OCCUPANCY ESTIMATION USING ENVIRONMENTAL SENSORS

Project Report submitted to

Department of Computer Science and Engineering

Dr. B.C. Roy Engineering College, Durgapur, WB

for the partial fulfillment of the requirement to award the degree

of

Bachelor of Technology

in

Computer Science and Engineering

by

Saptarshi Ghosh 12000120068

Sumit Dhar 12000120073

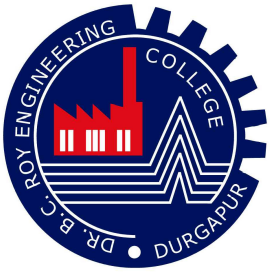
Abhik Mandal 12000120096

under the guidance

of

Supervisor: Dr. Arindam Ghosh, Associate Professor, Dept. of CSE, BCREC

Prof. Ruma Ghosh, Assistant Professor, Dept. of CSE, BCREC

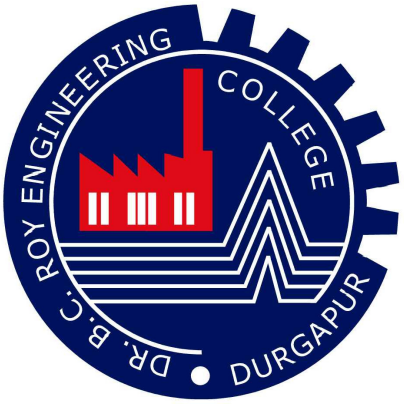


DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

DR. B.C. ROY ENGINEERING COLLEGE, DURGAPUR, WB

May, 2024

OCCUPANCY ESTIMATION USING ENVIRONMENTAL SENSORS



Saptarshi Ghosh

Sumit Dhar

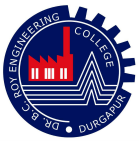
Abhik Mandal

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
DR. B.C. ROY ENGINEERING COLLEGE, DURGAPUR, WB

May, 2024

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

DR. B.C. ROY ENGINEERING COLLEGE, DURGAPUR, WB



DECLARATION

We the undersigned, hereby declare that our B.Tech final year Project entitled, **"Occupancy Estimation using Environmental Sensors"** is original and is our own contribution. To the best of out knowledge, the work has not been submitted to any other Institute for the award of any degree or diploma. We declare that we have not indulged in any form of plagiarism to carry out this project and/or writing this project report. Whenever we have used materials (data, theoretical analysis, figures, and text) from other sources, we have given due credit to them by citing in the text of the report and giving their details in the references. Finally, we undertake the total responsibility of this work at any stage here after.

Signature of the Students

Saptarshi Ghosh (12000120068)

Sumit Dhar (12000120073)

Abhik Mandal (12000120096)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

DR. B.C. ROY ENGINEERING COLLEGE, DURGAPUR, WB

RECOMMENDATION

This is to recommend that the work undertaken in this report entitled, **"Occupancy Estimation using Environmental Sensors"** has been carried out by **"Saptarshi Ghosh, Sumit Dhar, Abhik Mandal"** under my/our supervision and guidance during the academic year 2023-24. This may be accepted in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology (Computer Science and Engineering).

Dr. Arindam Ghosh

& Prof. Ruma Ghosh

Associate Professor

& Assistant Professor,

Department of CSE

Dr. Arindam Ghosh

Head of Department,

Department of CSE

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

DR. B.C. ROY ENGINEERING COLLEGE, DURGAPUR, WB

CERTIFICATE

This is to certify that, **Saptarshi Ghosh**, **Sumit Dhar** and **Abhik Mandal**, students in the Department of Computer Science & Engineering, worked on the project entitled **"Occupancy Estimation using Environmental Sensors".**

I hereby recommend that the report prepared by them may be accepted in partial fulfillment of the requirement of the Degree of Bachelors of Technology in the Department of Computer Science and Engineering, Dr. B. C. Roy Engineering College, Durgapur.

.....

.....

Examiners

.....

Dr. Arindam Ghosh

& Prof. Ruma Ghosh

(Supervisor)

.....

(Project Co-ordinator)

Date: .....

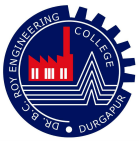
Place:

Dr. Arindam Ghosh

(HOD, CSE)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

DR. B.C. ROY ENGINEERING COLLEGE, DURGAPUR, WB



ACKNOWLEDGEMENT

It is our privilege to express our sincere regards to our project supervisor, Dr. Arindam Ghosh & Prof. Ruma Ghosh, for valuable inputs, able guidance, encouragement, whole-hearted cooperation, and constructive criticism throughout our project.

We deeply express our sincere thanks to the Head of Department, Dr. Arindam Ghosh, for encouraging and allowing us to present the project on the topic **"Occupancy Estimation using Environmental Sensors"** at our department premises for partial fulfillment of the requirements leading to the award of the B.Tech. Degree.

Furthermore, we would also like to acknowledge the crucial role of our teachers, whose instructions and guidelines acted as a foundation stone for this project.

Saptarshi Ghosh

Sumit Dhar

Abhik Mandal

Abstract

Occupancy can greatly affect the quality of air in a room. It has been found that high Occupancy levels with no proper ventilation and outdoor air pollution can significantly impact the occupant comfort, and productivity within built environments as it can lead to various health issues. In this work, we have studied the relationship of Occupancy with the various environmental parameters that includes CO_2, PM_1, $PM_{2.5}$, PM_{10}, $TVOC$, Temperature, Relative Humidity and Sound that affects Occupancy significantly. The real-time data acquisition has been done using an custom device which was developed by our team. This relationship helped us to find the primary parameters that will be needed for estimation and development of Machine Learning Classification models for accurate prediction of the Occupancy class. The occupancy estimation levels have classification accuracy in the range of 91% to 99% for random sampling. Finally, results show that multiple environmental sensor data performed well in predicting occupancy levels.

Keywords: Occupancy, Indoor air quality, Carbon Dioxide, Particulate matter, Coarse particles, Arduino

Contents

List of Figures

List of Tables

CHAPTER 1

Introduction

1.1 Overview

Occupancy simply refers to the number of students or individuals present in particular institution at any particular time, whether it's a school, office, hospital, hotel, or any other type of facility. As Occupancy in a classroom can greatly affect the Indoor Air Quality (IAQ) levels in a classroom, which plays a critical role in the health, comfort, and productivity of both students and teachers so managing it is very much essential. As students and teachers gather, the first noticeable change is often the rise in CO_2 levels. When a classroom is occupied, there is an increase in respiratory emissions, including CO_2 and moisture, due to the occupants' breathing. This elevation serves as a tangible marker of occupancy and can provide insights into ventilation adequacy. As occupancy increases, CO_2 levels in the classroom rise and thus elevated CO_2 levels can indicate poor ventilation thus affecting the

comfort and alertness of occupants. Beyond CO_2, occupancy affects the dispersion of respiratory aerosols—tiny droplets expelled through activities such as speaking, coughing, and sneezing. In a crowded classroom, these aerosols become more pervasive, carrying with them the potential for airborne transmission of pathogens and pollutants. The increased density of occupants heightens the risk of contagion, making proper ventilation and filtration essential for maintaining a healthy indoor environment.

The quality of air within a classroom can be highly variable, influenced by factors such as occupancy levels, ventilation, outdoor air pollution, building materials, and even the activities conducted within the space. The collective warmth generated by occupants further complicates the indoor climate. As more individuals fill the space, their metabolic heat contributions combine, impacting thermal comfort levels. In densely populated classrooms, this can lead to elevated temperatures, discomfort, and increased energy demands for cooling. Proper ventilation and temperature control systems are essential for managing indoor temperatures and ensuring a conducive learning environment for all occupants.

Occupancy levels impact IAQ, and IAQ can influence occupancy comfort and well-being. Proper ventilation is essential for managing indoor air quality in classrooms with varying occupancy levels. As the number of occupants increases, so too does the demand for fresh outdoor air to dilute indoor pollutants and maintain acceptable indoor air quality levels. Adequate ventilation rates are critical for removing contaminants, controlling CO_2 levels, and promoting occupant health and comfort. Poor IAQ can result in various health issues, including respiratory

problems, allergies, and reduced cognitive performance.

1.2 Significance and Applications of the Project

The significance of our project work includes Privacy Control which is a main concern in the various facial recognition technique implementations. Besides contactless, our project can also make sure that there is no chances of any unauthorised entry in the organization which was a great concern in the RFID implementation technique as RFID technique requires a tag for the person's allowance in the organisation, which may be lost. Studying the variability of air quality and developing prediction models have far-reaching implications. By understanding and forecasting air quality, we can take proactive steps to mitigate pollution and create healthier and more sustainable living environments. Here are some key areas where this knowledge is valuable:

1. **Security and Surveillance:** Occupancy sensors play a crucial role in building security systems by detecting unauthorized access or occupancy in restricted areas. Real-time monitoring of occupancy levels allows for timely response to security threats and enhances overall safety and security.

2. **Healthcare:** In healthcare settings, this work can assist in monitoring patient occupancy in rooms and common areas, optimizing facility management, and enhancing patient care through environmental adjustments.

3. **Education Environments:** In educational settings, this work can contribute to optimizing classroom occupancy, helping educators adapt teaching strategies

based on real-time data, and improving overall learning environments.

4. **Hospitality Industry:** Hotels and resorts can benefit from this work by optimizing room occupancy, streamlining housekeeping services, and enhancing guest experiences through personalized environmental controls.

5. **Workplace Management:** Employers can use this work to optimize office space utilization, facilitate flexible work arrangements, and create a more comfortable and productive work environment based on real-time occupancy data.

6. **Environmental Monitoring:** Occupancy data can be integrated with other environmental sensors to monitor indoor air quality, temperature, and humidity levels. This holistic approach to environmental monitoring allows for better understanding of indoor conditions and optimization of occupant comfort and well-being.

7. **Retail Analytics:** Retailers use occupancy data to analyze customer behavior and optimize store layouts and product placement. By understanding foot traffic patterns and dwell times, retailers can enhance customer experience, increase sales, and improve operational efficiency.

8. **Traffic Management:** In transportation hubs such as airports and train stations, occupancy sensors help monitor passenger flow and optimize resource allocation. This information can be used to improve crowd management, reduce congestion, and enhance overall efficiency of transportation systems.

9. **School Safety:** In educational settings, occupancy sensors contribute to school safety measures by monitoring occupancy levels in classrooms, hallways, and common areas. This information can be integrated with security systems to enhance emergency response protocols and ensure the safety of students and staff.

10. **Hospitality Industry:** In hotels and resorts, occupancy sensors help optimize guest services and resource allocation. Hoteliers use occupancy data to manage room turnover, allocate housekeeping resources efficiently, and enhance guest satisfaction by personalizing service delivery based on occupancy status.

1.3 About the Project

We propose a device-free framework for estimating the degree of occupancy using environmental sensors in this project. The proposed approach is unobtrusive, privacy-preserving, and requires fewer computational resources. Our understanding of the recent works revealed that the application of environment sensors for estimating occupancy is significantly less explored. Therefore, we put forth the following query: ***"What should be the key parameters to estimate classroom occupancy using environment monitoring sensors accurately?"***. This project's contribution lies in extending the set of features used for occupancy estimation modeling and in optimizing the model parameters to maximize occupancy estimation accuracy.

This project assumes that some features are more important than others for approximating occupancy in a classroom. For this purpose, the proposed work focuses on assessing the occupancy of a classroom by studying the variabil-

ity of IAQ-related features. Our approach consists of developing an IoT-based environment monitoring device that would enable us in real-time data collection, including levels of CO_2, particulate matter (PM), Temperature, Relative Humidity, and acoustics in a classroom using the various pollution-detection sensors. Further, the collected contextual information, helped us to explore the relationship between the parameters and the classroom occupancy. Then, we have calculated the overall correlation between the feature parameters and the Classified Occupancy and as a result we have considered CO_2 and PM as significant factors because of their strong correlation with the occupancy. Moreover, to determine the various dependencies between the parameters and the occupancy, we have studied multiple graphical plots between them. Lastly, we have developed multiple intelligent models to predict classroom occupancy accurately.

