open source library for gradient boosting on decision trees, 2021, URL https://catboost.ai/, [Online accessed 02-February-2021].

[19] J. Dutta, F. Gazi, S. Roy, C. Chowdhury, AirSense: Opportunistic crowd-sensing based air quality monitoring system for smart city, in: 2016 IEEE SENSORS, 2016, pp. 1–3, http://dx.doi.org/10.1109/ICSENS.2016.7808730.

[20] J. Dutta, C. Chowdhury, S. Roy, F. Gazi, A. Middya, Towards smart city: sensing air quality in city based on opportunistic crowd-sensing, in: Proceedings of the 18th International Conference on Distributed Computing and Networking, in: ICDCN '17, Association for Computing Machinery, New York, NY, USA, 2017, http://dx.doi.org/10.1145/3007748.3018286.

[21] Airveda Technologies Private Limited, Airveda: Air quality monitors, 2021, URL https://www.airveda.com/, [Online; accessed 24-January-2021].

[22] J. Dutta, S. Roy, Indoor air pollutant prediction using time series forecasting models, in: 2nd International Conference on Emerging Technologies in Data Mining and Information Security (IEMIS 2020), Springer Nature, 2020, http://dx.doi.org/10.1007/978-981-15-9927-9_50.

[23] J. Dutta, S. Roy, Iot-fog-cloud based architecture for smart city: prototype of a smart building, in: 2017 7th International Conference on Cloud Computing, Data Science Engineering - Confluence, 2017, pp. 237–242, http://dx.doi.org/10.1109/CONFLUENCE.2017.7943156.

[24] J. Dutta, Y. Wang, T. Maitra, S.H. Islam, B.S. Rawal, D. Giri, ES3B: Enhanced security system for smart building using IoT, in: 2018 IEEE International Conference on Smart Cloud (SmartCloud), 2018, pp. 158–165, http://dx.doi.org/10.1109/SmartCloud.2018.00034.