CHAPTER 2

Related Works

Numerous sophisticated methods and detailed modelling techniques have been employed to effectively estimate the level of indoor occupancy. Notably, in their thorough research, Esrafilian-Najafabadi *et al.* [1] have proposed a range of innovative features that could potentially be utilized for the prediction of occupancy. These include closely monitoring the energy consumption of various household appliances and accurately tracking the precise locations of occupants within a building. However, these methodologies often raise significant privacy concerns, as they entail the exposure of personal locations, and their effectiveness is frequently hindered by the unavailability of comprehensive and reliable data on appliance energy consumption. Additionally, alternative approaches utilize technologies such as PIR occupancy sensors [2], which may face challenges in detecting occupants who are stationary for prolonged periods. Likewise, WiFi-based systems [3] can experience connectivity issues that result in inaccuracies in occupancy determination, particularly in com-

plex scenarios where multiple occupants are present in the same room [4]. Another technological approach involves the use of Radio Frequency Identification (RFID) based systems [3], which are particularly applied in settings such as classrooms to monitor the presence of students. Despite their utility, these RFID systems are relatively expensive and are prone to security vulnerabilities, including potential attacks or tampering [5]. In addition, Face Recognition-based methods, which identify individuals by analysing human faces through images or videos captured by sophisticated digital cameras, are also frequently employed to monitor room occupancy. Although these image-based techniques are widely used across various sectors, they are notably invasive and require substantial computational power to process the visual data effectively.

In the specialized field of device-free methods for occupancy detection, as thoroughly explored and documented in various academic reviews and studies [6, 7], the strategic deployment of environmental monitoring sensors is found to be of utmost importance. These sensors, which are specially designed to accurately measure concentrations of gases and levels of dust within an environment, are invaluable due to their discreet and non-intrusive nature that also ensures the preservation of privacy for individuals within the monitored space. The data that these sensors procure is indispensable for the meticulous monitoring of Indoor Air Quality (IAQ). Additionally, this data is instrumental in providing detailed and significant insights into the prevailing environmental conditions [8]. Taking the example of educational settings such as classrooms, it is observed that the levels of Particulate Matter (PM) can be greatly influenced not only by a variety of

human activities taking place within these spaces but also by external environmental elements [9]. Furthermore, there has been consistent evidence suggesting that spaces that are highly occupied tend to have elevated concentrations of CO_2 [10]. Moreover, in certain geographic regions, the utilization of electricity consumption data has been identified as a highly effective method for accurately identifying occupancy states. This innovative approach not only provides critical contextual information but also significantly reduces the need for employing additional, specialized occupancy sensors, thereby optimizing the efficiency of the monitoring system [11].

A range of sophisticated machine learning-based methodologies are systematically employed to automate the complex process of estimating occupancy levels in diverse environments. In their comprehensive research, Sayed *et al.* [12] effectively harness a combination of cutting-edge deep learning (DL) and transfer learning (TL) techniques, alongside an array of diverse sensors, to accurately assess and quantify occupancy levels within various settings. Concurrently, Tien *et al.* [13] explore the utilization of specific advanced features from Convolutional Neural Networks (CNNs), particularly focusing on Region-based CNN (R-CNN), which they pair with robust transfer learning strategies. This approach is applied to detect the presence and activities of individuals in an office environment, where such individuals may be engaged in a variety of actions including walking, sitting, and standing, thereby providing dynamic occupancy data. Furthermore, Khalil *et al.* [14] also apply transfer learning to enhance the precision of their occupancy estimation methods. They perform a detailed comparative analysis, compare their results with those derived from other established machine learning algorithms such

as Random Forest (RF) and Support Vector Machine (SVM). Their findings indicate that the transfer learning models they developed are particularly effective, outperforming the traditional RF and SVM models in terms of accuracy in predicting occupancy levels, thereby underscoring the superiority and effectiveness of transfer learning in this specialized field.

CHAPTER 3

Methodology

The first part of this section explains the framework. The second part summarizes the structure, working, and calibration of the environment monitoring device. The third part discusses about the gathered data and how it is prepared for machine learning.

3.1 Framework

The work is divided into three stages [Fig. 3.1]. In the first stage, we start by examining the sampling area and record the various parameters related with it such as outdoor weather, floor number, floor type, etc. After this, we check the environment monitoring device to ensure the sensor stability for accurate readings. Finally, we start gathering data of the sampling area using the monitoring device and the real-time occupancy of the area is collected with the help of a custom website developed and hosted using Python and Streamlit [Fig. 3.2].

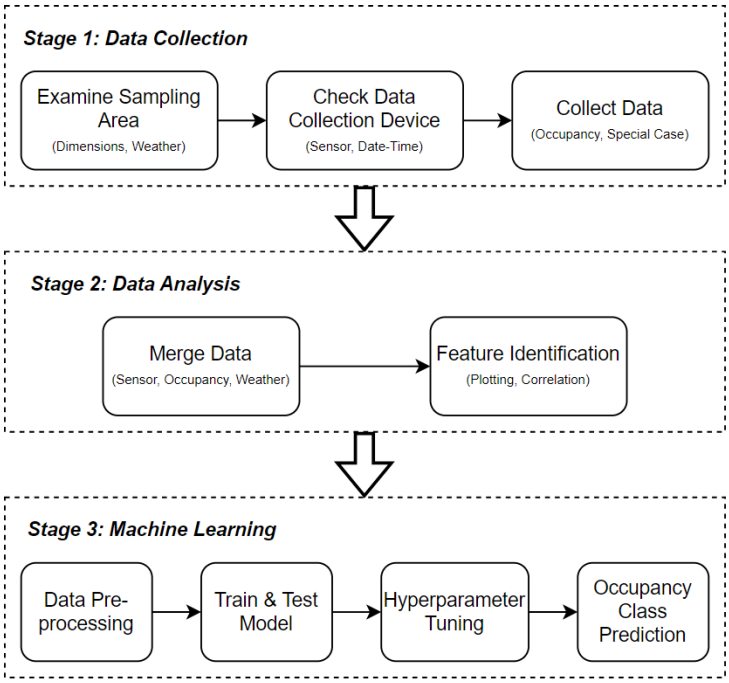


Figure 3.1: *Flowchart*

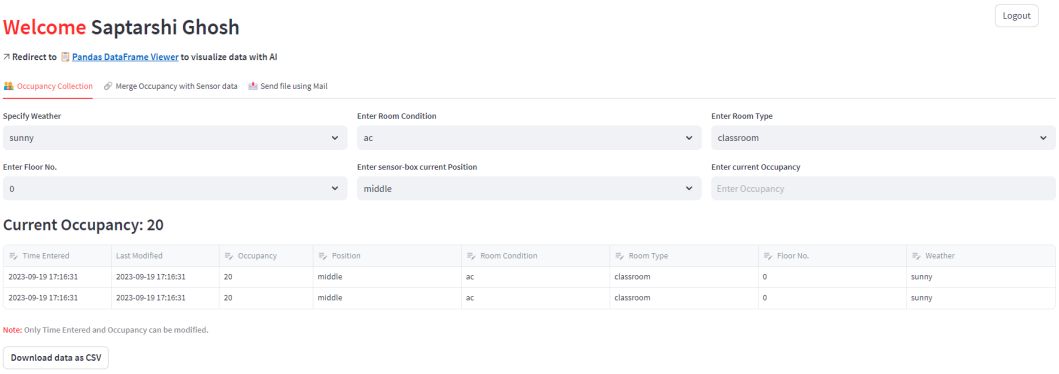


Figure 3.2: *Occupancy Collection using Website*

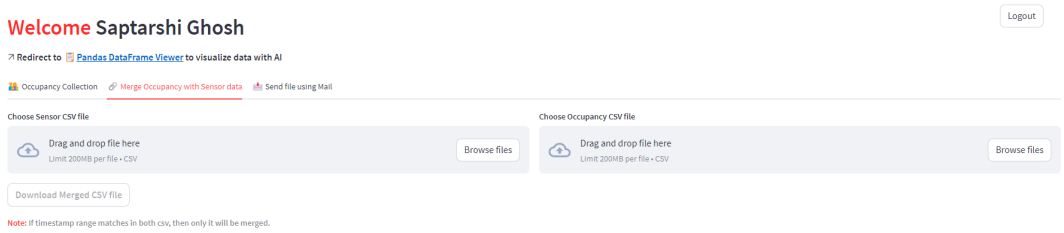


Figure 3.3: *CSV merging using Website*

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
1	Timestamp	CO (ppm)	NO2 (ppm)	CO2 (ppm)	TVOC (ppb)	PM1 (ug/m3)	PM2.5 (ug/m3)	PM10 (ug/m3)	Temperature (C)	Humidity (%)	Sound (dB)	Occupancy	Position	Room Condition	Room Type	Floor No.	Weather
2	16-Aug-2023 11:16:33	36.3	0.1	1701	0	44	65	70	26.1	14.1	72.1	26	middle	ac	lab	2	sunny
3	16-Aug-2023 11:16:38	36.1	0.1	1699	0	44	64	69	26.1	14.1	70.5	26	middle	ac	lab	2	sunny
4	16-Aug-2023 11:16:44	36.5	0.1	1695	0	44	64	69	26.1	14.1	77.2	26	middle	ac	lab	2	sunny
5	16-Aug-2023 11:16:50	34.2	0.1	1690	0	43	64	68	26.1	14.1	72.2	26	middle	ac	lab	2	sunny
6	16-Aug-2023 11:16:55	36.3	0.1	1684	0	43	64	67	26.1	14	75.3	26	middle	ac	lab	2	sunny
7	16-Aug-2023 11:17:01	36.1	0.1	1678	0	43	63	66	26.1	14	73.3	26	middle	ac	lab	2	sunny
8	16-Aug-2023 11:17:06	36.1	0.1	1672	0	44	64	67	26.1	14.1	71.8	26	middle	ac	lab	2	sunny
9	16-Aug-2023 11:17:12	36.3	0.1	1668	0	43	63	66	26.1	14.1	69.5	26	middle	ac	lab	2	sunny
10	16-Aug-2023 11:17:17	36.3	0.1	1667	0	44	63	66	26.1	14	78.6	26	middle	ac	lab	2	sunny
11	16-Aug-2023 11:17:23	36.3	0.1	1666	0	43	62	65	26.1	14	72.1	26	middle	ac	lab	2	sunny
12	16-Aug-2023 11:17:29	36.1	0.1	1666	0	43	62	65	26.1	14.1	70.9	26	middle	ac	lab	2	sunny
13	16-Aug-2023 11:17:34	36.3	0.1	1666	0	44	63	67	26.1	14.1	76.7	26	middle	ac	lab	2	sunny
14	16-Aug-2023 11:17:40	36.1	0.1	1667	0	44	62	66	26.1	14.1	71.6	26	middle	ac	lab	2	sunny
15	16-Aug-2023 11:17:45	36.5	0.1	1667	0	43	61	65	26.1	14.1	76.9	26	middle	ac	lab	2	sunny

Figure 3.4: *Dataset after Merging both CSV files*

In the second stage, we download two csv files, one from the environment monitoring device which holds the data about the environmental conditions and another from the website which holds the data about the area occupancy. Then, these two csv files are merged using the same website [Fig. 3.3] along with classification of the occupancy into their respective classes as summarized in Table 3.2. The resultant dataset [Fig. 3.4] is used for further data analysis before applying any machine learning algorithm.

After identifying the most important features from the dataset, in the third stage the dataset is pre-processed to remove any invalid, incorrect or erroneous data. The resultant cleaned dataset is used to train and test multiple machine learning algorithms to predict occupancy. Furthermore, each model is hypertuned to improve their performance and accurately predict the occupancy of an area.

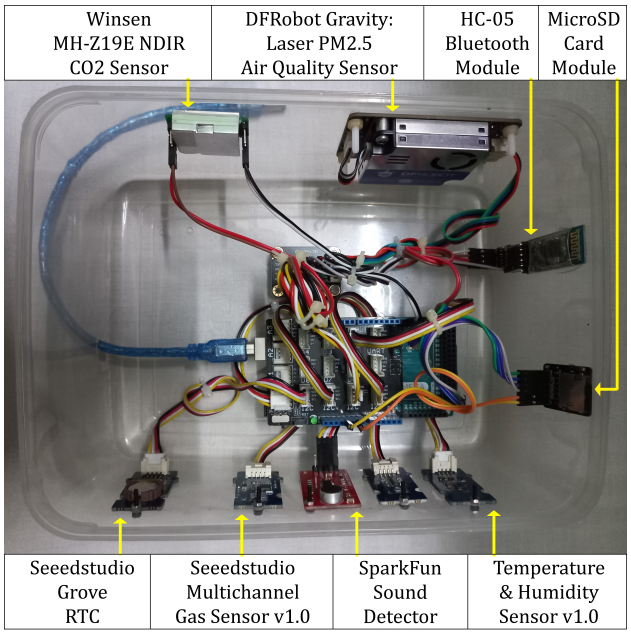


Figure 3.5: *Environmental Monitoring Device*

3.2 Sensing Module

We have assembled an environment monitoring device outfitted with various inexpensive sensors [Fig. 3.5]. This device has a Micro-SD card to store the measurements in .csv format. It also has an *HC-05 Bluetooth v4.0* module to send all device critical information to remote devices such as mobile phones/laptops. We equip it with a *SeeedStudio Grove RTC* module to track every measurement's actual date and time. The main parameters that we have considered for this work include CO_2, PM_1, $PM_{2.5}$, PM_{10}, CO, NO_2, Sound, Temperature, and Relative Humidity. All of these parameters are recorded every 5 seconds for an average duration of 1.5 hours in each sampling environment using multiple environmental sensors from various manufacturers such as *SeeedStudio*, *DFRobot*, *Winsen*, and *Sparkfun*. Table 3.1

shows detailed information about the sensors used in our experiments.

Sensor calibration: For sensor calibration, we have studied the techniques employed in prior comparable studies [15]. To maintain the least possible atmospheric pollution during calibration, sensors are factory-calibrated or calibrated using the manufacturer's program in an open environment between 12 O'clock Midnight and 6 AM. Other parameters we record before each sampling include outdoor weather at the time of sampling, floor number of the room, room dimensions, room type, room condition, board type used in the room (chalk-duster/marker), window type, and door type (of the room of experiment). Whenever there is a change in classroom occupancy, all the above are recorded in real-time. To allow the sensors to stabilize in the present sampling environment, the device is turned on ten minutes before the reading begins in each sampling. This is done to ensure steady data capture during the sampling. Since insufficient power supply can affect the working of the sensors, a portable 10000 mAh power bank is used to power the device while conducting the data collection. The device has an average current consumption rate of 200 mA. So, it can run for approximately 50 hours on a single recharge 10000 mAh power bank.

3.3 Data Processing

During the experimental data collection across multiple Air-Conditioned (AC) Labs, we have sampled cumulative data points consisting of 12071 rows. Based on the number of persons in the room, the complete dataset has been classified into uniquely identifiable classes to discover the most important feature parameters affecting occupancy detection. We then develop an accurate classification model

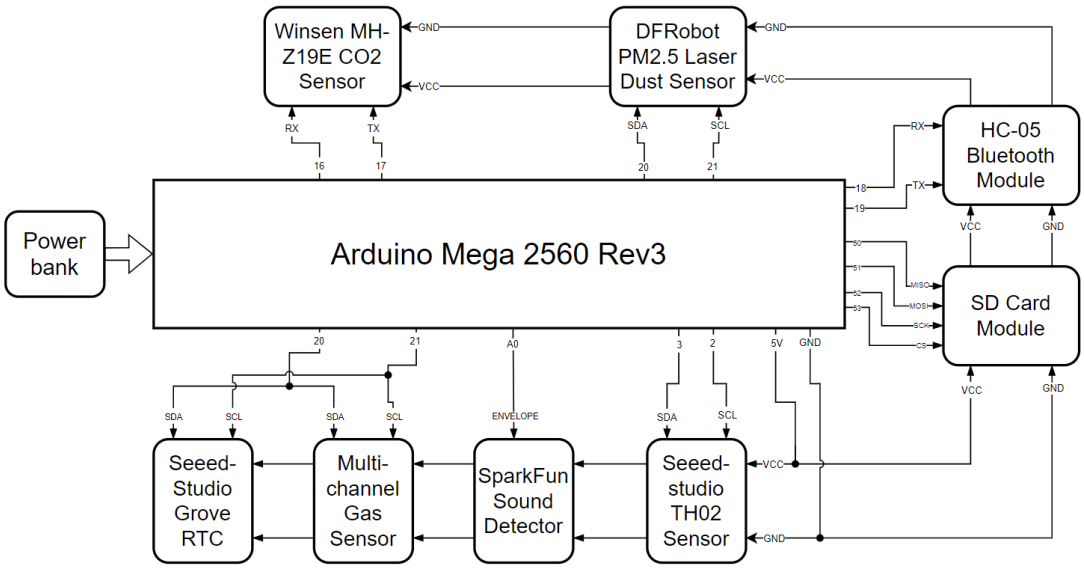


Figure 3.6: *Device Block Diagram*

Table 3.1: *Sensor Specification*

Name	Manufacturer	Sensing Pollutant	Range	Price (INR)
Winsen MH-Z19E NDIR CO2 Sensor	Winsen	CO_2	CO_2: 400-5000 ppm	1580
Gravity: Laser PM2.5 Air Quality Sensor	DFRobot	PM_1, $PM_{2.5}$, PM_{10}	PM: 0-500 $\mu g/m^3$	3299
Grove: Multichannel Gas Sensor v1.0	SeeedStudio	CO, NO_2	CO: 0-1000 ppm NO_2: 0.05-10 ppm	3600
SparkFun Sound Detector	Sparkfun	Sound	30-130 dB	1049
Grove: Temperature & Humidity Sensor v1.0	SeeedStudio	Temperature, Relative Humidity	T: 0-70 °C RH: 0-80 %	2200

Table 3.2: *Assigned Classes for Occupancy Estimation*

Class Name	Occupancy Range	Number of Rows before Pre-processing	Number of Rows after Pre-processing
Class: 0	0	165	19
Class: 1	1 to 10	3613	2428
Class: 2	11 to 20	2837	1773
Class: 3	21 to 30	1742	1401
Class: 4	31 to 40	1401	1177

based on the same. Table 3.2 summarizes each occupancy class in the dataset. Since Class 0 has a very small number of samples, we have excluded it from the train-test dataset.

1. **Relation of the Parameters with the Classified Occupancy:** At first, we calculate the overall correlation between the parameters and the Classified Occupancy [Fig. 3.7]. We take into consideration CO_2 and PM as our significant factors because of their strong positive and negative correlations with the Classified Occupancy. Further, we note that temperature and humidity positively correlate with classified Occupation, whereas sound has almost a negligible correlation. That is why we disqualify sound from consideration in our later studies.

   To illustrate the relation between different parameters and the occupancy, sample plots of CO_2 and PM_{10} are shown in [Fig. 3.8]. As we see during the beginning of the data accumulation, the laboratory has an Occupancy

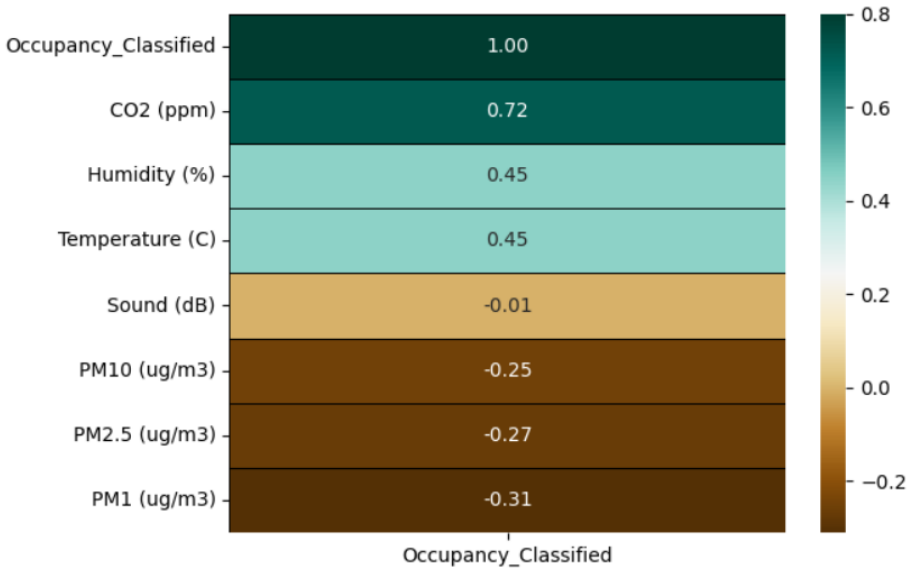
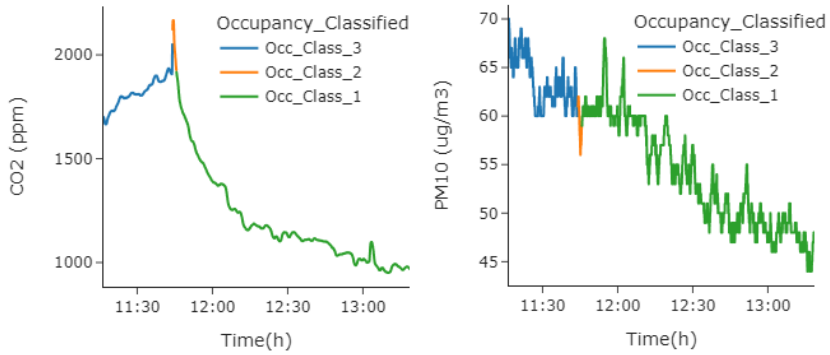


Figure 3.7: *Correlation of all the Parameters with the Classified Occupancy*

range within $21 - 30$, and the values of PM_{10} and the concentration levels of CO_2 are higher than the values of the same afterwards when the room has lesser Occupancy range. Soon after $11:40$ AM, we see a sharp drop in the Occupancy in the lab, which is the reason behind the values of PM_{10} and CO_2 declining throughout the rest of the period.



(a) (b)

Figure 3.8: *Illustration of CO_2 and PM_{10} with Classified Occupancy*

2. **Performance Metrics:** The model's performance can be evaluated using classification metrics such as precision, recall, F1-score and accuracy. A brief description of each metric is listed below.

    - ***Accuracy*** is the ratio of correct predictions to the number of predictions made by the algorithm.
    - ***Precision*** indicates the number of positive class predictions that actually belong to the positive class.
    - ***Recall*** measures how accurate the model is in identifying true positives.
    - ***F1-Score*** measures a model's accuracy by combining model' precision and recall scores.

CHAPTER 4

Experimental Results

4.1 Data Transformation and Cleaning

As explained in Chapter 3.1, after merging all the datasets of different dates, we preprocessed the data by converting all the string attributes features i.e., Position, Weather to categorical codes, i.e., mapping all data to categorical codes, the mapping is shown in Table 4.1. For handling null values, for features except CO_2 we can take the mean or median of the existing values present for filling out data of null values. As CO_2 is a primary attribute and holds maximum correlation with the occupancy class, we cannot introduce mean or median values to the null values. So we will drop all null values.

According to Fig. 3.7, all the features are quite important, so we are not applying any dimensional reduction technique, also dataset width in within the limit. Also we have introduced Position, Floor No., Weather as secondary attributes

Table 4.1: *Mapped Values*

Mapped Value	Position	Room Condition	Room Type	Weather
1	backside	ac	classroom	cloudy
2	frontside	non ac	lab	overcast
3	middle	-	-	rainy
4	-	-	-	sunny

which also has a quite impact on modeling.

We have scaled all the features using Standard Scaling method, which is used to standardize the features by removing the mean and scaling it to unit variance and also it transforms the data in a normal distribution (data near the mean are more frequent in occurrence than data far from the mean) as shown in Fig. 4.1.

$Z = \frac{x-\mu}{\sigma}$,

where μ = Mean, σ = Standard Deviance

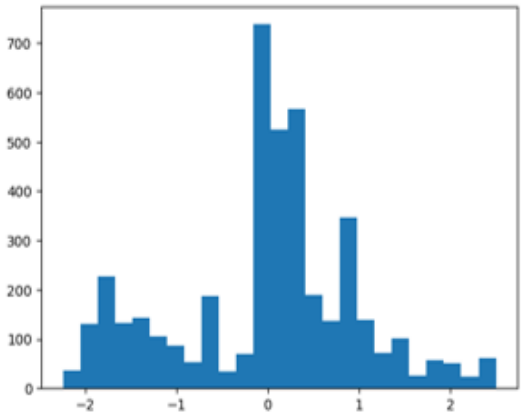


Figure 4.1: *Standardization Distribution*

After preprocessing, the preprocessed dataset will be used for further training. A sample preprocessed dataset is shown in Fig. 4.2.

	CO2 (ppm)	PM1 (ug/m3)	PM2.5 (ug/m3)	PM10 (ug/m3)	Temperature (C)	Humidity (%)	Position	Floor No.	Weather
0	-0.164599	1.560040	1.660422	1.695356	0.053942	0.126944	0.6543	0.0	0.907081
1	-0.166907	1.560040	1.575764	1.615034	0.053942	0.126944	0.6543	0.0	0.907081
2	-0.171524	1.560040	1.575764	1.615034	0.053942	0.126944	0.6543	0.0	0.907081
3	-0.177294	1.445456	1.575764	1.534713	0.053942	0.126944	0.6543	0.0	0.907081
4	-0.184219	1.445456	1.575764	1.454391	0.053942	0.023125	0.6543	0.0	0.907081
5	-0.191143	1.445456	1.491106	1.374070	0.053942	0.023125	0.6543	0.0	0.907081
6	-0.198067	1.560040	1.575764	1.454391	0.053942	0.126944	0.6543	0.0	0.907081
7	-0.202684	1.445456	1.491106	1.374070	0.053942	0.126944	0.6543	0.0	0.907081
8	-0.203838	1.560040	1.491106	1.374070	0.053942	0.023125	0.6543	0.0	0.907081
9	-0.204992	1.445456	1.406448	1.293748	0.053942	0.023125	0.6543	0.0	0.907081

Figure 4.2: *Preprocessed data*

The results are organized in following manner; the first part summarizes the evaluation metrics with the comparison of all models, feature importance, and each class's accuracy. The second part discusses the models which performed best with the evaluation metrics taken into consideration.

In the initial phase, the performance of several machine learning models, such as K-Nearest Neighbour (kNN), Decision Tree (DT), Support Vector Machine (SVM), Random Forest (RF), Gradient Boosting Machine (GBM), Extreme Gradient Boosting (XGB) and deep learning based models, such as Multilayer Perceptron (MLP), etc. for estimating occupancy levels is analyzed. In the next phase, hyper-parameter tuning for each model is done for further performance analysis and optimization. The best parameters are chosen using *GridSearchCV* and *RandomizedSearchCV* with five cross-validate folds where accuracy has been prioritized. All the hyper-parameters are selected using a well-documented and proven approach for tuning models. All experiments are designed and implemented in *Google Colab* using *Python v3.10* with built-in libraries.

4.2 Occupancy Estimation

The study uses the following features to make a prediction model for occupancy class estimation: CO_2 (ppm), PM_1 ($\mu g/m^3$), $PM_{2.5}$ ($\mu g/m^3$), PM_{10} ($\mu g/m^3$), Temperature (°C), Humidity (%), Position, Floor Number, and Weather.

To carry out the experimental analysis, we initially scale the features and divide the dataset into train-test datasets. For each model, we train it using the training dataset and evaluate it on the testing dataset with a split ratio of 60 : 40, indicating that 40% of the data has been assigned as test data and is chosen at random.

1. **Without Hyper-parameter Tuning:** As we can see in Table 4.2, XGB is the best-performing model with an accuracy of 99.23%. This reflects XGB's ability to make accurate predictions across different classes and indicates the excellent F1-score, recall, and precision that XGB routinely achieves at 0.99. RF closely follows, demonstrating a robust accuracy of 98.82% and balanced F1-score, recall, and precision at 0.99. DT and GBM also exhibit strong performance, with accuracy rates of 98.11% and 98.58%. Whereas, KNN and SVM display lesser accuracy at 96.7% and 91.16%. MLP and ANN provide competitive accuracy, with ANN showcasing improved F1-score, recall, and precision at 0.91, 0.92, and 0.92, respectively.

2. **With Hyper-parameter Tuning:** We can show from Table 4.3 that XGB is once again shows an average accuracy of 99.17% and F1-score of 0.99. This demonstrates how well it accomplishes a balance between recall and preci-

Table 4.2: *Performance Metrics of Different Models without Hyper-parameter Tuning*

Algorithm Name	Accuracy (%)	F1	Recall	Precision
KNN	96.7	0.97	0.97	0.97
SVM	91.16	0.91	0.91	0.91
Decision Tree	98.11	0.98	0.98	0.98
Random Forest	98.82	0.99	0.99	0.99
XGB	99.23	0.99	0.99	0.99
GBM	98.58	0.99	0.99	0.99
MLP	91.69	0.91	0.92	0.92
ANN	91.75	0.91	0.92	0.92

sion. DT, RF, and GBM follow closely, displaying strong performances with accuracy rates exceeding 98% and F1-scores around 0.99, highlighting their reliability for accurate classification. KNN and SVM exhibit commendable results with accuracy rates of 97.76% and 96.99%, respectively. While the MLP delivers competitive results with an accuracy of 97.94%, ANN shows satisfactory performance with an accuracy of 91.75%.

3. **Comparison of Estimation Results:** We deduce the following remarks by combining the observations from the results obtained and from Fig. 4.3. (i) XGB outperforms other models with a consistent accuracy of 99% and a high F1-score. (ii) RF, DT, and GBM provide consistently high accuracy of 98% irrespective of hyper-parameter tuning. (iii) SVM and MLP show

Table 4.3: *Performance Metrics of Different Models with Hyper-parameter Tuning*

Algorithm Name	Accuracy (%)	F1	Recall	Precision
KNN	97.76	0.98	0.98	0.98
SVM	96.99	0.97	0.97	0.97
DT	98.82	0.99	0.99	0.99
RF	98.41	0.98	0.98	0.98
XGB	99.17	0.99	0.99	0.99
GBM	98.76	0.99	0.99	0.99
MLP	97.94	0.98	0.98	0.98
ANN	91.75	0.91	0.92	0.92

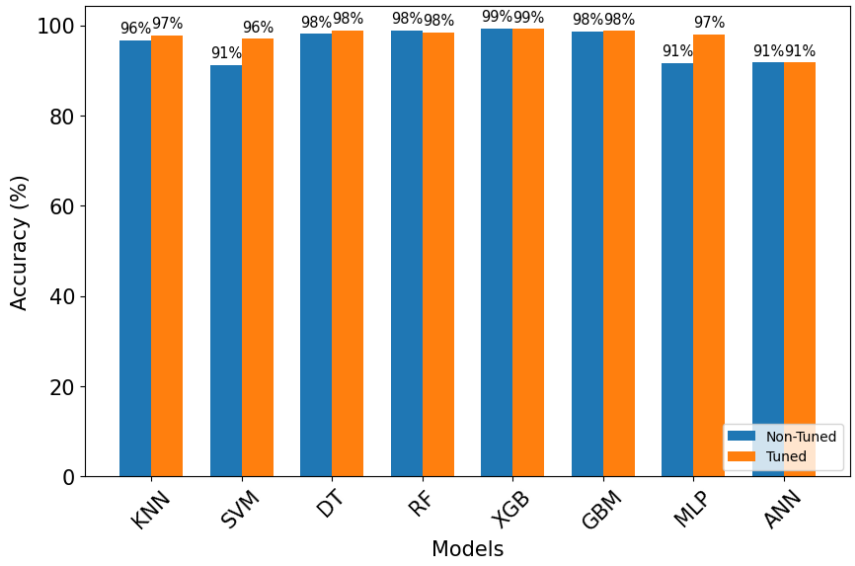


Figure 4.3: *Comparison between Non-Tuned vs Tuned models*

performance increases of $5-6\%$ in accuracy after hyper-parameter tuning and also show balanced recall and F1-scores. (iv) KNN, on the other hand, shows a slightly higher performance increase after hyper-parameter tuning and is overall consistent in predicting occupancy classes. (v) ANN shows the lowest accuracy score of 93% with slightly lower F1-scores.

For further analysis from Fig. 4.3, hyper-tuned models such as XGB, DT, RF, GBM, and MLP are employed. These models have been chosen due to their unique operating theories and excellent occupancy prediction capabilities.

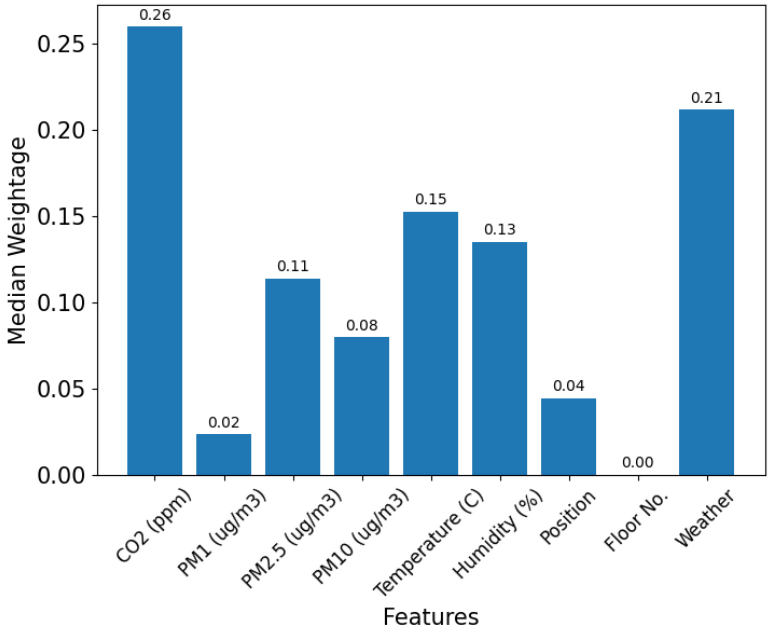


Figure 4.4: *Feature Importance of Selected Models*

4. **Feature Importance:** After performing all experiments, Fig. 4.4 shows the median weightage of feature importance for the selected models in predicting the occupancy class. The figure shows that CO_2, Weather, and Temperature are the most important features and hold the most weightage among all. This

tallies with the observations mentioned in Chitnis *et al.* [16] and Yun *et al.* [17]. Humidity shows significant weightage along with Particulate Matter, which also shows considerable weightage in predicting the occupancy classes as observed in Li *et al.* [9]. Though the position of the sensor box (e.g., front side, backside or middle of the classroom) is not a prime feature, it still has a small but significant impact on occupancy prediction.

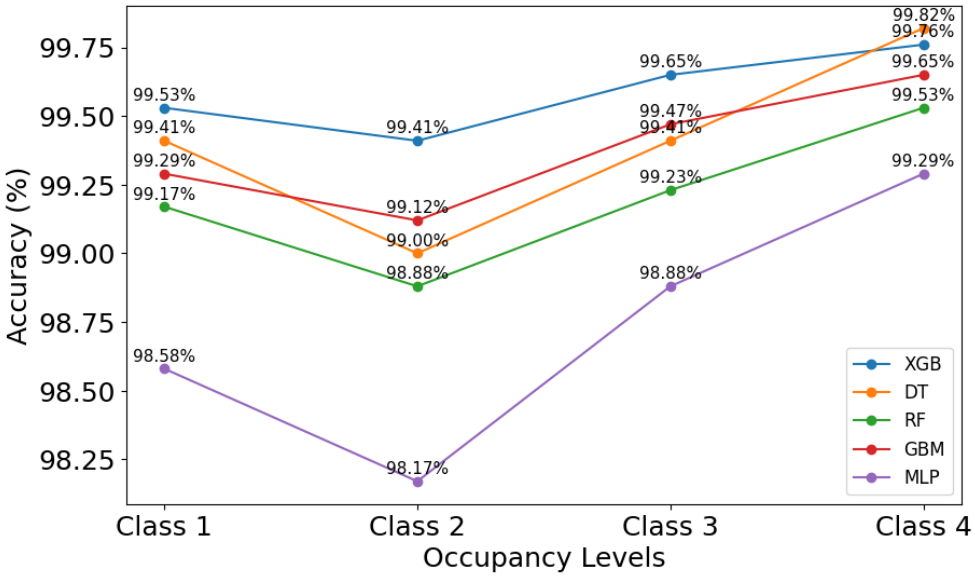


Figure 4.5: *Each Occupancy Class Accuracy Scores*

5. **Each Occupancy Class Estimations:** As noted in Fig. 4.5, XGB demonstrates high accuracy across all classes, with values ranging from 99.53% to 99.76%. DT and RF exhibit competitive results, with accuracies hovering around the high 90s. GBM and MLP also showcase strong accuracy, though slight variations exist across the different classes. This illustration provides insights into how each model performs concerning class-specific accuracy, aiding in evaluating and comparing their classification capabilities.

4.3 Discussion

For occupancy estimation, we have used environment sensors packaged in a custom-built environment sensing box for reading data from any air-conditioned labs in a classroom. We have applied different Machine Learning and Deep Learning algorithms with and without Hyper-parameter tuning to analyses the collected data. We have observed the best performance using XGB (Extreme Gradient Boosting) model with 99.17% without hyper-parameter tuning and 99.23% with hyper-parameter tuning. Further, ANN (Artificial Neural Network) shows satisfactory performance with an accuracy of 91.75% in either scenario. After comparing estimated results, XGB outperforms other models with a consistent accuracy of 99% and a high F1-score. In other case, ANN shows the lowest accuracy score of 93% with slightly lower F1-scores. Finally, after analyzing the collected data, we have observed that it is possible to predict the classroom occupancy in class range 0-4 as our lab capacity is 40.

CHAPTER 5

Conclusion

This research examines how environmental characteristics affect occupancy levels and detection accuracy. The experiment evaluates ML models with respect to the CO_2 (ppm), PM_1 ($\mu g/m^3$), $PM_{2.5}$ ($\mu g/m^3$), PM_{10} ($\mu g/m^3$), Temperature (°C), Humidity (%), Position, Floor Number, and Weather. The sensor box module's acquired dataset is used for the analysis. This allows us to investigate how environmental sensors may be used to forecast occupancy levels. Accuracy ranges from 91% to 99% have been reported in the experiments. With accuracies over 98%-99%, XGBoost, Decision Tree, and Random Forest offer the greatest results.

The work's constraints include occupancy class estimation in empty (unoccupied) rooms, different room setups such as non-air conditioned rooms, different floor levels and predicting the actual occupancy count. The sensor box module's placement location and the sensors' qualities might be further restrictions. Additionally, the size of the dataset (only 12 days in duration) makes it insufficient

to capture every trend.

Future research will focus on developing transfer learning models, creating an accurate regression model to estimate overall occupant values across various types of rooms, and forecasting occupancy in high-capacity rooms. Another future focus area might be combining the proposed model with other state-of-the-art indoor occupancy detection methods. Moreover, various systems exploiting WiFi and BLE signals have been proposed for building occupancy estimation or different building occupancy estimation. However, if wireless systems are used, the problem of privacy breaches cannot be overlooked. To reduce that threat, we can use forthcoming technologies such as edge devices, Blockchain-based IoT and Federated learning in combination with the proposed method. Another future line of work is occupancy estimation while people are engaged in various activities inside the building and frequently move in and out of the room.

Bibliography

[1] Mohammad Esrafilian-Najafabadi and Fariborz Haghighat. Impact of predictor variables on the performance of future occupancy prediction: Feature selection using genetic algorithms and machine learning. *Building and Environment*, 219:109152, 2022.

[2] Donya Sheikh Khan, Jakub Kolarik, Christian Anker Hviid, and Peter Weitzmann. Method for long-term mapping of occupancy patterns in open-plan and single office spaces by using passive-infrared (pir) sensors mounted below desks. *Energy and Buildings*, 230:110534, 2021.

[3] Reza Shahbazian and Irina Trubitsyna. Human sensing by using radio frequency signals: A survey on occupancy and activity detection. *IEEE Access*, 2023.

[4] Kailai Sun, Qianchuan Zhao, Ziyou Zhang, and Xinyuan Hu. Indoor occupancy measurement by the fusion of motion detection and static estimation. *Energy and Buildings*, 254:111593, 2022.

[5] Anurag Shrivastava, SJ Suji Prasad, Ajay Reddy Yeruva, P Mani, Pooja Nagpal, and Abhay Chaturvedi. Iot based rfid attendance monitoring system

of students using arduino esp8266 & adafruit. io on defined area. *Cybernetics and Systems*, pages 1–12, 2023.

[6] Zawar Hussain, Quan Z Sheng, and Wei Emma Zhang. A review and categorization of techniques on device-free human activity recognition. *Journal of Network and Computer Applications*, 167:102738, 2020.

[7] Arindam Ghosh, Amartya Chakraborty, Dhruv Chakraborty, Mousumi Saha, and Sujoy Saha. Ultrasense: A non-intrusive approach for human activity identification using heterogeneous ultrasonic sensor grid for smart home environment. *Journal of Ambient Intelligence and Humanized Computing*, pages 1–22, 2019.

[8] Arindam Ghosh, Kirti Kumari, Sagar Kumar, Mousumi Saha, Subrata Nandi, and Sujoy Saha. Noiseprobe: Assessing the dynamics of urban noise pollution through participatory sensing. In *2019 11th International Conference on Communication Systems & Networks (COMSNETS)*, pages 451–453. IEEE, 2019.

[9] Kangwei Li, Jiandong Shen, Xin Zhang, Linghong Chen, Stephen White, Mingming Yan, Lixia Han, Wen Yang, Xinhua Wang, and Merched Azzi. Variations and characteristics of particulate matter, black carbon and volatile organic compounds in primary school classrooms. *Journal of Cleaner Production*, 252:119804, 2020.

[10] Lavinia Chiara Tagliabue, Fulvio Re Cecconi, Stefano Rinaldi, and Angelo

Luigi Camillo Ciribini. Data driven indoor air quality prediction in educational facilities based on iot network. *Energy and Buildings*, 236:110782, 2021.

[11] Adnan Akbar, Michele Nati, Francois Carrez, and Klaus Moessner. Contextual occupancy detection for smart office by pattern recognition of electricity consumption data. In *2015 IEEE international conference on communications (ICC)*, pages 561–566. IEEE, 2015.

[12] Aya Nabil Sayed, Yassine Himeur, and Faycal Bensaali. Deep and transfer learning for building occupancy detection: A review and comparative analysis. *Engineering Applications of Artificial Intelligence*, 115:105254, 2022.

[13] Paige Wenbin Tien, Shuangyu Wei, John Kaiser Calautit, Jo Darkwa, and Christopher Wood. Vision-based human activity recognition for reducing building energy demand. *Building Services Engineering Research and Technology*, 42(6):691–713, 2021.

[14] Mohamad Khalil, Stephen McGough, Zoya Pourmirza, Mehdi Pazhoohesh, and Sara Walker. Transfer learning approach for occupancy prediction in smart buildings. In *2021 12th International Renewable Engineering Conference (IREC)*, pages 1–6. IEEE, 2021.

[15] Arindam Ghosh, Prithviraj Pramanik, Kartick Das Banerjee, Ashutosh Roy, Subrata Nandi, and Sujoy Saha. Analyzing correlation between air and noise pollution with influence on air quality prediction. In *2018 IEEE International Conference on Data Mining Workshops (ICDMW)*, pages 913–918. IEEE, 2018.

[16] Shubham Chitnis, Nivethitha Somu, and Anupama Kowli. Occupancy estimation with environmental sensors: the possibilities and limitations. *Energy and Built Environment*, 2023.

[17] Seoyeon Yun and Dusan Licina. Investigation of indicators for personal exposure and occupancy in offices by using smart sensors. *Energy and Buildings*, 298:113539, 2023.

Appendices

A.1 Paper Publication

- ***Title:*** OccuCon: A Context-aware Environment Sensing Approach Towards Indoor Occupancy Estimation
- ***Published in:*** 2024 IEEE International Conference on Computing, Power and Communication Technologies (IC2PCT)
- ***Date of Conference:*** 09-10 February 2024
- ***Date Added to IEEE Xplore:*** 08 April 2024
- ***DOI:*** https://doi.org/10.1109/IC2PCT60090.2024.10486464
- ***Publisher:*** IEEE
- ***Conference Location:*** Greater Noida, India

OccuCon: Context-aware Framework for Occupancy Estimation using Environmental Sensors

Ruma Ghosh*, Sumit Dhar†, Saptarshi Ghosh‡, Abhik Mandal§,
Arindam Ghosh¶, Partha Sarathi Paul‖, Sujoy Saha**
*†‡§¶Dr. B. C. Roy Engineering College, Durgapur, India,
‖KIIT University, Bhubaneswar, India, ***National Institute of Technology Durgapur, India
Email:*gh.ruma@gmail.com, †sumit10300203@gmail.com, ‡g2saptarshi@gmail.com, §mandalabhik75@gmail.com
, ¶arindam202@gmail.com, ‖mtc0113@gmail.com, **sujoy.ju@gmail.com

***Abstract*—High occupancy in a room with improper ventilation and outdoor air pollution can significantly impact the occupants' comfort and productivity which may also lead to various health-related issues. In this work, we have studied the relationship of occupancy in a classroom with the various environmental parameters prevailing in that room, such as concentration of CO_2, PM_1, $PM_{2.5}$, PM_{10}, and levels of *Temperature*, *Relative Humidity*, and *Acoustics*. All the above parameters affect indoor occupancy significantly. The real-time data acquisition has been done using a custom device. Collected contextual information helped us to find the relationships among primary features needed to develop estimation models for accurate classroom occupancy prediction. For random sampling, occupancy estimation accuracy ranges from 91% to 99%. Finally, results show that multiple environmental sensor data performed well in predicting occupancy levels.**

***Index Terms*—Occupancy, estimation, context identification, environmental sensors.**

I. Introduction

Occupancy refers to the number of individuals present in a particular facility at any particular time, say, the number of students/teachers present in a school or the number of employees present in an office, etc. [1]. As the degree of occupancy in a classroom can significantly affect the Indoor Air Quality (IAQ) [2], it plays a critical role in students' and teachers' health, comfort, and productivity. Understandably, whenever classroom occupancy increases, the CO_2 level in that classroom rises. An elevated CO_2 level is often an indication of higher occupancy and/or improper ventilation in the classroom. Exposure to such an environment for a long time may cause various health-related issues, including respiratory problems, allergies, dizziness, and reduced cognitive performance [3], [4]. So, managing the occupancy in a facility properly is essential for mutual benefit.

Different approaches and modeling techniques have been incorporated for estimating the degree of indoor occupancy. Studies like Esrafilian-Najafabadi *et al.* [5] proposed various features that may be used for occupancy prediction, such as appliance energy consumption and occupants' current locations. Such implementations compromise occupants' privacy, as their location is revealed, and/or they suffer from the unavailability of adequate appliance energy consumption data. Among other implementations, PIR occupancy [6] sensors may have difficulty in detecting occupants who are not moving. In contrast, WiFi-based systems [7] may face connectivity problems, which determine occupancy inaccurately in *many occupants in the room* scenario [8]. Radio Frequency Identification (RFID) based technologies [7] may also be used to track students in a classroom. Still, the cost of that technology is relatively high and can easily be attacked or tampered [9]. Face Recognition-based techniques are also used to monitor occupancy, identifying human faces from images/videos captured using a digital camera [10]. Though widely used, image-based techniques are highly obtrusive and require high computation resources.

In device-free approaches for occupancy detection [11], [12], environment monitoring sensors such as sensors used to monitor gaseous concentrations or dust levels can be helpful. These sensors are unobtrusive and preserve privacy. The contextual information gathered by such sensors allows us to monitor the IAQ and provides meaningful insights about the surroundings [13]. In a classroom, for example, the Particulate Matter (PM) level may be greatly caused by various human activities and the environment outside [14]. It has also been observed that high occupancy is associated with increased concentration levels of CO_2 [15]. Compared to other states, electricity consumption data has been employed [16] to identify occupancy states with high efficiency and offer contextual information simultaneously without the need for additional specialized occupancy sensors.

Different machine learning-based techniques are used for the automatic estimation of the degree of occupancy. In the work of Sayed *et al.* [17], deep learning (DL), and transfer learning (TL), along with different sensors are used for occupancy estimation. CNN features (R-CNN), and transfer learning is used by Tien *et al.* [18] for occupancy detection in an office space where persons are performing different activities like walking, sitting, and standing. Transfer learning is also used to optimize the accuracy of occupancy estimation in the work proposed by Khalil *et al.* [19]. The result is compared with Random Forest(RF) and Support Vector Machine(SVM) algorithms models and the authors predict the best result with the transfer learning model.

Considering the above-mentioned observations, we propose a device-free framework for estimating the degree of

occupancy using environmental sensors in this paper. The proposed approach is unobtrusive, privacy-preserving, and requires fewer computational resources. Our understanding of the recent works revealed that the application of environment sensors for estimating occupancy is significantly less explored. Therefore, we put forth the following query: ***"What should be the key parameters to estimate classroom occupancy using environment monitoring sensors accurately?"***. This paper's contribution lies in extending the set of features used for occupancy estimation modeling and in optimizing the model parameters to maximize occupancy estimation accuracy.

This study assumes that some features are more important than others for approximating occupancy in a classroom. For this purpose, the proposed work focuses on assessing the occupancy of a classroom by studying the variability of IAQ-related features. Our approach consists of developing an IoT-based device that would enable us in real-time data collection, including levels of CO_2, particulate matter (PM), Temperature, Relative Humidity, and acoustics in a classroom using the various pollution-detection sensors. Further, we explore the relationships between these parameters and classroom occupancy, identify correlations and various dependencies, and develop intelligent models to predict classroom occupancy accurately.

The remaining sections of the paper are organised as follows, methodology used in this work has been discussed in Section II. It includes the Sensing Module, which deals with developing an IOT-based device that will allow real-time data collection using the various sensors. It also describes approaches for sensor validation and inferring occupancy levels using the collected data. The methodology used for experiment and findings have been discussed in Section III. Finally, we conclude and have been discussed future work in Section IV.

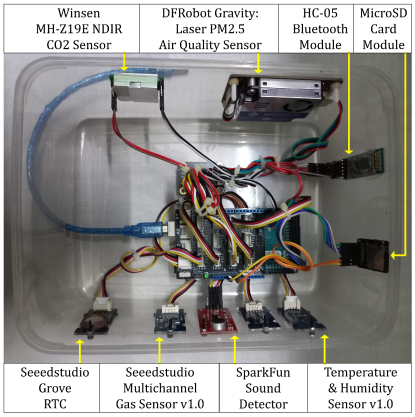


Fig. 1. Environmental Monitoring Device

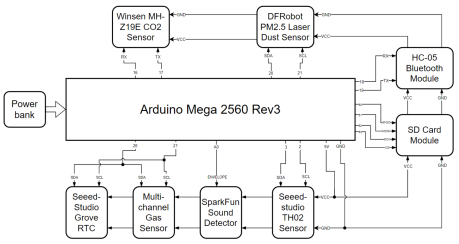


Fig. 2. Device Block Diagram

II. Methodology

The first part of this section summarizes the structure, working, and calibration of the environment monitoring device. The second part discusses about the gathered data and how it is prepared for machine learning.

A. Sensing Module

We have assembled an environment monitoring device [20] outfitted with various inexpensive sensors [Fig. 1]. This device has a Micro-SD card to store the measurements in .csv format. It also has an *HC-05 Bluetooth v4.0* module to send all device critical information to remote devices such as mobile phones/laptops. We equip it with a *SeeedStudio Grove RTC* module to track every measurement's actual date and time. The main parameters that we have considered for this work include CO_2, PM_1, $PM_{2.5}$, PM_{10}, CO, NO_2, Sound, Temperature, and Relative Humidity. All of these parameters are recorded every 5 seconds for an average duration of 1.5 hours in each sampling environment using multiple environmental sensors from various manufacturers such as *SeeedStudio*, *DFRobot*, *Winsen*, and *Sparkfun*. Table I shows detailed information about the sensors used in our experiments. ***Sensor calibration:*** For sensor calibration, we have studied the techniques employed in prior comparable studies [21]. To maintain the least possible atmospheric pollution during calibration, sensors are factory-calibrated or calibrated using the manufacturer's program in an open environment between 12 O'clock Midnight and 6 AM. Other parameters we record before each sampling include outdoor weather at the time of sampling, floor number of the room, room dimensions, room type, room condition, board type used in the room (chalk-duster/marker), window type, and door type (of the room of experiment). Whenever there is a change in classroom occupancy, all the above are recorded in real-time. To allow the sensors to stabilize in the present sampling environment, the device is turned on ten minutes before the reading begins in each sampling. This is done to ensure steady data capture during the sampling. Since insufficient power supply can affect the working of the sensors, a portable 10000 mAh power bank is used to power the device while conducting the data collection. The device has an average current consumption rate of 200 mA. So, it can run for approximately 50 hours on a single recharge 10000 mAh power bank.

TABLE I
SENSOR SPECIFICATION

Name	Manufacturer	Sensing Pollutant	Range	Price (INR)
Winsen MH-Z19E NDIR CO2 Sensor[1]	Winsen	CO_2	CO_2: 400-5000 ppm	1580
Gravity: Laser PM2.5 Air Quality Sensor[2]	DFRobot	PM_1, $PM_{2.5}$, PM_{10}	PM: 0-500 $\mu g/m^3$	3299
Grove: Multichannel Gas Sensor v1.0[3]	SeeedStudio	CO, NO_2	CO: 0-1000 ppm NO_2: 0.05-10 ppm	3600
SparkFun Sound Detector[4]	Sparkfun	Sound	30-130 dB	1049
Grove: Temperature & Humidity Sensor v1.0[5]	SeeedStudio	Temperature, Relative Humidity	T: 0-70 °C RH: 0-80 %	2200

B. Data Processing

During the experimental data collection across multiple Air-Conditioned (AC) Labs we have sampled cumulative data points consisting of 12071 rows. Based on the number of persons in the room, the complete dataset has been classified into uniquely identifiable classes to discover the most important feature parameters affecting occupancy detection. We then develop an accurate classification model based on the same. Table II summarizes each occupancy class in the dataset. Since Class 0 has a very small number of samples, we have excluded it from the train-test dataset.

TABLE II
ASSIGNED CLASSES FOR OCCUPANCY ESTIMATION

Class Name	Occupancy Range	Number of Rows before Pre-processing	Number of Rows after Pre-processing
Class: 0	0	165	19
Class: 1	1 to 10	3613	2428
Class: 2	11 to 20	2837	1773
Class: 3	21 to 30	1742	1401
Class: 4	31 to 40	1401	1177

1) ***Relation of the parameters with the Classified Occupancy****:* At first, we calculate the overall correlation between the parameters and the Classified Occupancy [Fig. 3]. We take into consideration CO_2 and PM as our significant factors because of their strong positive and negative correlations with the Classified Occupancy. Further, we note that temperature and humidity positively correlate with classified Occupation, whereas sound has almost a negligible correlation. That is why we disqualify sound from consideration in our later studies.

To illustrate the relation between different parameters and the occupancy, sample plots of CO_2 and PM_{10} are shown in [Fig. 4]. As we see during the beginning of the data accumulation, the laboratory has an Occupancy range within $21-30$, and the values of PM_{10} and the concentration levels of CO_2 are higher than the values of the same afterwards when

[1]https://www.winsen-sensor.com/product/mh-z19e.html
[2]https://www.dfrobot.com/product-2439.html
[3]https://wiki.seeedstudio.com/Grove-Multichannel_Gas_Sensor/
[4]https://www.sparkfun.com/products/12642/
[5]https://bit.ly/3QMhYFs

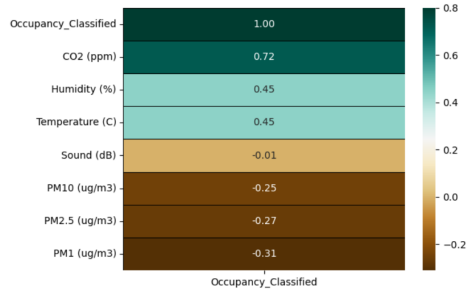
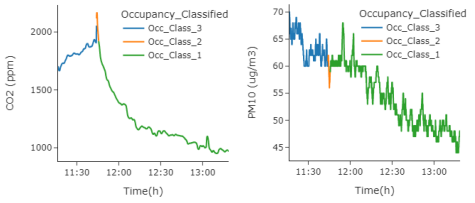


Fig. 3. Correlation of all the parameters with the Classified Occupancy

the room has lesser Occupancy range. Soon after $11:40$ AM, we see a sharp drop in the Occupancy in the lab, which is the reason behind the values of PM_{10} and CO_2 declining throughout the rest of the period.

2) ***Performance metrics****:* The model's performance can be evaluated using classification metrics such as precision, recall, F1-score and accuracy. A brief description of each metric is listed below.

- ***Accuracy*** is the ratio of correct predictions to the number of predictions made by the algorithm.
- ***Precision*** indicates the number of positive class predictions that actually belong to the positive class.
- ***Recall*** measures how accurate the model is in identifying true positives.
- ***F1-Score*** measures a model's accuracy by combining model' precision and recall scores.



(a) (b)

Fig. 4. Illustration of CO_2 and PM_{10} with Classified Occupancy

III. EXPERIMENTAL RESULTS

The results are organized in following manner; the first part summarizes the evaluation metrics with the comparison of all models, feature importance, and each class's accuracy. The second part discusses the models which performed best with the evaluation metrics taken into consideration.

In the initial phase, the performance of several machine learning models, such as K-Nearest Neighbour (kNN), Decision Tree (DT), Support Vector Machine (SVM),

Random Forest (RF), Gradient Boosting Machine (GBM), Extreme Gradient Boosting (XGB) and deep learning based models, such as Multilayer Perceptron (MLP), etc. for estimating occupancy levels is analyzed. In the next phase, hyper-parameter tuning for each model is done for further performance analysis and optimization. The best parameters are chosen using *GridSearchCV* and *RandomizedSearchCV* with five cross-validate folds where accuracy has been prioritized. All the hyper-parameters are selected using a well-documented and proven approach for tuning models. All experiments are designed and implemented in *Google Colab* using *Python v3.10* with built-in libraries.

A. Occupancy estimation

The study uses the following features to make a prediction model for occupancy class estimation: CO_2 (ppm), PM_1 ($\mu g/m^3$), $PM_{2.5}$ ($\mu g/m^3$), PM_{10} ($\mu g/m^3$), Temperature (°C), Humidity (%), Position, Floor Number, and Weather.

To carry out the experimental analysis, we initially scale the features and divide the dataset into train-test datasets. For each model, we train it using the training dataset and evaluate it on the testing dataset with a split ratio of $60:40$, indicating that 40% of the data has been assigned as test data and is chosen at random.

TABLE III
PERFORMANCE METRICS OF DIFFERENT MODELS WITHOUT HYPER-PARAMETER TUNING

Algorithm Name	Accuracy (%)	F1	Recall	Precision
KNN	96.7	0.97	0.97	0.97
SVM	91.16	0.91	0.91	0.91
Decision Tree	98.11	0.98	0.98	0.98
Random Forest	98.82	0.99	0.99	0.99
XGB	99.23	0.99	0.99	0.99
GBM	98.58	0.99	0.99	0.99
MLP	91.69	0.91	0.92	0.92
ANN	91.75	0.91	0.92	0.92

*1) **Without Hyper-parameter tuning**:* As we can see in Table III, XGB is the best-performing model with an accuracy of 99.23%. This reflects XGB's ability to make accurate predictions across different classes and indicates the excellent F1-score, recall, and precision that XGB routinely achieves at 0.99. RF closely follows, demonstrating a robust accuracy of 98.82% and balanced F1-score, recall, and precision at 0.99. DT and GBM also exhibit strong performance, with accuracy rates of 98.11% and 98.58%. Whereas, KNN and SVM display lesser accuracy at 96.7% and 91.16%. MLP and ANN provide competitive accuracy, with ANN showcasing improved F1-score, recall, and precision at 0.91, 0.92, and 0.92, respectively.

*2) **With Hyper-parameter tuning**:* We can show from Table IV that XGB is once again shows an average accuracy of 99.17% and F1-score of 0.99. This demonstrates how well

TABLE IV
PERFORMANCE METRICS OF DIFFERENT MODELS WITH HYPER-PARAMETER TUNING

Algorithm Name	Accuracy (%)	F1	Recall	Precision
KNN	97.76	0.98	0.98	0.98
SVM	96.99	0.97	0.97	0.97
DT	98.82	0.99	0.99	0.99
RF	98.41	0.98	0.98	0.98
XGB	99.17	0.99	0.99	0.99
GBM	98.76	0.99	0.99	0.99
MLP	97.94	0.98	0.98	0.98
ANN	91.75	0.91	0.92	0.92

it accomplishes a balance between recall and precision. DT, RF, and GBM follow closely, displaying strong performances with accuracy rates exceeding 98% and F1-scores around 0.99, highlighting their reliability for accurate classification. KNN and SVM exhibit commendable results with accuracy rates of 97.76% and 96.99%, respectively. While the MLP delivers competitive results with an accuracy of 97.94%, ANN shows satisfactory performance with an accuracy of 91.75%.

*3) **Comparison of estimation results**:* We deduce the following remarks by combining the observations from the results obtained and from Fig. 5. (i) XGB outperforms other models with a consistent accuracy of 99% and a high F1-score. (ii) RF, DT, and GBM provide consistently high accuracy of 98% irrespective of hyper-parameter tuning. (iii) SVM and MLP show performance increases of $5-6\%$ in accuracy after hyper-parameter tuning and also show balanced recall and F1-scores. (iv) KNN, on the other hand, shows a slightly higher performance increase after hyper-parameter tuning and is overall consistent in predicting occupancy classes. (v) ANN shows the lowest accuracy score of 93% with slightly lower F1-scores.

For further analysis from figure 5, hyper-tuned models such as XGB, DT, RF, GBM, and MLP are employed. These models have been chosen due to their unique operating theories and

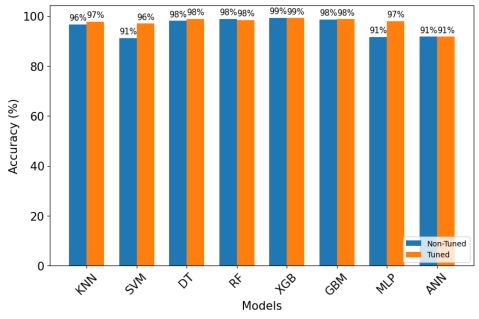


Fig. 5. Comparison between non-tuned vs tuned models

excellent occupancy prediction capabilities.

*4) **Feature Importance**:* After performing all experiments, Fig. 6 shows the median weightage of feature importance for the selected models in predicting the occupancy class. The figure shows that CO_2, Weather, and Temperature are the most important features and hold the most weightage among all. This tallies with the observations mentioned in Chitnis *et al.* [22] and Yun *et al.* [23]. Humidity shows significant weightage along with Particulate Matter, which also shows considerable weightage in predicting the occupancy classes as observed in Li *et al.* [14]. Though the position of the sensor box (e.g., front side, backside or middle of the classroom) is not a prime feature, it still has a small but significant impact on occupancy prediction.

*5) **Each Occupancy Class Estimations**:* As noted in Fig. 7, XGB demonstrates high accuracy across all classes, with values ranging from 99.53% to 99.76%. DT and RF exhibit competitive results, with accuracies hovering around the high 90s. GBM and MLP also showcase strong accuracy, though slight variations exist across the different classes. This illustration provides insights into how each model performs concerning class-specific accuracy, aiding in evaluating and comparing their classification capabilities.

B. Discussion

For occupancy estimation, we have used environment sensors packaged in a custom-built environment sensing box for reading data from any air-conditioned labs in a classroom. We have applied different Machine Learning and Deep Learning algorithms with and without Hyper-parameter tuning to analyses the collected data. We have observed the best performance using XGB (Extreme Gradient Boosting) model with 99.17% without hyper-parameter tuning and 99.23% with hyper-parameter tuning. Further, ANN (Artificial Neural Network) shows satisfactory performance with an accuracy of 91.75% in either scenario. After comparing estimated results, XGB outperforms other models with a consistent accuracy of

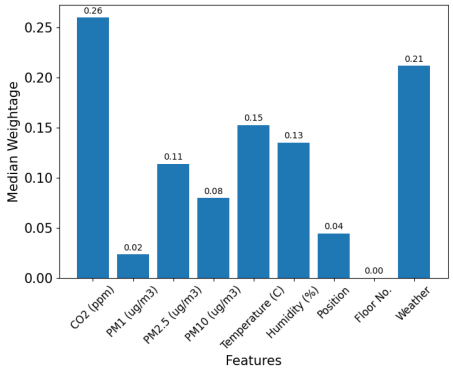


Fig. 6. Feature Importance of selected models

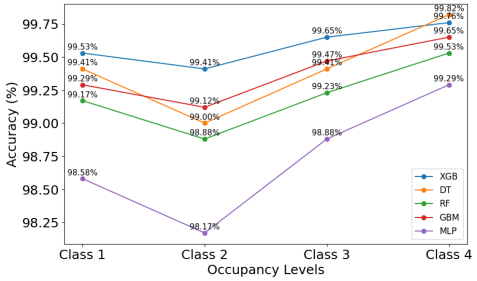


Fig. 7. Each Occupancy Class Accuracy Scores

99% and a high F1-score. In other case, ANN shows the lowest accuracy score of 93% with slightly lower F1-scores. Finally, after analyzing the collected data, we have observed that it is possible to predict the classroom occupancy in class range 0-4 as our lab capacity is 40.

IV. Conclusion and Future Work

This research examines how environmental characteristics affect occupancy levels and detection accuracy. The experiment evaluates ML models with respect to the CO_2 (ppm), PM_1 ($\mu g/m^3$), $PM_{2.5}$ ($\mu g/m^3$), PM_{10} ($\mu g/m^3$), Temperature (°C), Humidity (%), Position, Floor Number, and Weather. The sensor box module's acquired dataset is used for the analysis. This allows us to investigate how environmental sensors may be used to forecast occupancy levels. Accuracy ranges from 91% to 99% have been reported in the experiments. With accuracies over 98%-99%, XGBoost, Decision Tree, and Random Forest offer the greatest results.

The work's constraints include occupancy class estimation in empty (unoccupied) rooms, different room setups such as non-air conditioned rooms, different floor levels and predicting the actual occupancy count. The sensor box module's placement location and the sensors' qualities might be further restrictions. Additionally, the size of the dataset (only 12 days in duration) makes it insufficient to capture every trend.

Future research will focus on developing transfer learning models, creating an accurate regression model to estimate overall occupant values across various types of rooms, and forecasting occupancy in high-capacity rooms. Another future focus area might be combining the proposed model with other state-of-the-art indoor occupancy detection methods. As discussed in section I, various systems exploiting WiFi and BLE signals have been proposed for building occupancy estimation or different building occupancy estimation. However, if wireless systems are used, the problem of privacy breaches cannot be overlooked. To reduce that threat, we can use forthcoming technologies such as edge devices, Blockchain-based IoT and Federated learning in combination with the proposed method. Another future line of work is occupancy estimation while people are engaged in various

activities inside the building and frequently move in and out of the room.

REFERENCES

[1] Y. Jin, D. Yan, A. Chong, B. Dong, and J. An, "Building occupancy forecasting: A systematical and critical review," *Energy and Buildings*, vol. 251, p. 111345, 2021.

[2] L. Chatzidiakou, D. Mumovic, and A. J. Summerfield, "What do we know about indoor air quality in school classrooms? a critical review of the literature," *Intelligent Buildings International*, vol. 4, no. 4, pp. 228–259, 2012.

[3] K. Azuma, N. Kagi, U. Yanagi, and H. Osawa, "Effects of low-level inhalation exposure to carbon dioxide in indoor environments: A short review on human health and psychomotor performance," *Environment international*, vol. 121, pp. 51–56, 2018.

[4] I. Manisalidis, E. Stavropoulou, A. Stavropoulos, and E. Bezirtzoglou, "Environmental and health impacts of air pollution: a review," *Frontiers in public health*, vol. 8, p. 14, 2020.

[5] M. Esrafilian-Najafabadi and F. Haghighat, "Impact of predictor variables on the performance of future occupancy prediction: Feature selection using genetic algorithms and machine learning," *Building and Environment*, vol. 219, p. 109152, 2022.

[6] D. S. Khan, J. Kolarik, C. A. Hviid, and P. Weitzmann, "Method for long-term mapping of occupancy patterns in open-plan and single office spaces by using passive-infrared (pir) sensors mounted below desks," *Energy and Buildings*, vol. 230, p. 110534, 2021.

[7] R. Shahbazian and I. Trubitsyna, "Human sensing by using radio frequency signals: A survey on occupancy and activity detection," *IEEE Access*, 2023.

[8] K. Sun, Q. Zhao, Z. Zhang, and X. Hu, "Indoor occupancy measurement by the fusion of motion detection and static estimation," *Energy and Buildings*, vol. 254, p. 111593, 2022.

[9] A. Shrivastava, S. Suji Prasad, A. R. Yeruva, P. Mani, P. Nagpal, and A. Chaturvedi, "Iot based rfid attendance monitoring system of students using arduino esp8266 & adafruit. io on defined area," *Cybernetics and Systems*, pp. 1–12, 2023.

[10] S. M. Bah and F. Ming, "An improved face recognition algorithm and its application in attendance management system," *Array*, vol. 5, p. 100014, 2020.

[11] Z. Hussain, Q. Z. Sheng, and W. E. Zhang, "A review and categorization of techniques on device-free human activity recognition," *Journal of Network and Computer Applications*, vol. 167, p. 102738, 2020.

[12] A. Ghosh, A. Chakraborty, D. Chakraborty, M. Saha, and S. Saha, "Ultrasense: A non-intrusive approach for human activity identification using heterogeneous ultrasonic sensor grid for smart home environment," *Journal of Ambient Intelligence and Humanized Computing*, pp. 1–22, 2019.

[13] A. Ghosh, K. Kumari, S. Kumar, M. Saha, S. Nandi, and S. Saha, "Noiseprobe: Assessing the dynamics of urban noise pollution through participatory sensing," in *2019 11th International Conference on Communication Systems & Networks (COMSNETS)*. IEEE, 2019, pp. 451–453.

[14] K. Li, J. Shen, X. Zhang, L. Chen, S. White, M. Yan, L. Han, W. Yang, X. Wang, and M. Azzi, "Variations and characteristics of particulate matter, black carbon and volatile organic compounds in primary school classrooms," *Journal of Cleaner Production*, vol. 252, p. 119804, 2020.

[15] L. C. Tagliabue, F. R. Cecconi, S. Rinaldi, and A. L. C. Ciribini, "Data driven indoor air quality prediction in educational facilities based on iot network," *Energy and Buildings*, vol. 236, p. 110782, 2021.

[16] A. Akbar, M. Nati, F. Carrez, and K. Moessner, "Contextual occupancy detection for smart office by pattern recognition of electricity consumption data," in *2015 IEEE international conference on communications (ICC)*. IEEE, 2015, pp. 561–566.

[17] A. N. Sayed, Y. Himeur, and F. Bensaali, "Deep and transfer learning for building occupancy detection: A review and comparative analysis," *Engineering Applications of Artificial Intelligence*, vol. 115, p. 105254, 2022.

[18] P. W. Tien, S. Wei, J. K. Calautit, J. Darkwa, and C. Wood, "Vision-based human activity recognition for reducing building energy demand," *Building Services Engineering Research and Technology*, vol. 42, no. 6, pp. 691–713, 2021.

[19] M. Khalil, S. McGough, Z. Pourmirza, M. Pazhoohesh, and S. Walker, "Transfer learning approach for occupancy prediction in smart buildings," in *2021 12th International Renewable Engineering Conference (IREC)*. IEEE, 2021, pp. 1–6.

[20] A. Ghosh, S. Mondal, M. Saha, S. Saha, and S. Nandi, "Poster: Air quality monitoring using low-cost sensingdevices," in *Proceedings of the 14th Annual International Conference on Mobile Systems, Applications, and Services Companion*, 2016, pp. 27–27.

[21] A. Ghosh, P. Pramanik, K. D. Banerjee, A. Roy, S. Nandi, and S. Saha, "Analyzing correlation between air and noise pollution with influence on air quality prediction," in *2018 IEEE International Conference on Data Mining Workshops (ICDMW)*. IEEE, 2018, pp. 913–918.

[22] S. Chitnis, N. Somu, and A. Kowli, "Occupancy estimation with environmental sensors: the possibilities and limitations," *Energy and Built Environment*, 2023.

[23] S. Yun and D. Licina, "Investigation of indicators for personal exposure and occupancy in offices by using smart sensors," *Energy and Buildings*, vol. 298, p. 113539, 2023.

A.2 Device Code

```
#include <Wire.h>
#include <SPI.h>
#include <SD.h>
#include <TimeLib.h>
#include <RTClib.h>
#include <MutichannelGasSensor.h>
#include <MHZ19.h>
#include <sensirion_common.h>
#include <sgp30.h>
#include <DFRobot_AirQualitySensor.h>
#include <SoftwareI2C.h>
#include "src/TH02_dev.h"

void call_func(bool);
void error_led();
void rtc_1307(bool);
String str_md_hm();
String str_ymd_hms();
void multi_gas_sen(bool);
void z19e_co2_sen(bool);
void seed_sgp30_sen(bool);
void pm_sen(bool);
void th02_sen(bool);
void sound_sen(bool);

#define LED_BUILTIN 13
#define SD_MODULE_CS 53
#define SOUND_ENVELOPE A0
#define FREQUENCY 5000
#define HEAT_TIME 600000

unsigned long device_runtime;
int heat_lock=0;
short sgp_err;
unsigned short scaled_ethanol_signal, scaled_h2_signal;
String filename;

File myFile;
RTC_DS1307 rtc;
SoftwareI2C TH02_i2c;
MHZ19 myMHZ19;
DFRobot_AirQualitySensor particle(&Wire, 0x19);


void setup() {
    pinMode(LED_BUILTIN, OUTPUT);
    pinMode(SD_MODULE_CS, OUTPUT);
```

```
48        pinMode(SOUND_ENVELOPE, INPUT);
49
50        Wire.begin();
51
52        digitalWrite(LED_BUILTIN, LOW);
53
54        Serial.begin(9600);
55        Serial.println("Device starting...");
56        Serial1.begin(9600);
57        Serial1.println("Device starting...");
58        Serial2.begin(9600);
59
60        if (!rtc.begin()) {
61            Serial.println("RTC not found!");
62            Serial1.println("RTC not found!");
63            error_led();
64        }
65
66        gas.begin(0x04);
67        gas.powerOn();
68        delay(500);
69
70        myMHZ19.begin(Serial2);
71
72        if(myMHZ19.errorCode != RESULT_OK) {
73            Serial.println("CO2 sensor failed!");
74            Serial1.println("CO2 sensor failed!");
75            error_led();
76        }
77        delay(500);
78
79        while(sgp_probe() != STATUS_OK) {
80            Serial.println("SGP sensor failed!");
81            Serial1.println("SGP sensor failed!");
82            error_led();
83        }
84
85        sgp_err = sgp_measure_signals_blocking_read(&
          scaled_ethanol_signal, &scaled_h2_signal);
86        if (sgp_err != STATUS_OK) {
87            Serial.println("No SGP signals!");
88            Serial1.println("No SGP signals!");
89            error_led();
90        }
91        sgp_err = sgp_iaq_init();
92        delay(500);
93
```

```
94        while(!particle.begin()) {
95            Serial.println("PM not found!");
96            Serial1.println("PM not found!");
97            error_led();
98        }
99        delay(500);
100
101       TH02.initSoftwareI2C(&TH02_i2c, 3, 2);
102       delay(500);
103
104       if(!SD.begin(SD_MODULE_CS)) {
105           Serial.println("No SD card module!");
106           Serial1.println("No SD card module!");
107           error_led();
108       }
109
110       Serial.println("Creating Dataset file!");
111       Serial1.println("Creating Dataset file!");
112       filename = str_md_hm() + ".csv";
113       myFile = SD.open(filename, FILE_WRITE);
114
115       if(myFile) {
116           myFile.print("Date,Time,");
117           myFile.print("CO (ppm),NO2 (ppm),");
118           myFile.print("CO2 (ppm),");
119           myFile.print("TVOC (ppb),");
120           myFile.print("PM1 (ug/m3),PM2.5 (ug/m3),PM10
              (ug/m3),");
121           myFile.print("Temperature (C),Humidity (%),");
122           myFile.print("Sound (dB),");
123           myFile.print("Occupancy,");
124           myFile.println();
125           myFile.flush();
126           myFile.close();
127       }
128       else {
129           Serial.println("Dataset file creation failed!");
130           Serial1.println("Dataset file creation failed!");
131           error_led();
132       }
133
134       Serial.print("File created: ");
135       Serial.println(filename);
136       Serial1.print("File created: ");
137       Serial1.println(filename);
138
139       Serial.println("Device startup complete!");
```

```
140       Serial1.println("Device startup complete!");
141       Serial.println("Device in heating state!\n");
142       Serial1.println("Device in heating state!\n");
143       digitalWrite(LED_BUILTIN, HIGH);
144   }
145
146
147   void loop() {
148       device_runtime = millis();
149
150       if(device_runtime>HEAT_TIME && heat_lock==0) {
151           heat_lock = 1;
152
153           digitalWrite(LED_BUILTIN, LOW);
154
155           Serial.println("Device heating complete!\n");
156           Serial1.println("Device heating complete!\n");
157       }
158
159       if(device_runtime>HEAT_TIME && heat_lock==1) {
160           Serial.println("Opening Dataset file to write!");
161           Serial1.println("Opening Dataset file to write!");
162           myFile = SD.open(filename, FILE_WRITE);
163
164           if(myFile) {
165               call_func(true);
166               myFile.println();
167               myFile.flush();
168               myFile.close();
169               Serial.println("Closing Dataset file!");
170               Serial1.println("Closing Dataset file!");
171               Serial.println();
172               Serial1.println();
173               delay(FREQUENCY);
174           }
175           else {
176               Serial.println("Failed to open Dataset file!");
177               Serial1.println("Failed to open Dataset file!");
178               error_led();
179           }
180       }
181       else {
182           call_func(false);
183           Serial.println();
184           Serial1.println();
185           delay(FREQUENCY);
186       }
```

```
187   }
188
189
190   void call_func(bool sd) {
191       Serial.print("Device Runtime: ");
192       Serial.print(device_runtime/60000.0, 1);
193       Serial.println(" min");
194       Serial1.print("Device Runtime: ");
195       Serial1.print(device_runtime/60000.0, 1);
196       Serial1.println(" min");
197
198       rtc_1307(sd);
199       multi_gas_sen(sd);
200       z19e_co2_sen(sd);
201       seed_sgp30_sen(sd);
202       pm_sen(sd);
203       th02_sen(sd);
204       sound_sen(sd);
205   }
206
207
208   void error_led() {
209       while(1) {
210           digitalWrite(LED_BUILTIN, HIGH);
211           delay(250);
212           digitalWrite(LED_BUILTIN, LOW);
213           delay(250);
214       }
215   }
```

A.3 Machine Learning Code

```
with open("/content/ac_lab_original_models_object_40_test.pkl", "rb") as file:
    tmp1 = joblib.load(file)

tmp1 = evaluate_result(tmp1)

ob = torch.load("/content/ac_lab_ann_model_object_40_test.pkl")
tmp2 = evaluate_ann(ob, *train_test_split(ac_lab_x, ac_lab_y, test_size=0.4, random_state=42))[1]

qwe = tmp1.join(tmp2).loc['Test'].loc[['Accuracy', 'F1 Weighted', 'Recall Weighted', 'Precision Weighted']][
    ['KNN', 'SVM', 'DecisionTree', 'RandomForest', 'XGB', 'GradientBoosting', 'MLP Neural Net', 'ANN']].T
qwe['F1 Weighted'] = qwe['F1 Weighted'].map(lambda x: round(x / 100, 2))
qwe['Recall Weighted'] = qwe['Recall Weighted'].map(lambda x: round(x / 100, 2))
qwe['Precision Weighted'] = qwe['Precision Weighted'].map(lambda x: round(x / 100, 2))
qwe
```

Figure 1: *Non-Hypertuned models training and evaluation*

```
with open("/content/ac_lab_hypertuned_models_object_40_test.pkl", "rb") as file:
    tmp3 = joblib.load(file)

tmp3 = evaluate_result(tmp3)

ob = torch.load("/content/ac_lab_ann_model_object_40_test.pkl")
tmp4 = evaluate_ann(ob, *train_test_split(ac_lab_x, ac_lab_y, test_size=0.4, random_state=42))[1]

qwe_h = tmp3.join(tmp4).loc['Test'].loc[['Accuracy', 'F1 Weighted', 'Recall Weighted', 'Precision Weighted']][
    ['KNN', 'SVM', 'DecisionTree', 'RandomForest', 'XGB', 'GradientBoosting', 'MLP Neural Net', 'ANN']].T
qwe_h['F1 Weighted'] = qwe_h['F1 Weighted'].map(lambda x: round(x / 100, 2))
qwe_h['Recall Weighted'] = qwe_h['Recall Weighted'].map(lambda x: round(x / 100, 2))
qwe_h['Precision Weighted'] = qwe_h['Precision Weighted'].map(lambda x: round(x / 100, 2))
qwe_h
```

Figure 2: *Hypertuned models training and evaluation*

```python
models = qwe.rename(index = {
    'RandomForest': 'RF',
    'MLP Neural Net': 'MLP',
    'GradientBoosting': 'GBM',
    'DecisionTree': 'DT'}).index.tolist()  # Sample data
accuracy_nontuned = qwe['Accuracy']
accuracy_tuned = qwe_h['Accuracy']
fig, ax = plt.subplots(figsize=(10, 4)) # Set the figure size
bar_width = 0.35
index = np.arange(len(models))
# Plotting the bars
bars1 = ax.bar(index, accuracy_nontuned, width=bar_width, label='Non-Tuned')
bars2 = ax.bar(index + bar_width, accuracy_tuned, width=bar_width, label='Tuned')
# Annotate the bars with accuracy values
for bar, acc in zip(bars1, accuracy_nontuned):
    height = bar.get_height()
    ax.annotate(f'{int(acc)}%', xy=(bar.get_x() + bar.get_width() / 2, height),
                xytext=(0, 3),  # 3 points vertical offset
                textcoords="offset points",
                ha='center', va='bottom', fontsize=10.5)
for bar, acc in zip(bars2, accuracy_tuned):
    height = bar.get_height()
    ax.annotate(f'{int(acc)}%', xy=(bar.get_x() + bar.get_width() / 2, height),
                xytext=(0, 3),  # 3 points vertical offset
                textcoords="offset points",
                ha='center', va='bottom', fontsize=10.5)
# Set labels and title
ax.set_xlabel('Models', fontsize=15)
ax.set_ylabel('Accuracy (%)', fontsize=15)
# Rotate x-axis labels for better readability
ax.set_xticks(index + bar_width / 2)
ax.set_xticklabels(models, rotation=45, fontsize=15)
ax.tick_params(axis='y', which='both', labelsize=15)
# Add legend
ax.legend(loc='lower right')
plt.subplots_adjust(top=1.3)
# Show the plot
plt.show()
```

Figure 3: *Bar plot generation for comparing accuracy scores of models*

```python
with open("/content/ac_lab_hypertuned_models_object_40_test.pkl", "rb") as file:
    tmp3 = joblib.load(file)

est_mlp = lambda x: (x - np.min(x)) / (np.max(x) - np.min(x))

feature_importances_dt = tmp3.models['DecisionTree'].named_steps['classifier'].feature_importances_
feature_importances_rf = tmp3.models['RandomForest'].named_steps['classifier'].feature_importances_
feature_importances_xgb = tmp3.models['XGB'].named_steps['classifier'].feature_importances_
feature_importances_gb = tmp3.models['GradientBoosting'].named_steps['classifier'].feature_importances_

all_feature_importances = np.vstack((feature_importances_dt, feature_importances_rf,
                                     feature_importances_xgb, feature_importances_gb))
mlp_feature_importances = est_mlp([np.sum(np.abs(layer), axis=1) for layer in [layer / np.linalg.norm(layer, ord=2, axis=0)
                                   for layer in tmp3.models['MLP Neural Net'].named_steps['classifier'].coefs_]][0])
all_feature_importances = np.vstack((all_feature_importances, mlp_feature_importances))

fig, ax = plt.subplots(figsize=(8, 4))

# Plotting the bars
bars = ax.bar(df_tmp.columns[:-1].to_list(), np.median(all_feature_importances, axis=0))

ax.set_xlabel('Features', fontsize=15)
ax.set_ylabel('Median Weightage', fontsize=15)

plt.legend().set_visible(False)
ax.set_xticklabels(df_tmp.columns[:-1].to_list(), rotation=45, fontsize=12)
ax.tick_params(axis='y', which='both', labelsize=15)

# Annotate the bars with their values
for bar in bars:
    height = bar.get_height()
    ax.annotate(f'{height:.2f}', xy=(bar.get_x() + bar.get_width() / 2, height),
                xytext=(0, 3), textcoords="offset points",
                ha='center', va='bottom', fontsize=10)

# Show the plot
plt.subplots_adjust(top=1.2)
plt.show()
```

Figure 4: *Median weightage feature importances generation for each features in dataset*

```python
def calculate_class_accuracy(obj = None, confusion_matrix_ann = None):
    each_model_each_class_accuracy, res = {}, None
    if obj is not None:
        for model in obj.model_names:
            confusion_matrix = obj.get_metric_scores(model)['Test Confusion Matrix']
            num_classes, class_accuracies = len(confusion_matrix), {}
            for i in range(num_classes):
                TP = confusion_matrix[i, i]
                FP = sum(confusion_matrix[:, i]) - TP
                FN = sum(confusion_matrix[i, :]) - TP
                TN = np.sum(confusion_matrix) - TP - FP - FN
                total_samples = TP + TN + FP + FN
                accuracy = (TP + TN) / total_samples
                class_accuracies[f'Class {i + 1}'] = {'Accuracy (%)': round(accuracy * 100.00, 2),
                                                      'Total Samples': total_samples,
                                                      'Total Correct Samples Predicted': TP + TN}
            each_model_each_class_accuracy[model] = class_accuracies
    if confusion_matrix_ann is not None:
        num_classes, class_accuracies = len(confusion_matrix_ann), {}
        for i in range(num_classes):
            TP = confusion_matrix_ann[i, i]
            FP = sum(confusion_matrix_ann[:, i]) - TP
            FN = sum(confusion_matrix_ann[i, :]) - TP
            TN = np.sum(confusion_matrix_ann) - TP - FP - FN
            total_samples = TP + TN + FP + FN
            accuracy = (TP + TN) / total_samples
            class_accuracies[f'Class {i}'] = {'Accuracy (%)': round(accuracy * 100.00, 2),
                                              'Total Samples': total_samples,
                                              'Total Correct Samples Predicted': TP + TN}
        each_model_each_class_accuracy['ANN'] = class_accuracies
    for model_name in each_model_each_class_accuracy:
        tmp = pd.DataFrame(each_model_each_class_accuracy[model_name]).T.stack(0).reset_index().rename(
            columns = {'level_0': 'Class', 'level_1': 'Attributes', 0: model_name}).set_index(
                ['Class', 'Attributes'], drop = True)
        if res is None:
            res = tmp
        else:
            res = res.join(tmp)
    return res
```

```python
with open("/content/ac_lab_hypertuned_models_object_40_test.pkl", "rb") as file:
    tmp1 = joblib.load(file)

tmp2 = evaluate_ann(torch.load("/content/ac_lab_ann_model_object_40_test.pkl"),
                    *train_test_split(ac_lab_x, ac_lab_y, test_size=0.4, random_state=42))[0]['Test Confusion Matrix']

class_accuracy_df = calculate_class_accuracy(tmp1, tmp2).query("Attributes == 'Accuracy (%)'")

data_to_plot = class_accuracy_df[['XGB', 'DecisionTree', 'RandomForest', 'GradientBoosting', 'MLP Neural Net']].rename(
    columns = {'DecisionTree': 'DT', 'RandomForest': 'RF', 'GradientBoosting': 'GBM', 'MLP Neural Net': 'MLP'})

ax = data_to_plot.plot(marker='o', figsize=(10, 6))

for model in data_to_plot.columns:
    for index, value in enumerate(data_to_plot[model]):
        ax.annotate(f'{value:.2f}%', (index, value), textcoords="offset points", xytext=(0, 5), ha='center', fontsize=11)

new_labels = data_to_plot.index.get_level_values(0).to_list()
ax.set_xticks(range(len(new_labels)))
ax.set_xticklabels(new_labels, fontsize=18)
ax.set_xlabel('Occupancy Levels', fontsize=18)
ax.set_ylabel('Accuracy (%)', fontsize=18)
ax.tick_params(axis='y', which='both', labelsize=18);
ax.legend(fontsize='large')
plt.show()
```

Figure 5: *Each Classes Accuracy Score Generation Comparison Plot*